



UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE COMPUTACIÓN  
CENTRO DE INVESTIGACIÓN DE SISTEMAS DE INFORMACIÓN

**Desarrollo de una Aplicación de Apoyo a un  
Gestor de Redes Sociales, para la Captura de  
Datos desde la Red Social Twitter y su Respaldo  
en una Base de Datos NoSQL**

Trabajo Especial de Grado presentado ante la ilustre  
Universidad Central de Venezuela por  
Br. Guillén Omar.

Para optar al título de Licenciado en Computación

Tutor:

Prof. Franky Uzcátegui

Caracas, Mayo de 2016

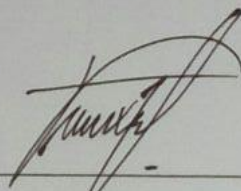
UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE COMPUTACIÓN

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "Desarrollo de una Aplicación de Apoyo a un Gestor de Redes Sociales, para la Captura de Datos desde la Red Social Twitter y su Respaldo en una Base de Datos NoSQL" y presentado por el bachiller: Br. Guillén Omar, C. I. 18003562, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 31 de Mayo de 2016 a las 14:00 horas, para que los autor lo defendiera en forma pública, lo que estos hicieron en la Aula 01 de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 20 puntos.

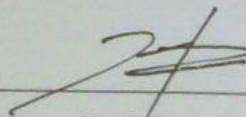
En fe de lo cual se levanta la presente Acta, en Caracas el día 31 de Mayo de 2016.



Prof. Franky Uzcátegui  
(Tutor)



Prof(a). Mercy Ospina  
(Jurado)



Prof. Jesús Lares  
(Jurado)

## **AGRADECIMIENTOS**

Durante toda mi etapa universitaria, conocí personas muy especiales que influyeron directamente en mejorar mi crecimiento personal y profesional. Sin estas personas la elaboración y desarrollo de este trabajo hubiese sido casi imposible de completar.

A mi familia, padres, hermanas, tíos, compañeros de estudio y recreación, les agradezco por todo el apoyo recibido durante estos años y hacer posible mis estudios y tener los principios y valores para poder alcanzar mis objetivos. No suelo decirles gracias, pero su existencia es uno de mis valores más apreciados.

Agradezco en gran medida a mi novia Inés Núñez ya que sin sus guías conceptuales y prácticas, el desarrollo de este documento no hubiese sido completado. También agradezco a la Señorita Verónica Jardim, la cual me dio guías en la implementación de la metodología para el desarrollo de la aplicación.

También doy las gracias a mí jurado, compuesto por la Profesora Mercy Ospina y el Profesor Franklin Sandoval, por su tiempo invertido, dedicación, esfuerzos donde básicamente su única recompensa es tener la satisfacción de participar en la formación de excelentes profesionales para nuestro país.

Y por último y no menos importante, agradezco a mi tutor, el profesor Franky Uzcátegui, el cual sin haber tenido algún tipo de relación académica o de amistad durante la culminación de mí carga académica, me dio un voto de confianza para guiarme y aconsejarme durante toda esta etapa, invirtiendo su tiempo y dedicación tanto para el desarrollo del seminario y el desarrollo de mi Trabajo Especial de Grado. Es un gran honor haber sido tutorado por un gran profesional.

Universidad Central de Venezuela.  
Facultad de Ciencias  
Escuela de Computación  
CISI

**Desarrollo de una Aplicación de Apoyo a un Gestor de Redes Sociales,  
para la Captura de Datos desde la Red Social Twitter y su Respaldo en  
una Base de Datos NoSQL**

**Autor:** Omar Napoleón Guillen Pérez

**Tutor:** Prof. Franky Uzcátegui

**Fecha:** Mayo 2016

**RESUMEN:**

En el presente Trabajo Especial de Grado se exhibieron un conjunto de conceptos, metodologías y herramientas que permitieron llevar a cabo el desarrollo de una solución que le permitiese al Gestor de Redes Sociales el manejo de datos no estructurados de la red social Twitter, lo cual generó información confiable que permitiera a futuro a el área de mercadeo interpretar fortalezas y debilidades de una determinada marca o producto.

**Palabras Claves:** Mercadeo, Twitter, NoSQL, Gestor de Redes Sociales.

---

## ÍNDICE DE CONTENIDO

|                                                                                      |           |
|--------------------------------------------------------------------------------------|-----------|
| <b>INTRODUCCIÓN.....</b>                                                             | <b>10</b> |
| <b>CAPÍTULO 1. PROBLEMA DE INVESTIGACIÓN.....</b>                                    | <b>12</b> |
| 1.1. Antecedentes. ....                                                              | 10        |
| 1.2. Planteamiento del problema. ....                                                | 11        |
| 1.3. Solución.....                                                                   | 13        |
| 1.4. Objetivos del TEG:.....                                                         | 15        |
| 1.4.1. Objetivo general: .....                                                       | 15        |
| <b>1.4.2. Objetivos específicos:.....</b>                                            | <b>15</b> |
| 1.5. Alcance: .....                                                                  | 16        |
| <b>MARCO CONCEPTUAL.....</b>                                                         | <b>17</b> |
| <b>2. Gestor de Redes Sociales (Community Manager):.....</b>                         | <b>17</b> |
| <b>2.1. Definición:.....</b>                                                         | <b>17</b> |
| <b>2.1.1. Departamentos a los que se adscribe el Gestor de Redes Sociales: .....</b> | <b>18</b> |
| <b>2.1.2. Responsabilidades del Gestor de Redes Sociales: .....</b>                  | <b>18</b> |
| 2.2. Proceso de Marketing:.....                                                      | 20        |
| 2.3.3. Modelos de datos a explorar: .....                                            | 23        |
| 2.4. Base de datos: .....                                                            | 26        |
| 2.4.1. Base de Datos Relacional: .....                                               | 26        |
| 2.4.1.1 <i>Componentes de la Base de Datos Relacional: .....</i>                     | <i>28</i> |
| 2.4.1.2 <i>Diseño de una base de datos: .....</i>                                    | <i>30</i> |
| 2.4.1.3 <i>Sistema de gestión de bases de datos Relacionales:.....</i>               | <i>31</i> |
| 2.4.2. Bases de datos No Relacional (NoSQL):.....                                    | 34        |

|                                                                                |           |
|--------------------------------------------------------------------------------|-----------|
| 2.4.2.1. <i>Ventajas de los sistemas No Relacionales (NoSQL):</i>              | 35        |
| 2.4.2.2. Principales diferencias con las bases de datos No Relacionales (SQL): | 36        |
| 2.4.2.3. <i>Tipos de base de datos No Relacional (NoSQL):</i>                  | 37        |
| 2.5. Sistema de información:                                                   | 40        |
| 2.5.1 Definición:                                                              | 40        |
| 2.5.2 Dimensiones de los sistemas de información:                              | 42        |
| 2.5.2.1. <i>Organizaciones:</i>                                                | 43        |
| 2.5.2.2 <i>Administración:</i>                                                 | 44        |
| 2.5.2.3 <i>Tecnología:</i>                                                     | 45        |
| 2.5.3 Importancia del SI:                                                      | 45        |
| 2.7. Tecnologías y su uso:                                                     | 46        |
| 2.7.2 Interfaz de Programación de Aplicaciones de Twitter (API de Twitter):    | 47        |
| 2.7.3. Base de datos NoSQL MongoDB:                                            | 49        |
| 2.7.3.1. <i>Instalación de la Base de datos NoSQL MongoDB:</i>                 | 49        |
| <b>CAPÍTULO 3</b>                                                              | <b>52</b> |
| <b>MARCO MÉTODOLÓGICO</b>                                                      | <b>52</b> |
| 3.1 Metodologías de desarrollo:                                                | 52        |
| 3.1.1 Métodos Tradicionales:                                                   | 52        |
| 3.1.1.1 <i>RUP:</i>                                                            | 52        |
| 3.1.1.2 <i>MSF:</i>                                                            | 53        |
| 3.1.2. Métodos ágiles:                                                         | 55        |
| 3.1.2.1 <i>El manifiesto ágil:</i>                                             | 56        |
| 3.1.2.2 <i>Principios:</i>                                                     | 57        |
| 3.1.2.4 <i>Proceso:</i>                                                        | 59        |
| 3.1.2.5 <i>Roles:</i>                                                          | 61        |

---

|                                                                                 |           |
|---------------------------------------------------------------------------------|-----------|
| 3.1.2.6. Artefactos .....                                                       | 62        |
| 3.1.2.7. Beneficios.....                                                        | 62        |
| <b>CAPITULO 4 .....</b>                                                         | <b>63</b> |
| <b>MARCO APLICATIVO .....</b>                                                   | <b>63</b> |
| 4.1. Plataformas y/o herramientas utilizadas:.....                              | 64        |
| 4.2. Sprint 1. Análisis detallado de requerimientos.....                        | 65        |
| 4.2.1 Historia de usuario.....                                                  | 65        |
| 4.2.2. Diagramas de flujo.....                                                  | 68        |
| 4.2.3. Diseño de la base de datos NoSQL.....                                    | 75        |
| 4.2.4. Diseño de las interfaces. ....                                           | 78        |
| 4.3. Sprint 2. Configuración de ambiente de desarrollo.....                     | 79        |
| 4.3.1. Configuración de la API de Twitter para desarrolladores. ....            | 79        |
| 4.3.2. Configuración de la Base de datos MongoDB.....                           | 80        |
| 4.3.3. Configuración de XAMPP.....                                              | 81        |
| 4.3.3. Configuración de conexión entre MongoDB y PHP. ....                      | 82        |
| 4.3.4. Configuración de conexión entre API de Twitter y PHP.....                | 83        |
| 4.4. Sprint 3. Desarrollo de la aplicación. ....                                | 85        |
| 4.4.1. Desarrollo de las Interfaces de Usuario. ....                            | 85        |
| 4.4.2. Desarrollo de la funcionalidad Identificación de usuario. ....           | 89        |
| 4.4.3. Desarrollo de la funcionalidad Búsqueda en línea.....                    | 89        |
| 4.4.4. Desarrollo de la funcionalidad Búsqueda programada. ....                 | 90        |
| 4.4.5. Desarrollo de la funcionalidad Exportar Tweets. ....                     | 91        |
| 4.4.6. Desarrollo de la funcionalidad Reportes.....                             | 92        |
| 4.4.7. Desarrollo de la funcionalidad Registrar usuario. ....                   | 92        |
| 4.5. Sprint 4. Realización de pruebas para certificación de la aplicación. .... | 93        |

|                                                                                                         |            |
|---------------------------------------------------------------------------------------------------------|------------|
| 4.6. Cronograma de actividades. ....                                                                    | 99         |
| 4.7. Manual de la aplicación.....                                                                       | 100        |
| <b>CONCLUSIONES Y REFLEXIONES DEL TRABAJO ESPECIAL<br/>DE GRADO .....</b>                               | <b>103</b> |
| <b>REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES .....</b>                                                     | <b>105</b> |
| <b>ANEXO A. ....</b>                                                                                    | <b>109</b> |
| <b>Captura de Imagen de <a href="https://apps.twitter.com/">https://apps.twitter.com/</a>.....</b>      | <b>109</b> |
| <b>ANEXO B. ....</b>                                                                                    | <b>111</b> |
| <b>Captura de Pantalla de una Aplicación de la API de Twitter .....</b>                                 | <b>111</b> |
| <b>ANEXO C. ....</b>                                                                                    | <b>113</b> |
| <b>Captura de Pantalla de los Parámetros a Llenar .....</b>                                             | <b>113</b> |
| <b>ANEXO D. ....</b>                                                                                    | <b>115</b> |
| <b>Captura de Pantalla de la Pestaña “Settings” .....</b>                                               | <b>115</b> |
| <b>ANEXO E. ....</b>                                                                                    | <b>117</b> |
| <b>Captura de pantalla de la pestaña “Update settings” .....</b>                                        | <b>117</b> |
| <b>ANEXO F.....</b>                                                                                     | <b>119</b> |
| <b>Captura de Pantalla de la Pestaña “Settings”.....</b>                                                | <b>119</b> |
| <b>ANEXO G. ....</b>                                                                                    | <b>121</b> |
| <b>Captura de Pantalla de la Pestaña “Keys and Acces Tokens” .....</b>                                  | <b>121</b> |
| <b>ANEXO H.....</b>                                                                                     | <b>123</b> |
| <b>Captura de Pantalla de las Credenciales para que Twitter Permita<br/>Utilizar sus Recursos. ....</b> | <b>123</b> |

## ÍNDICE DE FIGURAS

|                                                                                                                        |    |
|------------------------------------------------------------------------------------------------------------------------|----|
| <i>Figura 1.</i> Situación actual de Gestor de redes Sociales (Community Manager). .....                               | 11 |
| <i>Figura 2.</i> Solución general.....                                                                                 | 14 |
| <i>Figura 3.</i> Solución específica. ....                                                                             | 14 |
| <i>Figura 4.</i> Solución planteada para el Community Manager. ....                                                    | 15 |
| <i>Figura 5.</i> Modelo relacional. ....                                                                               | 28 |
| <i>Figura 6.</i> Elementos de una base de datos.....                                                                   | 28 |
| <i>Figura 7.</i> Diseño de una base de datos .....                                                                     | 30 |
| <i>Figura 8.</i> Modelo de base de datos NoSQL clave – valor .....                                                     | 38 |
| <i>Figura 9.</i> Base de dato documental.....                                                                          | 38 |
| <i>Figura 10.</i> Base de datos en grafo. ....                                                                         | 39 |
| <i>Figura 11.</i> Dimensiones relativas a la organización, administración y tecnología de información de sistemas..... | 42 |
| <i>Figura 12.</i> Estructura relativa a la organización, compuesta por diferentes niveles y especialidades.....        | 44 |
| <i>Figura 13.</i> Proceso RUP. ....                                                                                    | 53 |
| <i>Figura 14.</i> Ciclo de vida MSF. ....                                                                              | 54 |
| <i>Figura 15.</i> Proceso SCRUM. ....                                                                                  | 60 |
| <i>Figura 16.</i> Roles en SCRUM.....                                                                                  | 62 |
| <i>Figura 17.</i> Diagrama de flujo asociado a la historia de usuario “autenticación de usuario N° 1”.....             | 69 |
| <i>Figura 18.</i> Diagrama de flujo asociado a historia de usuario “registro de usuario N° 2”.<br>.....                | 70 |
| <i>Figura 19.</i> Diagrama de flujo asociado a historia de usuario “búsqueda en línea N° 3”. 71                        |    |
| <i>Figura 20.</i> Diagrama de flujo asociado a historia de usuario “búsqueda programada N° 4”.....                     | 72 |
| <i>Figura 21.</i> Diagrama de flujo asociado a historia de usuario “exporta usuario N° 5”....                          | 73 |

---

|                                                                                                                                              |    |
|----------------------------------------------------------------------------------------------------------------------------------------------|----|
| <i>Figura 22.</i> Diagrama de flujo asociado a historia de usuario “Reportes” . . . . .                                                      | 74 |
| <i>Figura 23.</i> Estructura del documento perteneciente a la colección “user” . . . . .                                                     | 76 |
| <i>Figura 24.</i> Estructura del documento perteneciente a las colecciones de búsquedas. ....                                                | 76 |
| <i>Figura 25.</i> Estructura del documento perteneciente a la colección “create_campaign” . . . . .                                          | 77 |
| <i>Figura 26.</i> Estructura del documento perteneciente a la colección “report” . . . . .                                                   | 77 |
| <i>Figura 27.</i> Interfaz de autenticación de usuarios. . . . .                                                                             | 78 |
| <i>Figura 28.</i> Diseño de la interfaz Búsqueda en línea. . . . .                                                                           | 78 |
| <i>Figura 29.</i> Diseño de la interfaz del reporte generado por el sistema. . . . .                                                         | 79 |
| <i>Figura 30.</i> Consola de comando de Windows . . . . .                                                                                    | 81 |
| <i>Figura 31.</i> Panel de control de XAMPP. . . . .                                                                                         | 81 |
| <i>Figura 32.</i> Página informativa de las características de PHP. . . . .                                                                  | 82 |
| <i>Figura 33.</i> Archivo de configuración de PHP. . . . .                                                                                   | 82 |
| <i>Figura 34.</i> Página informativa de PHP que muestra la configuración con MongoDB. . . . .                                                | 83 |
| <i>Figura 35.</i> Script PHP para realizar la prueba de conexión con Twitter. . . . .                                                        | 84 |
| <i>Figura 36.</i> Resultado de ejecutar el script de prueba de conexión con Twitter. . . . .                                                 | 84 |
| <i>Figura 37.</i> Interfaz de la funcionalidad de Identificación de usuario. . . . .                                                         | 86 |
| <i>Figura 38.</i> Interfaz de la funcionalidad Búsqueda en línea. . . . .                                                                    | 86 |
| <i>Figura 39.</i> Interfaz de la funcionalidad Búsqueda programada. . . . .                                                                  | 87 |
| <i>Figura 40.</i> Interfaz de la Funcionalidad Exportar Tweets. . . . .                                                                      | 87 |
| <i>Figura 41.</i> Interfaz de la funcionalidad Reportes. . . . .                                                                             | 88 |
| <i>Figura 42.</i> Interfaz de la funcionalidad Registrar usuario. . . . .                                                                    | 88 |
| <i>Figura 43.</i> Código para evaluación y autenticación de un usuario. . . . .                                                              | 89 |
| <i>Figura 44.</i> Extracto de código de la funcionalidad Búsqueda en línea. . . . .                                                          | 90 |
| <i>Figura 45.</i> Histograma de resultados de la funcionalidad de la aplicación . . . . .                                                    | 95 |
| <i>Figura 46:</i> Histograma de resultados del look and feel . . . . .                                                                       | 96 |
| <i>Figura 47.</i> Histograma de resultados de nivel de dificultad de uso de la aplicación y nivel de ayuda que ofrece la aplicación. . . . . | 97 |
| <i>Figura 48.</i> Histograma de resultados para funcionalidad. . . . .                                                                       | 98 |
| <i>Figura 49.</i> Histograma de resultados para el segundo periodo de recolección de información. . . . .                                    | 99 |

---

## ÍNDICE DE TABLAS

|                                                                                                      |     |
|------------------------------------------------------------------------------------------------------|-----|
| Tabla 1. <i>Comparación entre los modelos tipos de datos según su estructura.</i> .....              | 26  |
| Tabla 2. <i>Comparación entre Bases de Datos Relacionales (SQL) y No Relacionales (NoSQL).</i> ..... | 37  |
| Tabla 3. <i>Tabla comparativa de metodologías tradicionales.</i> .....                               | 55  |
| Tabla 4. <i>Comparación entre métodos ágiles y tradicionales.</i> .....                              | 58  |
| Tabla 5. <i>Historia de usuario N° 1.</i> .....                                                      | 65  |
| Tabla 6. <i>Historia de usuario N° 2.</i> .....                                                      | 66  |
| Tabla 7. <i>Historia de usuario N° 3.</i> .....                                                      | 66  |
| Tabla 8. <i>Historia de usuario N° 4.</i> .....                                                      | 67  |
| Tabla 9. <i>Historia de usuario N° 5.</i> .....                                                      | 67  |
| Tabla 10. <i>Historia de usuario N° 6.</i> .....                                                     | 68  |
| Tabla 11. <i>Opciones y valores de escala tipo Likert.</i> .....                                     | 93  |
| Tabla 12. <i>Puntuación máxima y mínima de cada factor.</i> .....                                    | 94  |
| Tabla 13. <i>Cronograma de actividades realizada con la metodología SCRUM.</i> .....                 | 100 |

## **INTRODUCCIÓN**

Debido al gran avance que existe día con día en las tecnologías de información, los Gestores de Redes Sociales se han tenido que enfrentar a nuevos desafíos que les permitan analizar, descubrir y entender de forma objetiva más allá de lo que sus herramientas tradicionales reportan sobre su información obtenida, al mismo tiempo que durante los últimos años el gran crecimiento de las aplicaciones disponibles en internet (redes sociales, etc.) han sido parte importante en las decisiones de negocio de las empresas. El presente Trabajo Especial de Grado, tiene como finalidad el desarrollo de una aplicación que trabaje con la red social Twitter, y que permita la capturar de forma objetiva y poco sesgada de los datos que no son tomados en cuenta de dicha red por tratarse de datos no estructurados y que son relevantes a la hora de realizar estudios de mercadeo para productos, con el fin de relacionar los mismos y obtener un alcance predictivo, que permita conseguir una gran cantidad de beneficios para la empresa y su toma de decisiones relacionadas con el área de mercadeo.

Todo lo previamente dicho, es expuesto en el presente Trabajo Especial de Grado, el cual está conformado por cuatro capítulos en total. En el primero, describe la problemática que presentan los Gestores de Redes Sociales al momento de manejar datos de manera manual. Aunado a ello se señalan los objetivos y alcances de este Trabajo Especial de Grado. El capítulo dos, siendo este el Marco Conceptual, se encuentran conceptos y componentes relacionados al tema, los cuales se utilizan como base teórica para el diseño y construcción de la propuesta. Dicho capítulo contiene desde lo más general a lo más específico, y se exponen definiciones como Gestor de Redes Sociales, Base de Datos, Mercadeo, tipos de datos, entre otros. En el tercer capítulo, el Marco Metodológico se explica a detalle la metodología escogida para la elaboración de la aplicación. Y en el capítulo 4 se describe el desarrollo de la aplicación a través de la metodología escogida y sus diferentes ciclos, en el cual se trabaja a partir de los objetivos específicos la realización integral de una para la captura de datos relevantes de mercadeo desde la Red Social Twitter. Por último se presentan las conclusiones y reflexiones a partir del desarrollo del Trabajo Especial de Grado.

## CAPÍTULO 1

### PROBLEMA DE INVESTIGACIÓN

#### 1.1. Antecedentes.

En la actualidad, el crecimiento de las redes sociales, haciendo énfasis en la red social Twitter, ha hecho que las empresas u organizaciones replanteen sus estrategias de mercadeo, con el fin de obtener una mayor difusión e intentar atraer y mantener a sus clientes. Es así, como surge un nuevo perfil profesional y abre paso a un nuevo cargo dentro de las mismas, siendo este el Community Manager o Gestor de Redes Sociales, el cual se encarga de entender qué estrategia se debe seguir para construir y mantener interacciones a través de redes sociales con los consumidores del producto al que se representa. Asimismo, una organización o empresa que decida contar con el cargo de Gestor de Redes Sociales, es una acción que requiere constancia y que es un trabajo que reclama cambios, ajustes e innovaciones persistentes dentro de sus estructuras, ya que influyen directamente en muchos aspectos, como por ejemplo, desde el tiempo que la empresa quiere dedicar a las diferentes webs, blogs y redes sociales, qué número de personas participarán en la labor, hasta la evolución constante de la forma en como se gestionan los datos obtenidos de las redes sociales para poder lograr los objetivos planteados por la empresa u organización.

Hoy en día, muchos individuos representando la figura del Community Manager dentro de diferentes empresas, tienen como una de sus principales funciones, gestionar de las redes sociales datos relevantes de acuerdo al producto o marca al cual representan, haciendo hincapié en la red social Twitter, ya que la misma posee ciertas características que la convierten en una de las principales herramientas para el Community Manager al momento de realizar su trabajo. Esta gestión en la mayoría de los casos la realiza de manera manual, es decir, primero acceden a la aplicación, para luego a través del buscador de Twitter, localizar una lista de los tweets de su interés. Luego seleccionan una muestra manualmente de acuerdo a su criterio, para posteriormente transcribirlos por lo general en una hoja de cálculo. Allí realizan procesos de ordenamientos, selección

y procesamiento de la data cualitativamente, con el fin de darle sentido a estos datos, es decir, estructurarlos y que se puedan obtener de los mismos, información útil en beneficio de la organización o empresa.

A continuación se muestra en la figura 1, un breve esquema de como gestiona el Community Manager los datos de la red Social Twitter.



Figura 1. Situación actual de Gestor de redes Sociales (Community Manager).

## 1.2. Planteamiento del problema.

Una vez descritos los antecedentes que se plantea dentro de las organizaciones o empresas que poseen la figura del gestor de redes sociales, es posible señalar debilidades

---

al momento de que esta figura realice su trabajo al gestionar los datos de la red social Twitter.

Entre ellos, destaca la necesidad de una herramienta que ayude a la automatización del manejo de los datos que se extraen de Twitter, para facilitar la obtención de información provechosa que pueda ser utilizada por el área de mercadeo.

En este sentido, cuando el Gestor de Redes Sociales hace la selección de muestra de tweets de forma manual, conlleva a los siguientes inconvenientes.

- (a) *Trabajo manual*: las tareas de extraer, seleccionar y procesar datos de la Red Social Twitter puede tener como consecuencia que este trabajo hecho a mano contenga múltiples debilidades, como por ejemplo una gran cantidad de tiempo laboral, alteración en los datos gestionados, no cumplir con los plazos establecidos, etc. Todo esto puede conllevar al no cumplimiento de los objetivos planteados por la empresa u organización debido a toma de decisiones no sustentadas correctamente que fueron basadas en estos datos gestionados anteriormente.
- (b) *Muestra poco representativa*: la cantidad de Tweets tomados de la población en total resulta poca, por ello no es posible determinar de forma precisa las tendencias reales que muestran los clientes en sus tweets. Esta problemática a su vez puede llevar a cometer sesgos. Es decir, la muestra de tweets lleva a una falacia al inferir una conclusión general a partir de una muestra insuficiente. Es decir, se realiza una generalización apresurada que puede ser entendida como una mala inducción.
- (c) *Sobrecarga cognitiva en los gestores de redes sociales*: debido a la falta de automatización en la gestión de data, y a la enorme cantidad de los mismos que tiene que procesar de manera manual por largas horas laborales, esto produce una sobrecarga en los procesos cognitivos del Gestor de Redes

---

Sociales sobre todo en los procesos atencionales y de memoria, lo cual puede conllevar a errores en la información que se maneja.

### **1.3. Solución.**

Después de haber señalado los respectivos problemas anteriormente relacionados con la situación actual planteada, se propuso como solución desarrollar una aplicación que permitiese la automatización de la gestión de los datos (extracción, almacenamiento y estructura de los mismos) generados en la Red Social Twitter que permitiera mejorar, optimizar y simplificar el trabajo de la figura del Community Manager. Dicha solución permitió un mejor manejo de los datos con la meta de que se cumplieran los objetivos del área de mercadeo de la empresa u organización, conllevando una mejor toma de decisiones dentro de las mismas.

Esta solución planteó conectar la red social Twitter a través de un ETL a una base de datos NoSQL, para almacenar los tweets que fueron seleccionados a través de una interfaz donde se introdujeron las palabras claves a buscar. Al introducir estas palabras claves, automáticamente se realizó la búsqueda y almacenamiento de los últimos tweets publicados en la red social.

Luego estos datos almacenados, a través de características propias que poseen las bases de Datos NoSQL, dieron la opción de filtrar, seleccionar, organizar y generar indicadores como lo son marca, zona geográfica, fecha y usuario para su posterior uso. Para finalizar estos datos gestionados fueron exportados a un formato Excel u hoja de cálculo de forma estructurada, para ser utilizados por la empresa u organización de la mejor forma que consideren para su toma de decisiones relacionadas con el área de mercadeo, es decir, pueden llegar a juntar sus datos preexistentes con este nuevo conjunto de datos.

A continuación se muestra la arquitectura de la solución general en la figura 2.

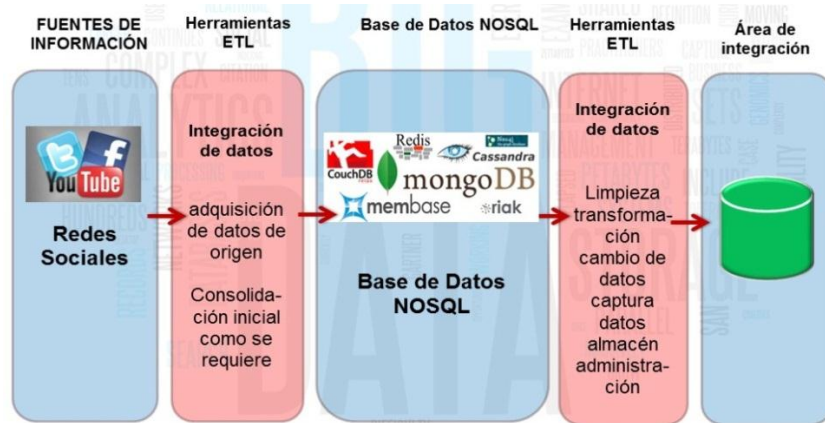


Figura 2.Solución general.

En la figura 3 se puede observar, las tecnologías que se utilizaron según la solución de la arquitectura general señalada en la figura 2. Donde se muestran: la red social Twitter, la base de datos NoSQL MongoDB, la API de Twitter, lenguaje de programación PHP, el servidor Apache XAMPP y el calendarizador System Scheduler .

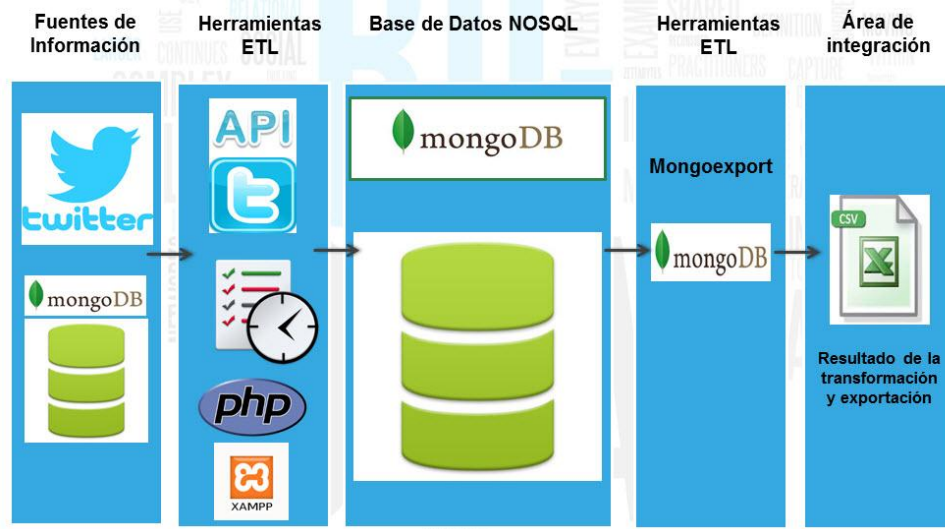


Figura 3.Solución específica.

En la figura 4 se puede observar un pequeño esquema donde se muestra la simplificación, automatización y manejo de la aplicación para el Gestor de Redes Sociales o Community Manager.



Figura 4. Solución planteada para el Community Manager.

## 1.4. Objetivos del Trabajo Especial de Grado:

### 1.4.1. Objetivo general:

Desarrollar una aplicación de apoyo para el gestor de redes sociales que permitiera la captura de datos desde la red social Twitter y su respaldo en una Base de Datos NoSQL

### 1.4.2. Objetivos específicos:

- Realizar un análisis detallado de requerimientos.
- Realizar una configuración de ambiente de desarrollo de la aplicación para la captura de datos de la Red Social Twitter.
- Desarrollar la aplicación para la captura de datos de la Red Social Twitter.
- Realizar pruebas para la certificación de la aplicación para la captura de datos de la Red Social Twitter.

---

## 1.5. Alcance:

Dado la problemática mencionada anteriormente, se propuso desarrollar un sistema que fuese útil dentro del área de mercadeo, que incluyese datos de la Red Social Twitter para posteriormente analizar dicha información en tiempo real y poder usar esto en el mercadeo de productos, cubriendo así las siguientes necesidades:

- (a) Capturar los datos no estructurados de determinado producto o marca de la Red Social Twitter a través de un aplicativo que automatice esta función.
- (b) Gestionar los datos almacenados dentro de una Base de Datos NoSQL (MongoDB).
- (c) Exportar datos de forma estructurada de la Base De Datos NoSQL (MongoDB). La misma estará organizada con los siguientes indicadores: (c.1) fecha, (c.2) ubicación geográfica, (c.3) usuario y (c.4) marca.

## CAPÍTULO 2

### MARCO CONCEPTUAL

#### **2. Gestor de Redes Sociales (Community Manager):**

##### **2.1. Definición:**

Son varias las definiciones que se han utilizado para arrojar luz sobre este perfil. Según Gallego (citado en Laíz, 2010), quien para el momento ejercía el cargo de presidente de la Asociación Española de Responsables de Comunidades Online (AERCO) el Gestor de Redes Sociales o Community Manager es “quien se encarga de cuidar y mantener la comunidad de fieles seguidores que la marca o empresa atraiga, y ser el nexo de unión entre las necesidades de los mismos y las posibilidades de la empresa” (p. 1).

Otro autor, Coghlan (citado en Laíz, 2010) se refiere al Gestor de Redes Sociales como “el arte de la gestión eficiente de la comunicación de otros online en las diferentes herramientas idóneas para el tipo de conversación que se crea conveniente con los potenciales clientes (ya sea un blog, una comunidad a medida, una cuenta en Twitter, una página de fans en Facebook...)” (p. 1). En otras palabras, para Coghlan, el Gestor de Redes Sociales debe ser la voz de la empresa en el contexto exterior, y la voz del cliente dentro de la compañía.

Aunado al concepto, también es importante señalar lo que no es un Gestor de Redes Sociales con el fin de aclarar los roles del mismo. Así, AERCO y Territorio Creativo (2009) en su publicación indican que un Community Manager, no es un empleado del departamento de comunicación y marketing, que mantiene un blog, abre una página en Facebook, o actualiza el estado de Twitter, sino más bien, dicho perfil incluye entre sus responsabilidades entender qué estrategia se debe seguir para construir relaciones alrededor de la marca y mantener interacciones personales con los miembros de la comunidad de la marca a la que se representa.

---

Por último, la AERCO define Community Manager como aquella persona encargada o responsable de sostener, acrecentar y, en cierta forma, defender las relaciones de la empresa con sus clientes en el ámbito digital, gracias al conocimiento de las necesidades y los planteamientos estratégicos de la organización y los intereses de los clientes. Una persona que conoce los objetivos y actuar en consecuencia para conseguirlos (AERCO y Territorio Creativo, 2009).

### **2.1.1. Departamentos a los que se adscribe el Gestor de Redes Sociales:**

Según AERCO y Territorio Creativo (2009), hoy en día no existe una convención a este respecto. El Community Manager suele estar adscrito al departamento más innovador de la empresa, aquel que toma la iniciativa en el uso de las redes sociales. Lo habitual es que el Community Manager pertenezca a los departamentos de comunicación o marketing, aunque en algunos casos dependen de tecnología o de innovación.

Una organización mediana o grande, debería empezar a replantearse su organización desde la base, y pensar que la gestión de los medios sociales comienza a ser una función en sí misma. De esta forma, el Community Manager podría llegar a tener un puesto de staff dependiendo de un director general. Por ello mencionan que hasta que llegue ese momento, parece razonable que los Community Manager se sitúen en la primera área, puesto que sus funciones y responsabilidades están ligadas con las tareas de las relaciones públicas y los directores de comunicación. Se analiza el mapa de públicos a los que dirigirse, objetivo, estrategia, mensaje, canal y acción a desarrollar, sólo que el canal se ha ampliado a internet. También cabe destacar que el director de marketing debería estar igualmente muy cercano a este perfil, independientemente de que fuese de su departamento o de otro.

### **2.1.2. Responsabilidades del Gestor de Redes Sociales:**

Como ya se mencionó, el papel del Gestor de Redes Sociales va mucho más allá de transmitirle a una audiencia las bondades de determinada empresa o servicio.

---

De acuerdo con AERCO y Territorio Creativo (2009), la tarea del Gestor de Redes Sociales está en establecer una relación de confianza con la comunidad de usuarios o simpatizantes de la marca, recoger el feedback de los mismos y utilizarlo para proponer mejoras internas.

Las responsabilidades del Gestor de Redes Sociales pueden resumirse en cinco tareas, siendo las siguientes:

- (a) *Escuchar*: Monitorizar constantemente la red en busca de comentarios sobre la empresa, sus competidores o su mercado.
- (b) *Circular esta información internamente*: A raíz de esta escucha, debe ser capaz de extraer lo relevante de la misma y dejar de un lado lo irrelevante, crear un discurso entendible y hacérselo llegar a las personas correspondientes dentro de la organización.
- (c) *Explicar la posición de la empresa a la comunidad*: El Community Manager es la voz de la empresa hacia la comunidad, una voz positiva y abierta que transforma la “jerga interna” de la compañía en un lenguaje inteligible. Responde y conversa activamente en todos los medios sociales en los que la empresa tenga presencia activa (perfil) o en los que se produzcan menciones relevantes. Escribe artículos en el blog de la empresa o en otros medios sociales, usando todas las posibilidades multimedia a su alcance. Y selecciona y comparte además contenidos de interés para la comunidad.
- (d) *Buscar líderes, tanto interna como externamente*: La relación entre la comunidad y la empresa está sustentada en la labor de sus líderes y personas de alto potencial. El Community Manager debe ser capaz de identificar y “reclutar” a estos líderes, no sólo entre la comunidad sino, y sobre todo, dentro de la propia empresa.
- (e) *Encontrar vías de colaboración entre la comunidad y la empresa*: La mayoría de directivos desconoce cómo la comunidad puede ayudar a hacer crecer su

empresa. No es algo que hayan utilizado nunca en su carrera, ni que hayan estudiado en las escuelas de negocios. El Community Manager les debe ayudar a diseñar una estrategia clara de colaboración.

De estas cinco funciones, hay dos que resultan imprescindibles, siendo estas la primera y la segunda. Lo más probable es que no contar con una monitorización activa de las conversaciones en internet conlleve hacia el fracaso. Pero tan importante es escuchar como saber transmitir lo aprendido a las personas apropiadas y en el formato apropiado, con un asesoramiento adecuado sobre las medidas a tomar. Cuando esto no sucede, y esta monitorización se deja en manos de personas sin conocimiento y experiencia en comunicación 2.0 suele llevar a la sobre reacción ante cualquier tipo de crítica en internet. Esta reacción excesiva ante la crítica lo único que hace es justificar y amplificar estas críticas. Cuando no hay un Community Manager para asesorar al respecto, las empresas cometen el error de reaccionar ante una crítica en un blog con acciones tan peculiares (y reales) como denunciar al que ha puesto el comentario, amenazarle, escribir comentarios falsos defendiendo a la empresa por parte de supuestos consumidores, y una larga serie de conductas no apropiadas (AERCO y Territorio Creativo, 2009).

Respecto a la tarea tercera, cuarta y quinta se trata de un camino que están emprendiendo con gran éxito las empresas más innovadoras del mundo, sean grandes o pequeñas.

## **2.2. Proceso de Marketing:**

De acuerdo con Ayala (2013), el Marketing o mercadotecnia es la disciplina que se dedica al análisis del comportamiento de los mercados y de los consumidores. Analiza la gestión comercial de las empresas con el objeto de captar, retener y fidelizar a los clientes a través de la satisfacción de sus necesidades.

El proceso de marketing es fundamental para cualquier empresa, pretende tener un conocimiento de las oportunidades que ofrece el entorno, se identifica el segmento

del mercado, se formulan estrategias y planes, además de ejecutar un control de los mismos.

Según Ayala (2013), los diferentes autores de marketing y los especialistas de mercadotecnia en la práctica pueden seguir un proceso muy singular que se adapte a un contexto específico; a pesar de esto hay una estructura básica que por lo general se encuentra en todas las organizaciones y que se resume en las siguientes fases del proceso:

- (a) *Análisis de oportunidades del mercadeo*: Esta fase se realiza por medio de una investigación, pretende descubrir una o varias necesidades latentes en el mercado, la percepción que la gente tiene acerca de la marca y de los competidores, los deseos insatisfechos que giran en torno al producto, las tendencias, etc. Es importante recalcar que la investigación por si misma debe tener un objetivo y es preciso se determine cuál es la información que se necesita obtener y su propósito, para evitar un mal uso de recursos económicos, informáticos y humanos en una investigación cuyos datos no sean de utilidad.
  
- (b) *Selección del mercado objetivo*: En base a la información mediante la investigación de mercados se determina cual es el perfil del consumidor, conociendo al cliente se puede ofrecer el producto o servicio que satisfaga sus necesidades y le permita percibir asociaciones positivas de la marca.
  
- (c) *Análisis del mercado*: en este punto se puede implementar un análisis FODA, es decir, se identifican respectivamente las fortalezas y debilidades de la organización (factores internos), de la misma forma que se reconocen cuáles son las oportunidades u amenazas que ofrece el mercado (Factores externos).
  
- (d) *Formulación de estrategias de marketing*: Una estrategia legítima incluye a todos los elementos de la mezcla de marketing, es decir, productos, precio plaza y promoción. Dichas estrategias van orientadas a satisfacer óptimamente al consumidor, basándose en la información encontrada en las fases anteriores.

---

(e) *Implementación de las estrategias de marketing*: Esta etapa puede variar según la estrategia lo permita, lo importante es definir las actividades que se realizan para ejecutar las estrategias, una agenda de donde, cuando y las metas que se pretenden alcanzar debe definirse. También se debe tener un registro que respalde el control y evaluación

Si se desea ser consecuente con las estrategias seleccionadas, tendrá que elaborarse un plan de acción para conseguir los objetivos propuestos en el plazo determinado. Cualquier objetivo se puede alcanzar desde la aplicación de distintos supuestos estratégicos y cada uno de ellos exige la aplicación de una serie de tácticas. Estas tácticas definen las acciones concretas que se deben poner en práctica para poder conseguir los efectos de la estrategia. Ello implica necesariamente el disponer de los recursos humanos, técnicos y económicos, capaces de llevar a buen término el plan de marketing.

Se puede afirmar que el objetivo del marketing es el punto de llegada, la estrategia o estrategias seleccionadas son el camino a seguir para poder alcanzar el objetivo u objetivos establecidos, y las tácticas son los pasos que hay que dar para recorrer el camino.

Es muy importante señalar que las tácticas se lleven a cabo deben ser consecuentes tanto con la estrategia de marketing a la que debe apoyar como con los recursos comerciales de los que dispone la empresa en el periodo de tiempo establecido.

(f) *Control y evaluación*: El control es el último requisito exigible a un plan de marketing. Permite saber el grado de cumplimiento de los objetivos a medida que se van aplicando las estrategias y tácticas definidas. A través de éste se pretende detectar los posibles fallos y desviaciones a tenor de las consecuencias que estos

vayan generando para poder aplicar soluciones y medidas correctoras de manera inmediata.

De no establecerse estos mecanismos de control, se tendría que esperar a que terminara el ejercicio y ver entonces si el objetivo marcado se ha alcanzado o no. En este último caso, sería demasiado tarde para reaccionar. Así pues, los mecanismos de control permiten conocer las realizaciones parciales del objetivo en periodos relativamente cortos de tiempo, por lo que la capacidad de reaccionar es casi inmediata.

### **2.3. Modelos de datos a explorar:**

De acuerdo con Camargo-Vega, Camargo-Ortega y Joyanes (2014), los *datos* son la mínima unidad semántica, y se corresponden con elementos primarios de información que por sí solos son irrelevantes como apoyo a la toma de decisiones. También pueden definirse como un conjunto discreto de valores, que por sí solos no dicen nada.

Según Barranco (2012), dichos datos no dependen tanto sobre la información que se debe analizar, sino más bien a qué problema es el que se está tratando de resolver.

Si bien existe una amplia variedad de modelos de datos a analizar, una buena clasificación siempre ayuda a entender mejor su representación, aunque según el mismo autor, es muy probable que estas categorías puedan extenderse con el avance tecnológico. Los modelos de datos pueden clasificarse según Elósegui (2014) como:

(a) Smart Data: Son todos los datos referentes al negocio (tanto online, como offline). Se pueden encontrar desde las cifras de ventas, datos sobre los clientes, datos sobre las actividades online, etc. Es decir, todo aquello que esté relacionado con la consecución de los objetivos de la empresa.

(b) Identity Data: Son todos los datos que permiten identificar a los clientes actuales y potenciales: datos sobre gustos de clientes, historial de compras, perfil de internauta,

tipo de interacción con los contenidos de la empresa (web, redes sociales, blog, mobile, etc.).

Toda esta información permite personalizar la oferta de producto, contenido y canal de comunicación al máximo.

(c) Open data: Agrupa al resto de datos externos a la empresa y que son accesibles por todo el mundo. Por su volumen y diversidad, nos será más difícil sacarles partido. Pero una vez encontrada la fuente de datos (e integrada), los beneficios que se pueden sacar de esta información son enormes.

Por otra parte, Camargo-Vega, Camargo-Ortega y Joyanes (2014), hacen referencia a el modelo de datos clasificados de acuerdo con su estructura, los cuales se clasifican en estructurados, semiestructurados y no estructurados. En este sentido, *los estructurados* a su vez se dividen en estáticos (array, cadena de caracteres y registros) y dinámicos (listas, pilas, colas, árboles, archivos). Se puede definir que los *datos estructurados* son aquellos de mayor facilidad para acceder, pues tienen una estructura bien especificada. Un array es una colección finita de elementos en formatos definidos del mismo tipo, es decir, son homogéneos, y ordenados por un índice; con estos formatos se facilita la administración de los datos; ejemplo de ellos, un campo que contiene una fecha DD, MM, AA, que contiene seis caracteres, o un formato con la dirección de la persona, que puede ser alfanumérico, con tamaño de 40 caracteres. En segundo lugar, los *datos semiestructurados* son aquellos que no tienen un formato definido. Lo que tienen son etiquetas que facilitan separar un dato de otro. Un dato de estos se lee con un conjunto de reglas de cierto nivel de complejidad y presentan las siguientes características:

- (a) Son datos irregulares, que pueden no tener un esquema en particular
- (b) se pueden presentar datos incompletos
- (c) Los componentes de este tipo de datos, pueden cambiar de tipo
- (d) pueden aparecer datos nuevos cuya estructura nada tiene que ver con la ya existente

---

Por último, *los datos no estructurados* son aquellos que no pueden ser normalizados, no tienen tipos definidos ni están organizados bajo algún patrón; tampoco son almacenados de manera relacional, o con base jerárquica de datos, debido a que no son un tipo de dato predefinido; es decir, no tienen un formato normalizado determinado. Sin embargo, los datos deben poder ser organizados, clasificados, almacenados, eliminados, buscados de alguna forma. Estos datos se pueden observar a diario en correos electrónicos, archivos de texto, un documento de algún procesador de palabra, hojas electrónicas, una imagen, un objeto, archivos de audio, blogs, mensajes de correo de voz, mensajes instantáneos, contenidos Web y archivos de video, entre otros.

En este caso de datos no estructurados, no tienen un identificador definido, no se puede reconocer su estado físico ni lógico; tampoco se puede identificar su tipo o clase; su tamaño no se puede encajar en una tabla predefinida, es el caso de los datos contenidos en una página web. Se puede tener el siguiente ejemplo: "Pedro nació el día 24 de noviembre de 1978, y el 20 de septiembre se graduó Julio en la universidad". Como se puede observar, no es tan fácil la administración de este tipo de información, no estructurada.

Lo que sí se puede respecto a los datos no estructurados es hacer uso de los metadatos, es decir, usar datos que puedan describir otros datos. Por ejemplo, en una biblioteca se tiene en fichas o en un sistema de información datos de los libros como: autor, título, editorial, ISBN y tema, entre otros. Lo anterior con el fin de hallar con facilidad un determinado libro; esta es la forma como los metadatos ayudan a buscar datos.

A continuación se muestra en la tabla 1 una comparación de los modelos de datos de acuerdo con su estructura.

Tabla 1. Comparación entre los modelos tipos de datos según su estructura.

| Modelos de Datos  | Formato                                                                                | Complejidad                                      | Características                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Estructurado      | Tienen una estructura bien especificada.                                               | Mayor facilidad para acceder.                    | Son homogéneos.                                                                                                                                                    |
| Semi-estructurado | No tienen un formato definido, Tienen etiquetas que facilitan separar un dato de otro. | Cierto nivel de complejidad para manejo de dato. | Pueden no tener un esquema en particular. Se pueden presentar datos incompletos. Pueden aparecer datos nuevos cuya estructura nada tiene que ver con la existente. |
| No estructurado   | No tienen estructura definida.                                                         | Nivel alto de dificultad para manejo de datos.   | No están organizados bajo algún patrón.                                                                                                                            |

## 2.4. Base de datos:

### 2.4.1. Base de Datos Relacional:

Según Cruz (s/f) una *base de datos* (BD) es una colección de archivos relacionados que permite el manejo de la información. Cada uno de dichos archivos puede ser visto como una colección de registros y cada registro está compuesto de una colección de campos. Cada uno de los campos de cada registro permite llevar información de algún atributo de una entidad del mundo real, por ejemplo, para una *universidad*, las entidades de interés podrían ser: *alumnos, profesores, salones, etc.* y para la entidad *alumno*, los atributos importantes podrían ser: *nombre, dirección, teléfono, carrera, matricula, etc.*

Un archivo de una base de datos también puede ser pensado como una tabla en la que se tienen renglones y columnas, cada renglón correspondiendo a un registro del archivo y cada columna correspondiendo a un campo (Cruz s/f).

En relación a ello, también es importante señalar que una BD también puede percibirse como un gran almacén de datos que se define y se crea una sola vez, y que se

---

utiliza al mismo tiempo por distintos usuarios. En una BD, todos los datos se integran con una mínima cantidad de duplicidad, por ello se dice que una finalidad de la BD es eliminar la redundancia o al menos minimizarla. De este modo, la BD no pertenece a un solo departamento de la organización sino que se comparte por toda empresa. Además, la misma no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción es lo que se denomina *metadatos*, se almacena en el diccionario de datos o catálogo y es lo que permite que exista independencia de datos lógica-física (Marqués, 2011).

El modelo relacional, desarrollado por Codd (de IBM) en 1970. Haciendo una analogía, el modelo relacional produjo una BD de *transmisión automática* que reemplazó a las BD de *transmisión estándar* que las precedieron. Su simplicidad conceptual preparó el camino para una revolución en el campo de las bases de datos (Rob y Coronel, 2006).

El modelo relacional se apoya en claves e índices para poder relacionar las tablas, adicionalmente permite relaciones *uno a uno*, *uno a muchos* y *muchos a mucho*, y el elemento a relacionar ya deja de llamarse nodo para convertirse en entidad, por lo que a los modelos que representan bases de datos relacionales se les suele llamar modelo *entidad-relación* (ver figura 5).

En este modelo, la información puede ser recuperada y almacenada mediante consultas que ofrecen una amplia flexibilidad para administrar la información. El lenguaje más habitual para construir estas consultas es SQL (Structured Query Language), el cual es el estándar más implementado por los principales motores o sistemas manejadores de base de datos relacionales.

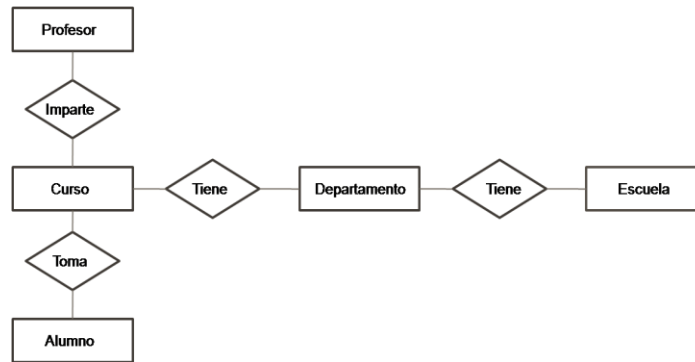


Figura 5. Modelo relacional. Tomado de <http://antoniovf.wordpress.com/2012/04/23/modelos-de-bases-de-datos/>

2.4.1.1 Componentes de la Base de Datos Relacional:

De acuerdo Peña (2006), una BD relacional está conformada por los elementos presentados en la figura 6, los cuales son descritos a continuación.

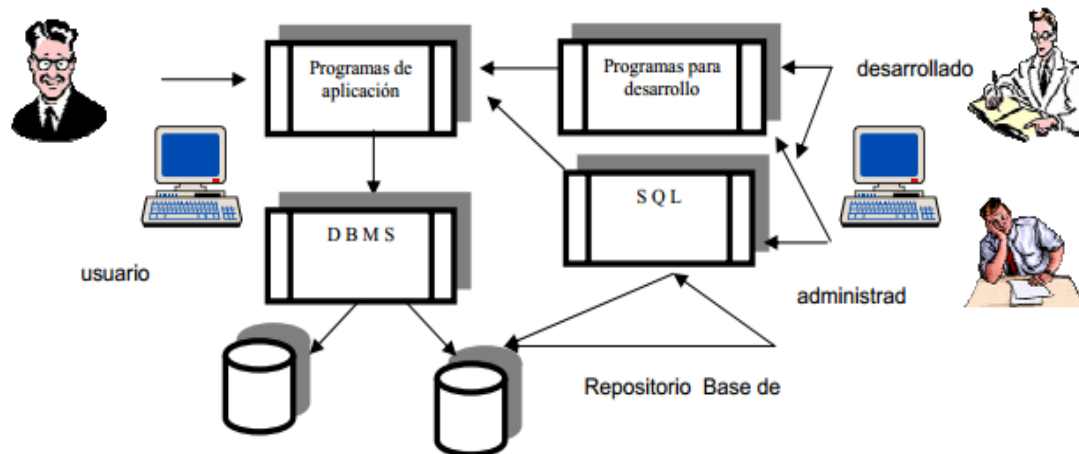


Figura 6. Elementos de una base de datos. Tomado de Peña (2006).

- 
- (a) *Sistema manejador de base de datos (DBMS)*: Es el programa que permite la definición y construcción de los elementos (tablas, reglas y procedimientos.) de la BD, con el fin de controlar el ingreso, almacenamiento, actualización, integridad y recuperación de la información.
  
  - (b) *Repositorio*: Son las definiciones de base de datos, tablas, tipos de dato, consultas, ventanas, reglas, valores por omisión “default”, procedimientos, reportes y otras definiciones que establecen la naturaleza del sistema y base de datos del usuario.
  
  - (c) *Lenguaje estructurado de consulta (SQL)*: Es un programa orientado a crear, administrar y explotar la base de datos, por medio de un lenguaje estándar equivalente al inglés que se puede usar en cualquier manejador de base de datos.
  
  - (d) *Programas para desarrollo de aplicaciones*: Son programas que facilitan la creación, prueba y mantenimiento de procesos de consulta, cálculo y explotación de la base de datos.
  
  - (e) *Programas de aplicación*: Son los procedimientos creados para servir de interface entre el usuario y la base de datos para introducir, validar, actualizar y explotar la información, ejecutar procesos de cálculo, conversión, exportación, replicación y administración de datos, los cuales emplean instrucciones de SQL y programas para desarrollo de aplicaciones.
  
  - (f) *Administrador*: Crea, mantiene y administra la BD, supervisa su operación y empleo de recursos, establece y aplica las políticas de acceso, seguridad e integridad en el uso de datos a cargo de los usuarios. También vigilia el rendimiento y tiempo de respuesta del sistema.
  
  - (g) *Desarrollador*: Es el personal técnico encargado de crear los programas para operar la BD.

- (h) *Usuario*: Son los interesados en introducir, actualizar y consultar los datos, conforme a las políticas establecidas por el administrador, utilizando los programas de aplicación.

#### 2.4.1.2. Diseño de una base de datos:

De acuerdo a Batini, Ceri y Navathe (1994), el diseño de una base de datos es un proceso complejo que abarca decisiones a distintos niveles. La complejidad se controla mejor si se descompone el problema en sub-problemas y se resuelve cada uno de manera independiente, utilizando diversas técnicas específicas. Así, el diseño de una base de datos se puede descomponer en diseño conceptual, diseño lógico y diseño físico tal como lo muestra la figura 7.

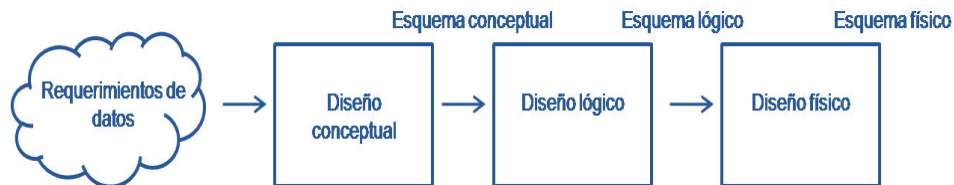


Figura 7. Diseño de una base de datos

- (a) El diseño conceptual parte de las especificaciones de requisitos de usuario, produciendo como resultado el esquema conceptual de la BD. Un esquema conceptual es una descripción de alto nivel de la estructura de BD (Batini et al, 1994). Se basa entonces en describir a la organización de la misma en un muy alto nivel, sin importar el sistema manejador de base de datos que se piense implementar para su manipulación.

El objetivo principal de este diseño es describir el contenido de información de la base de datos y no las estructuras que se usaran para almacenar y manejar esta información (Batini et al, 1994).

- (b) El *diseño lógico* parte del esquema conceptual y da como resultado un esquema lógico. Un esquema lógico es una descripción de la organización de la base de datos en términos de las estructuras de datos que puede procesar un tipo de

---

sistema manejador de base de datos (Batini et al, 1994). Se puede decir que el modelo lógico es el idioma para representar los esquemas lógicos.

A diferencia del diseño conceptual, el diseño lógico depende del tipo de modelo que implemente el sistema manejador de base de datos que se pretenda utilizar.

- (c) El *diseño físico* parte del esquema lógico y da como resultado un esquema físico, que vendría siendo una descripción de la implementación de una base de datos en memoria secundaria; las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos (Batini et al, 1994). Por ello, el diseño físico depende del sistema manejador de base de datos, en el cual se definen las entidades como tablas, los atributos como campos de las tablas y los elementos como los datos de las tablas.

Es importante destacar que existe una unión entre el diseño físico y lógico, ya que cualquier cambio implementado en el diseño físico afecta la estructura del esquema lógico.

#### 2.4.1.3. Sistema de gestión de bases de datos Relacionales:

De acuerdo con Marqués (2011), para implementar los diferentes modelos de datos explicados anteriormente, es necesaria una herramienta de software que ejerza el papel de interfaz entre el usuario, los modelos y el sistema físico. Esta es la función que ejercen los sistemas de gestión de base de datos (SGBD). Estos sistemas tienen la capacidad de crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma. Se denomina sistema de bases de datos al conjunto formado por la base de datos, el SGBD y los programas de aplicación que dan servicio a la empresa u organización.

Para Marqués (2011), el modelo seguido con los sistemas de bases de datos es muy similar al modelo que se sigue en la actualidad para el desarrollo de programas con lenguajes orientados a objetos, en donde se da una implementación interna de un objeto y una especificación externa separada. Los usuarios del objeto sólo ven la especificación externa y no se deben preocupar de cómo se implementa internamente el objeto. Una

---

ventaja de este modelo, conocido como abstracción de datos, es que se puede cambiar la implementación interna de un objeto sin afectar a sus usuarios ya que la especificación externa no se ve alterada. Del mismo modo, los sistemas de bases de datos separan la definición de la estructura física de los datos de su estructura lógica, y almacenan esta definición en la BD. Todo esto es gracias a la existencia del SGBD, que se sitúa entre la base de datos y los programas de aplicación.

Generalmente, un SGBD proporciona los servicios que se citan a continuación:

- (a) El SGBD permite la definición de la BD mediante un lenguaje de definición de datos. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos.
- (b) El SGBD permite la inserción, actualización, eliminación y consulta de datos mediante un lenguaje de manejo de datos. Hay dos tipos de lenguajes de manejo de datos los procedurales y los no procedurales. Estos dos tipos se distinguen por el modo en que acceden a los datos. Los lenguajes procedurales manipulan la base de datos registro a registro, mientras que los no procedurales operan sobre conjuntos de registros. En los lenguajes procedurales se especifica qué operaciones se debe realizar para obtener los datos resultados, mientras que en los lenguajes no procedurales se especifica qué datos deben obtenerse sin decir cómo hacerlo. El lenguaje no procedural más utilizado es el SQL (Structured Query Language) que, de hecho, es un estándar y es el lenguaje de los SGBD relacionales.

Según Marqués (2011), el SGBD proporciona un acceso controlado a la base de datos mediante:

- (a) Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos
- (b) Un sistema de integridad que mantiene la integridad y la consistencia de los datos
- (c) Un sistema de control de concurrencia que permite el acceso compartido a la base de datos.

- (d) Un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del hardware o del software.
- (e) Un diccionario de datos o catálogo, accesible por el usuario, que contiene la descripción de los datos de la base de datos.

De acuerdo con Marqués (2011), el SGBD se ocupa de la estructura física de los datos y de su almacenamiento. Con esta funcionalidad, el SGBD se convierte en una herramienta de gran utilidad. Sin embargo, desde el punto de vista del usuario, se podría discutir que los SGBD han hecho las cosas más complicadas, ya que ahora los usuarios ven más datos de los que realmente quieren o necesitan, puesto que ven la BD completa. Conscientes de este problema, los SGBD proporcionan un mecanismo de vistas que permite que cada usuario tenga su propia vista o visión de la BD. El lenguaje de definición de datos permite delimitar vistas como subconjuntos de las funciones que se acaban de citar e incluso más. Los sistemas modernos son conjuntos de programas complejos y sofisticados, con millones de líneas de código y con una documentación consistente en varios volúmenes. Lo que se pretende es proporcionar un sistema que permita gestionar cualquier tipo de requisitos y que tenga un 100% de confiabilidad ante cualquier tipo de fallo base de datos.

A continuación se presentan algunos sistemas manejadores de base de datos que están en el mercado actualmente.

- (a) *PostgreSQL* es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Berkeley Software Distribution) y con su código fuente disponible libremente. Al ser *software libre*, el desarrollo de este sistema es dirigido por una comunidad de desarrolladores apoyados por organizaciones comerciales.

Al ser un sistema objeto-relacional incluye las características de la orientación a objetos. Tiene soporte completo para claves primarias, foráneas, vistas, disparadores y procedimientos. Incluye la mayoría de los tipos de datos comunes (integer, boolean, varchar, date, entre otros) y también soporta almacenamiento de datos complejos como imágenes, sonidos y videos.

---

Además, permite desarrollar procedimientos en diferentes lenguajes de programación como Java, C++, Ruby y Python, lo que hace a PostgreSQL altamente personalizable. La última versión del producto en el mercado es la versión 9.1 y ocupa aproximadamente 50 Mbyte.

- (b) *Oracle* es un sistema manejador de base orientada en el modelo objeto-relacional, por lo cual adopta muchas características de PostgreSQL. Destaca por su entorno cliente/servidor, manejo de multiusuario, soporte de transacciones, estabilidad, escalabilidad y soporte de multiplataforma.

A diferencia de PostgreSQL, Oracle no es *software libre*, por lo que ofrece una edición gratuita destinada a proyectos académicos y de investigación pero con algunas limitaciones en su funcionalidad, y la versión de pago orientada a medianas y grandes empresas. Se puede considerar hoy en día que Oracle es la base de datos líder en el mercado y actualmente su última versión es la 11g.

- (c) *MySQL* es un sistema manejador de base de datos basado en el modelo relacional que fue desarrollado por la empresa Sun Microsystem pero paso a ser propiedad de Oracle mediante una operación de venta en el año 2009 que osciló entre los 5 millones de euros.

El software de base de datos MySQL consiste en un sistema cliente/servidor que se compone de un servidor SQL multi-hilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs).

Una de las características de MySQL es que es muy utilizado en aplicaciones web combinado con el lenguaje PHP debido a la facilidad de enlace entre ambos. MySQL ofrece una versión gratuita y una de pago.

#### **2.4.2. Bases de datos No Relacional (NoSQL):**

Según Ambriz (2013) hoy en día no existe una definición universalmente aceptada de NoSQL. En general, se refiere a aquellas bases de datos con modelos de consistencia más laxos que los establecidos por las bases de datos relacionales, con el fin

de tener una mejor escalabilidad al trabajar con altos volúmenes de información cuando la naturaleza de los datos no requiere un modelo relacional. Aunado a ello, otro autor, Acens (2014), menciona que dicho término apareció con la llegada de la web 2.0 ya que hasta ese momento sólo subían contenido a la red aquellas empresas que tenían un portal, pero con la llegada de aplicaciones como Facebook, Twitter o Youtube, cualquier usuario puede ser capaz de subir contenido, provocando así un crecimiento exponencial de los datos.

Por ello, con la llegada de la web 2.0, empiezan a aparecer los primeros problemas de la gestión de toda esa información almacenada en bases de datos relacionales. En un principio, para solucionar dicha problemática de accesibilidad, las compañías eligieron utilizar un mayor número de máquinas pero se dieron cuenta de que esto no solucionaba el problema, además de resultar en una inversión costosa. Una segunda solución era crear de sistemas pensados para un uso específico que con el paso del tiempo han dado lugar a soluciones robustas, apareciendo así el movimiento NoSQL. Es por esto que hablar de bases de datos NoSQL es hablar de estructuras que permiten almacenar información en aquellas situaciones en las que las bases de datos relacionales generan ciertos problemas debido principalmente a problemas de escalabilidad y rendimiento de las bases de datos relacionales donde se dan cita miles de usuarios concurrentes y con millones de consultas diarias.

Además de lo comentado previamente, las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Tampoco utilizan una estructura de datos en forma de tabla donde se van almacenando los datos sino que para el almacenamiento hacen uso de otros formatos como clave-valor, mapeo de columnas o grafos.

#### *2.4.2.1. Ventajas de los sistemas No Relacionales (NoSQL):*

Acens (2014), menciona que esta forma de almacenar la información ofrece ciertas ventajas sobre los modelos relacionales. Entre las ventajas se puede destacar:

- (a) Se ejecutan en máquinas con pocos recursos. Estos sistemas, a diferencia de los sistemas basados en SQL, no requieren de apenas computación, por lo que se pueden montar en máquinas de un coste menor.
- (b) Escalabilidad horizontal. Para mejorar el rendimiento de estos sistemas simplemente se consigue añadiendo más nodos, con la única operación de indicar al sistema cuáles son los nodos que están disponibles.
- (c) Pueden manejar gran cantidad de datos ya que utiliza una estructura distribuida, en muchos casos mediante tablas Hash.
- (d) No genera cuellos de botella. El principal problema de los sistemas SQL es que necesitan transcribir cada sentencia para poder ser ejecutada, y cada sentencia compleja requiere además de un nivel de ejecución aún más complejo, lo que constituye un punto de entrada en común, que ante muchas peticiones puede ralentizar el sistema.

#### 2.4.2.2. Principales diferencias con las bases de datos No Relacionales (SQL):

Algunas de las diferencias más destacables según Acens (2014) en los sistemas NoSQL y los sistemas SQL están:

- (a) No utilizan SQL como lenguaje de consultas. La mayoría de las bases de datos NoSQL evitan utilizar este tipo de lenguaje o lo utilizan como un lenguaje de apoyo. Por poner algunos ejemplos, Cassandra utiliza el lenguaje CQL, MongoDB utiliza JSON o BigTable hace uso de GQL.
- (b) No utilizan estructuras fijas como tablas para el almacenamiento de los datos. Permiten hacer uso de otros tipos de modelos de almacenamiento de información como sistemas de clave–valor, objetos o grafos.
- (c) No suelen permitir operaciones JOIN. Al disponer de un volumen de datos tan extremadamente grande suele resultar deseable evitar los JOIN. Esto se debe a que, cuando la operación no es la búsqueda de una clave, la sobrecarga puede llegar a ser muy costosa. Las soluciones más directas consisten en desnormalizar los datos, o bien realizar el JOIN mediante software, en la capa de aplicación.
- (d) Arquitectura distribuida. Las bases de datos relacionales suelen estar centralizadas en una única máquina o bien en una estructura máster–esclavo, sin

embargo en los casos NoSQL la información puede estar compartida en varias máquinas mediante mecanismos de tablas Hash distribuidas.

A continuación se presenta una tabla donde se pueden apreciar mejor las diferencias entre bases de datos SQL y NoSQL

Tabla 2. Comparación entre Bases de Datos Relacionales (SQL) y No Relacionales (NoSQL).

| Tipos de Base de Datos | Tipos de Modelos                                               | Forma de consulta                                                                            | Ventajas                                                                                                                               | Desventajas                                                                                               | Ejemplos                                   |
|------------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|--------------------------------------------|
| Relacionales (SQL)     | Relacional (Relación entre Tablas, claves primarias, foráneas) | Lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) | .Garantiza Herramientas para evitar duplicar registros.<br>Garantiza la integridad referencial.<br>Corto periodo de aprendizaje.       | . Costosa creación y mantenimiento del sistema<br>.Deficiencias con datos gráficos y multimedia           | Oracle, SQL Server, My SQL, PostgreSQL.    |
| NoSQL (Not Only SQL)   | Orientado a documentos. Orientadas a columnas. De clave valor. | Java Script (en la mayoría de los casos).                                                    | No hay redundancia.<br>Más liviana o ligera.<br>Mayor Velocidad de consulta.<br>Mayor escalabilidad.<br>No requieren estructuras fijas | No permiten Join's a cambio de velocidad.<br>Falta de experiencia.<br>Pocas Normas en común o Estándares. | Mongo DB, Casandra, Redis, CouchDB, Hbase. |

#### 2.4.2.3. Tipos de base de datos No Relacional (NoSQL):

Dependiendo de la forma en la que almacenen la información, es posible encontrar varios tipos distintos de bases de datos NoSQL. A continuación los tipos más usados.

- (a) *Bases de datos clave – valor:* Se puede observar en la figura 8. Son el modelo de base de datos NoSQL más popular, además de ser la más sencilla en cuanto a funcionalidad. En este tipo de sistema, cada elemento está identificado por una llave única, lo que permite la recuperación de la información de forma muy rápida, información que habitualmente está almacenada como un objeto binario (BLOB).

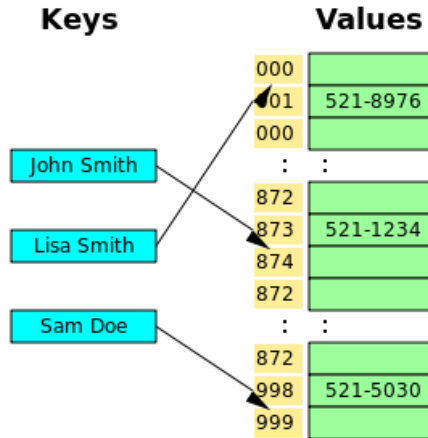


Figura 8. Modelo de base de datos NoSQL clave – valor

Se caracterizan por ser muy eficientes tanto para las lecturas como para las escrituras. Algunos ejemplos de este tipo son Cassandra, BigTable o HBase.

(b) *Bases de datos documentales*: Tal como se observa en la figura 9, este tipo de base de datos almacena la información como un documento, generalmente utilizando para ello una estructura simple como JSON o XML y donde se utiliza una clave única para cada registro. Este tipo de implementación permite, además de realizar búsquedas por clave–valor, realizar consultas más avanzadas sobre el contenido del documento.

| CLAVE | DOCUMENTO                                                                 |
|-------|---------------------------------------------------------------------------|
| 63    | Nombre: Esteban; Apellidos: Saavedra; Nacionalidad; Boliviana             |
| 64    | Nombre: Francisco; Apellidos: Torrez; Nacionalidad: Portuguesa; Edad: 39; |
| 65    | Nombre: Jorge; Nacionalidad: Chilena                                      |
| 66    | Nombre: Eduardo; Nacionalidad: Ecuatoriana                                |
| 67    | Nombre: Pedro; Nacionalidad: Peruana                                      |
| 68    | Nombre: Jose; Nacionalidad: colombiana; Edad: 44;                         |

Figura 9. Base de dato documental.

Son las bases de datos NoSQL más versátiles. Se pueden utilizar en gran cantidad de proyectos, incluyendo muchos que tradicionalmente funcionarían sobre bases de datos relacionales. Algunos ejemplos de este tipo son MongoDB o CouchDB.

- (c) *Bases de datos en grafo*: En este tipo de bases de datos, la información se representa como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se puede hacer uso de la teoría de grafos para recorrerla. Para sacar el máximo rendimiento a este tipo de bases de datos, su estructura debe ser totalmente normalizada, de forma que cada tabla tenga una sola columna y cada relación dos.

Este tipo de bases de datos ofrece una navegación más eficiente entre relaciones que en un modelo relacional. Algunos ejemplos de este tipo son Neo4j, InfoGrid o Virtuoso.

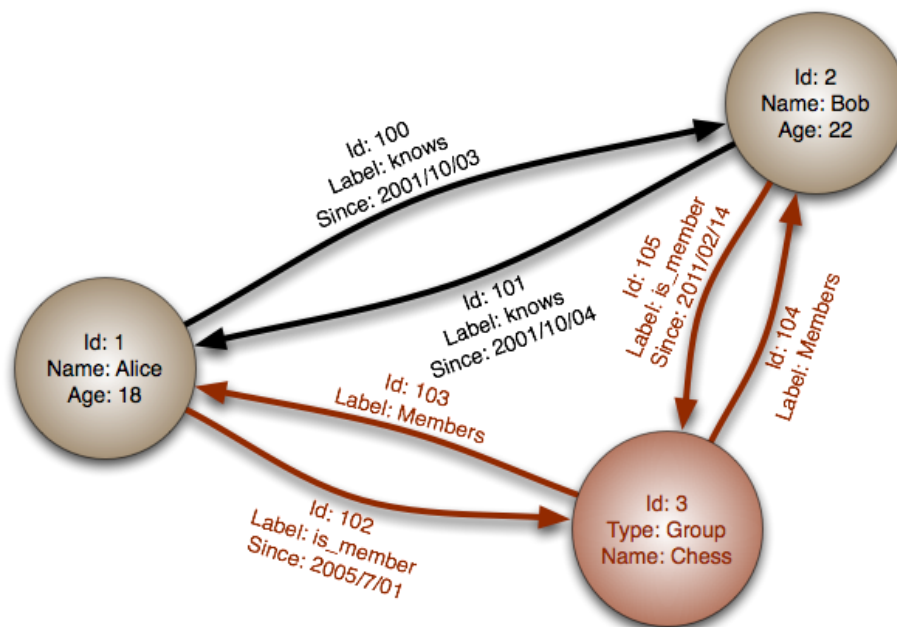


Figura 10. Base de datos en grafo.

(d) *Bases de datos orientadas a objetos*: En este tipo, la información se representa mediante objetos, de la misma forma que son representados en los lenguajes de programación orientada a objetos (POO) como ocurre en JAVA, C# o Visual Basic.NET. Algunos ejemplos de este tipo de bases de datos son Zope, Gemstone o Db4o.

## **2.5. Sistema de información:**

### **2.5.1 Definición:**

Según Colomina (1998), los Sistemas de Información (abreviados como SI) surgen en los años sesenta, en los Estados Unidos, dentro de los departamentos, y orientados a disciplinas informáticas (ciencias de la computación, ingeniería del software, etc.), para posteriormente generar otros grupos enmarcados en las escuelas de negocios con un bagaje de conocimientos diferente a los que inicialmente le habían asignado en las ciencias de computación. Esta diversidad de componentes es lo que hoy en día explica un rasgo característico de la disciplina, a saber, la carencia de teorías bien desarrolladas que generen consenso. Es más, se carece en general de un vocabulario aceptado. La terminología utilizada oscila entre expresiones comunes de la calle y; así, no es difícil incluso encontrar disputas entre investigadores sobre el significado exacto de la expresión "sistema de información", la dificultad para generar conceptos unánimes en Sistemas de Información deriva del entorno contextual siempre cambiante sobre el que especula (la tecnología) y, desde luego, el corto periodo de vida de los Sistemas de Información. (Colomina, 1998).

Laudon y Laudon (2008), define desde el punto de vista técnico a los Sistemas de información como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en un organización. Además los sistemas de información también

---

pueden ayudar a los gerentes y los trabajadores a analizar problemas, visualizar asuntos complejos y crear nuevos productos.

Los sistemas de información contienen información acerca de las personas, lugares y cosas importantes dentro de la organización o del entorno en que se desenvuelven. Por *información*, se entienden los datos que se han moldeado en una forma significativa y útil para los seres humanos. En contraste, los *datos* son secuencias de hechos en bruto que representan eventos que ocurren en las organizaciones o en el entorno físico antes de ser organizados y ordenados en una forma que las personas puedan entender y utilizar de manera efectiva (Laudon y Laudon 2008).

De acuerdo con Stair y Reynolds (2000) hay cuatro actividades en un sistema de información que producen la información que esas empresas requieren para tomar decisiones, controlar operaciones, analizar problemas y crea nuevos productos o servicios. Estas actividades son entradas, procesamiento, salida y retroalimentación de los cuales, dichos autores señalan que:

La *entrada*, captura o recolecta datos en bruto tanto de la organización como de su entorno externo.

El *procesamiento* convierte esta entrada de datos en una forma significativa.

La *salida* transfiere la información procesada a las personas que la usan o a las actividades para las que se utilizará.

Los sistemas de información también requieren *retroalimentación*, que es la salida que se devuelve al personal correspondiente de la empresa para ayudarle a evaluar o corregir la etapa de entrada.

En relación a ello, Laudon y Laudon (2008), señalan que, aunque los sistemas de información basados en computadora utilizan tecnología de cómputo para procesar datos en bruto y conseguir información significativa, hay una diferencia entre una computadora común y un programa de cómputo por una parte, y un sistema de información por otra. Las computadoras electrónicas y los programas de software relacionados constituyen la base técnica, las herramientas y materia prima, de los

modernos sistemas de información. Las computadoras proveen el equipo para almacenar y procesar la información. Los programas de cómputo, o software, con conjuntos de instrucciones funcionales que dirigen y controlan el procesamiento por computadora. Saber cómo funcionan las computadoras y los programas de cómputo es importante para diseñar soluciones a problemas de la organización, pero las computadoras son solo parte de un sistema de información.

### 2.5.2 Dimensiones de los sistemas de información:

De acuerdo con Laudon y Laudon (2008), para poder hablar de los sistemas de información, es necesario comprender que las dimensiones más amplias relativas a la organización, administración y tecnología de información de sistemas y su potencial para dar soluciones a retos y problemas del entorno de negocios (figura 11). Esta comprensión más amplia de los sistemas de información, que abarcan el entendimiento de las dimensiones administrativas, técnicas y organizacionales de los sistemas, se denomina conocimiento de los sistemas de información. Este tipo de conocimiento incluye un enfoque tanto conductual como técnico para el estudio de los sistemas de información. En contraste, el conocimiento de computación se enfoca sobre todo en el conocimiento de la tecnología de información.



Figura 11. Dimensiones relativas a la organización, administración y tecnología de información de sistemas. Tomado de Laudon y Laudon (2008).

---

El *campo de los sistemas de información gerencial* (o sus siglas en inglés MIS) se encarga de proporcionar este conocimiento más amplio de los sistemas de información. Los MIS abordan aspectos tanto conductuales como técnicos relacionados con el desarrollo, uso e impacto de los sistemas de información que utilizan los gerentes y los empleados de la empresa.

#### 2.5.2.1. Organizaciones:

Según Rangel (2009), los sistemas de información forman parte integral de las organizaciones. De hecho, algunas compañías como las empresas de investigación de crédito no podrían existir sin usar un sistema de información. Los elementos claves de una organización son sus empleados, estructura, procedimientos operativos, políticas y cultura.

Aunado a ello, las organizaciones tienen una estructura compuesta de diferentes niveles y especialidades, sus estructuras reflejan una clara división del trabajo. La autoridad y responsabilidad en una empresa se organizan como un jerarquía, o estructura de pirámide, de autoridad y responsabilidades ascendentes. Los niveles más altos de la jerarquía están conformados por los empleados administrativos, profesionales y técnicos, en tanto que los inferiores corresponden al personal operativo.

Para Laudon y Laudon (2008), la *alta dirección* toma las decisiones estratégicas más importantes relacionadas con productos y servicios y garantiza el desempeño financiero de la empresa, la *gerencia intermedia* pone en práctica los programas y planes de la alta dirección, la *gerencia de operaciones* es responsable de supervisar las actividades cotidianas de la empresa. *Los trabajadores de conocimiento*, como ingenieros, científicos o arquitectos, diseñan productos o servicios y crean nuevo conocimiento para la empresa, en tanto que los trabajadores de datos, como secretarías y oficinistas colaboran con el trabajo de oficina en todos los niveles de la empresa. *Los trabajadores de producción o servicio* son quienes en realidad elaboran el producto y proporcionan el servicio. Tal como se puede ver en la figura 12.



*Figura 12.* Estructura relativa a la organización, compuesta por diferentes niveles y especialidades. Tomado de Laudon y Laudon (2008).

Según los mismos autores, una organización coordina el trabajo por medio de sus jerarquías y sus procesos de negocios, de los cuales, la mayoría incluyen reglas formales que se han desarrollado con el paso del tiempo para la realización de las tareas. Estas reglas sirven para guía a los empleados en una diversidad de procedimientos.

Cada empresa tiene una cultura (organizacional) particular, a la cual se le podría denominar conjunto fundamental de supuestos valores y formas de hacer cosas, que ha sido aceptada por la mayoría de sus integrantes.

Los diversos niveles y especialidades de una organización crean intereses y puntos de vista diferentes. Con frecuencia, estos puntos de vista entran en conflicto sobre la manera en que debe dirigirse la compañía y como deben repartirse los recursos y las compensaciones, el conflicto es la base de las políticas de una organización. Los sistemas de información surgen de este caldero de perspectivas distintas, conflictos, compromisos y acuerdos que sean parte natural de todas las organizaciones (Laudon y Laudon, 2008).

#### *2.5.2.2 Administración:*

De acuerdo con Laudon y Laudon (2008), el trabajo de la administración es darle sentido a las diversas situaciones que enfrentan las organizaciones, toma de decisiones y formular planes de acción para resolver problemas de la organización.

---

Sin embargo, los gerentes deben hacer algo más que solo administrar lo que ya existe. También deben crear nuevos productos y servicios e incluso volver a crear la organización si lo amerita la situación. Es por ello que una parte importante de la responsabilidad de la administración es el trabajo creativo impulsado por el nuevo conocimiento y la nueva información. La tecnología de información puede jugar un rol preponderante para ayudar a los gerentes a diseñar y ofrecer nuevos productos y servicios y para rediseñar y cambiar el rumbo de las organizaciones.

### *2.5.2.3 Tecnología:*

Laudon y Laudon (2008) mencionan que la tecnología de información es una de las muchas herramientas que los gerentes utilizan para enfrentarse al cambio. Aunado a ello los autores hablan sobre (a) el hardware de cómputo, que vendría siendo el equipo físico utilizado para realizar las actividades de entrada, procesamiento y salida de un sistema de información, (b) el software de cómputo consiste en instrucciones detalladas, programadas por anticipado, que controlan y coordinan los componentes del hardware, (c) la tecnología de administración de datos consiste en el software que controla la organización de los datos en medios de almacenamiento físico.

### **2.5.3 Importancia del SI:**

La importancia de los sistemas de información la manifiestan Stair y Reynolds (1999), quienes señalan que la capacitación en sistemas de computación e información es un prerrequisito para obtener numerosas oportunidades de empleo, y no solo del campo SI. Los sistemas de computación pueden ejercer en profundo impacto en la estrategia corporativa y el éxito organizacional. De hecho señalan que “compañías del mundo entero disfrutaban en la actualidad de mejor seguridad y servicio, mayor eficiencia y eficacia, costos más bajos y mejor control y toma de decisiones gracias a los sistemas de información”. (p33)

Considerando la importancia que le atribuyen los autores a los SI, resulta evidente que el uso de un buen SI, beneficiaría la organización en varios aspectos, tales como mayor seguridad, servicio, eficiencia y control, estos aspectos favorecen sustancialmente el desarrollo de los procesos internos de las organizaciones.

Entre los objetivos del SI que se pueden encontrar, sobresalen (a) dar soporte a los objetivos y estrategias de la empresa, es decir que debe proveer toda la información necesaria para el funcionamiento del negocio en un momento dado, ya sea esta información referida a la actividad diaria directamente productiva, como a la planificación de la empresa a largo plazo (b) proporcionar a todos los niveles de la empresa la información necesaria para controlar las actividades de la misma, lo que indica que la información debe proporcionar no solo la ejecución de funciones o tareas, sino el control y verificación de que estas se han realizado como estaba previsto, (c) conseguir que se adapte a la evolución de la empresa, lo cual significa que debe ser un sistema capaz de evolucionar al ritmo de la empresa, pues las empresas cada vez son más dinámicas, y sus necesidades de información van cambiando a lo largo del tiempo, (d) utilizar la información como un recurso corporativo que debería ser planificado, gestionado y controlado para ser más efectivo a toda la organización, es decir que es sumamente útil en una empresa considerar la Información como otro recurso cualquiera, esto permite racionalizar y optimizar el uso de información, y por último, (d) definir la evolución del Sistema de Información actual hacia el SI necesario, por lo cual se entiende que en muy pocos casos se desarrolla un sistema de información antes de iniciar su actividad una empresa, por ello, casi siempre existe una realidad en información cuando se plantea la reorganización o puesta al día del mismo.

## **2.6. Tecnologías y su uso:**

### **2.6.1 Twitter:**

Es un servicio de microblogging. Dicha red permite enviar mensajes de texto plano de corta longitud, con un máximo de 140 caracteres, llamados tweets, que se muestran en la página principal del usuario. Los usuarios pueden suscribirse a los tweets de otros usuarios (lo cual se conoce como "seguir" y a los usuarios abonados se les da el nombre de "seguidores"). Los mensajes enviados son públicos, pudiendo difundirse privadamente mostrándolos únicamente a unos seguidores determinados. Los usuarios pueden twittear desde la web del servicio, con aplicaciones oficiales externas (como para teléfonos inteligentes), o mediante el Servicio de mensajes cortos (SMS) disponible

---

en ciertos países. Si bien el servicio es gratuito, su uso vía SMS soporta tarifas fijadas por el proveedor de telefonía móvil (Support Twitter, 2014).

### **2.6.2 Interfaz de Programación de Aplicaciones de Twitter (API de Twitter):**

Dicha interfaz, abreviada API es definida como una "llave de acceso" a funciones que permiten hacer uso de un servicio web provisto por un tercero, dentro de una aplicación web propia, de manera segura (Maquera, 2012).

A través de las API de Twitter cualquiera puede crear aplicaciones que comuniquen con el servicio de la mencionada red social.

Según Álvarez, García y Álvarez (2012), el uso de las APIs de Twitter está limitado, por lo que las aplicaciones no pueden conectarse un número indeterminado de veces para solicitar cualquier cosa. Sin embargo, los límites son más o menos aceptables para páginas personales, proyectos pequeños y medianos (además, siempre se puede cachear los resultados para no tener que solicitar lo mismo muchas veces seguidas). En el caso que se desee construir sistemas que hagan un uso intensivo del API de Twitter, estaría la posibilidad de registrar la aplicación. Los límites de acceso al API sin registro son 150 solicitudes por hora, mientras que para aplicaciones registradas en la "whitelist" podrían llegarse a hacer 20.000 solicitudes por hora.

Ahora bien, las solicitudes a veces se cuentan dependiendo de la IP del sistema que conecta con Twitter y a veces dependiendo de la cuenta de usuario Twitter que está solicitando un recurso, por lo que estos límites podrían ser un poco mayores si se tiene uno o varios usuarios Twitter. Hay también otros límites de uso del API, basados en los límites de paginación de las búsquedas que se pueden realizar, es decir, que limitan los resultados de búsquedas de estatus de un usuario o las referencias de una palabra clave en los post públicos.

Otro punto importante es que, cuando se comunica con el API de Twitter, se tiene que trabajar en UTF-8 y enviar cualquier parámetro codificado con formato de

---

URL. Estos formatos y juegos de caracteres se pueden conseguir fácilmente con diversas funciones (Álvarez, García y Álvarez, 2012).

El API de Twitter funciona por HTTP, accediendo a URLs que devuelven datos, todo por el protocolo HTTP. Para facilitar la solicitud de URLs por parte de un cliente y procesar las respuestas formatos del servidor a esas solicitudes, que devuelven contenidos en formatos distintos, principalmente en formato *JSON*.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos (Aguilera, Veléz y Echeverría, 2012).

JSON está constituido por dos estructuras:

- (a) Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un *objeto*, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- (b) Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

### **2.6.3. Base de datos NoSQL MongoDB:**

Se trata de un sistema de base de datos NoSQL, desarrollado bajo el concepto de código abierto y orientado a documentos. Forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema, haciendo que la integración de los datos sea más fácil y rápida (MongoDB Inc., 2015).

#### *2.6.3.1. Instalación de la Base de datos NoSQL MongoDB:*

Para esto, primero se debe acceder a la página oficial de MongoDB “<https://www.mongodb.org/>” y descargar el paquete instalador según el sistema operativo en el cual se desea instalar. En el caso del sistema operativo Windows, se descarga el paquete instalador con extensión “.msi”. Se realiza doble clic en el archivo “.msi”. Posterior a eso aparece un conjunto de pantallas encargadas de guiar en el proceso de instalación (MongoDB Inc., 2015).

El uso de MongoDB es considerado una de las mejores formas de almacenar datos no estructurados en tiempo real a través de una conexión creada en lenguaje Python de una red social, en este caso Twitter.

### **2.6.4. Lenguaje de programación PHP:**

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. También incluye una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

### 2.6.5. Framework Bootstrap

Bootstrap es un framework basado en HTML y CSS, fue creado por Twitter y lo liberó en 2012, desde entonces ha ganado muchos adeptos, hasta el nivel que cuenta con la mayor comunidad de Github del mundo.

Este framework ayuda a agilizar la creación del interfaz de las páginas web. Con la particularidad, que usando Bootstrap, el sitio estará adaptado a la pantalla de los dispositivos con el que se acceda, ya sea un ordenador, tablet, smartphone, televisión, etc. Esto significa que permite obtener una Web responsive o adaptativa.

Además de agilizar la creación de páginas Web, Bootstrap permite crear un diseño limpio, intuitivo, usable y de poco peso, por lo que la carga de las web son muy rápida. Es muy cómodo, porque muchas de las funcionalidades que necesitan ya están desarrolladas, y si no, tienes acceso a una gran cantidad de documentación en varios idiomas y una comunidad que dará respuestas a todas tus dudas.

### 2.6.6. Xampp:

Xampp es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

Solamente requiere descargar y ejecutar un archivo ZIP, tar, exe o fkl, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de

---

Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin.

---

## CAPÍTULO 3

### MARCO MÉTODOLÓGICO

#### 3.1 Metodologías de desarrollo:

##### 3.1.1 Métodos Tradicionales:

Las metodologías tradicionales según Uñoja (2012) son un conjunto de métodos para el desarrollo de software enfocado en documentación, planificación y procesos, imponen una disciplina de trabajo sobre el proceso de desarrollo del software, para realizar un producto eficiente. Es muy importante la planificación total de todo el trabajo a realiza para luego comenzar el ciclo de desarrollo del producto. Se centran especialmente en el control del proceso, actividades, artefactos, herramientas y notaciones para el modelado y una documentación detallada.

Entre las metodologías tradicionales más conocidas y usadas existe el Proceso Racional Unificado (RUP), en inglés *Rational Unified Process* y la metodología de Microsoft el Framework de Solución Microsoft (MSF) en inglés *Microsoft Solution Framework*, que se describirán a continuación.

##### 3.1.1.1 RUP:

Es un proceso formal que provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo, se divide en cuatro fases, concepción, elaboración, construcción y transición como se puede ver en la figura 13. (Uñoja, 2012).

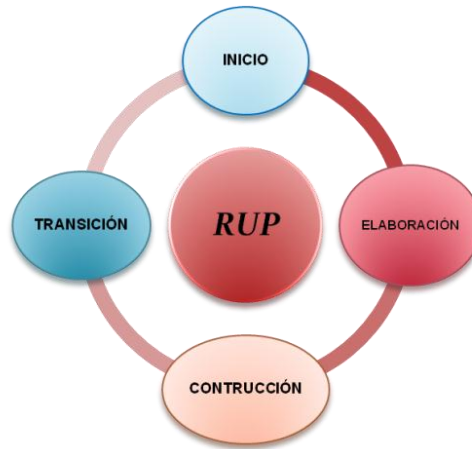


Figura 13. Proceso RUP. Tomado de Uñoja, 2012.

Entre los aportes que ofrece esta metodología se tiene que:

- La evaluación en cada fase permite cambios de objetivos.
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Se tiene un seguimiento detallado en cada una de las fases.

Y entre las limitaciones:

- La evaluación de riesgos es compleja.
- Existe una excesiva flexibilidad para algunos proyectos.
- Se pone al cliente en una situación que puede ser muy incómoda.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto.

### 3.1.1.2 MSF:

Es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos,

MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información, se divide en cinco fases principales como se puede ver en la figura 14, visión y alcances, planificación, desarrollo, estabilización e implantación. (Uñoja, 2012).



Figura 14. Ciclo de vida MSF. Tomado de Uñoja, 2012.

Se basa principalmente en los modelos espiral y cascada. Fue desarrollado con el objetivo de crear un modelo estructurado basado en una estructura de trabajo en desarrollo de software y tiene como principios fundamentales la comunicación entre cliente/usuario y los desarrolladores, un proceso de formación de personal y compartir los roles entre todo el equipo de trabajo.

En la tabla 3 se puede observar una comparación de las metodologías tradicionales explicadas anteriormente.

Tabla 3. *Tabla comparativa de metodologías tradicionales.*

|                    | RUP                                                                                                                                          | MSF                                                                         |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Proceso            | Mucho uso de UML, tiene un enfoque disciplinado, se entrega el ejecutable en el tiempo determinado, resuelve de forma proactiva los cambios. | No se hace uso de UML, abarca los riesgos, se desarrolla en componentes     |
| Tiempos de entrega | En el tiempo estipulado por el cliente y el equipo de desarrollo.                                                                            | Asegura calidad, por ello puede tardarse un poco más del tiempo estipulado. |
| Artefactos         | Entregables de UML, todas las técnicas, metodologías, casos de uso, diagramas de clases y otros                                              | No tiene mucha documentación                                                |

### 3.1.2. Métodos ágiles:

Según Letelier y Sánchez (2013), dichas metodologías son un conjunto de diversos métodos para desarrollar software basándose en el desarrollo iterativo e incremental, donde colabora todo un equipo auto-organizado y multidisciplinario destinado a un proyecto. Enfatizan las comunicaciones cara a cara y que el software funcional sea la primera medida del progreso.

Fueron pensadas especialmente para equipos de desarrollo pequeños, con plazos reducidos, requisitos volátiles y nuevas tecnologías. La filosofía de las metodologías ágiles pretende dar mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil, la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo pero con una alta calidad.

Algunas características que una metodología ágil aporta a un proyecto son:

- (a) Se hace repetitivo e incremental.
- (b) Se realiza una mitigación del riesgo mediante iteraciones fijas.
- (c) Se obtiene una mejora continua y calidad en el producto desde el primer día.

- (d) La prioridad de los requerimientos se realiza de acuerdo a su valor.
- (e) Los equipos son dedicados y auto-gestionados.
- (f) Existe una colaboración continua con el cliente.
- (g) Se realizan prácticas de desarrollo modernas.

### *3.1.2.1 El manifiesto ágil:*

De acuerdo con Letelier y Sánchez (2013), el manifiesto comienza enumerando los principales valores del desarrollo ágil, según el mismo, se valora al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado, sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. Es decir, es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente cuando es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

Desarrollar software que funciona más que conseguir una buena documentación. Aunque se parte de la base de que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo.

La colaboración con el cliente más que la negociación de un contrato. Las características particulares del desarrollo de software hacen que muchos proyectos hayan fracasado por intentar cumplir unos plazos y unos costes preestablecidos al inicio del mismo, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo.

---

Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta ya que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.

### *3.1.2.2 Principios:*

Los valores mencionados en el punto anterior inspiraron a crear los doce principios que se muestran a continuación (Beck, et al., 2001).

- (a) La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- (b) Se acepta que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- (c) Se entregan software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- (d) Los responsables de negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto.
- (e) Los proyectos se desarrollan en torno a individuos motivados. Se les debe dar el entorno y el apoyo que necesiten, y confiarles la ejecución del trabajo.
- (f) El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- (g) El software funcionando es la medida principal de progreso.
- (h) Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
- (i) La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.

- (j) La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- (k) Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- (l) A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Estos principios buscan eliminar los obstáculos que se puedan interponer en la realización del proyecto, integrar a todo el equipo en la realización del mismo, mejorar la comunicación, acelerar los tiempos de entrega y aumentar la calidad del proyecto.

A continuación se presentara en la tabla 4 muestra las diferencias entre métodos ágiles y tradicionales

Tabla 4. *Comparación entre métodos ágiles y tradicionales.*

| Metodologías Ágiles                                           | Metodologías Tradicionales                                             |
|---------------------------------------------------------------|------------------------------------------------------------------------|
| Especialmente preparados para cambios durante el proyecto     | Cierta resistencia a los cambios                                       |
| El cliente es parte del equipo de desarrollo                  | El cliente interactúa con el equipo de desarrollo mediante reuniones   |
| Grupos pequeños (<10 personas) y trabajando en el mismo sitio | Grupos grandes y posiblemente distribuidos                             |
| Menos énfasis en la arquitectura del software                 | La arquitectura del software es esencial y se expresa mediante modelos |

### 3.1.2.3 SCRUM:

A pesar de que existe el manifiesto y esto hace que las metodologías ágiles estén unificadas, existen muchas, y cada una tiene características propias, entre ellas resalta la metodología *SCRUM*.

*SCRUM* define un marco para la gestión de proyectos. Esta especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas Sprint, con una duración de 30 días. El resultado de cada Sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. Al igual que toda metodología, *SCRUM* depende del trabajo arduo y en conjunto de cada una de las partes que conforman el equipo de trabajo. (Orjuela y Rojas, 2008).

### 3.1.2.4 Proceso:

Como ya se mencionó, el desarrollo se realiza de forma iterativa e incremental. Para detallar el ciclo de vida se introducirán los siguientes términos explicados según Softeng (2014):

Requisitos Generales (*Product Backlog*): Conjunto de requisitos denominados historias descritos en un lenguaje no técnico y priorizados por valor de negocio. Los requisitos y prioridades se revisan y ajustan durante el curso del proyecto a intervalos regulares.

Plan del ciclo (*Sprint Planning*): Reunión durante la cual el propietario del producto (Product Owner) presenta las historias de los requisitos generales por orden de prioridad. El equipo determina la cantidad de historias a las que puede comprometerse a completar en ese ciclo, para en una segunda parte de la reunión, decidir y organizar cómo lo va a conseguir.

Ciclo (*Sprint*): Iteración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas. En cada nuevo ciclo, se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizándose siempre aquellas que aporten mayor valor de negocio. Las historias de los requisitos generales, se convierten en una nueva versión del software totalmente operativo.

Requisitos del ciclo (*Sprint Backlog*): Lista de las tareas necesarias para llevar a cabo las historias del ciclo.

Reunión diaria del ciclo (*Daily sprint meeting*): Reunión diaria de cómo máximo 15 minutos en la que el equipo se sincroniza para trabajar de forma coordinada. Cada miembro comenta que hizo el día anterior, que hará hoy y si hay impedimentos.

Demostración y retrospectiva: Reunión al final del ciclo y en la que el equipo presenta las historias conseguidas mediante una demostración del producto. Posteriormente, en la retrospectiva, el equipo analiza qué se hizo bien, qué procesos serían mejorables y discute acerca de cómo perfeccionarlos.

En la figura 15 se puede detallar el proceso de *SCRUM*.



Figura 15. Proceso SCRUM. Tomado de Softeng, 2014.

### 3.1.2.5 Roles:

En *SCRUM* el equipo se focaliza en construir software de calidad. La gestión de un proyecto se centra en definir cuáles son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en vencer cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo.

Según Softeng (2014), el equipo en una metodología *SCRUM* está formado por los siguientes roles (Ver figura 16):

- (a) *SCRUM master*: Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el *Product Owner* para maximizar la satisfacción del usuario.
- (b) *SCRUM owner*: Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y es responsable de la inversión del proyecto. Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el *Product Backlog* y las prioriza de forma regular.
- (c) *Team*: Grupo de profesionales con los conocimientos técnicos necesarios que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada *sprint*.

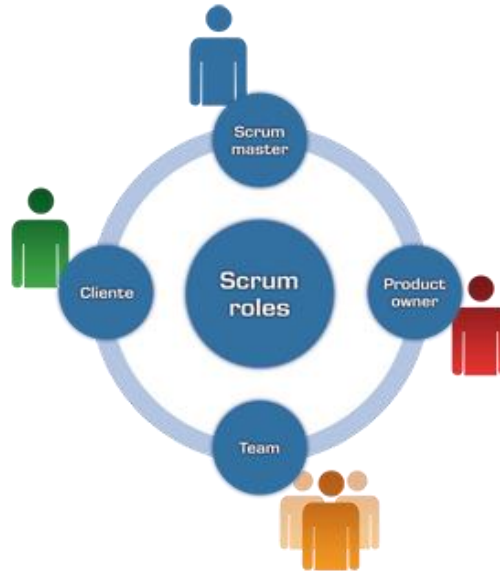


Figura 16. Roles en SCRUM. Tomado de Softeng, 2014.

#### 3.1.2.6. Artefactos

Hay dos artefactos en esta metodología según lo explicado anteriormente, el *Product Backlog* que es una lista ordenada de los requerimientos que se requieren de los productos, es creada por el *Product Owner* y el *Sprint Backlog*, que es un conjunto de elementos tomados del *Product Backlog* conjuntamente con un plan de entrega de incrementos, este *Backlog* es elaborado por el Equipo de Desarrollo (*Team*).

#### 3.1.2.7. Beneficios

- (a) Entrega en un tiempo establecido de los resultados.
- (b) Gestión regular de las expectativas del cliente que se basa en resultados tangibles.
- (c) Flexibilidad y adaptación.
- (d) Mitigación sistemática de los riesgos.
- (e) Productividad y calidad.
- (f) Alineamiento entre el cliente y el desarrollador.
- (g) Motivación del desarrollador.

## CAPITULO 4

### MARCO APLICATIVO

Hoy en día, el crecimiento de las redes sociales, en especial de Twitter, ha hecho surgir un nuevo perfil profesional dentro de las empresas, siendo este el Gestor de Redes Sociales, el cual, como ya se mencionó, se encarga de extraer de las redes sociales, por ejemplo Twitter, datos relevantes de acuerdo al producto o marca al cual representan. Este procesamiento suele ser problemático ya que la mayoría de las veces se realiza de forma manual, es decir, el Gestor de Redes Sociales hace la selección de muestra de tweets de esta forma, conlleva inconvenientes como por ejemplo una gran cantidad de tiempo laboral, alteración en los datos gestionados, no cumplir con los plazos establecidos, muestras poco representativas y sobrecarga cognitiva.

Posterior a señalar esta problemática, se desarrolló una aplicación que permite la automatización de la gestión de datos (extracción, almacenamiento y estructura de los mismos) generados en la red social Twitter, lo cual conlleva a mejorar, optimizar y simplificar el trabajo de la figura del Community Manager.

Este desarrollo conecta la red social Twitter a través de su API por medio del lenguaje PHP, a una base de datos NoSQL, para almacenar los tweets que fueron seleccionados a través de una interfaz donde se introdujeron los parámetros clave a buscar. Estos datos almacenados, a través de características propias de las bases de Datos NoSQL, dieron la opción de filtrar, seleccionar y organizar indicadores como lo son marca, zona geográfica, fecha y usuario para exportarlos a un formato de archivo .CSV de forma estructurada, para ser utilizados por la empresa u organización de la mejor forma que consideren para su toma de decisiones relacionadas con el área de mercadeo.

Con el objetivo de desarrollar esta aplicación, se utilizó una adaptación de la metodología SCRUM, aprovechando todas las ventajas que la misma nos brinda. Por ejemplo, permite obtener el software lo más rápido posible, trabaja en iteraciones cortas, incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto

---

administrado y la producción de un software de una forma consistente, sostenida y competitiva.

Por otra parte en este capítulo, se explica el diseño y construcción del desarrollo de la aplicación, el diseño de la base de datos NoSQL y la conexión entre la red social Twitter a través de su API con la aplicación generada en lenguaje de programación PHP en conjunto con HTML5.

#### **4.1. Plataformas y/o herramientas utilizadas:**

Entre las principales herramientas utilizadas, se encuentra la API para desarrolladores ofrecida por Twitter. Esta API permite la comunicación a través de un servicio de tipo Rest de la mencionada red social. El tipo de servicio Rest escogido dentro de varias opciones que ofrece la API, fue el nombrado por Twitter como “Search”, ya que es el que mejor se adapta a los requerimientos previamente definidos. Las respuestas obtenidas de este servicio luego de realizar las diferentes peticiones, vienen en formato JSON.

Otra herramienta seleccionada, fue MongoDB (base de datos NoSQL) para el almacenamiento de las búsquedas realizadas, ya que la misma ofrece un formato de almacenamiento conocido como Orientado a Documentos, el cual se adapta perfectamente al formato de respuesta o salida de la API de Twitter. También al ser una de las base de datos NoSQL más populares, posee una gran compatibilidad con diferentes herramientas y lenguajes.

Para realizar la conectividad entre la API de Twitter y la base de datos NoSQL, se seleccionó el lenguaje de programación conocido como PHP, ya que el mismo posee una muy buena documentación en relación con MongoDB y actualmente cuenta con librerías y /o extensiones para realizar conexiones con la API anteriormente mencionada.

Por ser PHP un lenguaje corrido del lado del servidor, se debió seleccionar un servidor para su implementación. Por lo tanto se seleccionó el sistema distribuido XAMPP, que cuenta con el servidor web Apache, el cual se integra perfectamente con PHP.

Para la parte de la vista (Front) o interfaz gráfica, se utilizó el Framework conocido como Bootstrap en conjunto con HTML5, debido a la simple integración entre ambas y PHP. Todo esto desarrollado bajo el patrón de Diseño Modelo Vista Controlador (MVC).

Motivado a una limitante que posee el lenguaje PHP, como lo es la ejecución en segundo plano de tareas programadas, se halló como solución a esta limitante la utilización del sistema System Scheduler, el cual se encarga de ejecutar en segundo plano las tareas que se le asignen sin la necesidad de esclavizar la aplicación principal. Esta se encarga de ejecutar un script PHP previamente implementado y diseñado para ejecutar tareas programadas.

## 4.2. Sprint 1. Análisis detallado de requerimientos.

De acuerdo a la metodología de desarrollo escogida (SCRUM), se lograron realizar los ciclos Sprint de acuerdo al número de objetivos específicos definidos en el capítulo 1. En este ciclo o Sprint se muestran de manera detallada el análisis de requerimiento asociado al sistema desarrollado.

### 4.2.1 Historia de usuario.

A continuación se muestran las diferentes historias de usuarios para la especificación de los requisitos necesarios para el desarrollo de la aplicación.

Tabla 5. *Historia de usuario N° 1.*

|                                |                                                                                                                                     |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Historia de usuario            | 1                                                                                                                                   |
| Módulo-Funcionalidad           | Autenticación de usuario.                                                                                                           |
| Rol                            | Yo como usuario registrado.                                                                                                         |
| Característica / Funcionalidad | Necesito un módulo donde pueda autenticarme para hacer uso de las funcionalidades de la aplicación.                                 |
| Criterios de aceptación        | El usuario debe estar debidamente registrado y colocar correctamente los datos para acceder a las funcionalidades de la aplicación. |

El diagrama de flujo asociado a esta primera historia de usuario puede observarse en la figura 17.

Tabla 6. *Historia de usuario N° 2.*

|                                |                                                                                                                                                                                               |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Historia de usuario            | 2                                                                                                                                                                                             |
| Módulo-Funcionalidad           | Registro de usuario.                                                                                                                                                                          |
| Rol                            | Yo como usuario registrado.                                                                                                                                                                   |
| Característica / Funcionalidad | Necesito un módulo donde pueda registrar nuevos usuarios con dos tipos de roles diferentes. Por ejemplo, súper usuarios (privilegios para crear nuevos usuarios o súper usuarios) y usuarios. |
| Criterios de aceptación        | El usuario debe autenticarse en la aplicación y acceder a la funcionalidad de registro usuario. Para crear un nuevo usuario, se debe poseer el rol de súper usuario.                          |

El diagrama de flujo relacionado a la segunda historia de usuario previamente expuesta se muestra en la figura 18.

Tabla 7. *Historia de usuario N° 3.*

|                                |                                                                                                                                                                              |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Historia de usuario            | 3                                                                                                                                                                            |
| Módulo-Funcionalidad           | Búsqueda en línea.                                                                                                                                                           |
| Rol                            | Yo como usuario registrado.                                                                                                                                                  |
| Característica / Funcionalidad | Necesito una funcionalidad donde pueda realizar la búsqueda de una palabra o frase en la red social Twitter de forma automatizada y luego almacenarlos en una base de datos. |
| Criterios de aceptación        | El usuario debe autenticarse en la aplicación y acceder a la funcionalidad de Búsqueda en línea, donde llenará los campos necesarios para realizar dicha búsqueda.           |

Más adelante, en la figura 19 puede observarse el diagrama de flujo relacionado a la tercera historia de usuario.

Tabla 8. *Historia de usuario N° 4.*

|                                |                                                                                                                                                                                                                                                            |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Historia de usuario            | 4                                                                                                                                                                                                                                                          |
| Módulo-Funcionalidad           | Búsqueda programada.                                                                                                                                                                                                                                       |
| Rol                            | Yo como usuario registrado.                                                                                                                                                                                                                                |
| Característica / Funcionalidad | Necesito una funcionalidad donde pueda programar la búsqueda (definir la hora, cantidad de tweets a buscar cada cierto periodo de tiempo) de una palabra o frase en la red social Twitter de forma automatizada y luego almacenarlos en una base de datos. |
| Criterios de aceptación        | El usuario debe autenticarse en la aplicación y acceder a la funcionalidad de Búsqueda programada, donde llenará los campos necesarios para realizar dicha búsqueda.                                                                                       |

El diagrama de flujo asociado a esta cuarta historia de usuario se encuentra graficada en la figura 20.

Tabla 9. *Historia de usuario N° 5.*

|                                |                                                                                                                                                                                                                                                              |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Historia de usuario            | 5                                                                                                                                                                                                                                                            |
| Módulo-Funcionalidad           | Exportar Tweets.                                                                                                                                                                                                                                             |
| Rol                            | Yo como usuario registrado.                                                                                                                                                                                                                                  |
| Característica / Funcionalidad | Necesito una funcionalidad donde pueda exportar los tweets almacenados de las búsquedas realizadas en un archivo de tipo .CSV de forma estructurada con la información del texto del Twitter, quién lo creó, su ubicación geográfica y la fecha de creación. |
| Criterios de aceptación        | El usuario debe autenticarse en la aplicación y acceder a la funcionalidad de Exportar tweets, donde seleccionara la búsqueda a realizada que desea exportar.                                                                                                |

El diagrama de flujo relacionado a la quinta historia de usuario se muestra en la figura 21.

Tabla 10. *Historia de usuario N° 6.*

|                                |                                                                                                                                                                        |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Historia de usuario            | 6                                                                                                                                                                      |
| Módulo-Funcionalidad           | Reportes.                                                                                                                                                              |
| Rol                            | Yo como usuario registrado.                                                                                                                                            |
| Característica / Funcionalidad | Necesito una funcionalidad donde pueda visualizar las búsquedas realizadas, qué usuario realizó la búsqueda, la cantidad de tweets encontrados y si fueron exportados. |
| Criterios de aceptación        | El usuario debe autenticarse en la aplicación y acceder a la funcionalidad de Reporte para visualizar los datos generados.                                             |

Por último, en la figura 22 se puede observar el diagrama de flujo relacionado a la sexta historia de usuario.

#### **4.2.2. Diagramas de flujo.**

A continuación se expondrán los diagramas de flujos relacionados a las historias de usuario previamente descritas.

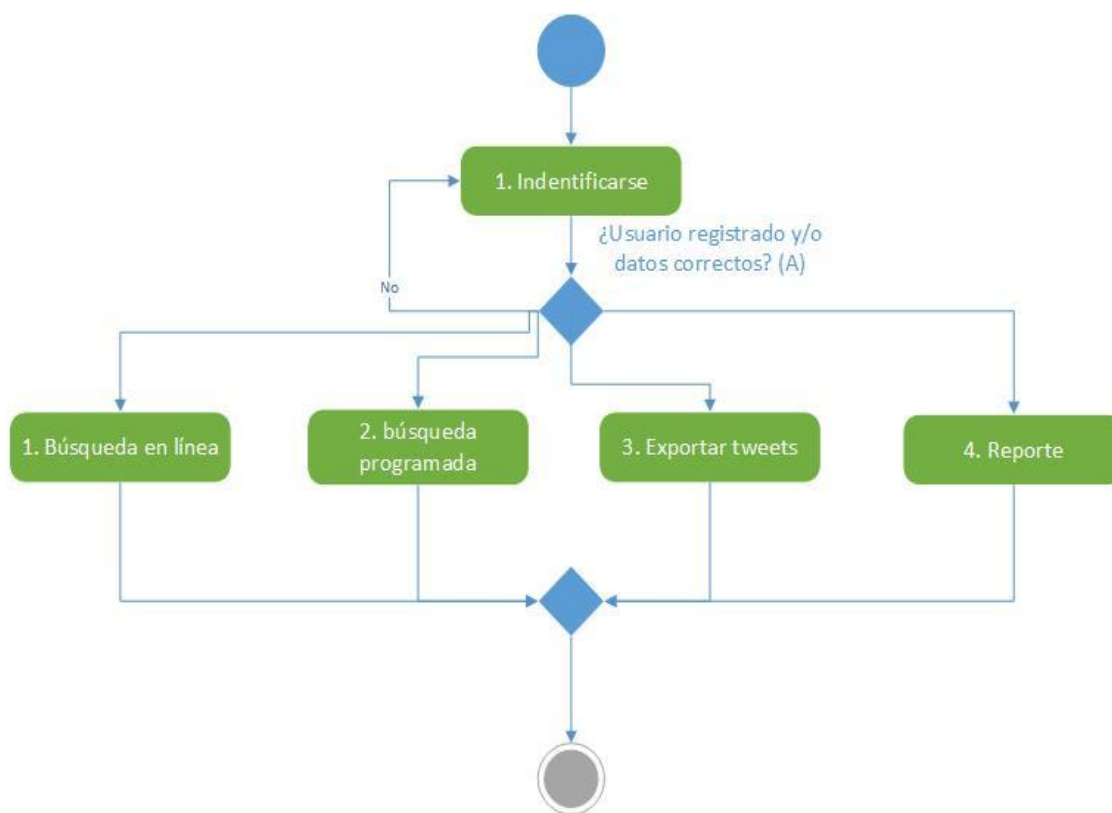


Figura 17. Diagrama de flujo asociado a la historia de usuario “autenticación de usuario N° 1”.

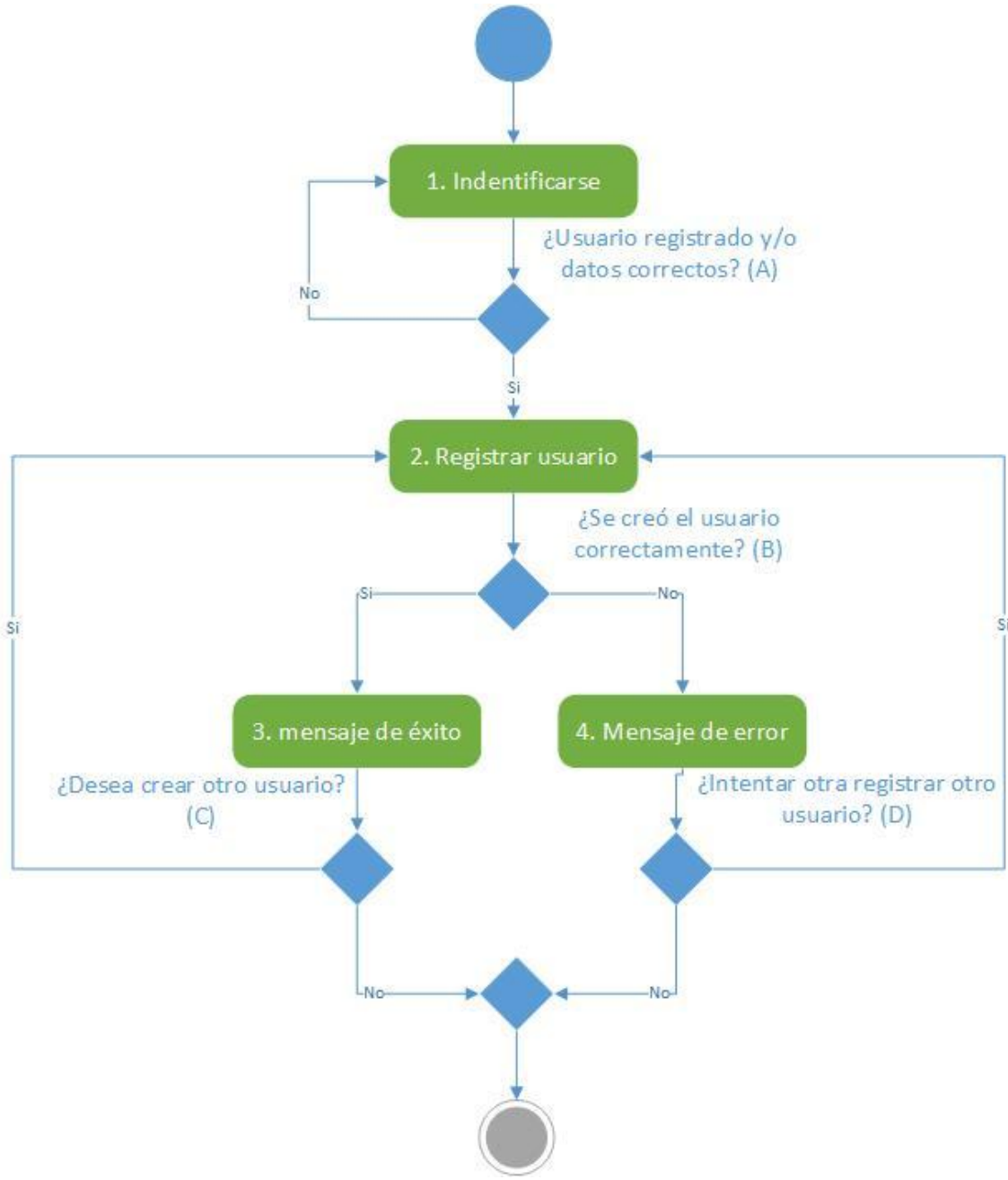


Figura 18. Diagrama de flujo asociado a historia de usuario “registro de usuario N° 2”.

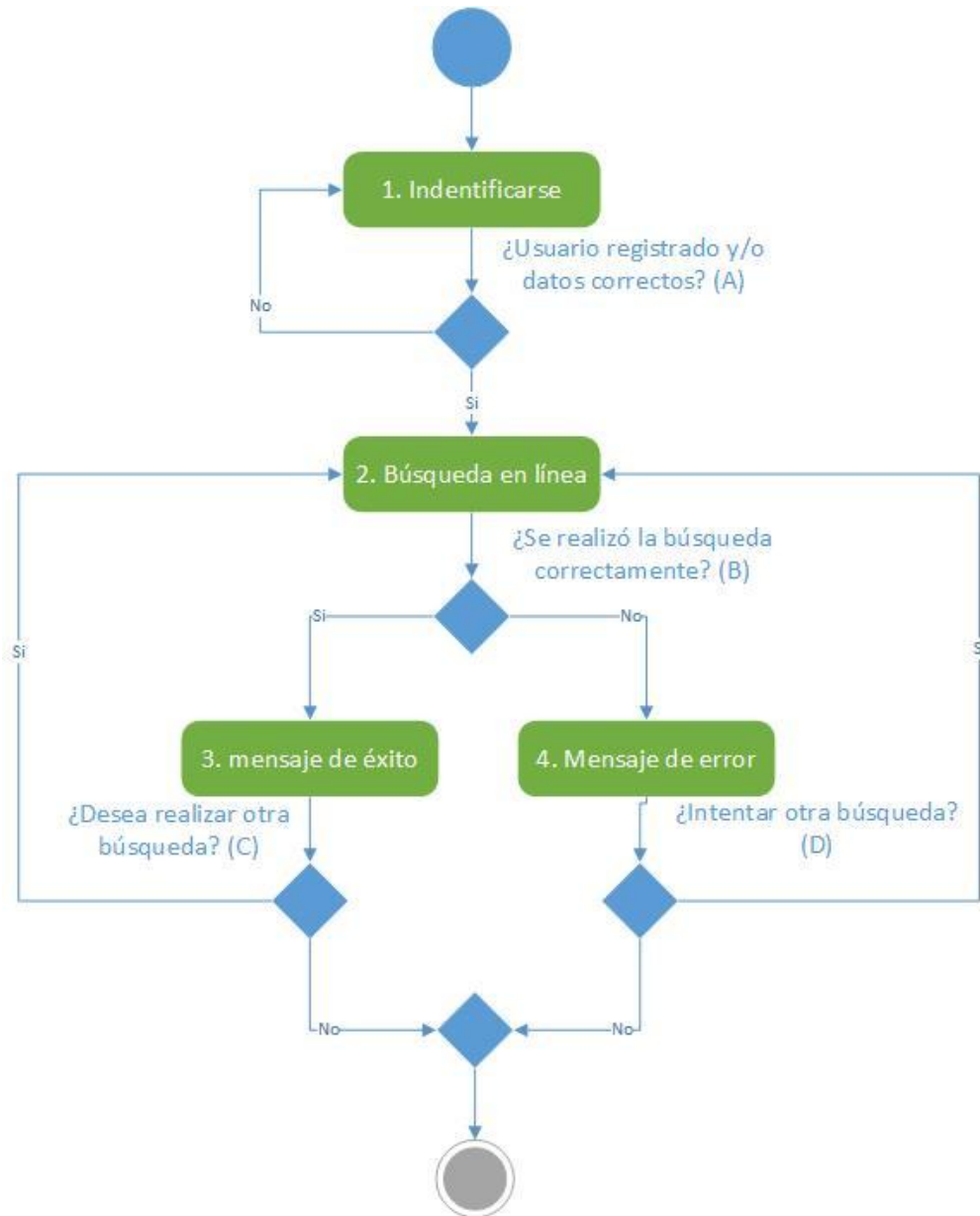


Figura 19. Diagrama de flujo asociado a historia de usuario “búsqueda en línea N° 3”.

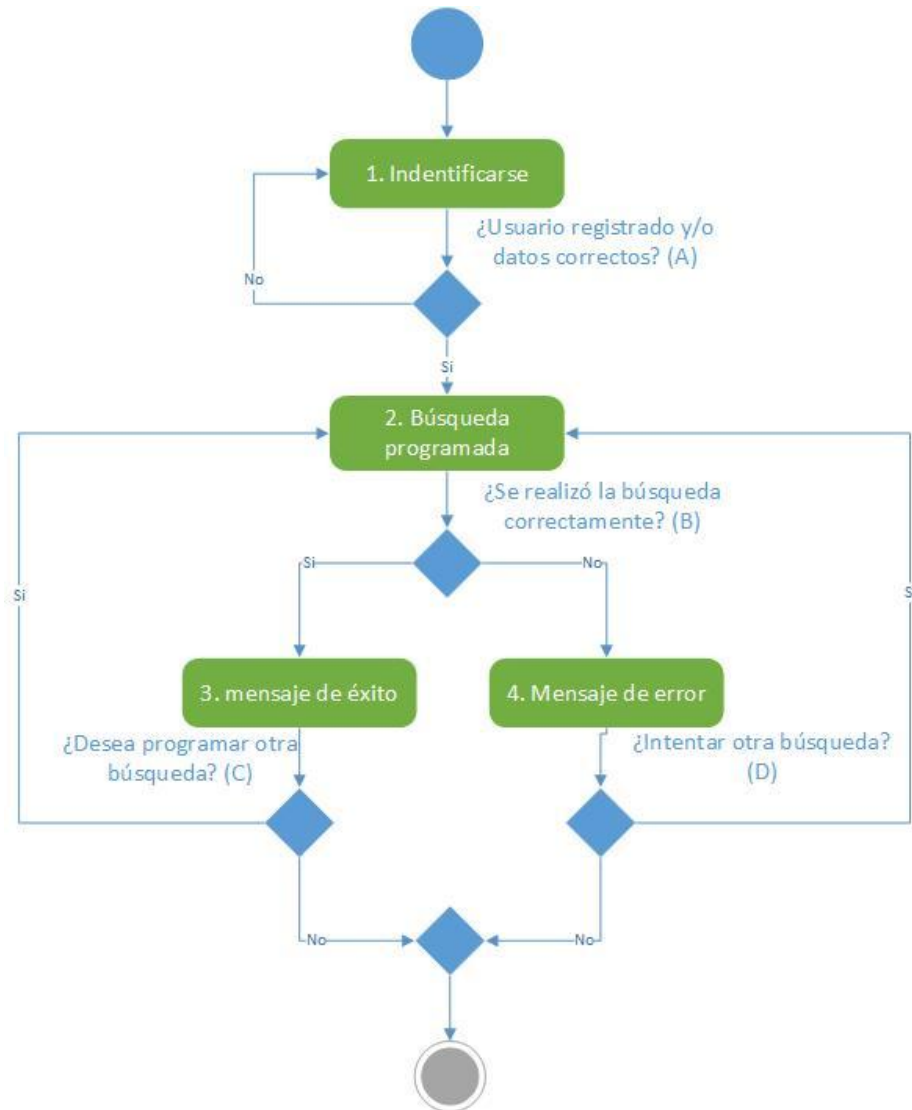


Figura 20. Diagrama de flujo asociado a historia de usuario “búsqueda programada N° 4”.

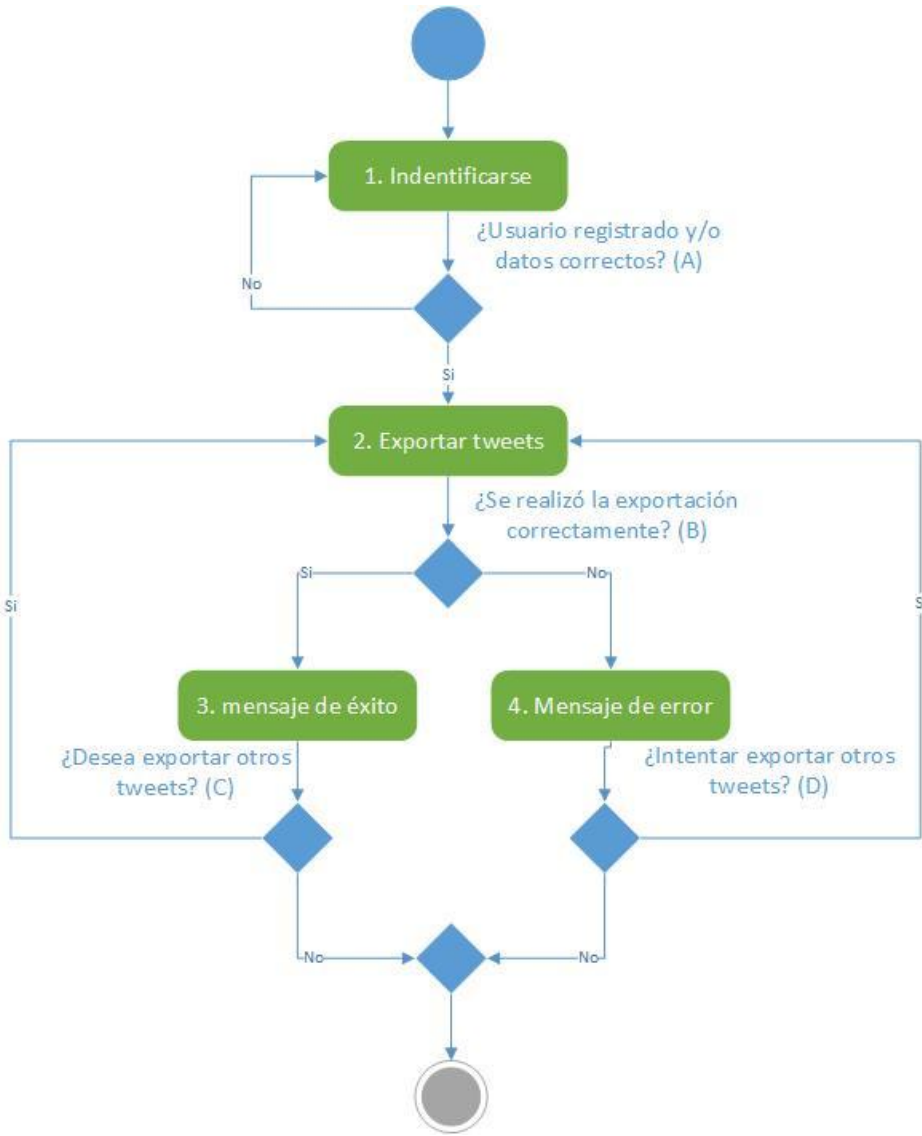


Figura 21. Diagrama de flujo asociado a historia de usuario “exporta usuario N° 5”.

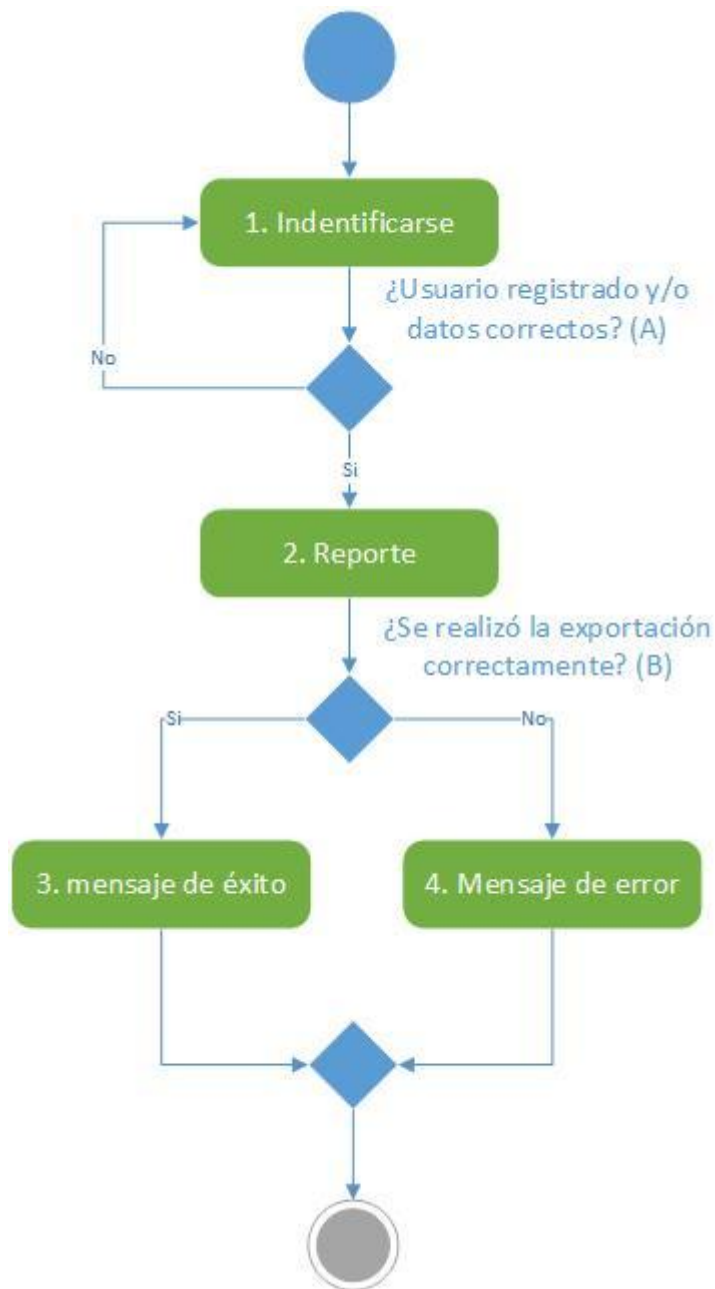


Figura 22. Diagrama de flujo asociado a historia de usuario “Reportes”.

### 4.2.3. Diseño de la base de datos NoSQL.

Como se ha mencionado anteriormente, debido a la naturaleza, así como el volumen de los datos obtenidos como respuesta de las peticiones realizadas a la API de Twitter en formato JSON, se decidió utilizar como base de datos NoSQL MongoDB, la cual proporciona una estructura de almacenamiento orientada a Documentos. Esta adapta perfectamente al formato de salida de la API antes mencionada, ya que MongoDB trabaja en concreto con el formato BSON, que es una forma de representar de forma binaria objetos JSON.

Una de las grandes ventajas que brinda MongoDB (como base de datos NoSQL) es la no definición de un esquema en específico. Los documentos (concepto similar a los registros de una tabla de una base de datos relacional) de una misma colección (concepto similar a una tabla de una base de datos relacional), pertenecientes a una base de datos específica pueden tener esquemas diferentes. Aunado a esto, la creación de la base de datos, las diferentes colecciones a utilizar y los campos pertenecientes a cada documento, son creados dinámicamente, es decir, si todos estos elementos no existen en la base de datos, con tan solo hacer las sentencias para insertar nuevos datos, si no existen en la misma son creados automáticamente.

La base de datos que se creó para uso de la aplicación, le fue asignado el nombre de “Twitter”. Para dar solución a las historias de usuario descritas anteriormente, se implementaron diferentes colecciones:

- Para la historia de usuario N° 1 y N° 2, creó la colección llamada “user” en donde se almacenan los diferentes usuarios registrados en la aplicación y se creó los campos mostrados a continuación en la figura 23, para definir su nombre de usuario, contraseña, rol del usuario en la aplicación (privilegios que posee), etc.

```
{
  "_id": ObjectId(" "),
  "nombre": " ",
  "nombre_usuario": " ",
  "password": " ",
  "nacimiento": " ",
  "direccion": " ",
  "privilegios": " ",
  "apellido": " "
}
```

Figura 23. Estructura del documento perteneciente a la colección “user”.

- Se implementó la creación de diferentes colecciones, correspondientes a cada búsqueda realizada en la aplicación, es decir, el nombre de la colección corresponde a la frase o palabra buscada. En la figura 24, se muestra los campos más importantes almacenados en la colección. Estas colecciones son la forma de dar solución al almacenamiento planteado en la historia de usuario N° 3.

```
{
  "_id": ObjectId(" "),
  "user": {
    "screen_name": " ",
    "location": " "
  },
  "created_at": " ",
  "text": " ",
  "insert_user": " ",
  "insert_date": " "
}
```

Figura 24. Estructura del documento perteneciente a las colecciones de búsquedas.

- Se creó la colección “create\_campaign” en donde se almacenan los datos necesarios para realizar las búsquedas programadas en los campos mostrados en la figura 25. Esta colección se encuentra asociada a la historia de usuario N° 4.

```
{
  "_id": ObjectId(" "),
  "frase": " ",
  "query": {
    "q": " ",
    "count": " ",
    "lang": " ",
    "geocode": " ",
    "result_type": " ",
    "until": " "
  },
  "date_init": " ",
  "min_duracion": " ",
  "min_intervalo": " ",
  "user": " ",
  "cron": " "
}
```

Figura 25. Estructura del documento perteneciente a la colección “create\_campaign”.

- Se implementó la colección “report” la cual almacena el número de tweets conseguidos por búsqueda, el usuario que realizó la búsqueda, la palabra o frase buscada y si fueron exportados los datos. El modelo del documento asociado a la colección se muestra en la figura 26 y se relaciona directamente con la historia de usuario N° 6.

```
{
  "user": "",
  "date": "",
  "count_tweets": NumberInt(""),
  "collection_name": "",
  "export": ""
}
```

Figura 26. Estructura del documento perteneciente a la colección “report”.

#### 4.2.4. Diseño de las interfaces.

En este punto se da a conocer el diseño inicial de las interfaces de usuario de la aplicación más importantes del sistema según las funcionalidades y requerimientos planteados en las historias de usuario.

- En la figura 27, se puede apreciar el diseño de interfaz para la autenticación de los usuarios registrados. La misma consta de 2 campos de formularios y un botón.



Identificate

**Usuario:**  
Escriba su nombre de usuario

**Contraseña:**  
Escriba su contraseña

Iniciar Sesión

Figura 27. Interfaz de autenticación de usuarios.

- La figura 28, muestra los diferentes campos (6 en total) que deben ser llenados, para realizar la búsqueda en línea de la palabra o frase que se quiera.



Inicio Buscar Tweets Crear Campaña Exportar Tweets Reportes

**Palabra o frase a buscar:**  
Escriba la palabra o frase a buscar

**Cantidad de Tweets a buscar (No mayor a 100):**  
Cantidad de Tweets a buscar

**Lugar de origen de los Tweets:**  
Seleccione la ubicación

**Lenguaje de los Tweets:**  
Seleccione el lenguaje a buscar

**Tipos de resultados:**  
Seleccione el tipo de resultado

**Fecha de inicio de la búsqueda:**  
Seleccione una fecha igual o anterior a la actual

Buscar

Figura 28. Diseño de la interfaz Búsqueda en línea.

-Y en la figura 29, se visualiza el reporte generado por el sistema.



The screenshot shows a web interface with a navigation bar at the top containing the following items: Inicio, Buscar Tweets, Crear Campaña, Exportar Tweets, and Reportes. Below the navigation bar is a section titled 'Reportes del Aplicativo' which contains a table with the following data:

| Número de Twitter almacenados | Usuario que realizó la búsqueda | Palabra o frase buscada | Fecha de la Búsqueda | ¿Fueron exportados los Tweets? |
|-------------------------------|---------------------------------|-------------------------|----------------------|--------------------------------|
| 20                            | napoleon                        | metro de caracas        | 07-10-2015 00:06     | No                             |
| 10                            | napoleon                        | metro de los teques     | 07-10-2015 00:08     | No                             |
| 7                             | napoleon                        | imbecil                 | 07-10-2015 04:52     | No                             |

Figura 29. Diseño de la interfaz del reporte generado por el sistema.

### 4.3. Sprint 2. Configuración de ambiente de desarrollo.

Durante este Sprint, se realizó diferentes pruebas y configuraciones para dejar todas las plataformas y herramientas descritas anteriormente en un correcto funcionamiento antes de empezar el desarrollo del sistema.

#### 4.3.1. Configuración de la API de Twitter para desarrolladores.

Para el uso y configuración de dicha API, se realizaron los siguientes pasos:

- (a) Se accedió a la URL de aplicaciones para twitter: <https://apps.twitter.com/>
- (b) Se registró o ya se poseía una cuenta activa en twitter y se procedió a loguearse. Tal como se muestra en el Anexo A.
- (c) Posteriormente se creó una nueva aplicación tal como se muestra en el Anexo B.
- (d) A continuación se llenaron diversos parámetros, los cuales se señalan en el Anexo C.
- (e) Luego se buscó la pestaña “settings” la cual se muestra en el Anexo D.
- (f) Posteriormente se seleccionó la opción resaltada en el Anexo E, y se pulsa “Update settings”, el cual también aparece remarcado en dicha ilustración.

- (g) Luego se accedió a la pestaña “Permissions”, se seleccionó read and Write y pulsar Update Setting, tal como aparece en el Anexo F.
- (h) Una vez realizado lo anterior, se pudo observar la pestaña “Keys and Acces Tokens”, la cual se debe pulsar, tal como se ejemplifica en el Anexo G.
- (i) Este paso anterior es uno de los más importantes, ya que con éste se obtienen las credenciales para que Twitter permita utilizar sus recursos tal como se puede apreciar en el Anexo H.

#### **4.3.2. Configuración de la Base de datos MongoDB.**

Motivado a que el desarrollo del sistema se realizó en un entorno Windows, primero se debió descargar de la página oficial de MongoDB (<https://www.mongodb.org/downloads>) el archivo ejecutable para instalar la base de datos.

Luego de descargar el archivo ejecutable, se instaló de manera simple, haciendo clic siempre en el botón “siguiente” hasta finalizar la instalación.

Después de haber instalado la base de datos, se configuró la variable de entorno PATH para poder iniciar el servidor de la base de datos desde la consola de comandos de Windows. Al realizar la instalación en el disco duro C:\ se agregó la siguiente ruta a la variable PATH: “C:\mongodb”.

Al haber realizado esta configuración se verificó si fue instalada correctamente la base de datos abriendo y ejecutando en la consola de Windows el siguiente comando: “mongod”. Esto muestra la conexión abierta al puerto 27017, con lo que se decidió que se encuentra funcionando correctamente y que era posible conectarse a ella, como se muestra en la siguiente figura 30.

```

C:\Windows\system32\cmd.exe - mongod
C:\Users\Home>mongod
2015-08-02T17:49:36.753-0700 I CONTROL Hotfix KB2731284 or later update is not
installed, will zero-out data file
2015-08-02T17:49:36.805-0700 I JOURNAL [initandlisten] journal dir=C:\data\db\j
ournal
2015-08-02T17:49:36.807-0700 I JOURNAL [initandlisten] recover : no journal fil
es present, no recovery needed
2015-08-02T17:49:36.895-0700 I JOURNAL [durability] Durability thread started
2015-08-02T17:49:37.005-0700 I JOURNAL [journal writer] Journal writer thread s
tarted
2015-08-02T17:49:37.017-0700 I CONTROL [initandlisten] MongoDB starting : pid=3
112 port=27017 dbpath=C:\data\db\ 64-bit host=Home-PC
2015-08-02T17:49:37.017-0700 I CONTROL [initandlisten] targetMinOS: Windows 7/W
indows Server 2008 R2
2015-08-02T17:49:37.017-0700 I CONTROL [initandlisten] db version v3.0.3
2015-08-02T17:49:37.018-0700 I CONTROL [initandlisten] git version: b40106b36ee
cd1b4407eb1ad1af6bc60593c6105
2015-08-02T17:49:37.018-0700 I CONTROL [initandlisten] OpenSSL version: OpenSSL
 1.0.1n-fips 19 Mar 2015
2015-08-02T17:49:37.018-0700 I CONTROL [initandlisten] build info: windows sys
.getwindowsversion(major=6, minor=1, build=7601, platform=2, service_pack='Servic
e Pack 1') BOOST_LIB_VERSION=1_49
2015-08-02T17:49:37.019-0700 I CONTROL [initandlisten] allocator: system
2015-08-02T17:49:37.020-0700 I CONTROL [initandlisten] options: {}
2015-08-02T17:49:37.039-0700 I INDEX [initandlisten] allocating new ns file C
:\data\db\local.ns, filling with zeroes...
2015-08-02T17:49:37.455-0700 I STORAGE [FileAllocator] allocating new datafile
C:\data\db\local.0, filling with zeroes...
2015-08-02T17:49:37.456-0700 I STORAGE [FileAllocator] creating directory C:\da
ta\db\tmp
2015-08-02T17:49:37.677-0700 I STORAGE [FileAllocator] done allocating datafile
C:\data\db\local.0, size: 64MB, took 0.217 secs
2015-08-02T17:49:37.695-0700 I NETWORK [initandlisten] waiting for connections
on port 27017

```

Figura 30. Consola de comando de Windows

### 4.3.3. Configuración de XAMPP.

Al realizarse el desarrollo en entorno Windows, se accedió a la página oficial de XAMPP (<https://www.apachefriends.org/es/index.html>) se realizó la descarga del archivo ejecutable, e instaló dicho archivo en nuestro entorno.

Luego de instalarlo se accedió al panel de control de XAMPP e inició el servidor web Apache como se muestra en la figura 31.

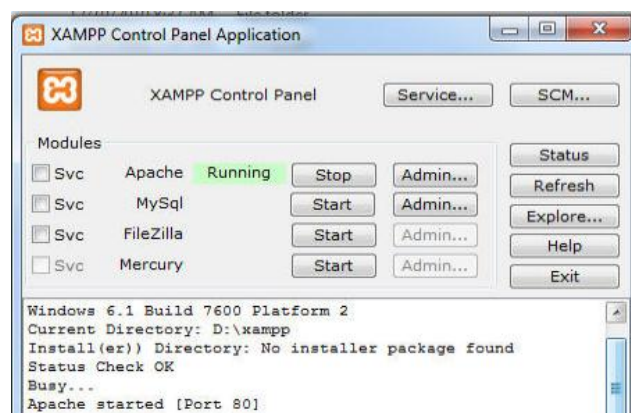
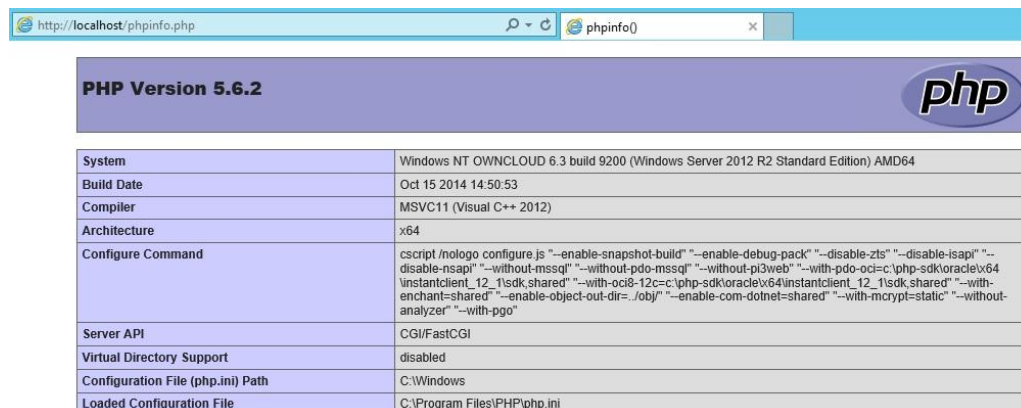


Figura 31. Panel de control de XAMPP.

### 4.3.3. Configuración de conexión entre MongoDB y PHP.

Al seleccionar Apache como servidor web que interpreta el lenguaje PHP para la comunicación con la base de datos MongoDB, se instaló la extensión para la comunicación entre ambos de la siguiente manera:

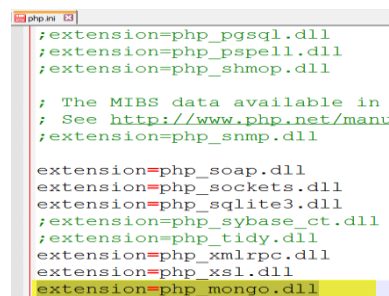
Primero se validó la versión de PHP que posee XAMPP accediendo a la página: <http://localhost/phpinfo.php> como se muestra en la figura 32.



| PHP Version 5.6.2                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System                            | Windows NT OWNCLOUD 6.3 build 9200 (Windows Server 2012 R2 Standard Edition) AMD64                                                                                                                                                                                                                                                                                                                                                                                               |
| Build Date                        | Oct 15 2014 14:50:53                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Compiler                          | MSVC11 (Visual C++ 2012)                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Architecture                      | x64                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Configure Command                 | csript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--disable-isapi" "--disable-nsapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\64\instantclient_12_1\sdk,shared" "--with-enchant-shared" "--enable-object-out-dir=.obj" "--enable-com-dotnet-shared" "--with-mcrypt-static" "--without-analyzer" "--with-pgo" |
| Server API                        | CGI/FastCGI                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Virtual Directory Support         | disabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Configuration File (php.ini) Path | C:\Windows                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Loaded Configuration File         | C:\Program Files\PHP\php.ini                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

Figura 32. Página informativa de las características de PHP.

Luego de validar la versión de PHP, se descargó los conectores (drivers) entre MongoDB y PHP según la versión antes mencionada de la siguiente página: <https://github.com/mongodb/mongo-php-driver>. Posterior a ello se descomprimió en la siguiente ruta el archivo descargado: C:\xampp\php\ext, se abrió el archivo “php.ini” ubicado en la ruta: C:\xampp\php y se declaró la variable: extensión=php\_mongo.dll como se muestra en la siguiente figura 33.



```

;extension=php_pqsql.dll
;extension=php_pspell.dll
;extension=php_shmop.dll

; The MIBS data available in
; See http://www.php.net/manu
;extension=php_snmp.dll

extension=php_soap.dll
extension=php_sockets.dll
extension=php_sqlite3.dll
;extension=php_sybase_ct.dll
;extension=php_tidy.dll
extension=php_xmlrpc.dll
extension=php_xsl.dll
extension=php_mongo.dll

```

Figura 33. Archivo de configuración de PHP.

Para finalizar hubo que asegurarse que se configuró correctamente los conectores accediendo nuevamente a la página: <http://localhost/phpinfo.php> y buscar la configuración de conexión de MongoDB como se muestra en la figura 34.

**mongo**

| MongoDB Support |         | enabled |
|-----------------|---------|---------|
| Version         | 1.4.5   |         |
| SSL Support     | enabled |         |
| Streams Support | enabled |         |

| Directive                | Local Value | Master Value |
|--------------------------|-------------|--------------|
| mongo.allow_empty_keys   | 0           | 0            |
| mongo.chunk_size         | 262144      | 262144       |
| mongo.cmd                | \$          | \$           |
| mongo.default_host       | localhost   | localhost    |
| mongo.default_port       | 27017       | 27017        |
| mongo.is_master_interval | 15          | 15           |
| mongo.long_as_object     | 0           | 0            |
| mongo.native_long        | 0           | 0            |
| mongo.ping_interval      | 5           | 5            |

Figura 34. Página informativa de PHP que muestra la configuración con MongoDB.

#### 4.3.4. Configuración de conexión entre API de Twitter y PHP.

Twitter OAuth es un protocolo de autenticación (librerías) que permite a los usuarios conectarse con el sistema mediante una confirmación previa en Twitter.

Para establecer la conexión se necesitaron estos archivos:

- OAuth.php: Clase para manejar el protocolo OAuth.
- twitteroauth.php: Clase para interactuar con Twitter y el protocolo OAuth.

Estos archivos vienen por defecto en la librería y deben editarse con los datos obtenidos anteriormente en la configuración de la API de Twitter (Customer key,

Customer secret, Access token y Access token secret), en conjunto con la opción “Search” que se utilizó para realizar las búsquedas.

El script PHP para realizar la prueba de conexión con Twitter puede verse en la figura 35 y para ver la respuesta del API de Twitter puede verse en la figura 36 al ejecutar dicho script.

```

<?php
require_once("twitteroauth/twitteroauth/twitteroauth.php");

$twitteruser = "twitterusername";
$notweets = 30;
$consumerkey = "1234xcxc5";
$consumersecret = "12345xcxc6789";
$accesstoken = "1xcxc23456789";
$accesstokensecret = "1xcxc2345";

function getConnectionWithAccessToken($cons_key, $cons_secret, $oauth_token, $oauth_token_secret) {
    $connection = new TwitterOAuth($cons_key, $cons_secret, $oauth_token, $oauth_token_secret);
    return $connection;
}

$connection = getConnectionWithAccessToken($consumerkey, $consumersecret, $accesstoken, $accesstokensecret);
$tweets = $connection->get("https://api.twitter.com/1.1/search/tweets.json?", "busqueda => 'frase a buscar!');
echo json_encode($tweets);
?>

```

Figura 35. Script PHP para realizar la prueba de conexión con Twitter.

```

"created_at": "Fri Jan 23 23:57:36 +0000 2015",
"id": 558775589612437504,
"id_str": "558775589612437504",
"text": "Thanks, polar vortex: Attendance dips at major Chicago museums in 2014 http://t.co/jQ1LEurk9s http://t.co/3bbs2nGemx",
"source": "\u003ca href=\"http://twitter.com\" rel=\"nofollow\"\u003eTwitter Web Client\u003c/a\u003e",
"truncated": false,
"in_reply_to_status_id": null,
"in_reply_to_status_id_str": null,
"in_reply_to_user_id": null,
"in_reply_to_user_id_str": null,
"in_reply_to_screen_name": null,
"user": {
  "id": 7313362,
  "id_str": "7313362",
  "name": "Chicago Tribune",
  "screen_name": "chicagotribune",
  "location": "Chicago, IL",
  "url": "http://www.chicagotribune.com/",
  "description": "Chicago Tribune news, features and so much more live from our newsroom. A part of your life since 1847.",
  "protected": false,
  "verified": true,
  "followers_count": 321548,
  "friends_count": 523,
  "listed_count": 8074,
  "favorites_count": 34,
  "statuses_count": 47367,
  "created_at": "Sat Jul 07 14:10:07 +0000 2007",
  "utc_offset": -21600,
  "time_zone": "Central Time (US & Canada)",
  "geo_enabled": false,
  "lang": "en",

```

Figura 36. Resultado de ejecutar el script de prueba de conexión con Twitter.





## 4.4. Sprint 3. Desarrollo de la aplicación.

En este Sprint, se realizó el desarrollo completo de la aplicación, desde el desarrollo de las distintas interfaces de usuario, hasta las funcionalidades descritas durante el Sprint N° 1 en las historias de usuario.

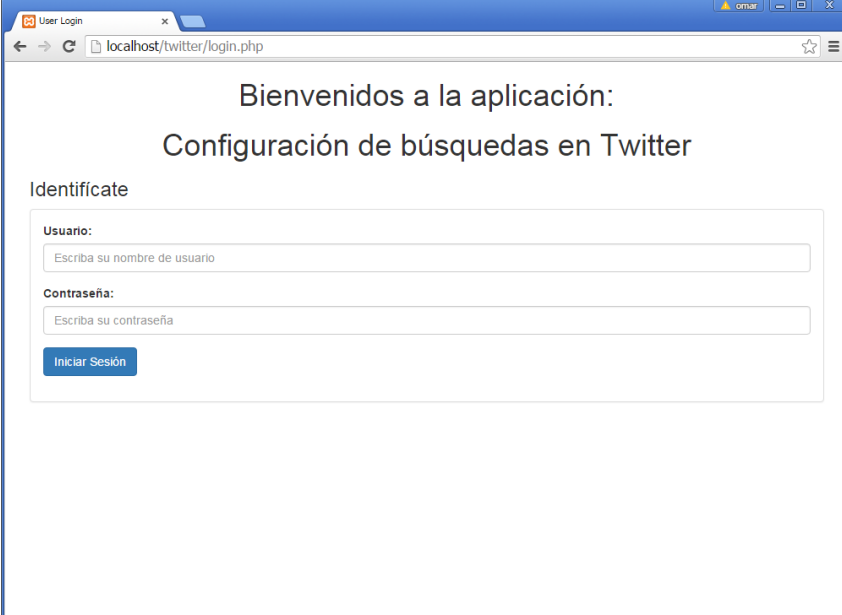
### 4.4.1. Desarrollo de las Interfaces de Usuario.

Para cumplir con el desarrollo de las funcionalidades de la aplicación, primero se optó por elaborar las diferentes interfaces de usuario. Como se mencionó anteriormente, se tomó como plataformas para las vistas, el framework llamado Bootstrap y HTML5, los cuales se integran perfectamente y brindan una apariencia sencilla, amigable y de fácil entendimiento para el usuario. Básicamente todas las interfaces mostraron formularios para ser rellenos por el usuario dependiendo de la funcionalidad a utilizar, además de contar con un menú superior horizontal donde se puede seleccionar la funcionalidad a utilizar.

Los colores seleccionados para la aplicación, fueron aquellos que brindan armonía entre ellos, sencillez y un buen look and feel para el usuario, los cuales fueron los siguientes:


-  Como color de fondo se estableció el color blanco (#FFFFFF), al igual que para los títulos del menú y los diferentes textos que contienen los botones.
-  El color seleccionado para los botones y el menú fue el color azul (#337ab7).
-  Para los diferentes títulos se seleccionó el color negro (#000000).
-  Los bordes que enmarcan los diferentes campos de formulario, se crearon con el color (#CCC).

A continuación se muestran en las siguientes figuras, las imágenes de las interfaces de usuario creadas para la aplicación.



The screenshot shows a web browser window with the URL `localhost/twitter/login.php`. The page title is "Bienvenidos a la aplicación: Configuración de búsquedas en Twitter". Below the title, there is a section labeled "Identifícate" with two input fields: "Usuario:" (with placeholder "Escriba su nombre de usuario") and "Contraseña:" (with placeholder "Escriba su contraseña"). A blue button labeled "Iniciar Sesión" is positioned below the password field.

Figura 37. Interfaz de la funcionalidad de Identificación de usuario.



The screenshot shows a web browser window with the URL `localhost/twitter/profile.php`. The page title is "Configuración de búsquedas en Twitter". In the top right corner, there are links for "Cerrar Sesión" and "Bienvenido napoleon". A dark blue navigation bar contains the following menu items: "Inicio", "Búsqueda en línea", "Búsqueda programada", "Exportar tweets", "Reportes", and "Registrar usuario". The main content area contains several search configuration options: "Palabra o frase a buscar:" (input field with placeholder "Escriba la palabra o frase a buscar"), "Cantidad de Tweets a buscar (No mayor a 100):" (input field with placeholder "Cantidad de Tweets a buscar"), "Lugar de origen de los Tweets:" (dropdown menu with placeholder "Seleccione la ubicación"), "Lenguaje de los Tweets:" (dropdown menu with placeholder "Seleccione el lenguaje a buscar"), "Tipo de búsqueda:" (dropdown menu with placeholder "Seleccione el tipo de búsqueda"), and "Fecha de inicio de la búsqueda:" (input field with placeholder "2015-10-18" and a calendar icon). A blue "Buscar" button is located at the bottom left of the form.

Figura 38. Interfaz de la funcionalidad Búsqueda en línea.

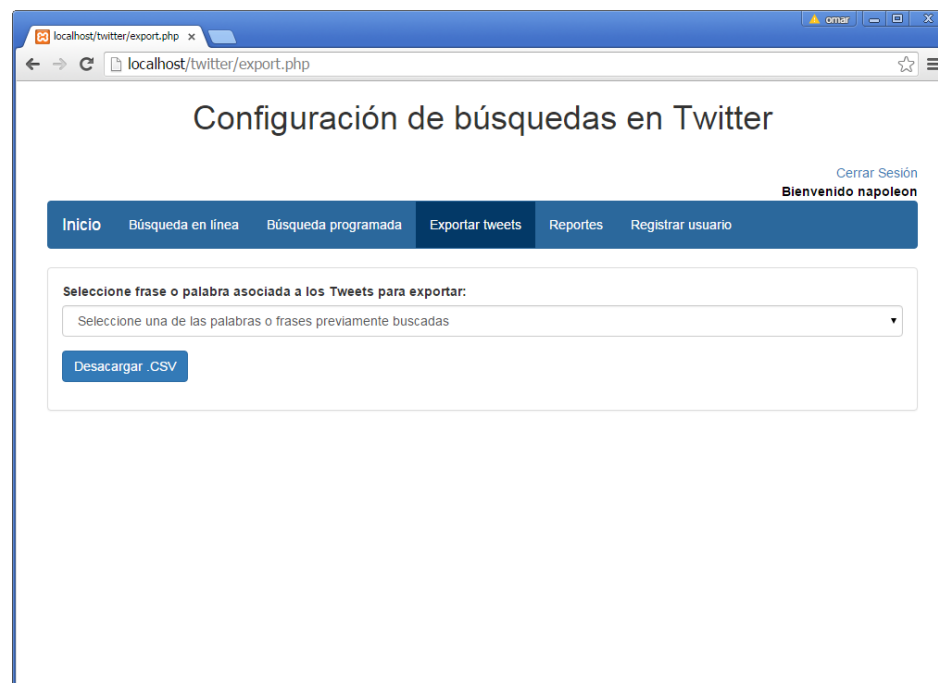


The screenshot shows a web browser window with the URL `localhost/twitter/create.php`. The page title is "Configuración de búsquedas en Twitter". In the top right corner, there are links for "Cerrar Sesión" and "Bienvenido napoleon". A navigation menu includes "Inicio", "Búsqueda en línea", "Búsqueda programada" (which is highlighted), "Exportar tweets", "Reportes", and "Registrar usuario". The main content area contains several form fields for configuring a scheduled search:

- Palabra o frase a buscar:** A text input field with the placeholder "Escriba la palabra o frase a buscar".
- Cantidad de Tweets a buscar (No mayor a 100):** A text input field with the placeholder "Cantidad de Tweets a buscar".
- Lugar de origen de los Tweets:** A dropdown menu with the placeholder "Seleccione la ubicación".
- Lenguaje de los Tweets:** A dropdown menu with the placeholder "Seleccione el lenguaje a buscar".
- Tiempo de duración(minutos):** A text input field with the placeholder "tiempo que durará la campaña".
- Fecha de inicio de la búsqueda:** A date input field showing "2015-10-18" with a calendar icon.
- Fecha y hora de inicio de ejecución de la búsqueda:** A date and time input field showing "18-10-2015 12" with a calendar icon.
- Intervalos de tiempo para realizar la búsqueda(No puede ser mayor al tiempo de duración):** A text input field with the placeholder "cada cuanto tiempo se realizará la búsqueda".

A "Crear" button is located at the bottom left of the form area.

Figura 39. Interfaz de la funcionalidad Búsqueda programada.



The screenshot shows a web browser window with the URL `localhost/twitter/export.php`. The page title is "Configuración de búsquedas en Twitter". In the top right corner, there are links for "Cerrar Sesión" and "Bienvenido napoleon". A navigation menu includes "Inicio", "Búsqueda en línea", "Búsqueda programada", "Exportar tweets" (which is highlighted), "Reportes", and "Registrar usuario". The main content area contains a single form field for selecting a phrase to export:

- Seleccione frase o palabra asociada a los Tweets para exportar:** A dropdown menu with the placeholder "Seleccione una de las palabras o frases previamente buscadas".

A "Descargar .CSV" button is located below the dropdown menu.

Figura 40. Interfaz de la Funcionalidad Exportar Tweets.



Figura 41. Interfaz de la funcionalidad Reportes.



Figura 42. Interfaz de la funcionalidad Registrar usuario.

#### 4.4.2. Desarrollo de la funcionalidad Identificación de usuario.

Para el desarrollo de este punto, primero se tomó en cuenta el modelo del documento mostrado anteriormente en la figura 23 perteneciente a la colección “user”. Como primer paso establecemos la conexión con nuestra base de datos MongoDB llamada “twitter” para evaluar los datos capturados a través de los campos mostrados en la figura 37 (usuario y contraseña) y se evaluó contra la base de datos si el usuario existe, y de existir, si la contraseña es válida. Si los datos eran correctos llevaba a la figura 38 y en caso contrario mostraba un mensaje de error. En la figura 43 se apreciar cómo se realizaba esta operación en lenguaje PHP.

```
class User
{
    const COLLECTION = 'user';
    private $_mongo;
    private $_collection;
    private $_user;

    public function __construct()
    {
        $database = $connection->selectDB('twitter');
        $this->_mongo = DBConnection::instantiate();
        $this->_collection = $this->_mongo->getCollection(User::COLLECTION);
        if ($this->isLoggedIn()) $this->_loadData();
    }

    public function isLoggedIn()
    {
        return isset($_SESSION['user_id']);
    }

    public function authenticate($nombreusuario, $password)
    {
        $query = array('nombreusuario' => $nombreusuario, 'password' => md5($password));
        $this->_user = $this->_collection->findOne($query);
        if (empty($this->_user)) return False;
        $_SESSION['user_id'] = (string) $this->_user['_id'];
        return True;
    }
}
```



Figura 43. Código para evaluación y autenticación de un usuario.

#### 4.4.3. Desarrollo de la funcionalidad Búsqueda en línea.

Para esta funcionalidad, como factores importantes se estableció la conexión con MongoDB y con la API de Twitter. Luego de haber instanciado estas conexiones, se armó en lenguaje PHP con un arreglo que contiene todos los campos establecidos en la interfaz mostrada en la figura 39, ya que estos representaban las opciones de búsqueda que ofrecía la opción “Search” de dicha API. Luego de haber llenado el arreglo

mencionado anteriormente con los datos suministrados por el usuario a través del formulario ofrecido en la interfaz, se llamó la opción de la API “Search” donde se pasó como parámetro el arreglo anterior. Esto devolvía una cierta cantidad de tweets en formato JSON que luego eran almacenados en la base de datos con el modelo de documento mostrado en la figura 24 y con el nombre de la colección utilizado en el campo de búsqueda de la palabra o frase. En la figura 44, se puede apreciar un extracto del código de la funcionalidad.

```

case 'search':
    $report = 'report_search';
    $frase = rtrim(trim($_POST['frase']));
    $count = $_POST['count'];
    $lang = $_POST['lang'];
    $geocode = $_POST['location'];
    $result = $_POST['result'];
    $until = $_POST['date'];
    if($lang == 'Español'){
        $lang = 'es';
    }else if($lang == 'Inglés'){
        $frase = 'en';
    }else{
        $lang = '';
    }

    if($geocode == 'Caracas'){
        $geocode = '10.5,-66.93';
    }else if($geocode == 'Maracaibo'){
        $geocode = '10.63,-71.63';
    }else{
        $geocode = '';
    }

    if($result == 'Reciente'){
        $result = 'recent';
    }else if($result == 'Popular'){
        $result = 'popular';
    }else{
        $result = 'mixed';
    }

    $locati = $_POST['location'];
    $query = array(
        "q" => $frase,
        "count" => $count,
        "lang" => $lang,
        "geocode" => $geocode,
        "result_type" => $result,
        "until" => $until
    );

    $arreglo = $connection->get("search/tweets", $query);
    $collection = $database->selectCollection($frase);
    $collection->insert($arreglo);
} catch (MongoConnectionException $e) {
    die("No se ha podido conectar a la base de datos " . $e->getMessage());
} catch (MongoException $e) {
    die("No se han podido insertar los datos " . $e->getMessage());
}

```

Arreglo armado para la búsqueda que contiene la frase a buscar, el idioma, la fecha de creación, etc.

Llamada a la API, donde se pasa como parametro el arreglo

Se insertan los datos en la BD

Figura 44. Extracto de código de la funcionalidad Búsqueda en línea.

#### 4.4.4. Desarrollo de la funcionalidad Búsqueda programada.

Para el desarrollo de esta funcionalidad, encontramos la limitante relacionada, a que el Lenguaje PHP no puede dejar ejecutando una acción en segundo plano sin esclavizar la aplicación principal. Para superar este obstáculo o limitante, utilizamos como apoyo una herramienta llamada “System Scheduler” la cual nos ofrece la ayuda de

---

la ejecución en segundo plano de diferentes acciones o tareas programadas. En este sentido creamos un Script de apoyo en PHP (muy similar a la figura 38) que se estará ejecutando permanentemente y tomará datos como por ejemplo: fecha de inicio de la búsqueda junto con la hora de inicio, además del tiempo que va durar la búsqueda y cuantas veces se va a realizar búsquedas en ese intervalo de tiempo. Estos datos son llenados en los campos del formulario mostrados anteriormente en la figura 39. Estos datos son almacenados en MongoDB en una colección llamada “create\_campaign”. Por lo tanto el script PHP mencionado anteriormente “preguntará” a la base de datos cada cierto periodo de tiempo si debe ejecutar una búsqueda programada.

#### **4.4.5. Desarrollo de la funcionalidad Exportar Tweets.**

Para realizar la exportación de una de las palabras o frase buscadas previamente, hacemos uso de una de las alternativas de exportación que nos brinda MongoDB, llamada “mongoExport”. Primeramente mostramos en un campo de tipo selección la palabra o frase que se desea exportar como se mostró en la figura 40. Al seleccionar una de las opciones, a nivel de código establecemos conexión con la base de datos, en concreto con la colección llamada igual a la opción seleccionada y le asignamos a un arreglo todos los datos almacenados en dicha colección. Luego Filtramos los campos de los documentos pertenecientes a la colección que necesitamos, los cuales son, fecha de creación del tweet, usuario que creó el tweet, texto del tweet y ubicación desde donde se generó. Por último, utilizamos la sentencia “mongoExport” en la cual estructuramos los datos dentro de un archivo con formato .CSV con los campos anteriormente descritos y generamos el archivo en una ruta específica.

---

#### **4.4.6. Desarrollo de la funcionalidad Reportes.**

Esta funcionalidad fue desarrollada básicamente consultando los documentos de la colección “report”, donde los campos de los documentos se van actualizando, dependiendo de las operaciones que se vayan realizando en la aplicación, como por ejemplo si una búsqueda fue exportada o no. En la figura 16 mostrada con anterioridad se aprecian los diferentes campos que componen el reporte.

#### **4.4.7. Desarrollo de la funcionalidad Registrar usuario.**

Su desarrollo fue bastante básico. Esta funcionalidad se conecta directamente con la colección “user” al momento de que un usuario identificado previamente y que posea privilegios como súper usuario quiera registrar un nuevo usuario. Se basa en rellenar el formulario mostrado en la figura 42 y almacenar un nuevo documento en la colección nombrada anteriormente relacionada al nuevo usuario creado. Esta funcionalidad trabaja en principio de la misma manera como si se hubiese desarrollado con una base de datos Relacional. Pero posee la particularidad que esta opción solo será mostrada para usuarios que posean privilegios de “súper usuario”. Esto se logra colocando una bandera a nivel de código para mostrar o no la opción dependiendo del rol o privilegio del usuario que accedió a la aplicación.

#### 4.5. Sprint 4. Realización de pruebas para certificación de la aplicación.

Para probar la funcionalidad del aplicativo, se realizó un instrumento que se encargó de medir el look and feel, si el mismo es funcional, su nivel de complejidad y si ofrece ayuda en su utilización.

Dicho instrumento se trató de una Escala Tipo Likert, la cual ofreció estudiar el grado de acuerdo y desacuerdo de los usuarios a través de preguntas redactadas en ítems cerrados de manera sencilla y efectiva.

Este instrumento estuvo conformado por 13 ítems escritos de forma afirmativa, los cuales se encargaban de medir 10 factores distintos siendo estos autenticación de usuario (ítem 1), registro de usuario (ítem 2), búsqueda en línea (ítem 3), búsqueda programada (ítem 4), exportar tweets (ítem 5), reportes (ítem 6), Look and feel (ítem 7, 8, 9, 10), nivel de dificultad de uso de la aplicación (ítem 11), nivel de ayuda que ofrece la aplicación (ítem 12), y funcionalidad (ítem 13).

Dicha escala ofrecía 5 opciones las cuales eran (a) totalmente de acuerdo, (b) de acuerdo, (c) indiferente, (d) en desacuerdo y (e) totalmente en desacuerdo a las cuales se les asignó el siguiente valor al momento de ser usadas en una base de datos para su estudio.

Tabla 11. *Opciones y valores de escala tipo Likert.*

| Calificador cualitativo  | Valor asignado |
|--------------------------|----------------|
| Totalmente de acuerdo    | 5              |
| De acuerdo               | 4              |
| Indiferente              | 3              |
| En desacuerdo            | 2              |
| Totalmente en desacuerdo | 1              |

Esto quiere decir, que a mayor puntuación en la escala, mejor calificada era el factor que obtenía la puntuación. La puntuación máxima y mínima que podía obtener cada factor era como se muestra a continuación.

Tabla 12. *Puntuación máxima y mínima de cada factor.*

| Factor                                                                                                     | Puntuación mínima | Puntuación máxima |
|------------------------------------------------------------------------------------------------------------|-------------------|-------------------|
| Autenticación de usuario                                                                                   | 20                | 100               |
| Registro de usuario                                                                                        | 20                | 100               |
| Búsqueda en línea                                                                                          | 20                | 100               |
| Búsqueda programada                                                                                        | 20                | 100               |
| Exportar tweets                                                                                            | 20                | 100               |
| Reportes                                                                                                   | 20                | 100               |
| Look and feel (con los subfactores agradable a la vista, localización de botones, colores y tipo de letra) | 80                | 400               |
| Nivel de dificultad de uso de la aplicación                                                                | 20                | 100               |
| Nivel de ayuda que ofrece la aplicación                                                                    | 20                | 100               |
| Funcionalidad                                                                                              | 20                | 100               |

La muestra utilizada estuvo conformada por 20 personas, a los cuales se les aplicó 2 veces el instrumento en tiempos distanciados.

Primeramente, una vez finalizado el aplicativo se le pidió que hicieran uso del mismo, dándoles un total de 15 minutos. Dentro de ese periodo de tiempo, los sujetos podían hacer uso de todas las funcionalidades de la aplicación a fin de familiarizarse con la misma y su interfaz. Luego de eso se les entregó a cada uno el cuestionario donde se les pidió que lo respondieran con el fin de que sus opiniones ayudaran a detectar errores en la aplicación y fuese posible realizar los arreglos pertinentes de la misma.

A continuación se exponen los resultados arrojados por las encuestas para la primera fase de exploración de las funcionalidades de la aplicación.

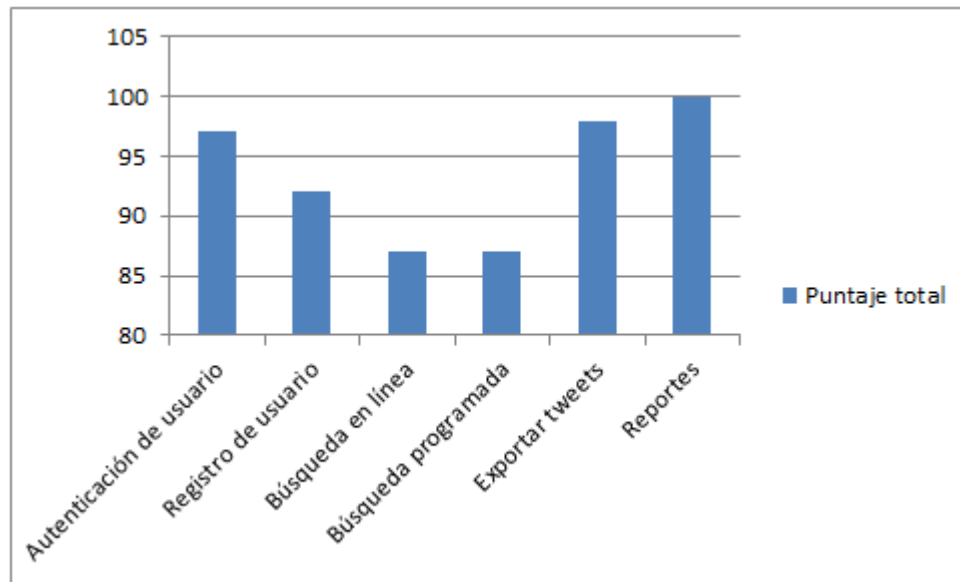


Figura 45. Histograma de resultados de la funcionalidad de la aplicación

Como se puede observar en el gráfico, la “autenticación de usuario” casi obtuvo la puntuación máxima total, lo cual indica que su funcionamiento es adecuado y el módulo permite autenticar al usuario para hacer uso de las funcionalidades de la aplicación, esto quiere decir que el módulo donde se registran nuevos usuarios con dos tipos de roles diferentes (súper usuarios y usuarios) funciona de forma adecuada en la aplicación.

En relación a “búsqueda en línea”, se observa que la misma obtuvo una puntuación promedio-baja, lo cual indicó que al realizar la búsqueda de una palabra o frase en la red social Twitter de forma automatizada se estaba presentando un problema. Al explorar a más profundidad se descubrió que cuando se intentaba realizar la búsqueda de una frase (en vez de una única palabra) se presentaba una falla, la cual fue solventada para mejorar la funcionalidad de la aplicación.

Al visualizar los resultados arrojados para “búsqueda programada”, se observa una puntuación alta, indicando que la aplicación es capaz de programar la búsqueda (definir la hora, cantidad de tweets a buscar cada cierto periodo de tiempo) de una

palabra o frase en la red social Twitter de forma automatizada y luego almacenarlos en una base de datos.

Al observar los resultados para “exportar tweets” se visualiza que la aplicación es capaz de exportar los tweets almacenados de las búsquedas realizadas en un archivo de tipo .CSV de forma estructurada con la información del texto del Twitter, quién lo creó, su ubicación geográfica y la fecha de creación del tweet.

En relación a los reportes, se observó que los mismos obtuvieron una puntuación perfecta, lo cual es señal de que la aplicación permite visualizar las búsquedas realizadas, qué usuario las realizó, la cantidad de tweets encontrados y si fueron exportados.

En relación al factor “Look and feel” se puede observar los resultados de la encuesta en el siguiente gráfico.

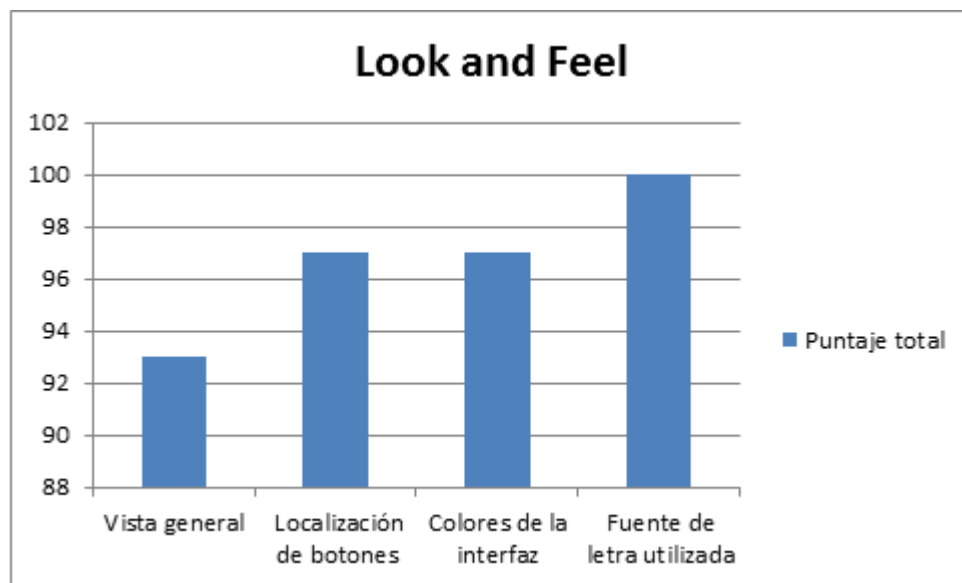


Figura 46: Histograma de resultados del look and feel

En general se observa que los sub-factores de Look and feel (vista general, colores de la interfaz, localización de botones y fuente de letra) tiene una puntuación que

oscila entre muy alto y perfecto, lo cual significa que la interfaz de la aplicación resultó agradable para los usuarios y por ello no necesitó realizar modificaciones posteriores.

En lo que respecta al factor “nivel de dificultad de uso de la aplicación” y “nivel de ayuda que ofrece la aplicación” se obtuvieron puntuaciones medio-bajo y bajo, tal como se puede apreciar en el gráfico.

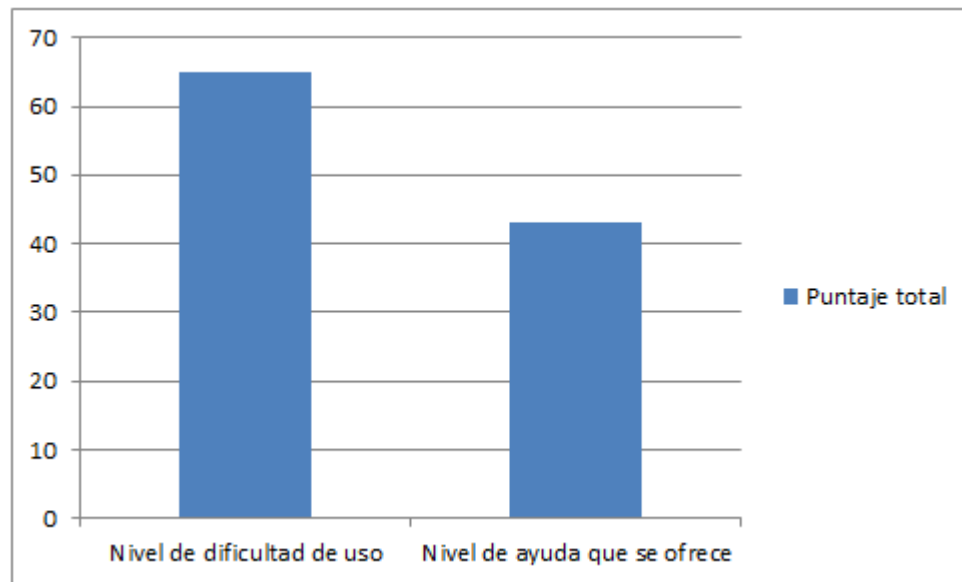


Figura 47. Histograma de resultados de nivel de dificultad de uso de la aplicación y nivel de ayuda que ofrece la aplicación.

Esto indicó que existía un problema que se relaciona con dichos factores de la aplicación.

Se pudo apreciar que el hecho de que la aplicación tuviese un nivel de dificultad medio-alto estaba relacionado con que la aplicación ofreciese poca ayuda al ser utilizada. Para resolver esto se le agregó a la aplicación un tooltip que aparece cuando se pasa el mouse sobre uno de los campos del formulario a rellenar. Dicho tooltip ofrece una explicación y/o descripción de la funcionalidad de los campos.

Por último en relación a la funcionalidad, se observa en el gráfico los siguientes resultados:

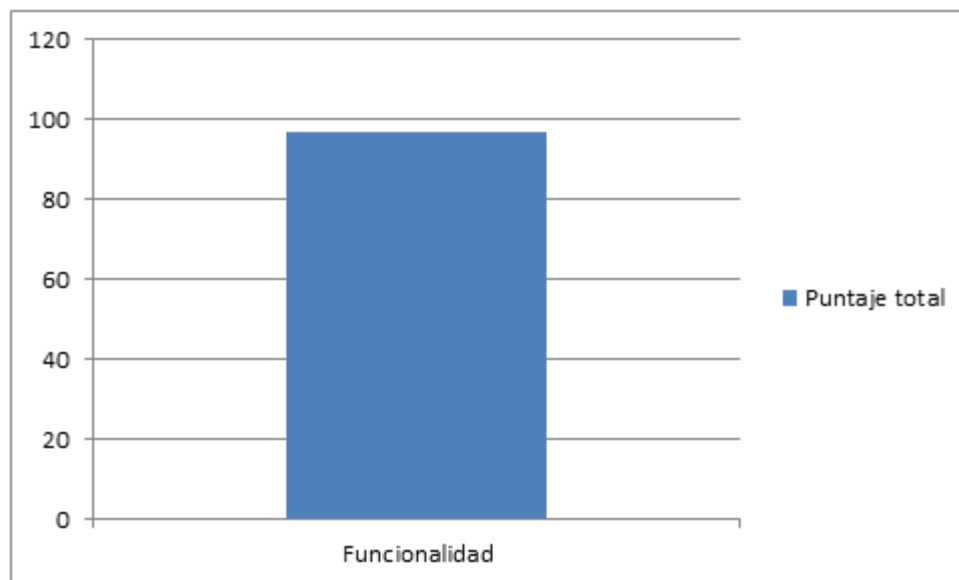


Figura 48. Histograma de resultados para funcionalidad.

La aplicación fue considerada como útil y funcional, sin embargo era posible aumentar el rendimiento de la aplicación. Se consideró que al mejorar los puntos ya mencionados (nivel de dificultad de uso de la aplicación y nivel de ayuda que ofrece la aplicación) el índice de funcionalidad aumentaría.

Posterior a esta primera aplicación del instrumento se procedió a mejorar los problemas detectados, por ejemplo “el nivel de dificultad de uso de la aplicación” y “el nivel de ayuda que ofrece la aplicación”. Posterior de haber optimizado la aplicación, dejando que transcurrieran 27 días, se pidió nuevamente a los 20 sujetos anteriores que utilizaran por 15 minutos más la aplicación y que nuevamente dieran su opinión en la misma encuesta. Los resultados dieron como se muestra a continuación:

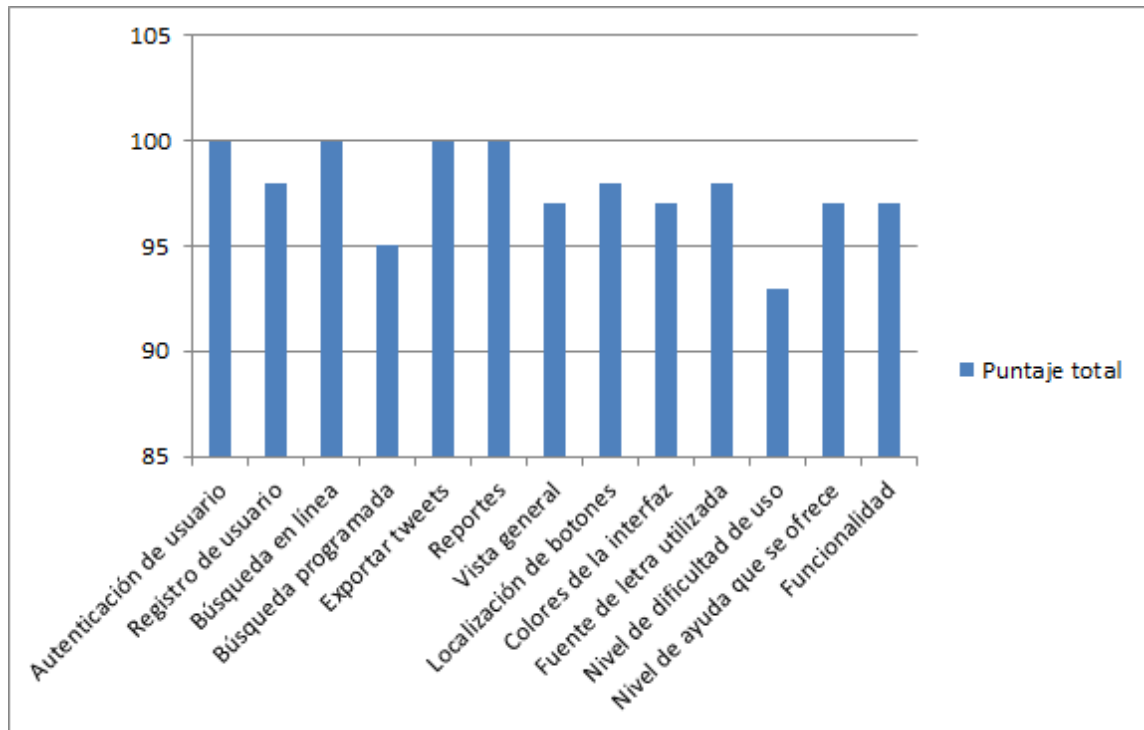


Figura 49. Histograma de resultados para el segundo periodo de recolección de información.

Las brechas detectadas en la primera aplicación fueron superadas en la segunda, lo cual se evidencia al observar los resultados, los cuales obtuvieron casi la puntuación perfecta.

#### 4.6. Cronograma de actividades.

A continuación se muestra a través de la siguiente tabla el cronograma de actividades que se cumplió para el desarrollo de la aplicación según la metodología SCRUM con la cual se trabajó.

Tabla 13. *Cronograma de actividades realizada con la metodología SCRUM.*

| <b>Sprint</b> | <b>Actividades realizadas</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <b>Duración</b>  |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Sprint N° 1   | Análisis detallado de requerimientos. <ul style="list-style-type: none"> <li>• Historia de usuario.</li> <li>• Diagramas de flujo.</li> <li>• Diseño de la base de datos.</li> <li>• Diseño de las interfaces.</li> </ul>                                                                                                                                                                                                                                                                              | <b>4 semanas</b> |
| Sprint N° 2   | Configuración de ambiente. <ul style="list-style-type: none"> <li>• Configuración de la API de Twitter.</li> <li>• Configuración de la base de datos.</li> <li>• Configuración de XAMPP.</li> <li>• Configuración de conexión entre PHP y MongoDB.</li> <li>• Configuración de conexión entre la API de Twitter y PHP.</li> </ul>                                                                                                                                                                      | <b>4 semanas</b> |
| Sprint N° 3   | Desarrollo de la aplicación. <ul style="list-style-type: none"> <li>• Desarrollo de las Interfaces de Usuario.</li> <li>• Desarrollo de la funcionalidad Identificación de usuario.</li> <li>• Desarrollo de la funcionalidad Búsqueda en línea.</li> <li>• Desarrollo de la funcionalidad Búsqueda programada.</li> <li>• Desarrollo de la funcionalidad Exportar Tweets.</li> <li>• Desarrollo de la funcionalidad Reportes.</li> <li>• Desarrollo de la funcionalidad Registrar usuario.</li> </ul> | <b>4 semanas</b> |
| Sprint N° 4   | Realizar pruebas para la certificación de la aplicación.                                                                                                                                                                                                                                                                                                                                                                                                                                               | <b>4 semanas</b> |

#### **4.7. Manual de la aplicación.**

A continuación mostramos los pasos a seguir para la utilización de la aplicación:

- 
- Acceder a la URL asignada a la aplicación, para nuestro ejemplo usaremos “local host/twitter/login.php”. Y nos aparecerá la figura 37.
  - Una vez realizado este primer paso se abrirá la pantalla de inicio en la cual aparecen diferentes opciones. Para comenzar debes colocar en el campo “Palabra o frase a buscar” tal como su nombre lo indica, la información que se desea obtener.
  - Aunado a ello, se debe especificar los parámetros que aparecen en los demás campos, siendo estos : (a) Lugar de origen de los tweets, que permite especificar la zona geográfica de donde proviene el tweet, (b) tipo de resultados, que permite hacer una búsqueda aleatoria (mixto), reciente o popular, (c) Cantidad de tweets a buscar (no pueden ser más de 100) (d) Lenguaje de los tweets (inglés o español) y por último (e) Fecha de culminación de la búsqueda. Estos campos los podemos apreciar en la figura 38.
  - Para el uso de la funcionalidad Búsqueda programada, se deben llenar los mismos datos explicados en el punto anterior ya además los siguientes nuevos campos: (a) Tiempo de duración, en la cual se especifica cuánto tiempo durará la búsqueda, la cual tiene que colocarse en minutos, (b) Fecha y hora del inicio de la búsqueda, en la que se especifica a partir de qué fecha y hora se empezará a realizar la búsqueda, (c) Intervalos de tiempo para realizar la búsqueda, define cada cuanto tiempo se van a realizar las búsquedas dentro del tiempo de duración. Estos campos los podemos apreciar en la figura 39.
  - Posterior a ello, se pulsa la opción de menú “Exportar Tweets” para que se abra la siguiente ventana mostrada en la figura 40. En el campo que aparece, se selecciona la frase o palabra que ya se había buscado con anterioridad, y se hace clic en el botón “Descargar .CSV” este paso permite exportar la data que fue previamente buscada en Twitter en un formato estructurado.

- También, en las opciones del menú se puede hacer clic en “Reportes” que permite realizar un seguimiento de las búsqueda dando información como (a) Número de Tweets almacenados, (b) Usuario que realizó la búsqueda, (c) Fecha de la búsqueda, (d) Frase o palabra buscada y (e) si fueron o no exportados los Tweets. Esta descripción se aprecia en la figura 41.

## **CONCLUSIONES Y REFLEXIONES DEL TRABAJO ESPECIAL DE GRADO**

El desarrollo de esta aplicación les brindan a las organizaciones y a sus Gestores de Redes Sociales la posibilidad de monitorear y analizar las opiniones de sus clientes basándose en los datos obtenidos en la red social Twitter para la toma de decisiones a nivel de mercadeo.

El uso de esta aplicación ofrece numerosas herramientas, conceptos y metodologías que permiten que la generación de la información sea más confiable y sencilla para el Gestores de Redes Sociales y que la obtención de la misma se valga menos de sus propios sesgos.

En este sentido, la automatización de la gestión o manejo de estos datos brinda numerosos beneficios a la organización que los implementa como por ejemplo reducción de tiempo laboral, toma de decisiones de mercado mejor sustentadas.

Aunado podemos dar las siguientes reflexiones, donde se describen las ventajas y limitaciones que se presentaron durante el desarrollo de la aplicación.

- Facilita enormemente la captura y transformación de los datos generados en la red Social Twitter, aumentando la productividad de la empresa a nivel de tiempos y de la calidad de los datos.
- La aplicación entrega un formato de datos estructurados que puede ser compatible prácticamente para cualquier sistema de inteligencia de negocios que lo desee consumir.
- El desarrollo de esta aplicación ayuda y motiva la implementación de nuevas tecnologías que apunta a ser el futuro en las tecnologías de Información.
- Como principal limitante se presentan el número de peticiones que se pueden hacer a través de la API de Twitter, ya que para tener acceso

ilimitado a los datos que posee Twitter en cuanto a sus tweets, se debe realizar un pago por el uso de esta modalidad de servicio.

- Para el mantenimiento y correcta funcionalidad de la aplicación, la organización o empresa que desee aplicarla y/o usarla, debe contar con una persona encargada de hacer un correcto mantenimiento de la base de datos MongoDB, es decir, esta persona debe gestionar los datos almacenados en los mismos, como por ejemplo borrar los datos antiguos o innecesarios.
- Para futuras expansiones o nuevas versiones de la aplicación se recomienda usar también otras opciones de la API de Twitter para potenciar la aplicación.
- También para nuevas versiones de la aplicación se recomienda usar como entorno de desarrollo y ejecución plataformas de 64bit, para lograr explotar al máximo todas las bondades que brinda MongoDB, como lo es por ejemplo la escalabilidad horizontal para aumentar la velocidad de las consultas cuando la cantidad de datos que se manejen sean de un muy alto volumen.

## REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES

Acens (2014). Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar. Acenswhitepapers. Recuperado de <http://www.acens.com/wp-content/images/2014/02/bbdd-NoSQL-wp-acens.pdf>

AERCO & Territorio Creativo. (2009). La función del Community Manager. Recuperado de <http://www.maestrosdelweb.com/images/2010/04/communitymanager.pdf>.

Aguilera, M., Veléz, S. & Echeverria, F. (2012) Implementación de un componente para el préstamo de material bibliográfico digital para la Biblioteca Virtual en el CIB-ESPOL. Recuperado de <http://core.ac.uk/download/pdf/12414311.pdf>

Álvarez, S. García, E & Álvarez, M. (2012). Manual de iniciación a la programación. *Desarrolladorweb.com*. Recuperado de <http://www.infobrandsen.com.ar/archivos/iniciacionprogramacion.pdf>

Ayala, S. (2013).Cuál es el proceso de Marketing. Recuperado de <http://www.todomkt.org.com/2013/04/cual-es-el-proceso-del-marketing.html>

Barranco, R. (2012), ¿Qué es Big Data?. *IBM developerWorks*. Recuperado de <http://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/>

Batini, C., Ceri, S. & Navathe, S. (1994). Diseño Conceptual de bases de datos: un enfoque de entidades-interrelaciones. Delaware, EUA: Addison Wesley

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. et, al. (2001). Doce principios para el desarrollo ágil del software. Artículo en línea. Recuperado de: <http://www.agilemanifesto.org/iso/es/principles.html>. Consultado el 2-01-2014.

BofA Merrill Lynch. (2012). Big Data II – New Themes, New Opportunities, White paper.

Cardoso, A (2010). Base de datos relacional Oracle 10g. Recuperado de <http://es.scribd.com/doc/26782386/Objeto-Relacional-Oracle-10g>

Colomina, E. (1998). Adopción de Sistemas de Información en las PYME. Teoría y evidencia empírica. Tesis doctoral, Facultad de Ciencias Económicas y Doctorales, Universidad de Alicante. Recuperado de <http://rua.ua.es/dspace/handle/10045/3393>.

Cowburn, P. (2015) Lenguaje de programación PHP. Recuperado de <http://php.net/manual/es/index.php>.

Cruz, M. (s/f). Conceptos básicos de bases de datos. Universidad Autónoma del Estado de México. Maestría en Ingeniería Industrial. Recuperado de <http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/bd1.pdf>

Elósegui, T. (2014), Big Data Analytics: El futuro de la analítica digital. Recuperado de <http://tristanelosegui.com/2014/05/05/big-data-analytics-el-futuro-de-la-analitica-digital/>

Fidelity Worldwide Investmen (2012), *Big data: una "revolución industrial" en la gestión de los datos digitales*. Recuperado de [https://www.fondosfidelity.es/static/pdfs/informesfondos/Fidelity\\_ArgInvSXXI\\_BigData\\_Sept12\\_ES.pdf](https://www.fondosfidelity.es/static/pdfs/informesfondos/Fidelity_ArgInvSXXI_BigData_Sept12_ES.pdf)

Kimball, R. (1998). The Data Warehouse Lifecycle Toolkit. (1ª edición) John Wiley & Sons. Inc. New York.

Kimball, R. & Caserta, J. (2008). The Data Warehouse ETL. Toolkit (2ª edición). Wiley. New York.

Laíz, I. (2010). Un nuevo perfil en la empresa: el community manager. *Managersmagazine*. Recuperado de <http://managersmagazine.com/index.php/2010/11/que-es-el-community-manager/>.

Laudon K. & Laudon J. (2008). Sistemas de información gerencial administración de la empresa digital. 10ª Edición. Pearson Educación. México, S.A

Letelier, P. & Sánchez, E. (2013). *Metodologías Ágiles en el Desarrollo de Software*. Grupo Ingeniería del Software y Sistemas de Información (ISSI). Recuperado de <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>

Cano Giner, J. (2007). Business Intelligence: competir con información. *Fundación Cultural Banesto*. Madrid.

Marqués, M. (2011). Bases de datos. Publicaciones Universitat Jaume I. España. Recuperado de <http://www.uji.es/bin/publ/edicions/bdatos.pdf>

MongoDB (2015). Release Notes for MongoDB 3.0. Recuperado de <http://docs.mongodb.org/manual/>.

Orjuela, A & Rojas, M (2008). *Las Metodologías del Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo*. Revista Avances en Sistemas e Informática. 5 2, 1657-7663.

Peña, A. (2006). Inteligencia de Negocios: Una Propuesta para su Desarrollo en las organizaciones. 1ª edición. Instituto Politécnico Nacional. México DF.

Pip.pypa.io (2015). PIP. Recuperado de <https://pip.pypa.io/en/stable/>.

Python (2015). Python's documentation, tutorials, and guides are constantly evolving. Recuperado de <https://www.python.org/downloads/>.

Rangel, W. (2009), Sistemas de Información. Universidad Central de Venezuela. Facultad de Ciencias. Escuela de Computación.

Rob, P., & Coronel, C. (2006). Sistemas de bases de datos, diseño, implementación y administración. Thomson. México DF.

Rodríguez, B. (2010) Estudio de las plataformas tecnológicas para el apoyo, control y seguimiento de los estudiantes en cuanto al desarrollo de las actividades académicas

Rouda, N. (2014). Getting Real About Big Data: Build Versus Buy. *The Enterprise Strategy Group, Inc.* Recuperado de <http://www.oracle.com/us/corporate/analyst-reports/esg-getting-real-bigdata-2228170.pdf>

Sánchez, F. (2013). Big data: los datos como valor de negocio. Recuperado de <http://bigdata.ticbeat.com/big-data-los-datos-como-valor-de-negocio/>

Sinnexus (2012). Data warehouse. Recuperado de [http://www.sinnexus.com/business\\_intelligence/datawarehouse.aspx](http://www.sinnexus.com/business_intelligence/datawarehouse.aspx).

Softeng (2014). Softeng. Metodología SCRUM para desarrollo de software - aplicaciones complejas. Recuperado de <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>.

Stair, R. & Reynolds, G. (2000) *Principios de sistemas de información: enfoque administrativo*. 4ta edición. Thomson. México DF.

Support Twitter (2014). Centro de ayuda de la aplicación Twitter. Recuperado de <https://support.twitter.com/articles/76621>

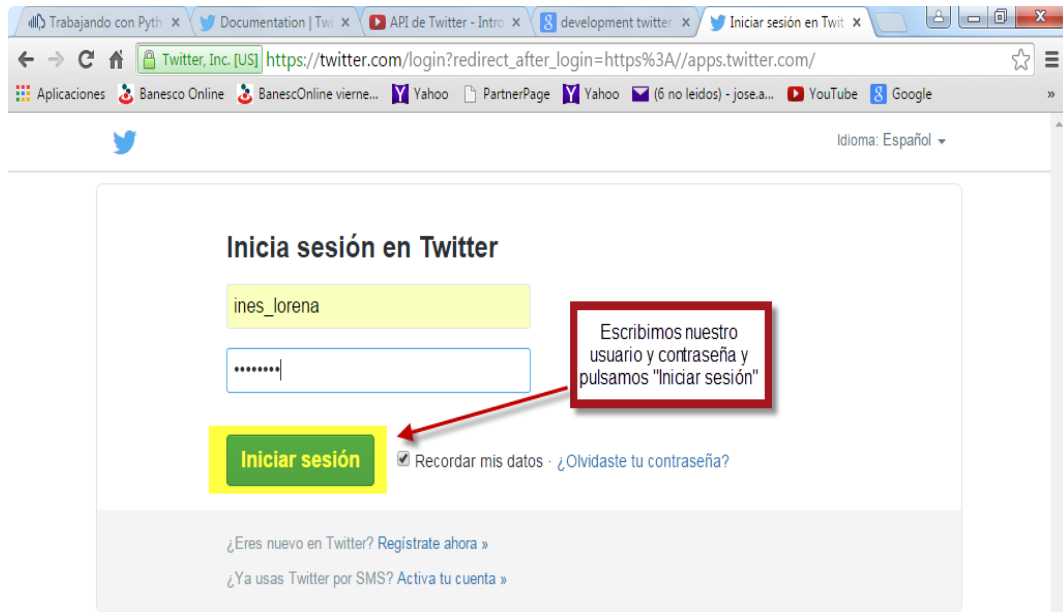
Uñoja, R (2012). Metodologías para el desarrollo de software. Recuperado de <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-desoftware.html>.

William, I. (2007). The Father of Data Warehousing. Inmon Consulting Services. San Diego.

Wolff, C. (2002). Modelamiento multidimensional. Recuperado de <http://www.inf.udec.cl/~revista/ediciones/edicion4/modmulti.pdf>.

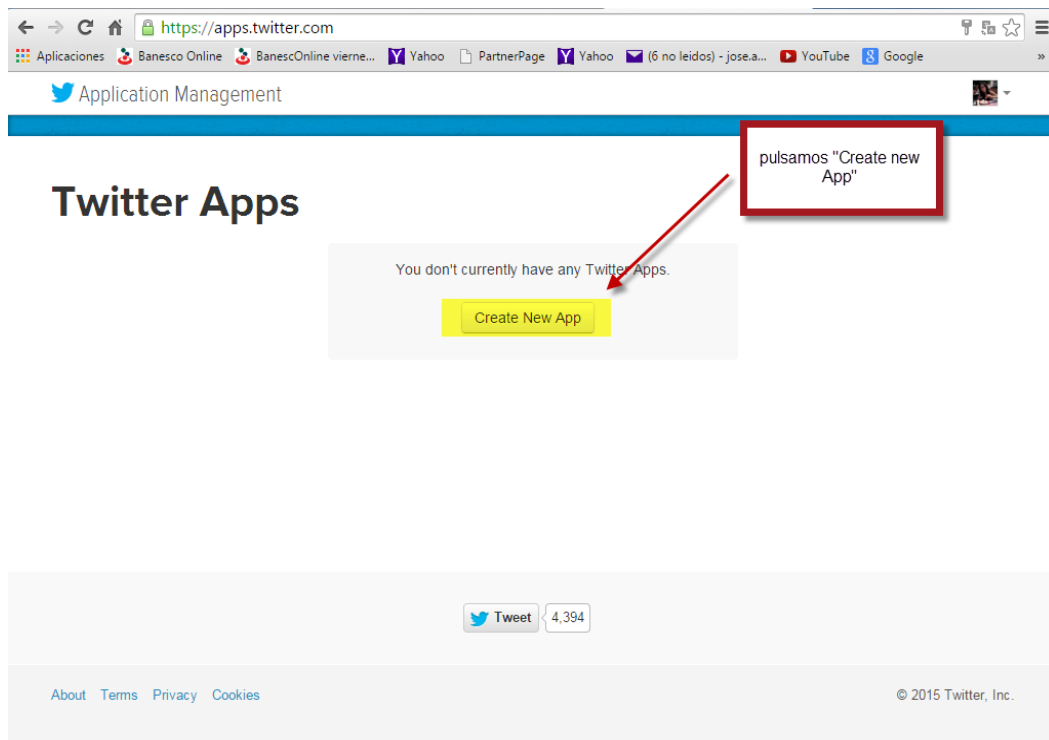
**ANEXO A.**

**Captura de Imagen de <https://apps.twitter.com/>**



**ANEXO B.**

**Captura de Pantalla de una Aplicación de la API de Twitter**



## **ANEXO C.**

### **Captura de Pantalla de los Parámetros a Llenar**



## Create an application

Lenamos los campos marcados en amarillo, aceptamos los terminos y condiciones y pulsamos "Create your twitter application"

### Application Details

**Name \***

Prueba

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

**Description \***

prueba de conexión y almacenamiento con Mongodb

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

**Website \***

www.prueba.com

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

**Callback URL**

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth\_callback URL on the request token step, regardless of this here. To restrict your application from using callbacks, leave this field blank.

### Developer Agreement

written consent of Twitter. Any attempted assignment in violation of this paragraph is null and void, and Twitter may terminate this Agreement. This Agreement does not create or imply any partnership, agency or joint venture. This Agreement will be governed by and construed in accordance with the laws of the State of California, without regard to or application of conflicts of law rules or principles. All claims arising out of or relating to this Agreement will be brought exclusively in the federal or state courts of San Francisco County, California, USA, and you consent to personal jurisdiction in those courts. Despite the foregoing, you agree that money damages would be an inadequate remedy for Twitter in the event of a breach or threatened breach of a provision of this Agreement protecting Twitter's intellectual property or Confidential Information, and that in the event of such a breach or threat, Twitter, in addition to any other remedies to which it is entitled, is entitled to such preliminary or injunctive relief (including an order prohibiting Company from taking actions in breach of such provisions), without the need for posting bond, and specific performance as may be appropriate. The parties agree that neither the United Nations Convention on Contracts for the International Sale of Goods, nor the Uniform Computer Information Transaction Act (UCITA) shall apply to this Agreement, regardless of the states in which the parties do business or are incorporated. No waiver by Twitter of any covenant or right under this Agreement will be effective unless memorialized in a writing duly authorized by Twitter. If any part of this Agreement is determined to be invalid or unenforceable by a court of competent jurisdiction, that provision will be enforced to the maximum extent permissible and the remaining provisions of this Agreement will remain in full force and effect.

Yes, I agree

Create your Twitter application

**ANEXO D.**

**Captura de Pantalla de la Pestaña “Settings”**

The screenshot shows a web browser window with the URL <https://apps.twitter.com/app/8130935>. The page title is "Application Management". A green notification bar at the top states: "Your application has been created. Please take a moment to review and adjust your application's settings." The application name is "Prueba\_big\_data\_mongoDB\_twitter" with a "Test OAuth" button. Below the name are tabs for "Details", "Settings", "Keys and Access Tokens", and "Permissions". The "Settings" tab is highlighted in yellow, and a red box with the text "pulsamos en la pestaña 'Settings'" has an arrow pointing to it. Under the "Settings" tab, there is a Twitter logo, the text "prueba de conexión y almacenamiento con MongoDB", and a link "https://www.facebook.com/ines.nunez.545/about". Below this is the "Organization" section with the text "Information about the organization or company associated with your application. This information is optional." and a table with two rows: "Organization" (None) and "Organization website" (None). At the bottom is the "Application Settings" section with the text "Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform."

Trabajando con Pyth x Documentation | Twi x API de Twitter - Intro x Prueba\_big\_data\_mo x Ines Nuñez x

← → ↻ 🏠 <https://apps.twitter.com/app/8130935> 🌟 ☰

Aplicaciones Banesco Online BanescoOnline viernes... Yahoo PartnerPage Yahoo (6 no leídos) - jose.a... YouTube Google »


🐦 Application Management

Your application has been created. Please take a moment to review and adjust your application's settings.

## Prueba\_big\_data\_mongoDB\_twitter

 Test OAuth

Details **Settings** Keys and Access Tokens Permissions

 prueba de conexión y almacenamiento con MongoDB  
<https://www.facebook.com/ines.nunez.545/about>

### Organization

Information about the organization or company associated with your application. This information is optional.

|                      |      |
|----------------------|------|
| Organization         | None |
| Organization website | None |

### Application Settings

Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.

pulsamos en la pestaña "Settings"

**ANEXO E.**

**Captura de pantalla de la pestaña “Update settings”**



# Prueba\_big\_data\_mongoDB\_twitter

Test OAuth

Details Settings Keys and Access Tokens Permissions

## Application Details

### Name \*

Prueba\_big\_data\_mongoDB\_twitter

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

### Description \*

prueba de conexión y almacenamiento con Mongodb

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

### Website \*

https://www.facebook.com/ines.nunez.545/about

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

### Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth\_callback URL on the request token step, regardless of th here. To restrict your application from using callbacks, leave this field blank.

Allow this application to be used to Sign in with Twitter

Seleccionamos la siguiente opción

## Application Icon

### Change icon



Seleccionar archivo Ningún archivo seleccionado

Maximum size of 700k. JPG, GIF, PNG.

## Organization

### Organization name

The organization or company behind this application, if any.

### Organization website

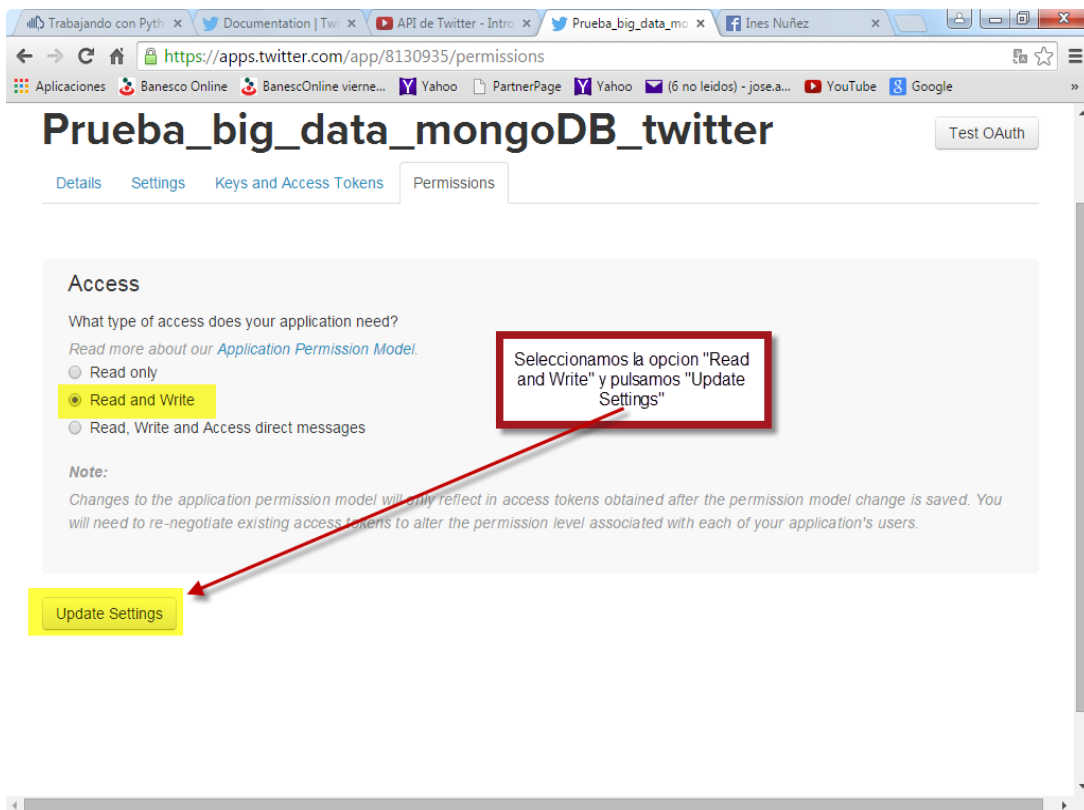
The organization or company behind this application's web page, if any.

Update Settings

Y pulsamos "Update Settings"



**ANEXO F.**

**Captura de Pantalla de la Pestaña “Settings”**



**ANEXO G.**

**Captura de Pantalla de la Pestaña “Keys and Acces Tokens”**

Application Management  

## Prueba\_big\_data\_mongoDB\_twitter

Test OAuth

[Details](#) [Settings](#) [Keys and Access Tokens](#) [Permissions](#)

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

|                              |                                                    |
|------------------------------|----------------------------------------------------|
| Consumer Key (API Key)       | XggamU5QludVMvEE33NTmmdK5                          |
| Consumer Secret (API Secret) | GDSxaEvesnETwkWmYN0BFEOX3q86BYN6RK0rLMbhmeMijeSfK0 |
| Access Level                 | Read-only (modify app permissions)                 |
| Owner                        | ines_lorena                                        |
| Owner ID                     | 293084045                                          |

### Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

### Your Access Token


You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

### Token Actions

[Create my access token](#)

pulsamos "Create my acces token"



[About](#) [Terms](#) [Privacy](#) [Cookies](#) © 2015 Twitter, Inc.

**ANEXO H.**

**Captura de Pantalla de las Credenciales para que Twitter Permita Utilizar sus Recursos.**



### Status

Your application access token has been successfully generated. It may take a moment for changes you've made to reflect. [Refresh](#) if your changes are not yet indicated.

## Prueba\_big\_data\_mongoDB\_twitter

Test OAuth

[Details](#) [Settings](#) [Keys and Access Tokens](#) [Permissions](#)

Las credenciales resaltadas en amarillos son las mas importantes en tener en cuenta

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

|                              |                                                           |
|------------------------------|-----------------------------------------------------------|
| Consumer Key (API Key)       | XggamU5QludVMvEE33NTmmdK5                                 |
| Consumer Secret (API Secret) | GDSxaEvesnETwkWmYN0BFEOX3q86BYN6RK0rLMbhmeMljeSfK0        |
| Access Level                 | Read and write ( <a href="#">modify app permissions</a> ) |
| Owner                        | ines_lorena                                               |
| Owner ID                     | 293084045                                                 |

### Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

### Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

|                     |                                                   |
|---------------------|---------------------------------------------------|
| Access Token        | 293084045-3VHQQTxjafghANtd7fWFUBifYDhnfyBfURkrYac |
| Access Token Secret | N0q0FeYhosiBxPQgGans9pHOUEX8Prrr2TPR3ZaLKIHCT     |
| Access Level        | Read and write                                    |
| Owner               | ines_lorena                                       |
| Owner ID            | 293084045                                         |