UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA WEB DE ANÁLISIS DE OPINIONES EN TWITTER INTEGRANDO ALGORITMOS DE DATA MINING

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

ANDRÉS ALEJANDRO CÓRDOVA GALLEGUILLOS

PROFESOR GUÍA:
JUAN DOMINGO VELÁSQUEZ SILVA

MIEMBROS DE LA COMISIÓN:
FRANCISCO JAVIER MOLINA JARA
EDGARDO JULIO SANTIBAÑEZ VIANI

Este trabajo ha sido parcialmente financiado por el proyecto CORFO 13IDL2-23170

SANTIAGO DE CHILE
2015

RESUMEN DE LA MEMORIA PARA OPTAR AL
TITULO DE: Ingeniero Civil Industrial
POR: Andrés Alejandro Córdova Galleguillos
FECHA: 21/12/2015
PROFESOR GUIA: Juan Domingo Velásquez Silva

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA WEB DE ANÁLISIS DE OPINIONES EN TWITTER INTEGRANDO ALGORITMOS DE DATA MINING

El objetivo General de esta memoria de título es "Diseñar y Construir un prototipo funcional de sistema de análisis de opiniones en Twitter integrando algoritmos de Data Mining".

Este trabajo se enmarca en el proyecto “OpinionZoom – Plataforma de análisis de sentimientos e ironía a partir de información textual en redes sociales para la caracterización de la demanda de productos y servicios”. Este es un proyecto de I+D aplicada concursado por InnovaChile de CORFO dirigido por el Web Intelligence Centre (WIC) de la Universidad de Chile.

Este proyecto intenta satisfacer una necesidad de las organizaciones; la de conocer mejor a su público demandante y a sus opiniones con respecto a la marca, los productos o servicios que ofrece, o sobre algún tópico en particular. Si bien es frecuente que se hagan estudios de mercado para intentar resolver estas inquietudes, éstos resultan caros y presentan sesgos de distintas clases.

Por otro lado, existe mucho contenido en la Web generado por usuarios de diferentes servicios, y a cada minuto se agregan miles de gigabytes de este tipo de datos. Solo en Twitter, red social de microblogging, se generan aproximadamente 340.000 *tweets* por minuto. Si se consideran las poderosas herramientas desarrolladas en el último tiempo en el campo de Data Mining, existe un gran costo de oportunidad al no aprovechar las información de primera fuente que se puede obtener de allí para responder a las búsquedas de las organizaciones.

Esta memoria de título pretende comprobar que es posible crear un sistema de análisis de opiniones en Twitter integrando algoritmos de Data Mining que por separado detecten entre otras cosas la orientación sentimental de una opinión, la influencia de los usuarios de Twitter y los intereses de estos usuarios.

La solución a implementar es un prototipo funcional que permite revisar y proyectar la funcionalidad de la aplicación final que tendrá el proyecto en cuestión. Una de las novedades de este trabajo es la construcción de un Data Warehouse para colectar las opiniones vertidas en Twitter y proveer de información útil para la gestión al usuario del sistema.

Se logran los objetivos al diseñar y construir un sistema de análisis de opiniones en Twitter integrando algoritmos de Data Mining a nivel de prototipo, mostrando resultados coherentes y satisfactorios, que instan a nuevas mejoras con vistas a un producto final. Se valida de esta forma la hipótesis y se aporta con una novedosa aplicación de un Data Warehouse que ocupe los datos que gratuitamente otorga Twitter para la mejor gestión de productos y servicios de una organización.

A Jesús

Agradecimientos

A Dios, por quien todo lo bueno ha sido creado.

A mi abuela Teresa, quien me ha acompañado con perseverancia y fortalecido con su alimento y su cariño. Ella me enseñó a ser responsable, trabajador y a hacer las cosas con amor.

A mis padres, quienes me han sostenido con amor incondicional y me han dado tres hermosas hermanas, íntimas compañeras de viaje, a quienes también les agradezco por su paciencia y abnegado cariño.

A Patricia, por soportarme con amor por tantos años.

A la Pastoral de Ingeniería, que me ha acompañado en este largo peregrinar por la facultad.

A la Salita Sur, que fue un segundo hogar. A Felipe, Rocío, Nico, Romi, Pipe, Pangui, Juano y Naty.

A la Salita Norte, expertos en tecnología y admirables en bondad. En especial a la ayuda generosa e inteligente de Jorge y Yerko, sin la cual este trabajo no podría ser lo que es.

A tantos amigos que me han apoyado con palabras de aliento y me han sostenido con sus oraciones. En especial a Andrea, correctora oficial de esta memoria.

Al profesor Edgardo Santibañez, quien me enseñó a pensar y diseñar un trabajo de título.

Al profesor Francisco Molina, quien con su apoyo siempre amable ha llevado a este trabajo a su mejor versión y ha reconfortado esta extensa labor.

Al profesor Juan Velásquez por darme la oportunidad de realizar mi memoria bajo su tutela y por iluminarme con sus experiencias de vida y sus valores, tanto dentro del aula durante la carrera como en los Nomikai durante este último tiempo de mayor cercanía.

Tabla de Contenido

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Las comunicaciones en el mundo han cambiado vertiginosamente en las últimas décadas, sobre todo con la masificación del uso de Internet a nivel global. Con la aparición de la Web 2.0 y el crecimiento de los medios sociales, los usuarios están aportando cada vez más información y contenidos de manera gratuita. Esto engendra una oportunidad para las organizaciones en cuanto pueden tomar mejores decisiones al obtener mayor información sobre sus clientes. En particular pueden ampliar o mejorar su oferta, pronosticar mejor su demanda, incrementarla e ir a buscar prospectos gastando menos recursos y obteniendo mejores resultados.

En el caso particular de las empresas, para conocer a sus clientes estas incurren en grandes inversiones para realizar estudios de mercados, que por lo demás no son certeros, debido a lo costoso que es llegar a una masa crítica para obtener niveles de error aceptables, los sesgos de los destinatarios al participar de encuestas y la mala aplicación de las técnicas. Soslayando estas dificultades, el producto de un buen estudio de mercado es estático y puede quedar obsoleto en el mediano plazo. Si se obtiene información a partir de los datos que gratuitamente se generan en la Web, se puede caracterizar a los clientes, actuales y potenciales, de forma dinámica y continua.

En Chile existen más de 5.000 grandes empresas que cuentan con perfiles en redes sociales. Muchas de ellas no sospechan el tremendo potencial que esto posee, que es la posibilidad de conocer a sus usuarios, permitiendo un mejor pronóstico de demanda, ofrecer mejores productos u ofrecerle más personalizadamente a un usuario especifico los productos que son de su interés. Las que si conocen los beneficios que el buen uso de estas herramientas pueden traer no tienen, en general, las capacidades de obtener la información relevante a partir de la gran cantidad de datos que poseen, por lo que intentan leer las opiniones de un pequeño porcentaje de usuarios para hacerse una idea parcial de los tipos de consumidores existentes y sus preferencias.

El desafío es por tanto crear sistemas avanzados e integrados que permitan recopilar, organizar y extraer conocimiento desde grandes bases de datos, que aprovechen los datos generados diariamente por millones de consumidores en la Web.

El Web Intelligence Centre o WIC , del departamento de Ingeniería Industrial de la Universidad de Chile, ha estado investigando temas de procesamiento de texto y análisis de opiniones hace ya algunos años, pero los trabajos investigativos han tenido objetivos individuales desde su

concepción por lo que no es posible capitalizar estos esfuerzos con un servicio completo y real que responda cabalmente a los requerimientos de las empresas, aun teniendo todos los insumos científicos necesarios para hacerlo al mejor nivel.

Se hace necesario entonces integrar o unir estos algoritmos para formar un sistema tecnológico concreto que, recibiendo como entrada opiniones en la Web, logre producir información valiosa para un cliente real dinámicamente, de tal forma que éste, conociendo mejor a sus propios o potenciales clientes, pueda gestionar su interacción con ellos de mejor manera, incrementando sus utilidades.

1.1. Antecedentes

Este trabajo se enmarca en el Proyecto OpinionZoom, financiado por INNOVA CORFO, el cual se realiza en el centro de inteligencia Web de la Universidad de Chile, Web Intelligence Centre.

1.1.1. WIC

El Web Intelligence Centre (WIC) [1] es el centro de inteligencia Web de la Universidad de Chile. Éste nació hace aproximadamente una década y ha estado siempre a cargo del profesor Juan Velásquez, phd.

El WIC cuenta con numerosas publicaciones en revistas científicas internacionales, produce varias memorias de pregrado y tesis de posgrado anualmente y crece año a año gracias a los aportes de practicantes y profesionales, principalmente ingenieros, que llegan a hacer crecer el Centro.

El WIC es miembro del Web Intelligence Consortium[1], el que agrupa distintos centros de investigación y desarrollo en Inteligencia Web de todo el mundo.

En su página Web el WIC declara su misión, su visión y sus objetivos:

- Misión: Desarrollar investigación de frontera en el campo de Tecnologías de Información creando nuevas soluciones para abordar problemas complejos de ingeniería utilizando herramientas basadas en la Web de las Cosas
- Visión: Ser un líder a nivel internacional en la investigación de tecnologías de información y comunicaciones aplicadas a la resolución de problemas del mundo real.
- Objetivos:
  - Publicar en las principales revistas, conferencias y editoriales relacionadas con Web Intelligence.
  - Proveer un servicio profesional, excelente y rápido para todos nuestros clientes.
  - Dictar cursos de orientación práctica acerca de las Tecnologías de Información y su aplicación en los negocios.

[1] `http://wi-consortium.org/`, 26 de octubre de 2015

El WIC dicta el contexto de investigación y desarrollo en que se enmarca el proyecto OpinionZoom y por consiguiente el presente Trabajo de Título.

1.1.2. Proyecto OpinionZoom

El Proyecto OpinionZoom es un proyecto del WIC financiado por INNOVA CORFO bajo el código 13IDL2-23170, según concurso adjudicado en 2013. Este es un concurso Línea 2, nomenclatura usada para indicar investigación más desarrollo (I+D) aplicada. El Objetivo General de este proyecto es desarrollar una plataforma de análisis de sentimientos e ironía a partir de información textual en redes sociales para la caracterización de la demanda de productos y servicios. OpinionZoom nació a partir de la investigación [2] de miembros del WIC sobre la aplicación del modelo de minería de opiniones basado en aspectos de Bing Liu [3] en la industria del turismo en Chile [4]. Para dicha investigación se construyó y utilizó una herramienta novedosa de minería de opiniones [5] cuyo impacto dio pie para que se concibiera un proyecto mayor. Este trabajo de título viene a ser un prototipo funcional de mayor alcance, y el primero de OpinionZoom con un fin comercial ulterior. Es un avance fundamental considerando los alcances que luego se explicitan.

Objetivos de OpinionZoom

El proyecto OpinionZoom presenta un objetivo general y cuatro específicos [6] .

Objetivo General de OpinionZoom

El objetivo general de OpinionZoom es el siguiente:

Extraer y analizar opiniones, sentimientos e ironía a partir de la información textual en redes sociales para la caracterización de la demanda de productos y servicios

Objetivos Específicos de OpinionZoom

Existen cuatro objetivos específicos para cumplir el objetivo general:

1. Construir un repositorio de palabras claves etiquetadas (corpus) sobre la base del análisis lingüistico de los textos de una comunidad afín usuaria de redes sociales.

2. Adaptar e integrar algoritmos de data mining para extraer patrones que permitan interpretar los datos y generar modelos de predicción de demanda de productos a partir de la información textual en redes sociales.

3. Diseñar, construir y evaluar un prototipo de plataforma de software que integre los algoritmos, los modelos y el repositorio para la predicción de la demanda de productos a partir del análisis de sentimientos e ironía a partir de la información textual en redes sociales.

4. Valorizar el mercado y la propiedad intelectual, y definir una estrategia para el empaquetamiento y transferencia de la tecnología.

Visión, Misión y Valores de OpinionZoom

OpinionZoom no solo presenta objetivos, sino que además tiene una misión que cumplir, una visión que le orienta en su accionar y valores que respetará siempre [7], con miras a proyectarse más allá del período de financiamiento ofrecido por CORFO en esta etapa.

Visión

OpinionZoom tiene claro a donde quiere llegar en el mediano plazo, y declara qué quiere ser en el futuro:

Ser la empresa de más alta reputación en la industria, reconocida por entregar un servicio de análisis de opiniones preciso, confiable y rápido.

Misión

En la misión OpinionZoom declara el servicio que realiza, o en otras palabras, cuál es su aporte para sus clientes:

Entregar un servicio de extracción y análisis de opiniones, sentimientos y polaridades, que contribuye a acercar a las distintas organizaciones proveedoras de servicios y productos con los consumidores, propiciando así mejores resultados a las empresas y aportando a la calidad de vida de la comunidad en general a través de la mejor satisfacción de sus necesidades.

Valores

OpinionZoom publica también sus valores para que todos los involucrados en el proyecto y en especial los usuarios del servicio, sepan dónde se encuadran las acciones necesarias para cumplir con la misión arriba declarada.

Respeto por la privacidad e identidad de las personas

Veracidad y transparencia en el tratamiento de los datos con el objeto que los resultados obtenidos sean confiables

Servicios de OpinionZoom

Los servicios de OpinionZoom son una salida del modelo de negocios, el cual fue creado en una tesis del magíster en ingeniería de negocios [7]. La lista no es detallada pero los servicios que debe ofrecer la plataforma se agregan en tres secciones: Inteligencia de clientes, Trending Alert e Impacto de campañas.

- INTELIGENCIA DE CLIENTES
  - Identificación: ¿Quiénes y cómo son mis clientes? Algoritmo Líderes de Opinión
  - Conocimiento: ¿Qué les gusta? Algoritmo Interés Complementario
  - Escucha: ¿Qué y cómo están hablando de tu marca y la competencia? Métricas de Medición

- TRENDING ALERT
  - Alertas de Temas Hot: reclamos puntuales, reclamos generalizados, contingencia
  - Seguimiento de reclamos mediante: análisis de opiniones ex post y generación de encuestas automáticas.
  - Análisis de Trend LifeCycle: duración, peaks/valles, velocidades
- IMPACTO DE CAMPAÑAS
  - Impacto de Campaña:
  - Medición de métricas antes/después de acción de marketing de tu marca y su competencia.
  - Conceptos asociados a la campaña y la marca durante la campaña.
  - Validación de Productos: Polaridad y conceptos asociados a un producto determinado. Uso de hashtag e imagenes para capturar información.
  - Otros servicios: Automatización de reportes personalizables de RRSS. Apoyo en investigaciones científicas, etc.

El prototipo resultante del presente trabajo de título sólo aborda los primeros 2 servicios: Inteligencia de Clientes y Trending Alert.

1.2. Descripción del Trabajo de Título

El trabajo consiste en diseñar y construir un prototipo funcional de un sistema de análisis de opiniones vertidas en Twitter. Para este prototipo se tomará cada *tweet* como una opinión. Para crear este sistema es necesario ocupar algoritmos de Data Mining, ya sea de *Opinion Mining* u otros específicos para Twitter. Éste sistema de análisis ofrecerá dos servicios para potenciales clientes: caracterizar usuarios y generar alertas a partir de las opiniones vertidas respecto a un tema.

Para ambos servicios se necesitarán indicadores y métricas que ayuden a los futuros clientes de OpinionZoom a entender a su público objetivo para gestionar mejor sus propios servicios. La respuesta tecnológica por antonomasia para este problema es un Data Warehouse, como se explicará en los capítulos venideros. Se construirá entonces un Data Warehouse, que recopile datos y ponga a disposición del cliente indicadores orientados a la gestión.

Un Data warehouse incluye la visualización de información útil a partir de la base datos. En este caso la entrega de estos servicios se hará a través de una página Web. El trabajo incluye entonces, además del Data Warehouse, la creación de un sitio Web para mostrar los resultados.

Finalmente, el presente trabajo de título comprende la evaluación de los resultados obtenidos tanto del sistema en sí como del análisis que es capaz de arrojar.

1.3. Justificación

Este trabajo de título es esencial para el proyecto OpinionZoom, financiado por CORFO. Las primeras etapas ya están realizadas: el estado del arte, la construcción del corpus en español y el desarrollo de algoritmos. Lo que viene naturalmente es integrar estos algoritmos a un sistema de análisis de opiniones en Twitter.

Los algoritmos desarrollados en el centro tienen actualmente un valor puramente académico. Este trabajo de título desarrolla su potencial otorgándoles valor comercial al construir un producto útil para clientes del mercado, que requieran innovaciones tecnológicas en esta área.

Por último la industria chilena tiene necesidad de servicios como los que ofrecerá OpinionZoom, pues las empresas gastan mucho dinero en estudios de mercados que tienen muchos sesgos. Con los servicios ofrecidos podrán categorizar mejor a sus clientes, con todos los beneficios que ello implica.

1.4. Objetivos

Este trabajo de título tiene un objetivo general y cuatro objetivos específicos.

1.4.1. Objetivo General

Diseñar y Construir un prototipo funcional de sistema de análisis de opiniones en Twitter integrando algoritmos de Data Mining.

1.4.2. Objetivos Específicos

1. Diseñar un sistema de análisis de opiniones en Twitter.
2. Construir un Data Warehouse que recopile datos y permita poner a disposición del cliente indicadores orientados a la gestión.
3. Construir una plataforma Web que ofrezca un servicio de análisis de opiniones.
4. Evaluar el rendimiento del prototipo de sistema.

1.5. Hipótesis de Investigación

La hipótesis de investigación que se pretende validar y que orienta el presente trabajo de título es la siguiente:

Es posible crear un sistema de análisis de opiniones en Twitter integrando algoritmos de Data Mining que por separado detecten entre otras cosas la orientación sentimental de una opinión, la influencia de los usuarios de Twitter y los intereses de estos usuarios.

1.6. Alcances

El sistema construido es un prototipo funcional y es solo una parte del proyecto OpinionZoom. De este modo quedan fuera de este trabajo etapas previas y posteriores que son realizadas por otros memoristas y tesistas del WIC. Explícitamente quedan fuera de este trabajo de título:

- La construcción de un repositorio de 'Tweets'. El centro ya cuenta con una base de datos, llamada 'La Gorda', que será explicada más adelante. Un crawler de Twitter la alimenta con los tweets emitidos por usuarios chilenos.
- API de Polaridad: Esta interfaz de programación fue creada en el WIC. Su valor reside en el algoritmo, el cual fue creado en primera instancia con fines de investigación y su rendimiento es sobresaliente con respecto al estado del arte de clasificación subjetiva en español.

Además quedan fuera del alcance de este trabajo:

- Evaluación económica del proyecto.
- Búsqueda de un cliente real.
- Test de usabilidad de la página Web.

Lo que sí se espera de este trabajo está en la siguiente sección de Resultados Esperados.

1.7. Resultados Esperados

Los resultados esperados tienen directa relación con los objetivos específicos. Los resultados 'a)' y 'b)' corresponden al primero, 'c)' al segundo, 'd)' al tercero y 'e)' y 'f)' al último.

a) Selección de algoritmos a usar

b) Ubicación geográfica y lógica del sistema y sus componentes

c) Data Warehouse

d) Página Web funcional

e) Evaluación de resultados de análisis para un caso ficticio

f) Evaluación de rendimiento del sistema

1.8. Metodología

Se aborda este trabajo como uno de desarrollo de software [8], y desde allí se escoge una metodología de desarrollo ágil. Se explican las implicancias de esta elección y el método particular seleccionado, Scrum.

1.8.1. Metodología Ágil

Una metodología de desarrollo es una forma validada de crear un software y consiste en (1) una filosofía de desarrollo, (2) múltiples herramientas, modelos y métodos, (3) una buena documentación y (4) una organización que la sustente. Para el desarrollo de este sistema se ocupará la metodología ágil de desarrollo. Las metodologías ágiles de desarrollo comienzan a surgir poco antes de la década de los '90 en contraposición a las metodologías tradicionales de desarrollo de software de programación, tales como CMM (Modelo de Madurez de Capacidades o *Capability Maturity Model* en inglés) o el desarrollo en cascada (*Waterfall* en inglés).

Los métodos tradicionales se centran en la planificación y ejecución secuencial de etapas de desarrollo (con poco solapamiento en general), en la correcta documentación del trabajo desarrollado y en el cumplimiento de la carta gantt y del presupuesto. En general sus etapas son análisis, diseño, desarrollo, pruebas, implementación y mantienimiento, pudiendo ser éstas iterativas o en espiral según la metodología ocupada. Éstos métodos clásicos han sido muy útiles para grandes proyectos de desarrollo de software, pero para el presente trabajo de diseño y construcción de un prototipo es claramente más conveniente ocupar una metodología ágil de desarrollo. Una de las primeras razones, es que por ser un trabajo que se alimenta de investigación, no se puede hacer un acuerdo inmutable previo como exigen los métodos tradicionales. Es decir, no se conocen *a priori* ni las entradas ni menos las salidas del sistema a desarrollar. Las metodologías ágiles tienen la gracias de contar con iteraciones cortas, incluso de un par de semanas, pudiendo planificar el trabajo hasta 2 veces por mes. En cuatro ítems se puede describir el enfoque de las metodologías ágiles:

Las metodologías ágiles de desarrollo siguen un manifiesto de cuatro valores [9]: (1) Individuos e interacciones sobre procesos y herramientas; (2) Software funcionando sobre documentación extensiva; (3) Colaboración con el cliente sobre negociación contractual; (4) Respuesta ante el cambio sobre seguir un plan. Para una mayor comprensión se explica cada punto de la lista.

1. Individuos e interacciones sobre procesos y herramientas: la prioridad es la calidad profesional del equipo y la entrega temprana y continua. Esto hace frecuente la interacción con el cliente y posibilita la retroalimentación oportuna.

2. Software funcionando sobre documentación extensiva: la prioridad es satisfacer al cliente. Con las metodologías de desarrollo tradicional, si el cliente llega a buscar su producto en la

mitad del plazo estipulado se encontrará con media documentación, la cual debe ser extensiva. Con una metodología ágil el cliente se encontraría con el código bien documentado (que es parte importante de la documentación) y con el software funcionando en una parcialidad de sus posibilidades.

3. Colaboración con el cliente sobre negociación contractual: la prioridad es participar con el cliente, lo que permite que el desarrollador entienda el valor de lo que hace y pueda proponer también al cliente a partir de su experiencia y de las posibilidades que observa.

4. Respuesta ante el cambio sobre seguir un plan: se aceptan cambios de requerimientos, e incluso son deseables, pudiendo adaptar al software a los eventos.

Algunas de las metodologías ágiles de desarrollo de software más conocidas son Scrum (1986), KANBAN, Cristal Transparente (en inglés *Crystal Clear*), programación extrema (en inglés *eXtreme Programming* o *XP*, 1996), desarrollo de software adaptativo (*feature driven development*) y Método de desarrollo de sistemas dinámicos (*Dynamic Systems Development Method* o *DSDM*, 1995).

Scrum

Scrum [10] es un proceso para manejar proyectos complejos; un proceso que es fácil de aprender, junto con sus prácticas, artefactos y reglas. Es el método de desarrollo ágil escogido.

Se ocupa Scrum por la naturaleza del proyecto, sus requerimientos y por las ventajas de la metodología, la primera ya mencionada: lo fácil que resulta aprenderla. El proyecto a desarrollar es complejo, su itinerario es desconocido, se necesita tener entregables constantemente y se construye y modifica en base a los requerimientos del usuario, siempre cambiantes porque las investigaciones del WIC van ofreciendo nuevas posibilidades para el software.

Por otro lado Scrum hace crecer al equipo, el cual se está preparando para otros proyectos en el futuro. Los problemas no se resuelven individualmente, sino que colectivamente, pues dentro de la filosofía del método está que los problemas se resuelven mejor y más rápido con muchas mentes. En los equipos que trabajan según este paradigma no es raro ver a dos programadores sentados en el mismo computador, cuando uno solamente está con las manos en el teclado y el segundo sólo observa. Al contrario de la intuición que puede surgir, esto hace el desarrollo más rápido pues una segunda visión inmediata advierte al momento errores de tipeo y contrasta ideas para encontrar las mejores alternativas.

Scrum permite que el producto se defina entre el cliente y el desarrollador, lo cual es muy conveniente en este caso porque el equipo puede encontrar nuevos requerimientos para mejorar el producto. Esto responde a otro paradigma en que se basa Scrum, que es la búsqueda del mejoramiento continuo. Dentro de los procesos de Scrum se descubrirán otras bondades de la metodología atingentes al proyecto.

Dentro de las herramientas, modelos y métodos que tiene una metodología de desarrollo de software, Scrum define Roles, Artefactos y Procesos.

Roles

En Scrum hay solamente 3 roles: el Dueño del Proyecto, el Equipo y el *ScrumMaster*. El Dueño del Proyecto es el que representa los intereses de todos los afectados por el resultado final. Está a cargo del *backlog*, que se explicará más adelante pero que es en pocas palabras la lista de requerimientos de usuario. En este caso, el rol de Dueño del Proyecto es ocupado por el director del WIC.

El Equipo es el que desarrolla las funcionalidades. Debe gestionarse y organizarse a si mismo y debe poder trabajar como equipo, no simplemente repartiéndose las tareas. El éxito de cada iteración y del proyecto en general es responsabilidad de todo el equipo colectivamente.

El *ScrumMaster* es el responsable de implementar Scrum correctamente y asegurarse de que todos sigan las reglas y prácticas propias de la metodología, dentro de la cultura organizacional.

Hay una jerga propia de Scrum, que impone Schwaber, para separar a quienes están comprometidos con el proyecto y quienes solo se ven afectados por él. Los primeros son llamados 'cerdos' y los segundos 'pollos'. La prioridad es darle autoridad y responsabilidad a los 'cerdos' y ahorrar interferencias inecesarias de los 'pollos'.

Procesos

Una de las características más llamativas de Scrum es el celo que se tiene en las ceremonias. Las ceremonias son las reuniones que dan comienzo a un proceso. Se tienen varios subprocesos dentro del gran proceso que enmarca todo el proyecto.

El primer paso es obtener una visión del producto, lo que se traduce en la primera ceremonia: la reunión icinial. Aunque la idea primeramente sea vaga, ésta dará una lista de requerimientos de usuarios, llamada *Backlog* para el proyecto.

Luego todo el trabajo es hecho en *Sprints*. Los Sprints son iteraciones mensuales de trabajo. Éstos son planificados con una ceremonia mensual. Se recomienda que esta reunión, a la que asiste el Dueño del Proyecto, no dure más de ocho horas.

Diariamente está la ceremonia que da comienzo al trabajo de todo el Equipo. Esta reunión, liderada por ScrumMaster, dura no más de 15 minutos. En ella cada miembro del equipo dice qué ha estado haciendo, qué va a hacer ese día y qué impedimentos, o bloqueantes, ha encontrado al intentar hacer su trabajo. De esta forma se sincroniza el trabajo de todos los miembros y se planifican reuniones excepcionales que algunos miembros requieran con otros miembros del equipo.

A fin del mes está la ceremonia de revisión, con la presencia del Dueño del Proyecto. En menos de cuatro horas el equipo presenta lo que ha desarrollado en los últimos 30 días y se determina el trabajo que debe planificarse para el próximo mes.

Finalmente está la ceremonia retrospectiva entre el ScrumMaster y el Equipo. En ésta se revisa el seguimiento de la metodología y se proponen modificaciones para hacer las ceremonias más provechosas y en general el trabajo más eficiente.

La aplicación de este proceso en el presente trabajo presenta varios cambios en el tiempo, adap-

tándose al equipo que, por la naturaleza del WIC, es siempre cambiante. Así, se tiene que la ceremonia inicial de Sprint se hace cada 2 meses, la ceremonia diaria se hace más corta y la ceremonia de revisión no considera todas las áreas del desarrollo, si no que sólo lo prioritario a revisar por el Dueño del Proyecto cuando éste tuviera tiempo. Sin embargo, acomodar el desarrollo al contexto, al equipo y a los plazos es parte del método, que permite estas flexibilidades.

Artefactos

Scrum propone el uso de tres herramientas, bautizadas como 'artefactos', que sirven para planificar y controlar el desarrollo del proyecto. Intentan ser simples para facilitar su manejo para todos los roles. A continuación se listan estos artefactos:

- Backlog del producto: es la lista de los requerimientos del sistema a desarrollar. Nunca está completo pues es dinámico. Sus registros iniciales corresponden a la primera idea que se tuvo en la planificación, pero pueden ir cambiando. El Dueño del Proyecto es el responsable de los contenidos y su priorización. En este caso, el ScrumMaster tomó esta última atribución parcialmente.
- Backlog de Sprint: es la lista de los requerimientos del sistema para el mes al que pertenece el Sprint. De aquí nacen las tareas para el equipo. Cada requerimiento es atendido por tareas de no más de 16 horas de duración estimada, ni menos de 4.
- Incremento de funcionalidad: Es la tabla que contiene los requerimientos funcionales y su estado de realización, considerando su documentación correspondiente. Si el Dueño del Proyecto necesita el producto en cualquier momento, puede contar con los requerimientos que están en estado de 'Completado'. Este artefacto no fue ocupado en este trabajo pues los requerimientos no eran tan cuantiosos ni requerían mayor documentación.

Nótese nuevamente que esta metodología fue aplicada adaptada al contexto. Entre otras circunstancias se cuenta el menor tamaño del equipo y del proyecto y la variación de los miembros del equipo. Sin embargo, la mayor novedad metodológica del trabajo, fue hacer conciliar la metodología ágil de desarrollo con la construcción de un Data Warehouse, cuya metodología propia responde más bien a las tradicionales.

1.9. Estructura del Trabajo de Título

El presente trabajo se divide en 8 capítulos. Esta sección termina el capítulo introductorio. El segundo capítulo es el marco teórico que introduce el contexto científico y tecnológico pertinente en el que se ha desarrollado el trabajo. Luego, los 5 capítulos siguientes corresponden a cada uno de los objetivos específicos, abordándose el tercer objetivo específico en 2 capítulos (capítulos 5 y 6). Se concluye al fin en el capítulo 8 proponiendo también posibles trabajos futuros. Le siguen a la conclusión la bibliografía y los anexos.

Capítulo 2

Marco Referencial

El presente capítulo pretende dar a conocer algunos elementos importantes para contextualizar al lector y así también familiarizarlo con ciertos conceptos propios del campo de las tecnologías, relevantes para una comprensión cabal del trabajo desarrollado y en particular del título de éste. Este marco referencial comprende dos apartados. La primera parte será el marco conceptual, que abarca 6 secciones, y la segunda el marco teórico compuesto por las secciones Prototipo Funcional y Sistema de Análisis.

2.1. API

La mejor forma encontrada para encapsular los algoritmos de tal manera que entreguen los resultados esperados de ellos es a través de una interfaz de programación de aplicaciones, o mejor conocida como API por sus siglas en inglés (Application Programming Interface).

Básicamente una API permite contar con un servicio cuyo funcionamiento se desconoce, o no es visible para el usuario que para estos efectos es el sistema de análisis en desarrollo. De esta forma se sabe que hay una entrada y una salida y simplemente se ocupan los resultados sin necesidad (ni capacidad) de meterse en lo que sucede al interior de la API. Las APIs creadas se acceden a través de la Web, bajo el mismo dominio que la página web de OpinionZoom, que se explicará más adelante.

2.2. Twitter

Twitter es una plataforma Web de *microblogging*, concepto que se refiere el servicio de enviar y publicar mensajes de texto muy breves. Twitter acota el largo de estos mensajes a 140 catacteres, pudiéndose además de texto enviar pequeños dibujos representativos de una idea común llamados *emojis*. A los mensajes se les llama *tweets*, que será un término ocupado por el resto del trabajo.

Twitter.com fue creado en 2006 en Estados Unidos y funciona actualmente en ese país, específicamente su sede central se encuentra en San Francisco, California. Es considerada una de las redes sociales más populares del mundo. Actualmente cuenta con más de 304 millones de usuarios activos[1].

Se ha propuesto en los alcances que sólo se ocupa esta red social para este trabajo. Esto se explica por las cualidades de Twitter: su uso frecuente en Chile, la gran cantidad de datos y el fácil acceso a éstos a través de sus APIs. Por otro lado, el WIC ha desarrollado y está desarrollando sus propias APIs para esta red social, por lo que resulta natural su elección.

2.2.1. Twitter en Chile

En Chile hay miles de usuarios activos en Twitter. Identificarlos no es tarea fácil, consiguientemente cuantificarlos tampoco. Si bien es cierto que algunos explicitan su país, ciudad o pueblo de origen en el campo que Twitter destina para que el usuario publique su ubicación, esta información puede ser modificada a discreción, haciendo difícil para una máquina determinar con certeza la ubicación real del usuario. Por ejemplo un habitante de Santiago de Chile puede poner en el campo Ubicación: 'Santiago de Chile', 'Stgo', 'SCL' u otra nomenclatura inventada por él. Para identificar a los usuarios chilenos se realizó una heurística: se clasifican como chilenos los usuarios que siguen a algún usuario famoso chileno como un comunicador social o un medio de comunicación exclusivamente chileno.

2.2.2. *Streaming* API

Twitter tiene tres APIs[2]: AdsAPI, REST API y Streaming API. La primera es para publicidad, la segunda para buscar tweets históricos y con la tercera se obtienen los tweets que se emiten a cada instante. Esta última es la API que alimenta la gran base de datos 'La Gorda' que se presenta en el siguiente capítulo. Funciona a través de un código en Java y unas llaves de autentificación. Arroja *tweets* en forma de JSON.

2.2.3. Influencia en Twitter

Este sistema quedaría gravemente incompleto si no se considera la influencia del opinante. Naturalmente a los clientes les interesarán más las opiniones (tanto positivas como negativas) de las personas más influyentes. De ese modo querrán agradar más a los influyentes y evitar sus juicios negativos hacia ellos o sus productos.

Existen algunas aproximaciones para medir la influencia de un usuario en Twitter, entre ellas funciones que consideran la cantidad de seguidores y usuarios seguidos, la frecuencia de uso de la

[1] `http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/`, 26 de octubre de 2015

[2] `https://dev.twitter.com/overview/documentation`, visitado el 26 de octubre de 2015

Red Social o la cantidad de 'retweets' que obtienen sus publicaciones. Otra aproximación puede venir de modelar las relaciones de Twitter como un grafo y usando algoritmos tipo Page Rank asignar una influencia por usuario. Incluso en el WIC se han hecho buenos intentos para medir la influencia por la generación de contenido interesante y los tópicos de opinión [11].

Hay dos problemas que surgen en este trabajo al asignarle un indicador de influencia a los usuarios y a los tweets. El primero es que no todo tweet es una opinión, por lo que para ser certeros necesariamente hay que descubrir el contenido para intentar asignarle una influencia. El segundo problema es que la influencia de un tweet, aunque relacionada, no debería ser la misma que la influencia de quién la emite. Esto sucede cuando un especialista que tiene mucha influencia en su área emite una opinión respecto de un área en la que es completamente lego. Naturalmente esta opinión debería tener menos influencia que las opiniones que emite sobre su especialización. Ambos problemas son asumidos como una oportunidad de mejora para próximas iteraciones del producto desarrollado y serán abordados por la API de influencia correspondiente que a la entrega de este trabajo continúa en desarrollo en el WIC.

2.2.4. *Klout* API

Para subsanar la carencia de una métrica propia de influencia se recurre temporalmente a la API de Klout[3], una empresa estadounidense de tecnología que ofrece esta interfaz a desarrolladores, gratuita pero con restricciones. No se le pueden hacer más de 10 consultas por segundo ni más de 20.000 por día. Klout entrega una puntuación entre 0 y 100 para cada usuario de Twitter que ellos tienen en sus registros, donde 0 representa a un usuario sin influencia y 100 a un usuario absolutamente influyente. Como dato ejemplificador, el presidente de los Estados Unidos de América, Barack Obama, tiene 99 puntos de influencia Klout.

Además del nivel de influencia, Klout ofrece a través de su API una lista de intereses de los usuarios, lo que constituye uno de los servicios de OpinionZoom: interés complementario.

Pese a que actualmente ambas salidas, de influencia y de intereses, son parte indispensable para OpinionZoom, Klout debe ser sustituido completamente. Los problemas con los servicios ofrecidos por Klout son los siguientes:

1. No se sabe cómo son construidos sus algoritmos, por lo tanto no se tiene control sobre la precisión de ellos. En un caso hipotético, entendiendo que los usuarios de Twitter desean sentirse importantes [12], Klout podría recibir dinero de clientes para aumentar la influencia de ciertas cuentas de Twitter lo que falsearía los datos que recibe OpinionZoom. Esto haría no tan verídicos los resultados lo que va en directa contraposición a los valores de OpinionZoom.

2. No se tiene control sobre la disponibilidad de la API. Si por algún motivo caprichoso Klout decide bajar su API, OpinionZoom se quedaría sin conocer la influencia y los intereses de nuevos usuarios, o no podría actualizar sus datos.

3. Los intereses, aunque coherentes, no son precisos. Como dato curioso, se tiene que a la gran mayoría de los usuarios, sin importar la nacionalidad, están relacionados con el interés

[3] `https://klout.com/s/developers/home`, 26 de octubre de 2015

'Richard Nixon', lo que es absolutamente incongruente con la realidad. Además, los intereses genéricos a veces resultan de poca utilidad. Por ejemplo a la mayoría de los usuarios chilenos se los relaciona con el interés 'Chile', lo que, aunque real, puede ser poco útil.

Es por ello que se hace necesaria la pronta construcción de APIs de influencia y de intereses en el WIC, cuyo proceso sea conocido, cuyos servicios estén siempre disponibles sin restricciones artificiales y cuyos resultados sean precisos y de confianza.

2.3. JSON

JSON[4] es un formato ligero de intercambio de datos. Es una forma de transmitir información. Es el acrónimo para *JavaScript Object Notation*. A parte de ser liviano, un JSON tiene la característica de que es fácil de escribir y de leer por los humanos y es fácil de generarlo e interpretarlo por una máquina. Sus ventajas lo han hehco reemplazar al formato XML y organizaciones que manejan gran cantidad de información, como Twitter o Facebook[5], lo ocupan para guardar y transmitir datos.

Básicamente se trata de un arreglo de pares 'nombre:valor', como se puede ver en la figura 2.1. Este JSON corresponde a un *tweet* encontrado en la base de datos 'La Gorda'.

Figura 2.1: JSON correspondiente a un *tweet*


```
{"id": 587728880137535488, "lang": "es", "text": "Bolsa de
Santiago sigue al alza y SQM anota el mayor avance diario en 5
meses http://t.co/vrCGZzDHe5", "place": null, "source": "<a
href=\"http://24farandula.com\" rel=\"nofollow\">Farandula
App</a>", "user_id": 2539714687, "entities": {"urls": [{"url":
"http://t.co/vrCGZzDHe5", "indices": [78, 100], "display_url":
"goo.gl/HH0bxc", "expanded_url": "http://goo.gl/HH0bxc"}],
"trends": [], "symbols": [], "hashtags": [], "user_mentions":
[]}, "truncated": false, "created_at": "Mon Apr 13 21:27:39
+0000 2015", "coordinates": null, "filter_level": "low",
"timestamp_ms": "1428960459156", "possibly_sensitive": false,
"in_reply_to_user_id": null, "in_reply_to_status_id": null,
"in_reply_to_screen_name": null}
```


Fuente: Elaboración Propia

La base de datos 'La Gorda' estaba en Postgresql, pero cuando fue migrada a SOLR los JSON cambiaron su formato, para que la búsqueda fuera más directa.

2.4. Data Warehouse

El manejo de datos históricos ha ido cambiando a través de los siglos. A mediados del siglo XX aparecen los computadores y los sistemas de información, en particular primitivos sistemas de

[4] `http://www.json.org/json-es.html`, 26 de octubre de 2015

[5] `http://www.sitepoint.com/facebook-json-example/`, 26 de octubre de 2015

apoyo a la decisión. En un comienzo los problemas de integridad y disponibilidad de los datos se solucionaba con una redundancia de 'archivos maestros' (master files) que a su vez traían otros problemas: la necesidad de sincronizar la información constantemente, la complejidad de mantener los programas, la dificultad para desarrollar nuevos programas y la necesidad de gran cantidad de *hardware* para almacenar todos los archivos maestros. Un par de décadas después, intentando capear estos inconvenientes, las organizaciones empiezan a usar modelos de extracciones de los datos originales para funcionar en los departamentos y luego extracciones de las extracciones. Entre más madura y grande la organización, mayores los problemas de esta arquitectura que fue llamada 'Arquitectura naturalmente evolucionada'. Los problemas de ésta arquitectura son tres: la justificada falta de credibilidad que tiene el usuario que más lejos se encuentra del núcleo, la enorme disminución en la productividad, sobre todo al intentar crear programas que recopilen información de las distintas fuentes repartidas en la organización; y, por último, la dificultad de obtener información de esos datos dispersos. A finales del siglo XX aparecen una nueva forma de concebir una arquitectura de procesamiento de información dentro de una organización y el término 'Data Warehouse', de la mano de Bill Inmon [13].

Un data warehouse (almacén de datos en español) es un sistema que extrae, limpia, ajusta y suministra datos originales a un almacenamiento de datos dimensional para apoyar e implementar consultas y análisis para la toma de decisiones [14].

Según Ralph Kimball, líder en la industria de Data Warehouse, el concepto de Data Warehouse se malentiende, por lo que especifica lo que no es un Data Warehouse (DW). Un DW no es:

- un producto. Un DW no se puede comprar, pues incluye analisis, manipulación de datos, modelamiento dimensional y acceso a la data. No existe un producto que pueda contener estos elementos.
- un lenguaje. No se trata de un código ni un lenguaje que se puede enseñar a usar como XML, SQL o JAVA.
- un proyecto. Para construir un DW se necesitan uno o varios proyectos. Es mejor pensarlo como proceso que como proyecto.
- un modelo de datos. Un modelo de datos no hace un DW. Se descarta esta idea pensando en un modelo de datos sin ETL y por lo tanto sin datos.
- una copia de un sistema transaccional. Algunos creen que un sistema que haga reportes de su sistema transaccional ya es un DW, pero la estructura del almacenamiento es distinta.

Independientemente de la metodología de desarrollo del sistema del que el Data Warehouse es parte, la construcción de un DW tiene su propia secuencia de pasos. Típicamente las cuatro grandes etapas son: Requerimientos, Arquitectura, Implementación y Pruebas y lanzamiento. A continuación se explican brevemente las dos primeras etapas, que luego serán vistas en detalle en los capítulos sucesores junto con la tercera y la cuarta.

2.4.1. Requerimientos

Es el primer paso y definirá todo el trabajo posterior. Los requerimientos son de diversa índole. Los primeros son las necesidades de negocio, que no se pueden definir sólo por una persona, sino

que se recomienda entrevistarse con diferentes actores de la organización y de modo especial con los usuarios finales del DW. Los requerimientos de conformidad guardan relación con los datos y la información en sí, y aseguran la veracidad, el respaldo y el flujo de los datos durante el proceso de extracción, transformación y carga (ETL, por sus siglas en inglés[6]). Otros requerimientos apuntan a la seguridad, a la integración de los datos repartidos en distintos sistemas, a la baja latencia de datos, etcétera. Uno de los requerimientos más presentes por los usuarios finales son las interfaces, las cuales deben ser simples y fáciles de entender. También se tiene como requisito general no construir un Data Warehouse que requiera habilidades que no existen en la organización. Si no existe quien maneje JAVA en la organización, no se puede hacer un módulo que lo requiera como lenguaje.

2.4.2. Arquitectura

La decisión de la arquitectura es fundamental y temprana. Ella definirá la forma del DW y el proceso de ETL. Hay dos decisiones importantísimas que aqui se toman, la primera de ella es si ocupar una herramienta ya hecha para hacer el ETL o programar el ETL desde cero. Para el presente trabajo se optará por el segundo camino. Las ventajas del primero son, entre otras, el buen manejo de la 'metadata' que presentan estos programas, la documentación automática, el buen rendimiento para grandes cantidades de datos y funcionalidades de balance de carga para distintos servidores. Una herramienta así estaría completamente sobredimensionada para el tamaño del DW a construir. Además, se valoran más las ventajas de optar por la segunda vía. Éstas son, entre otras, que las tecnicas de programación orientada a objetos ayudan a hacer las transformaciones consistentes y verificables, los programadores están en la organización y probablemente se queden allí, no se carga con los límites de habilidades y conocimientos de un terceros y se cuenta con flexibilidad ilimitada. La segunda decisión importante que se toma en esta etapa de arquitectura, es si considerar o no un área de preparación de datos (DSA[7]). Un DSA está a cargo del equipo que hace el ETL y sólo depende de ellos. En ningún caso se pueden obtener vistas de ella ni estará disponible para los usuarios finales. Esta opción surge de la resolución del conflicto entre 2 objetivos excluyentes en la práctica:

- Llevar los datos desde la fuente hasta el punto final lo más rápido posible
- Tener la habilidad de recuperar los datos en caso de una caída sin tener que reiniciar el proceso por completo.

Se verá en el capítulo siguiente que esta tensión será resuelta de dos formas distintas: se creará un DSA para los usuarios de Twitter, pero no para los *tweets* en sí.

2.5. Proceso de Extracción de Conocimiento

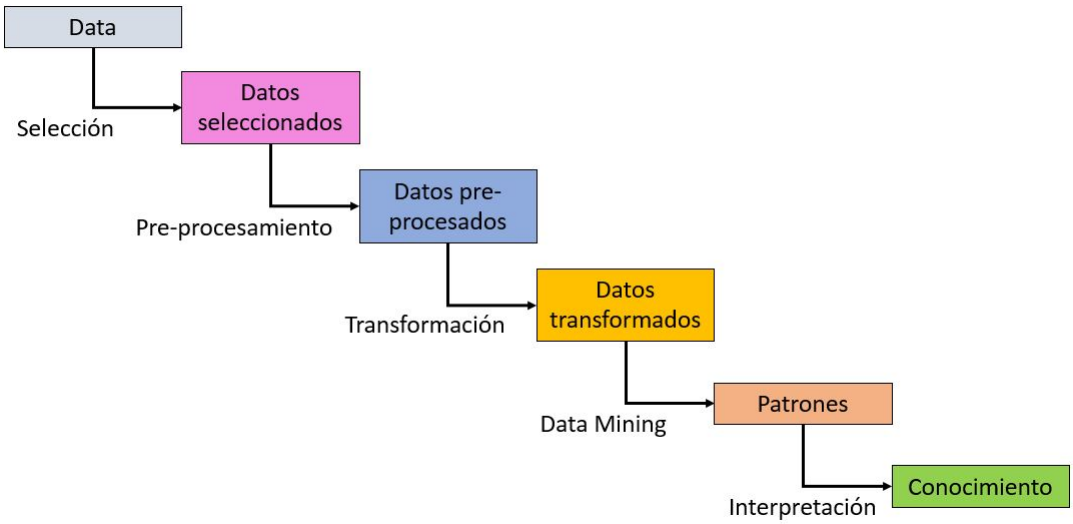
Será importante separar tres conceptos de uso común que se manejarán intencionadamente este trabajo como conceptos semánticamente distintos: datos, información y conocimiento [15]:

[6]Extract, Transform and Load
[7]*Data Staging Area*

- Datos: son valores que no explican nada y que no orientan a una acción y por tanto no ayudan a la toma de decisiones. Por ejemplo, para el prototipo en construcción las opiniones vertidas en Twitter son datos, pues por sí mismas no aportan nada nuevo ni se puede emprender ninguna gestión.
- Información: es un conjunto de datos procesados, el cual tiene un significado. Este significado puede venir de un contexto, una categorización, un cálculo o una agregación. La información puede cambiar los juicios o los comportamientos de un receptor. En el mismo ejemplo, una opinión sobre una marca es un dato, pero varias opiniones que se categorizan como positivas puede significar que los productos de la empresa están teniendo buena recepción.
- Conocimiento: es la mezcla de información con la experiencia que sirve para incorporar iterativamente nuevas experiencias y nueva información. El conocimiento es útil para la acción y se origina y aplica en la mente de las personas 'conocedoras'. Organizacionalmente, el conocimiento no sólo está en documentos, sino que en procesos formales o no formales resultantes de un aprendizaje previo. Para transformar información a conocimiento se puede comparar elementos, predecir consecuencias, buscar conexiones y conversar con otros portadores de conocimiento. Siguiendo con el ejemplo, un analista que vea que después de una campaña publicitaria la polaridad de las opiniones en Twitter respecto a su marca ha aumentado, identificará la campaña como una causa de este aumento y propondrá repetir la campaña para mejorar los resultados de la empresa.

Hoy hay sobreabundancia de datos, gracias a las nuevas tecnologías de información y comunicación. Por otra parte, se necesita conocimiento para tomar decisiones. El proceso KDD (*Knowledge Discovery in Databases* o Proceso de Extracción de Conocimiento, como se le conoce en español) viene a salvar esta brecha entre los datos sobreabundantes y la necesidad de conocimiento. Las etapas del proceso KDD [16], en su forma más general, se pueden ver en la figura 2.2

Figura 2.2: Proceso KDD



Fuente: Elaboración Propia

En el título de este trabajo se propone usar algoritmos de Data Mining, etapa del proceso KDD. Data Mining (o Minería de Datos en español) es el campo de las ciencias de la computación que

se refiere al desarrollo de algoritmos y modelos matemáticos para extraer patrones y obtener conocimiento útil [17]. En este se ocupan algoritmos de Data Mining, es decir, algoritmos que utilizan métodos de inteligencia artificial, aprendizaje automático, métodos estadísticos y sistemas de bases de datos para obtener algún indicador útil para el sistema en construcción. En el capítulo siguiente se muestran los algortimos utilizados.

2.5.1. Web Opinion Mining

Una opinión es una creencia, juicio, o forma de pensar acerca de algo. Es lo que alguien piensa sobre una cosa en particular[8]. Las opiniones encuentran asidero en la experiencia del opinante. Por ejemplo, una persona puede formarse una opinión sobre una película luego de verla o porque tuvo experiencias con películas del mismo género, del mismo director o con los mismos actores. Por eso un consumidor puede buscar opiniones de consumidores anteriores para tomar decisiones, la cual en este caso podría ser ir a ver la película o recomendarla. En el ejemplo de la película, el proceso KDD se hace de forma automática, intuitiva y transparente, pero finalmente a partir de los datos, que pueden ser opiniones vertidas en un "Blog"de la Web, se llegó a un conocimiento que invita a una acción, a tomar una decisión. Con la sobreabundancia de datos en la Web, las organizaciones necesitan extraer conocimiento de las opiniones para saber qué características de sus productos o servicios mejorar, descubrir nuevas oportunidades de negocio, predecir mejor la demanda, etcétera.

Opinion Mining es el campo encargado de extraer opiniones de texto sin estructura combinando técnicas de NLP (procesamiento de lenguajes naturales) y ciencias de la computación [18]. Es entonces la etapa de Data Mining aplicada a texto (*Text Mining*) cuando este texto representa una opinión. Web Opinion Mining no es nada más qué Opinion Mining sobre opiniones veertidas en la Web.

Opinion Mining es también conocido como Sentiment Analysis (Análisis de Sentimientos en castellano) [19] pues en el análisis de una opinión se puede identificar y clasificar el contenido emocional, subjetivo, opinado. En este trabajo, se considerará todo *tweet* como una opinión y se hará el supuesto fuerte de que cada opinión tiene sentimientos implícitos, pudiendo ser éstos negativos, neutros o positivos. Frente a la variedad de sentimientos se considera la polaridad como esta actitud que resume los sentimientos escondidos en la opinión.

2.6. Tecnologías Utilizadas y Herramientas de Trabajo

- Lenguajes
  - JAVA
  - PHP
  - SQL
  - HTML
  - VOLT

[8] `www.merriam-webster.com/dictionary/opinion`, 26 de octubre de 2015

- Gestores de Bases de Datos (DBMS)
  - MySQL
  - MariaDB
  - Postgresql
  - SOLR
- Entorno de desarrollo
  - NetBeans
- Servidor independiente
  - XAMPP
- Repositorio
  - Bitbucket
- Servicios Web
  - AWS (Amazon Web Services)
- Editor de texto
  - Sublime Text
- Browser
  - Mozilla Firefox
- Administrador de Tareas
  - Asana

2.6.1. Justificación de Tecnologías Utilizadas y Herramientas de Trabajo

A continuación se justifica el uso de las tecnologías y herramientas escogidas exponiendo sus bondades y ventajas.

Java

Java es un lenguaje de programación. Es uno de los más populares según el ranking que hace la compañía TIOBE software[9]. Entre sus cualidades más valiosas se encuentra la concurrencia, el ser de propósito general y el ser orientado objetos [20]. Estas tres características son necesarias para el proyecto desarrollado. La concurrencia se necesita para ejecutar distintos procesos como el de *streaming* de tweets y el generación de alertas, que correrán simultáneamente. Como es de propósito general puede tanto establecer conexión con las bases de datos como hacer cálculos y ejecutar rutinas programadas en el tiempo. Por último el que sea orientado a objetos ayuda a la reutilización de código en el mismo proyecto pues los objetos definidos van cambiando sus atributos a medida que son requeridas funcionalidades específicas de cada clase.

Otra ventaja de Java es que es independiente de la plataforma, es decir que corre en cualquier hardware. De esta forma el código funciona tanto en el computador en el que fue desarrollado como en los servidores de Amazon que fueron contratados para contener el Data Warehouse.

[9] `http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html`, 26 de octubre de 2015

Java fue escogido además por ser el lenguaje enseñado en la facultad de ciencias físicas y matemáticas de la Universidad de Chile al memorista y por estar excelentemente documentado por millones de usuarios en todo el mundo.

PHP

Para construir la página Web se usará PHP [21], por la familiaridad con el lenguaje (contenido visto en la carrera), la documentación abundante, por ser gratuito, abierto (*open source*), simple, intuitivo y porque se integra muy bien a través de XAMPP con MySQL. Además tiene una curva de aprendizaje muy baja, sus entornos de desarrollo son de rápida y fácil configuración, tiene fácil despliegue (paquetes totalmente autoinstalables), fácil acceso a bases de datos y una comunidad muy grande.

SQL

SQL (*Structured Query Language* en inglés) es un lenguaje estructurado de consulta [22]. Aparte de las consultas CRUD (*Create, Read, Upload, Delete*) tiene manejo de cálculos algebraicos sencillos.

Con este lenguaje se puede accesar a diversos gestores de bases de datos como PostgreSQL, MySQL y MariaDB, todas ellas ocupadas en este proyecto.

HTML

HTML (*HyperText Markup Language* en inglés) es un lenguaje interpretado para la construcción de páginas Web [23]. Pese a que no se cuenta con documentos con formato HTML dentro de la página resultante de este trabajo, este lenguaje sí se ocupa en los que tienen extensión tanto '.php' como '.volt'.

Volt

Volt es un lenguaje de programación [24] que intenta estar entre un nivel intermedio entre un lenguaje plenamente interpretado (como HTML) y más de cómputo (como PHP). Su objetivo es hacerle más fácil la tarea al programador, con funciones propias del lenguaje. Las vistas de la página Web están escritas en este lenguaje.

MySQL

MySQL es un sistema de administración de bases de datos para bases de datos relacionales [25]. MySQL es gratuito y viene instalado por defecto con el servidor también gratuito e independiente

llamado XAMPP, que se presenta más adelante. Es robusto, escalable y de confianza. Sus usuarios se multiplican, pues en su versión básica es gratuito. Tiene una interfaz amigable en PHP llamada phpmyadmin, corre en diversos sistemas operativos y es potenciado con el uso de 'MySQL Workbench', también gratuito.

Se ocupa en este proyecto porque viene muy de la mano con Apache, un servidor HTTP de codigo abierto, en XAMPP, facilitando el uso de PHP, y con éste el desarrollo de la página Web.

MariaDB

Los mismos creadores de MySQL crearon también MariaDB [26], que es otro sistema de administración de bases de datos para bases de datos relacionales. La creación de ésta fue fruto de la compra de la primera por la empresa de tecnología Oracle. Con esta compra, MySQL dejó de ser *Open Source*, lo que motivó a los fundadores a crear un sistema mejor que el primero que mantuviera el sello colaborativo, siendo consecuentes con su filosofía. MariaDB mantiene la misma escalabilidad y es tan robusto y confiable como su predecesor. Las ventajas de MariaDB[10] sobre el primero son principalmente la velocidad, los nuevos comandos que resultan muy útiles (particularmente para la eliminación de tablas, registros, índices e incluso consultas corriendo) y una disminución sustantiva de las alertas y errores que presenta constantemente MySQL. Además el ser *Open Source* ya es una ventaja para los usuarios, pues está constantemente creciendo en nuevas funcionalidades y en el rendimiento de las ya existentes.

PostgreSQL

PostgreSQL es un sistema administrador de bases de datos para bases de datos objeto-relacional. Es *Open Source* y ha sido desarrollada por más de 15 años. Tiene una arquitectura probada y poderosa en términos de integridad y rectitud en el tratamiento de datos [27]. Es uno de los sistemas gratuitos con mejor reputación y corre en una variedad de sistemas operativos, incluidos Ubuntu y Windows, que son los ocupados en este proyecto. Tiene muy buen manejo de llaves (primarias, foráneas, únicas) y funciones útiles y robustas para manejar variables temporales (timestamp, date, etc.). También tiene muy buena documentación. Se ocupa en este proyecto por su excepcional manejo de JSON, tipo de datos que se explicará más adelante. Esto es posible porque una de sus características es que puede tener atributos de gran tamaño.

Solr

Solr es altamente fiable, escalable y tolerante a fallos. Al consultar proporciona indexación distribuida, replicación y equilibrio de carga [28]. También tiene recuperación automática cuando se cae y configuración centralizada.

Su ventaja reside en la rapidez para hacer consultas *Create* y *Read*, es decir, insertar y leer registros. La base de datos más grande del proyecto se cambió de PostgreSQL a Solr porque lo que

[10] `https://mariadb.com/kb/en/mariadb/mariadb-vs-mysql-features/`, 26 de octubre de 2015

necesita hacer es insertar los datos que recibe de Twitter y a la vez alimentar a la base de datos más pequeña, la del sitio www.opinionzoom.cl, pero no se modifican ni borran filas.

Es un motor de código abierto y por lo tanto gratuito. El notorio aumento de rendimiento se ve en el capítulo de resultados.

NetBeans IDE

NetBeans IDE es un ambiente de desarrollo integrado o IDE en inglés (Integrated Development Environment) para Java. También soporta otros lenguajes, pero para el presente trabajo sólo se ocupa para el código en Java. NetBeans IDE permite desarrollar programas en Java de forma fácil y rápida. Una de sus ventajas radica en la magnitud de la red de usuarios y desarrolladores en todo el mundo, promovido por su gratuidad y por ser de código abierto. La versión de NetBeans IDE utilizada es la 8.0.2.

NetBeans IDE es el ambiente de desarrollo oficial de Java 8 [29]. Posee un editor, un analizador de código y convertidores. Tiene muchas características, plantillas y herramientas útiles. En su riqueza constantemente creciente, NetBeans impone el standard de su rubro. NetBeans destaca en la edición de texto y en la organización de proyectos.

NetBeans es mucho más que un editor de texto. Dentro de sus funcionalidades está la identación de líneas, 'match' de palabras y paréntesis, el realce de código sintáctica y semánticamente y recomendaciones acertadas de generación y completitud de código.

NetBeans IDE permite administrar grandes proyectos de forma eficiente y ordenada. Por como organiza y muestra la información en ventanas laterales se puede tener una mirada general clara del Proyecto, aunque hayan muchísimas carpetas y archivos y millones de líneas de código. También tiene herramientas de integración, pudiendo compartir el trabajo desarrollado por Git. De esta manera un nuevo desarrollador puede entrar a editar el trabajo y entender el código que encuentra pues está estructurado y bien organizado, por lo que le será fácil navegar por el Proyecto y ponerse a trabajar en él.

XAMPP

XAMPP es un servidor independiente de plataforma. Es un software libre que contiene MySQL, Apache e intérpretes para PHP y Perl. Tiene licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas [30]. XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X. Su nombre es un acrónimo de 'cualquier sistema operativo' (representado con X), Apache, MySQL, PHP y Perl.

Bitbucket

Bitbucket es un servicio de *hosting* o almacenamiento web de proyectos que usan *Mercurial* o *Git*. Bitbucket [31] tiene una versión gratuita, que es la que se usa en este proyecto. Bitbucket está escrito en Python y usa Django como framework en su página web.

Bitbucket se usa para almacenar todo el código del proyecto en Java. Una característica ventajosa es lo bien aliniado que está con NetBeans. Desde allí se puede subir el proyecto entero al repositorio. Luego, desde una consola, se puede subir el proyecto al servidor de Amazon y desde allí mismo echarlo a correr.

Amazon Web Services

Amazon Web Services (AWS) es una colección de servicios de computación en la nube (también llamados servicios web) [32]. Estos servicios pertenecen a la empresa Amazon.com. Es considerado uno de los más grandes, y de los pioneros, siendo sus competidores Microsoft Azure y Google Cloud Platform.

De todos los servicios ofrecidos por AWS, el proyecto Opinion Zoom ocupará *EC2: Virtual Servers in the Cloud*, *Route 53: Scalable DNS and Domain Name Registration* y *SES: Email Sending Service*. El segundo es sólo para relacionar el nombre de dominio (www.opinionzoom.cl) con los servidores correspondientes. El último se usará para poder enviar emails a los usuarios, tanto para el registro en el sitio como para el sistema de alertas. El presente prototipo, sin embargo, sólo usará los dos primeros.

Sobre los servidores arrendados al servicio EC2, en un comienzo se usó el más pequeño ofrecido por AWS, de 'Propósito General', de un procesador y de 1GB de memoria RAM, pero luego, por los requisitos de rendimiento, se mejoró a uno de 3,75GB de memoria RAM. Esta es una de las características más satisfactorias de AWS: se pueden cambiar los recursos rápidamente, pagando sólo lo que se usa, haciendo el servicio plenamente escalable.

Sublime Text

Sublime Text es el editor de texto elegido por sobre otros para este trabajo en su parte concerniente al desarrollo de la página Web y su unión con el Data Warehouse. Para estos efectos, se programa con 3 lenguajes: HTML, PHP y Volt. La versión utilizada es Sublime Text 3. Sublime Text fue proveído por el WIC, sin embargo es preferible por sobre otros editores, incluso sobre Atom o Notepad++.

Las características más sobresalientes o distintivas de Sublime Text [33] son las siguientes:

- Fácil navegación entre archivos: Es una de las mejores características de este editor, pues tiene una barra lateral con las carpetas y sus archivos de un directorio seleccionado. Además, una barra superior mantiene los archivos abiertos disponibles a un click. De todas maneras con "Ctrl+P" se abre un novedoso buscador donde poniendo parte de un nombre de un archivo

se puede encontrar en milisegundos sin tener que entrar a bucear a un conjunto de carpetas. En este buscador, se ocupan símbolos como '@', '#' o ':' con los cuales buscar dentro de un archivo e incluso ir a una línea en particular.

- Atajos o shortcuts: No sólo "Ctrl+P" es un atajo del teclado, sino que Sublime Text provee muchos atajos útiles similares e incluso más: se le pueden agregar nuevos atajos del teclado, haciendo de Sublime Text una herramienta personalizable. Sin ir más lejos, en este trabajo se agregó el comando "Ctrl+Shift+F" para espaciar el código automáticamente, a imagen de NetBeans IDE.
- Selecciones múltipes: La idea es hacer muchos cambios de una vez y no un cambio muchas veces. Gracias al atajo "Ctrl+D" se van seleccionando las siguientes apariciones de una variable seleccionada, pudiendo así cambiar el nombre de éstas o borrarlas fácilmente. También está la opción de hacerlo con el mouse.
- Modo Libre de Distracción: Una de las características más apreciadas por los desarrolladores es el modo libre de distracción. Apretando "Shift+F11" desaparece la barra de inicio del Sistema Operativo, la barra lateral izquierda del directorio y la ayuda de navegación transparente de la derecha. Así, el desarrollador puede centrarse solamente en el código. También está el modo pantalla completa en el que no desaparece la barra lateral.
- Edición separada: Se puede separar la ventana principal de edición de código en varias partes. Con esta funcionalidad se pueden comparar dos archivos en paralelo, copiar información particular de uno en el otro con facilidad sin perder de vista ninguno de los dos e incluso programar en 2 monitores separados si se cuenta con ellos. Ha sido de mucha utilidad para el desarrollo de este trabajo.
- Cambio instantáneo de proyectos: Una agradable característica de Sublime Text es que siempre tiene disponible los archivos abiertos la última vez. Incluso no es necesario guardar los cambios. Se puede cambiar de proyecto o apagar el computador y volver al proyecto cerrado con naturalidad y con todos los archivos tal como se dejaron la última vez instantáneamente.
- Personalización: Ya se dijo que los atajos son personalizables, pero también lo son los Menus, los 'Snippets', las 'Macros' y las completitudes. La forma de hacerlo es muy simple, a través de archivos JSON. Esta flexibilidad se extiende hasta poder especificar que determinadas personalizaciones sean válidas para ciertos tipos de archivos o directamente por proyecto.

En conclusión Sublime Text es un editor de texto de primera línea, insoslayable en cualquier lista de ranking de los mejores editores, con funcionalidades únicas y útiles. Para este trabajo, nada le ha faltado y no se ha echado en falta las ventajas de otros editores, incluida la estética.

Mozilla Firefox

La selección del navegador no es algo trivial cuando se desarrolla una página web, pues no todos los navegadores tienen el mismo comportamiento. Por ejemplo, Google Chrome va consumiendo memoria RAM crecientemente a través del tiempo, Mozilla Firefox no abre sitios inseguros, Safari está optimizado para Mac e Internet Explorer resulta un poco más lento que su competencia. Se escogió Mozilla Firefox [34], versión 39.0, por ser el navegador más rápido, por su inspector de elementos, por poner énfasis en la privacidad y por no consumir tanta memoria RAM como Chrome. Además es personalizable y de código abierto. Las características de privacidad de Firefox son las siguientes:

- No ser rastreado: 'No ser rastreado' es una innovación de Firefox que permite indicar cómo recopilar y utilizar la información personal en línea.
- Navegación privada: Protege el historial de navegación.
- Botón Olvidar: Elimina del equipo la información de navegación ipso facto. Se usa cuando se haya visitado un sitio que no se quiere registrar en el historial.

Las características de seguridad de Firefox son las siguientes:

- Conexiones seguras: Se puede usar la ID instantánea del sitio Web para verificar cuál es su identidad y verificar que la conexión con el sitio es segura.
- Protección de primer nivel: Advertencias de sitios potencialmente fraudulentos, protegiendo de suplantación de identidad (phishing), software malintencionados (malware), troyanos y spyware.
- Actualizaciones de seguridad automáticas: Firefox se actualiza automáticamente, lo que garantiza tener siempre las soluciones de seguridad más recientes.

Las características de personalización de Firefox son las siguientes:

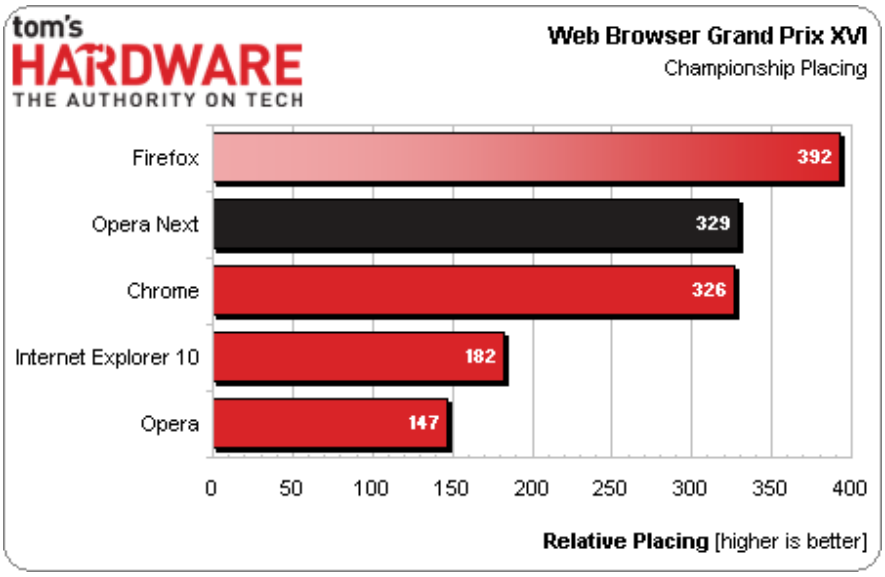
- Diseñado para ser rediseñado: Se puede personalizar el menú añadiendo, moviendo o eliminando cualquier botón. Así se pueden mantener las características más usadas a sólo un click.
- Temas: Se puede elegir entre miles de temas para 'vestir' al navegador.
- Complementos: Aplicaciones que se instalan para añadir características adicionales a Firefox, como enviar *tweets* y bloquear la publicidad.
- Barra alucinante: Aprende de la navegación para encontrar los sitios favoritos para volver a ellos sin tener que recordar la dirección.
- Sincronización: Acceder a los marcadores, historial y contraseñas desde cualquier dispositivo.

Las características de velocidad de Firefox son las siguientes:

- Más rápido que nunca: Esta es la versión más rápida de Firefox. Sobresale en parámetros técnicos y en tareas diarias frecuentes, según los resultados de Hardware Web Browser Grand Prix XVI [35]. La comparación con otros navegadores se puede ver en la figura 2.3.
- Velocidad en juegos: Desde la Web se puede jugar sin tener que sacrificar la velocidad ni el rendimiento. Firefox está a la vanguardia de los juegos en línea. Esta característica no es ocupada en este trabajo de título.
- Velocidad que se siente: Al abrir una pestaña, cambiar de una pestaña a otra u obtener resultados de la Barra Alucinante, el contenido llega con rapidez, sensiblemente para el usuario.

En conclusión, Mozilla Firefox 39.0 es el navegador ideal para desarrollar la página Web de 'Opinion Zoom'. Esto no quita que la plataforma deba estar disponible para otros navegadores, incluidos los más populares en los dispositivos móviles.

Figura 2.3: Velocidad de Navegadores



Fuente: www.tomshardware.com

Asana

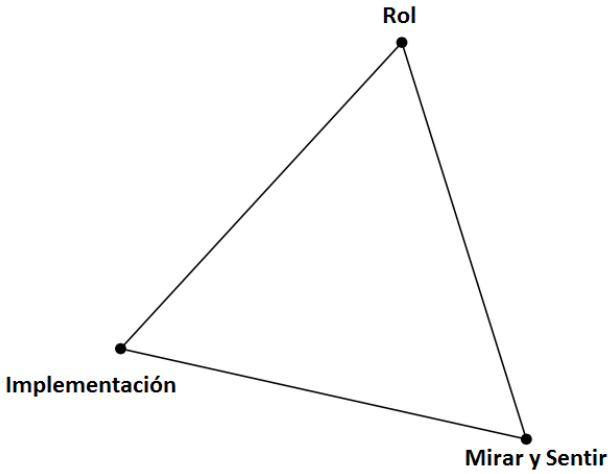
Según la metodología descrita en el capítulo 1, la organización del equipo es crítica para avanzar eficientemente en este proyecto. Para gestionar las tareas se ocupa Asana en vez de las pizarras y los post-it recomendados por la metodología *Scrum*. Asana es una aplicación web y móvil diseñada para mejorar la comunicación y colaboración en equipo [36]. En ella se pueden asignar tareas a personas específicas como también a varios miembros del equipo, con un plazo límite y asignado a un proyecto en particular. Estas tareas pueden ser marcadas con palabras claves que faciliten su búsqueda y la contextualicen, tienen una descripción y pueden ser comentadas por los miembros del equipo. Dentro de otras funcionalidades es muy útil que envíe correos a los miembros del equipo con sus tareas, sobre todo cuando se acerca la fecha de entrega, y muestre gráficamente el avance del proyecto en total en base a las tareas declaradas y cumplidas en el tiempo. Se ocupa la versión gratuita disponible para equipos de menos de 15 personas, como es en el caso de OpinionZoom.

2.7. Prototipo Funcional

La definición de prototipo resulta ambigua. Lo que se entienda por prototipo dependerá de la disciplina en la que el concepto se use y del contexto para el que se requiera. La responsabilidad de definir un prototipo, incluso en el sentido etimológico de la palabra, que es ponerle límites, es de la organización en la que éste se desarrolla. De todas maneras, en términos generales, un prototipo es un objeto que sirve para probar caracteríticas con vistas a una mejor versión del mismo y, últimamente, a una versión final. Como dirían Stephanie Houde y Charles Hill [37] un prototipo es cualquier diseño de la representación de una idea. El modelo que ellos ocupan para caracterizar un artefacto, que es el fin último de un prototipo, es de tres dimensiones: rol, mirar y sentir, e

implementación. Rol se refiere a la funcionalidad del artefacto, mirar y sentir corresponde a la experiencia sensorial del uso del artefacto, e implementación guarda relación con los componentes y técnicas con las que el artefacto realiza su función. Ninguna relación es más importante que otra. El esquema se puede ver en la figura 2.4.

Figura 2.4: Modelo de artefacto de Houd y Hill



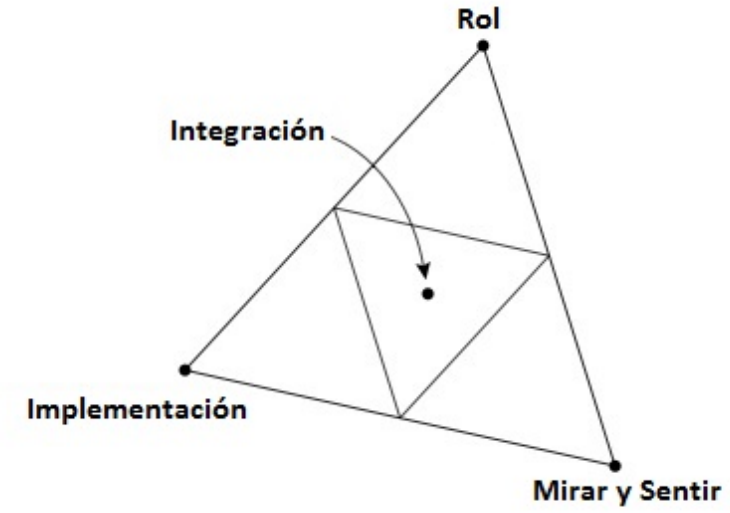
Fuente: Elaboración Propia a partir del Modelo de Houd & Hill

El modelo de artefacto permite plantearse por separado las interrogantes correspondientes a cada dimensión y tratarlas con métodos distintos. El prototipo se hace entonces para ayudar a responder esas preguntas, y en tal misión, se puede ubicar en cualquier parte del triangulo dependiendo de qué tan cercano esté a responder las preguntas de determinada dimensión. Por ejemplo, se puede querer contruir un prototipo sólo de diseño para explorar la interacción sensible del usuario con el objeto. En tal caso el prototipo estaría muy cercano a la esquina inferior derecha del modelo. Existe un cuarto punto de atracción ubicado en el centro del triangulo, el cual representa la integración de las 3 dimensiones. Un prototipo cercano a ese punto tiene por misión responder a los 3 tipos de interrogantes de forma balanceada. La figura 2.5 muestra con mayor claridad las zonas de ubicación.

Es justamente en esta zona de integración en donde se ubica el prototipo construido en este trabajo. Pero las preguntas a las que se quiere responder son eminentemente del tipo funcional y sobre las herramientas, por lo que la ubicación más exacta estaría en la esquina superior izquierda del triángulo interior que demarca la zona de integración. En otras palabras, el prototipo funcional que se presenta, y como su epíteto señala, no está enfocado en el diseño si no en su funcionalidad y en la construcción de éstas, por lo que en el modelo se ubica cerca del eje de la implementación y el rol, como se muestra en la figura 2.6.

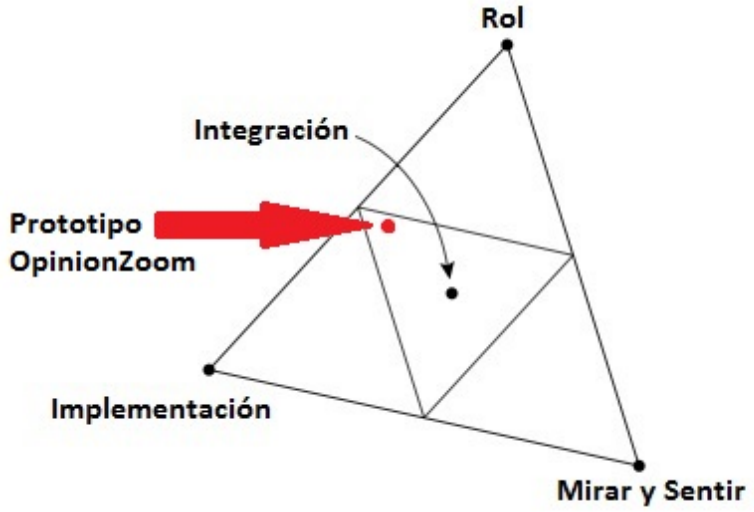
La conclusión natural que se desprende de esto, es que el objeto construido en este trabajo debe funcionar, es decir, debe tener funcionalidades que sirvan a un usuario final, y las herramientas utilizadas para su construcción serán también objeto de análisis a fin de proponer y levantar cambios y mejoras con vistas al producto final que será OpinionZoom.

Figura 2.5: Modelo de Houd y Hill para la categorización de un prototipo



Fuente: Elaboración Propia a partir del Modelo de Houd & Hill

Figura 2.6: Modelo de Houd y Hill aplicado al prototipo de OpinionZoom



Fuente: Elaboración Propia a partir del Modelo de Houd & Hill

2.8. Sistema de Análisis

Según la Teoría General de Sistemas, un sistema "es el conjunto de elementos interrelacionados de modo tal que producen como resultado algo superior y distinto a la simple agregación de los elementos" [38]. A partir de ahí se reconocen elementos, relaciones y objetivos. El objetivo final de este sistema será el análisis; aunque si se le quiere encontrar un fin ulterior, éste puede ser el análisis para la gestión. Los elementos principales que componen este sistema son el Data Warehouse y la

página Web, siendo su relación, más que sinérgica, de tipo simbiótica, pues no se puede capitalizar el valor del primero sin el segundo, y la página literalmente queda vacía sin el primero que lo alimenta.

Un sistema por antonomasia se encuentra delimitado por un entorno o un ambiente. Según Schoderbek [39] un ambiente se reconoce por estar constituido por variables no controlables por la organización que gestiona el sistema y porque los factores que lo constituyen son relevantes para la organización. En este caso, el sistema está inmerso en un ambiente compuesto principalmente por (1) Twitter, cuyas variables (*tweets*) están completamente fuera de control para la organización pero son la materia prima del proceso contenido en el sistema; (2) Klout, que aporta con influencia e intereses, y (3) los clientes de OpinionZoom por partida doble, pues ellos aportan con las variables que el sistema necesita para funcionar y además son el destinatario del objetivo final del sistema. En este sentido, este sistema es abierto en cuanto la fuerte relación que tiene con el ambiente, el cual es entrada y salida del sistema.

Todo sistema tiene entradas, un proceso y salidas. Hay 2 tipos de sistemas: perfectamente definido, cuando se conoce el proceso de transformación de las entradas en salidas, y caja negra, cuando no se conoce el proceso. En este caso se hablará de que el sistema creado es uno perfectamente definido, en cuanto cada parte del proceso, y él en su conjunto, ha sido creado durante el presente trabajo. Las cajas negras serán las APIs ya mencionadas, pero éstas forman parte del ambiente.

En particular este es un sistema de información, teniendo este concepto varias acepciones análogas que dependen de su contexto. En amplio sentido este es un sistema de información (SI) porque consiste en el tratamiento, almacenamiento y distribución de datos o información para el apoyo de decisiones gerenciales o de gestión de una organización. Existen distintos tipos de SI [40] según el grupo de una organización al que sirven. En este caso el SI pertenece al nivel de conocimiento, y apoyará a los trabajadores del conocimiento y de datos de las organizaciones clientes de OpinionZoom en su labor de integrar el nuevo conocimiento en los negocios. Los otros tipos de SI son de nivel operativo, administrativo y estratégico. Si bien el SI en cuestión puede ser ocupado en otros niveles, puesto que es difícil definir límites teóricos en la práctica, éste será eminentemente a nivel de conocimiento.

Existe otra categorización, dentro de los 4 tipos de SI recién señalados, sobre el fin específico de cada SI. Existen así los ESS[11] a nivel estratégico, los MIS[12] y los DSS[13] a nivel administrativo, los KWS para el nivel del conocimiento y los TPS[14] a nivel operativo. Por naturaleza el SI creado es un Sistema del Trabajo del Conocimiento (KWS por sus siglas en inglés, *Knowledge Work System*).

Los Sistemas del Trabajo del Conocimiento crean e incorporan conocimiento a la organización canalizándolo a los nodos correspondientes. Pese a ser un activo intangible, a veces difícil de administrar, el conocimiento es fundamental para las organizaciones y crítico su buen manejo para obtener ventajas competitivas. El Sistema de análisis diseñado y creado en el presente trabajo de título creará conocimiento a través del análisis de opiniones en Twitter, pero la distribución de éste y las decisiones que de él se desprenden, serán tarea de la organización que requiera los servicios de OpinionZoom.

[11] Sistemas de Apoyo a Ejecutivos (*Executive Support Systems*)
[12] Sistemas de Información Gerencial (*Management Information Systems*)
[13] Sistemas de Apoyo a las Decisiones (*Decision Support Systems*)
[14] Sistemas de Procesamiento de Transacciones (*Transaction Processing Systems*)

En el siguiente capítulo se abordará con mayor detalle el sistema de análisis construido, y en el resto de capítulos la explicación detallada de sus componentes.

Capítulo 3

Diseño del Sistema

Un sistema se puede esquematizar por sus entradas, proceso y salidas. Ya se adelantó en el capítulo anterior que el objetivo del sistema es el análisis de opiniones en Twitter, el dueño del sistema será el WIC y el ambiente está dado por Twitter, Klout, y los clientes de OpinionZoom. Es este ambiente el que define las entradas del sistema. Luego se explica el proceso de transformación, detallando los elementos involucrados. Las salidas, al ser básicamente los servicios de OpinionZoom, se detallan en un capítulo posterior. Se termina este capítulo mostrando la ubicación de la página Web, los scripts en ejecución y la API de polaridad.

3.1. Entradas

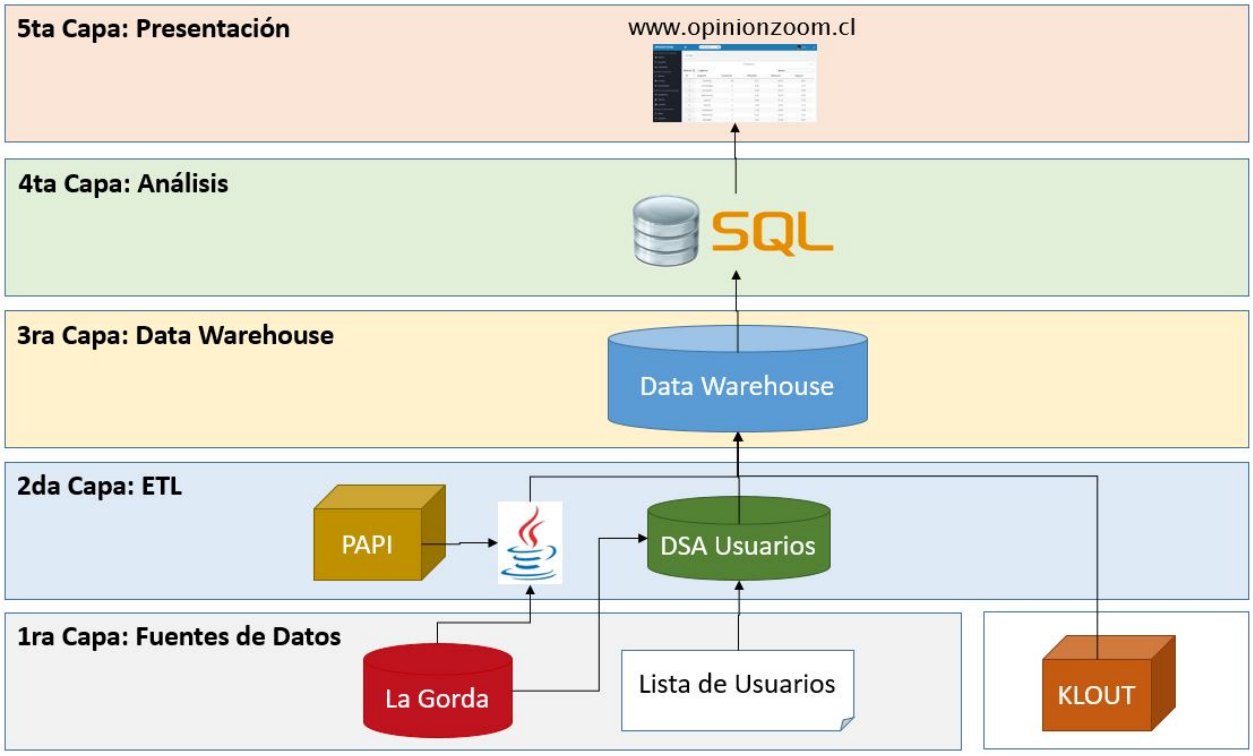
Las entradas son el inicio del flujo del sistema. Son variables no controladas por el sistema. Específicamente los tweets son una incógnita antes de entrar al sistema, el puntaje Klout para la influencia y los intereses de los usuarios, son un producto de Klout ignorado por OpinionZoom hasta que entran estas variables al sistema, y los datos de los usuarios de OpinionZoom también son desconocidos hasta que entran al sistema.

- Tweets. Son las opiniones que recibirá el sistema para el análisis. Son textos de no más de 140 caracteres.
- API de Klout. Otorgan un puntaje de influencia y una lista de intereses para cada usuario de Twitter registrado por ellos en su base de datos.
- Usuarios de OpinionZoom. Los usuarios aportan dos tipos de datos: para la página Web y para el Data Warehouse. Para el Data Warehouse aportarán las keywords, o palabras claves (con menos de 50 caracteres), que quieren analizar y que determinarán que *tweets* entrarán en el sistema. Para la página Web aportan con sus datos personales: nombre, nombre de usuario, email y contraseña. También un nombre de cliente, el cual servirá para asociar al usuario de OpinionZoom.cl con las keywords del Data Warehouse, y consecuentemente con los *tweets* a analizar.

3.2. Proceso

El proceso transforma datos en información. Para ello extrae los datos, se les aplican algoritmos a éstos, se ordenan de cierta manera y luego se analizan con consultas pertinentes orientadas a la obtención de información útil para el cliente final para por fin presentarla a los usuarios de OpinionZoom. El proceso del sistema se puede ver en la figura 3.1, y a la postre será equivalente a la arquitectura del Data Warehouse [41] con la página Web como capa de presentación. Son 5 las capas de este sistema, donde el flujo es siempre ascendente, desde la primera capa hasta la capa superior.

Figura 3.1: Sistema OpinionZoom



Fuente: Elaboración Propia

La primera capa, como su nombre lo dice, envuelve a todas las fuentes de datos: los *tweets*, los usuarios de Twitter, la lista de identificadores de usuarios de Twitter a seguir y la influencia y los intereses dados por Klout. La API de Klout no es estrictamente parte del sistema, y es por eso que es descrita en la sección anterior como parte del entorno y en el esquema aparece anexada a la capa.

La segunda capa corresponde al ETL, descrito con mucho detalle en el próximo capítulo. Aquí se extraen los datos de las fuentes de datos, se transforman mediante scripts de Java y algunos se almacenan temporalmente en un DSA.

La tercera capa contiene al Data Warehouse, como el centro del proceso, fin de las capas precedentes y base de análisis para luego presentar los resultados. En el siguiente capítulo se ahonda en la estructura del Data Warehouse y los procesos que le son más propios.

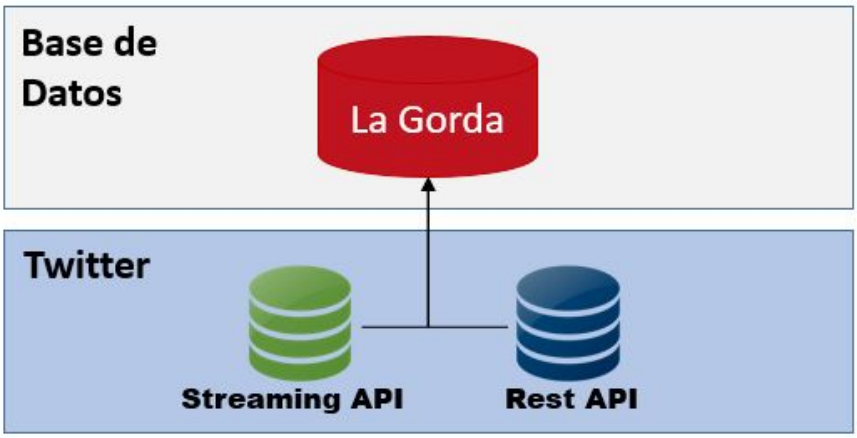
La cuarta capa corresponde al análisis, el cual se hace en base a consultas SQL. Ahí radica toda la inteligencia de la siguiente capa, en donde se muestran los resultados.

La quinta capa es la de presentación, en donde se muestran los resultados frutos del análisis de la capa anterior. Se profundiza en ella en capítulos posteriores. A continuación se explica cada uno de los componentes de la primera, segunda y cuarta capa, incluyendo las relaciones entre ellas.

3.2.1. La Gorda

La Gorda es la base de datos más primitiva del sistema. La Gorda tiene dos tablas: 'tweet' y 'users'. En la primera se guardan los *tweets* provenientes de la Streaming API. En la segunda tabla se muestran los usuarios de Twitter provenientes de la Rest API. La tabla 'tweet' se actualiza en tiempo real y suma diariamente una cantidad aproximada de un millón de *tweets*, mientras que la incorporación de nuevos registros en la tabla 'users' es menos frecuente. Un sencillo esquema de construcción de la base de datos La Gorda se puede ver en la figura 3.2.

Figura 3.2: Base da datos "La Gorda"



Fuente: Elaboración Propia

Durante el transcurso de este proyecto La Gorda fue alojada en PostgreSQL y en Solr secuencialmente. Se muestran a continuación las diferencias de diseño entre ambos.

La Gorda en PostgreSQL

La antigua forma en que los datos se guardaban en La Gorda tanto *tweets* como usuarios es principalmente en formato JSON, no obstante se añadieron unas pocas columnas para mejorar la búsqueda. La tabla para los tweets, que se puede ver en la tabla 3.1, tenía 4 columnas: idtweet, data, idtweetinterna y dia.

La tabla 'users' contaba con 3 columnas: idusuario, snapshot y data, descritas en la tabla 3.2.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idtweet	bigint	identificador del tweet según Twitter
data	jsonb	json con la información del tweet
idtweetinterna	bigint	identificador del tweet añadido al insertar el tweet en La Gorda
dia	date	fecha del Tweet

Tabla 3.1: Tabla 'tweet' para La Gorda en PostgreSQL.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idusuario	bigint	identificador del usuario según Twitter
snapshot	timestamp	tiempo (fecha y hora) en que fue insertado el usuario
data	jsonb	json con la información del usuario

Tabla 3.2: Tabla 'users' para La Gorda en PostgreSQL.

Como se señala en el marco conceptual, una de las bondades de PostgreSQL es el buen manejo que tiene del formato JSON. Aquí se comprueba al contar con tipo de atributo especial, que no está en otros DBMS, llamado 'jasonb' que sirve para alojar JSON y hacer búsquedas eficientemente dentro de ellos.

La Gorda en Solr

Mientras PostgreSQL es un sistema de gestión para bases de datos relacionales, Solr es un motor de búsqueda, hecho para la inserción y la búsqueda de información a velocidades inauditas para un DBMS tradicional. Esto le significa tener una estructura diferente a PostgreSQL, MySQL o MariaDB. Por ello, La Gorda tuvo que reestructurarse, y aprovechándose de ello, su estructura final quedó optimizada para la búsqueda. Los usuarios ya no son relevantes para el sistema, pues ya fueron cargados desde La Gorda en PostgreSQL. Las información del *tweet* quedó en 8 columnas:

1. id: es el identificador del *tweet* según Twitter
2. lang: es el idioma en que viene el *tweet*
3. text: es el texto del *tweet*
4. created_at: es la fecha y la hora en que fue emitido el *tweet*
5. retweeted_status_id: es un campo presente en los retweets y contiene el identificador del *tweet* original

6. user_id: es el identificador del usuario de Twitter

7. others: contiene toda la otra información propia del *tweet* que no se ocupa en el prototipo

8. version: es una especie de identificador interno de Solr que no se ocupa en el prototipo

Las búsquedas de esta forma son mucho más rápidas, principalmente porque la rapidez de las consultas de búsqueda es la característica principal de Solr, pero también por la forma en que se almacena la información.

3.2.2. Lista de Usuarios

Naturalmente La Gorda no pretende, ni por capacidad ni por objetivos, almacenar toda la información de Twitter. El objetivo es capturar la mayor información de Twitter de usuarios chilenos. Para ello se realizó en el WIC un estudio del comportamiento de los usuarios chilenos en Twitter. El primer desafío fue identificar a los usuarios chilenos de Twitter. Para ello se ocupó una heurística que consiste en definir usuarios semillas y luego ir incorporando a la base de datos a usuarios que *retweetean*, siguen y comentan sobre los usuarios semillas. Las semillas se escojen consistentemente como usuarios que tienen mucha relevancia a nivel nacional pero cuyo seguimiento masivo en el extranjero carece de un fundamento lógico. El segundo paso fue ordenar estos usuarios supuestamente chilenos según su nivel de uso de la red social. Con este fin se creó un simple ratio de uso que se calcula como el número de tweets emitidos dividido el tiempo en que la cuenta ha estado activa.

$$Uso(tweets, Fcreado) = \frac{tweets}{Factual - Fcreado} \tag{3.1}$$

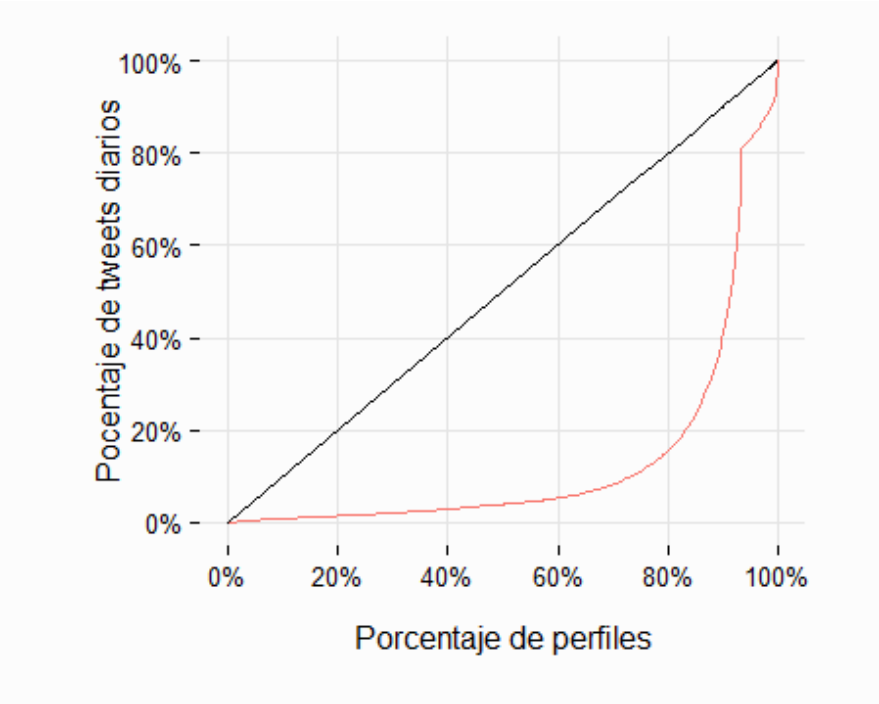
Ordenando estos usuarios según este índice de tweets diarios de menor a mayor se llegó a una forma particular de la curva de Lorenz, como muestra la figura 3.3. La línea roja muestra claramente que sólo el 20 % de los usuarios (el 20 % más activo) aporta con aproximadamente el 80 % de los tweets que se diariamente se emiten en Chile.

Con este resultado se optó por seguir a esta fracción de usuarios más asiduos a Twitter cuyo número se cortó en 500.000. Luego de algunos meses, de este medio millón de usuarios sólo seguían realmente activos poco más de 178.000, que fueron los escogidos para la construcción del prototipo. Se hizo un archivo de texto que en cada linea contiene el número identificador de Twitter para cada uno de los usuarios a incluir al Data Warehouse.

3.2.3. Algoritmos

El objetivo general es diseñar y construir un sistema de análisis de opiniones en Twitter integrando algoritmos de Data Mining. Los algoritmos a utilizar tienen fines distintos y excluyentes, por lo que no cabe la posibilidad de hacerlos competir. Se ocupan cuatro algoritmos, 3 de los cuales están encapsulados como APIs y uno es desarrollado dentro de este mismo trabajo.

Figura 3.3: Curva de Lorenz en el porcentaje diario de tweets para usuarios chilenos



Fuente: Web Intelligence Centre

Polaridad

El algoritmo de polaridad utilizado para este prototipo fue desarrollado en el WIC y es uno de los mejores algortimos de *Opinion Mining* para texto en español [19]. Está encapsulado como una API que se instaló en un servidor de Amazon Web Services. Esta API es conocida como PAPI, por ser el acrónimo en inglés de *Polarity API* (API de polaridad). La forma de consulta es a través de una dirección Web a la que se le añade el texto por el que se le quiere calcular la polaridad. A modo de ilustración, si se quiere conocer la polaridad de la frase 'me gusta escribir mi tesis' se visita la dirección `http://52.27.93.26/tag?sentence=megustaescribirmitesis` y se obtiene un JSON con la frase consultada y su polaridad, que en este caso tiene el valor 2.

La API no ofrece una lectura posterior, pero se entiende por polaridad negativa una menor a 0, donde entre más lejana a 0 se considera más negativa. Por el contrario, sobre 0 es una opinión positiva y entre más alto el valor es más positiva. No hay límites inferiores ni superiores para la polaridad obtenida a través de PAPI, sino que dependerá proporcionalmente del largo del texto el valor absoluto de la polaridad obtenida.

Influencia

La influencia viene de la API de Klout, y es un misterio para OpinionZoom el funcionamiento interno y el algoritmo que utiliza. Recuérdese que Klout no es parte del sistema. La influencia de Klout se obtiene accediendo a la API a través de Java por la dependencia 'klout-java-wrapper-

1.0.jar'. La influencia es un número racional entre 0 y 100.

Intereses

Al igual que el algoritmo de influencia, el de intereses también es una caja negra. Los intereses se obtienen de igual manera que la influencia. Los intereses son muchísimos y están en 3 niveles, pero para efectos del prototipo se tratará cada salida con el mismo peso.

Sexo de usuarios

Pese a estar fuera del alcance de esta memoria, para probar la funcionalidad del prototipo de OpinionZoom se confeccionó un sencillo algoritmo para identificar el sexo de los usuarios de Twitter que pueblan el Data Warehouse. Existen distintas aproximaciones para conocer el sexo de una persona en Twitter. Algunas solo toman en cuenta el texto [42] mientras otras también otros campos como el nombre, la descripción, la URL y la ubicación [43]. Las metodologías también son diversas y se han usado desde *N-gramas* [44] hasta métodos más lingüisticos [45]. En este trabajo se optó por el tipo de algoritmo más simple para disminuir el costo temporal asociado a la construcción de éste. Además sólo se quiere probar la funcionalidad a nivel de prototipo dejando en segundo lugar el buen rendimiento entendido como precisión.

El método más simple es augurar el sexo a partir del nombre del usuario. Esto presenta la dificultad de la libertad del usuario de Twitter para definir tanto su nombre de usuario como su nombre de cuenta (*screen name*). Existe también una segunda dificultad más general que la primera, que es la presencia de cuentas de Twitter de organizaciones que son naturalmente asexuadas, y a las que por coincidencia se les podría asignar un sexo. Sin embargo, por ser un prototipo se omitirá el error que esto puede producir. La primera dificultad se resuelve con un campo tipo de sexo 'indefinido' que no se considerará para hacer los análisis comparativos entre hombres y mujeres. Esto se hace bajo el supuesto fuerte de que tanto hombres como mujeres se asignan nombres ambiguos en sus cuentas de Twitter en igual proporción.

Básicamente existen dos caminos para obtener desde una máquina el sexo de una persona a partir de su nombre. La primera es utilizar una API existente [46] y la segunda es construir desde cero una función que compare el nombre ingresado con una lista de nombres masculinos y otra lista de nombres femeninos. Se optó por la segunda vía para no incluir más cajas negras en el sistema que arriesgaran la estabilidad del sistema. Además con la opción escogida se puede mejorar fácilmente la precisión a través del tiempo simplemente agregando más nombres en las listas mencionadas. La función construida, llamada *GetSexo*, en pseudocódigo queda como sigue:

```
GetSexo(name) {
    nombre = name.AMinusculas();

    Archivo hombres = CargarArchivo("ninos.txt"));
    Archivo mujeres = CargarArchivo("ninas.txt"));

    line = null;
```

```
        while((line = hombres.LeerLinea()) != null){
            if(nombre.contiene(line.AMinusculas())){
                retorna 1;
            }
        }
        while((line = mujeres.LeerLinea()) != null){
            if(nombre.contiene(line.AMinusculas())){
                retorna 2;
            }
        }
        retorna 3;
    }
```

Son tres las posibles salidas del algoritmo: 1 si es hombre, 2 si es mujer y 3 si no se puede determinar. Esta función se ocupa de la siguiente forma:

1. Se aplica la función *GetSexo* sobre el nombre de usuario

2. Si el resultado es 1 o 2 se sale. Si es 3 se sigue.

3. Se aplica la función *GetSexo* sobre el nombre de *screen_name*

En el Data Warehouse el usuario se queda con el sexo resultante de este algoritmo.

3.3. Diseño Físico

Se cuenta con 2 servidores en Amazon, a través de AWS. En el primero se aloja la página Web y se hace el proceso de ETL. Se trata de un servidor de propósito general, el más básico de la lista, cuya capacidad alcanza con abundancia los requisitos actuales de procesamiento. Un segundo servidor contiene la API de polaridad, PAPI. Este sí requiere mayor capacidad y por tanto, como se puede ver en la tabla, es un servidor especialista en la optimización de cálculos, necesario para todos los costosos pasos del algoritmo.

Fuente: Elaboración Propia

Instancia	Familia	Tipo	vCPUs	Memory (GiB)
Página Web y ETL	General purpose	m3.medium	1	3,75
PAPI	Compute optimized	c4.large	2	3,75

Tabla 3.3: Servidores AWS

Capítulo 4

Construcción del Data Warehouse

El segundo objetivo específico de este trabajo es la construcción de un Data Warehouse, que es una colección ordenada de datos orientada a la gestión. Para la construcción de este Data Warehouse se tienen los siguientes pasos: Definición de indicadores, Modelo de Base de Datos, ETL (Extracción, Transformación y Carga) y creación de vistas para el usuario final, las que se harán en la página Web. En este capítulo se abordarán sólo las tres primeras etapas, dejando la cuarta para los capítulos siguientes. Esta metodología tradicional del Data Warehouse debe convivir con la metodología ágil de desarrollo Scrum.

4.1. Definición de Indicadores

La construcción de un Data Warehouse viene a satisfacer la demanda por un instrumento de gestión. Para un analista esto se traduce básicamente en una serie de indicadores que, redundantemente, le indiquen o sugieran una acción. Los indicadores pueden ser, entre otros, cuantitativos en cuanto entreguen un número, gráficos o categorizadores. Lo importante es identificarlos desde un comienzo, independientemente a que luego se descubran otros indicadores de interés para el usuario que puedan ser incluidos en el Data Warehouse. Resulta necesario advertir que si bien los indicadores serán la primera entrada del diseño en cuanto a requerimientos y salida esperable en las vistas, no es en esta etapa en donde se definen a cabalidad en cuanto a sus configuraciones. Por ejemplo, se puede definir lo que se entenderá por frecuencia, pero luego la ventana de tiempo en que está medida puede ser sujeta a alteración en la ejecución según los requerimientos del usuario, por lo que no se define *a priori* en esta sección.

Para la construcción del prototipo los indicadores fueron establecidos por el Equipo en conjunto con el ScrumMaster, y luego visados por el Dueño del Proyecto. Éstos indicadores responden al modelo de negocios del proyecto, al proyecto OpinionZoom presentado a CORFO y a la creatividad del Equipo. Del primero surge también la necesidad de identificar los intereses de los usuarios. De los indicadores prometidos en el proyecto se cumple con la influencia y la polaridad, pero no se tiene un indicador de ironía. El indicador 'impacto' es creado por el Equipo. A continuación se presentan los indicadores cuantitativos iniciales que se buscan por *tweet*:

1. **Polaridad**: Es el sentimiento contenido en una opinón clasificado como positivo o negativo [47]. Para este trabajo este indicador está dado por la API de polaridad desarrollada en el WIC.

2. **Influencia**: Representa lo significativo que resultan las opiniones de un usuario para sus seguidores en cuanto a la capacidad de cambiar tendencias, esparcir ideas o generar más seguidores (*word-of-mouth*). Lo ideal sería tener una medida de influencia por cada tweet.

3. **Impacto**: Este es el indicador más complejo a entender. Se trata de la polaridad de una opinión ponderada por la influencia que le corresponde. Intenta mostrarle al usuario de OpinionZoom el impacto negativo o positivo que tendrá una opinión para los seguidores del usuario que la emite. La influencia obtenida de la aplicación de Klout mencionada en el capítulo anterior varía en teoría de 0 a 100. En la práctica lo hace de 10 a 99. Para normalizar esta escala a la de la polaridad se le hace un ajuste geométrico. Luego, el impacto será el producto entre la polaridad y esta influencia normalizada que varía entre 0 y 10.

$$Impacto(polaridad, influencia) = polaridad * \frac{influencia^2}{1000} \tag{4.1}$$

Los *tweets* pertenecerán a una palabra clave o keyword. Los *tweets* se reciben como hechos dado que contienen a la keyword a la que luego pertenecerán. Para estas keyword también se definen indicadores.

1. **Frecuencia**: Es simplemente la cantidad de *tweets* que se han recibido para determinada keyword (o un conjunto de ellas en el caso de las alertas) en una ventana de tiempo.

2. **Polaridad**: Es el promedio simple de la polaridad de cada *tweet* asociado a la keyword.

3. **Influencia**: Es el promedio simple de la influencia asociada a cada *tweet* asociado a la keyword.

4. **Impacto**: En este caso el promedio del impacto de cada *tweet* se pondera por la frecuencia, de tal modo que el impacto de una opinión influyente no neutra sea mucho menor que el impacto que puedan generar varias personas con igual influencia y similar polaridad en magnitud. El impacto vectorial que aparece en la ecuación es el promedio simple del impacto de todos los *tweets* asociados a la keyword.

$$Impacto(frecuencia, \overline{impacto}) = ln(frecuencia) * \overline{impacto} \tag{4.2}$$

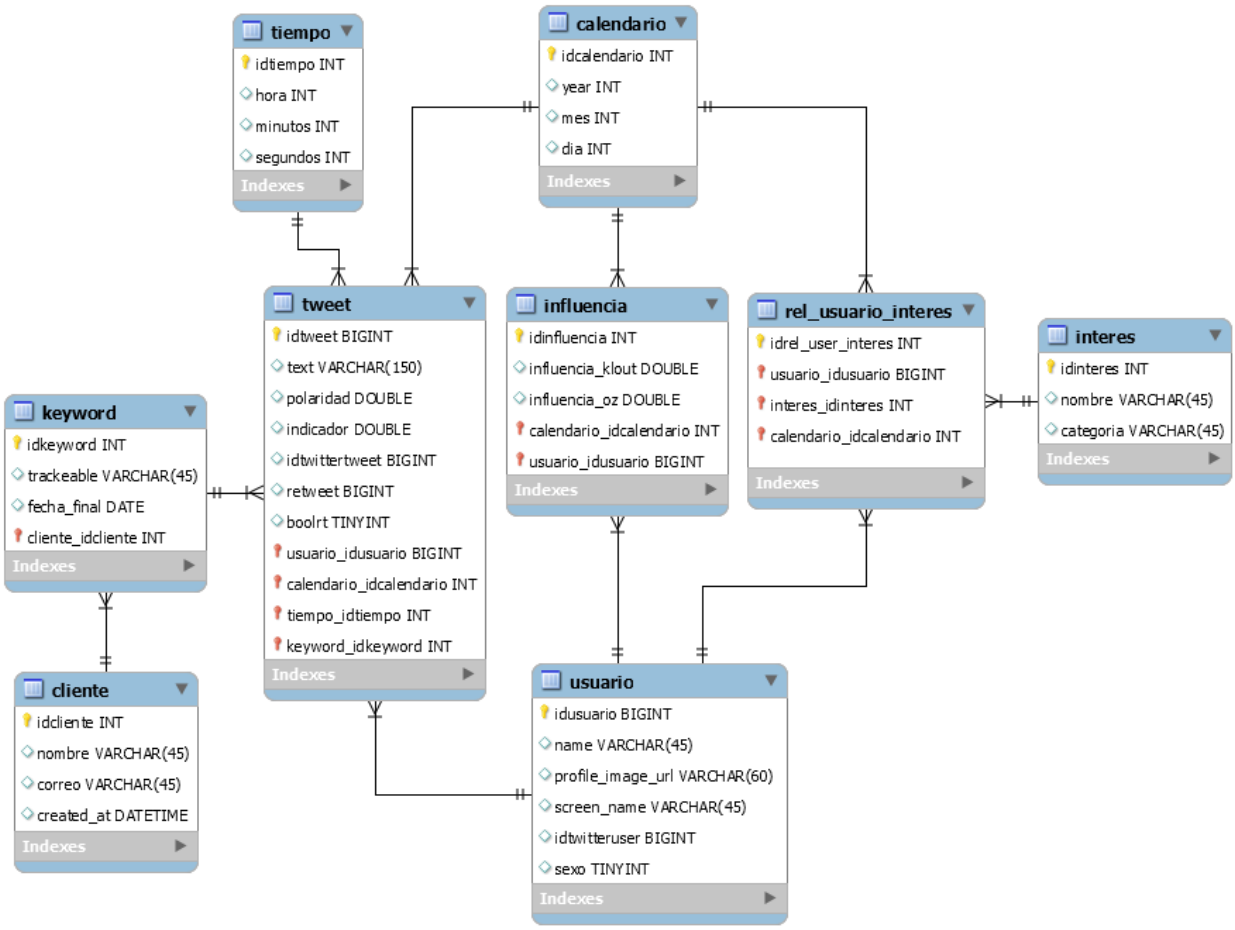
4.2. Modelo Constelación

Tradicionalmente, el modelamiento de un Data Warehouse sigue un modelo estrella, que consiste en una tabla de hechos y sus dimensiones circundantes. En un comienzo se pensó un modelo estrella, el cual se puede ver en el apéndice A, con la tabla 'tweet' como única *fact table*. Del mo-

delo constelación se iteró cortamente a un modelo copo de nieve, donde una de las dimensiones también pasa a ser una tabla de hechos, pero finalmente se llegó a un modelo constelación, que es un modelo con varias tablas de hechos las cuales no son dimensiones las unas de las otras.

En este modelo constelación los hechos considerados para modelar son principalmente los *tweets*. Las otras *fact tables* representan la influencia de un usuario y la relación entre un usuario y sus intereses, como se puede ver en la figura 4.1.

Figura 4.1: Modelo Constelación



Fuente: Elaboración Propia

4.3. Tablas del Modelo

A continuación se explican las tablas que componen el modelo de data warehouse y se detallan sus atributos. Primero se describen las 6 tablas de dimensiones y luego las 3 tablas de hechos.

4.3.1. Tablas de Dimensiones

El modelo tiene 5 tablas de dimensiones. Hay una sexta tabla que no es de hechos, pero que se incluye dentro del diagrama, que es la tabla 'clientes'. Esta tabla no es propia del Data Warehouse pues no tiene relación con los indicadores a obtener, pero es necesaria para organizar las 'keywords', que sí son parte del modelo, y relacionarlas con las vistas del usuario final. Las tablas de dimensiones son las siguientes: keyword, usuario, interes, calendario y tiempo.

Keyword

La tabla keyword contiene los 'trackeables', es decir las secuencias de carácteres, sean éstas un hashtag, una cuenta de Twitter o simplemente palabras seleccionadas, con las que se buscarán los *tweets*. La consulta a la gorda tiene en la cláusula 'WHERE' que el texto contenido en el *tweet* contenga algún 'trackeable'. La tabla 4.1 muestra todos los atributos.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idkeyword	int	identificador de la keyword
trackeable	varchar(45)	nombre de la keyword
fecha_final	date	última fecha en la que será buscada la keyword
cliente_idcliente	int	llave foránea que relaciona la keyword con el cliente que la posee

Tabla 4.1: Tabla Keyword

Usuario

La tabla usuario contiene los usuarios de Twitter. El número de usuarios seguidos en un comienzo eran 500.000. Este medio millón de cuentas, en diciembre del 2014, se suponía que generaba el 90 % del contenido de Twitter en Chile. Sin embargo, en julio del 2015 se conservaron sólo los usuarios que se mantenían activos, llegando estos a 242.000, de los cuales so guardaron los 178.000 más activos. La dimensión usuario guarda relación con los *tweets*, pues cada *tweet* tiene asociado un usuario, con la tabla influencia, que contiene la influencia de los usuarios en el tiempo, y con la tabla que relaciona a los usuarios con los intereses. La tabla 4.2 muestra todos los atributos.

Interés

La tabla interes contiene los intereses de los usuarios de Twitter. Pese a ser muchos intereses, es un número acotado por la API de intereses usada. La forma de llenado de la tabla fue justamente con los intereses obtenidos por los usuarios, por lo que es muy poco probable que haya un interés de algún usuario que no se encuentre aquí. Podría darse sólo si el usuario cambia de interés a uno

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idusuario	int	identificador de la keyword
name	varchar(45)	nombre del usuario
profile_img_url	varchar(60)	url de imagen de usuario en Twitter
screen_name	varchar(45)	nombre de la cuenta en Twitter
idtwitteruser	bigint	identificador original de Twitter
sexo	tinyint	número binario para determinar sexo del usuario

Tabla 4.2: Tabla Usuario

que no se encuentre en la lista, pero esto es bastante improbable dado que se registran más de 2.500 intereses. La tabla 4.3 muestra todos los atributos.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idinteres	int	identificador de cada interés
nombre	varchar(45)	nombre del interés
categoría	varchar(45)	cada interés estará asociado a una categoría. Las categorías serán conjuntos disjuntos. Por ahora se comienza con este atributo en nulo.

Tabla 4.3: Tabla Interés

Calendario

La tabla calendario tiene el día calendario de todos los días desde el año 2006 hasta el 2050, diferenciándolos por año, mes y día. La tabla 4.4 muestra todos los atributos.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idcalendario	int	identificador de cada interés
year	int	número del año
mes	int	número del mes, de 1 a 12
dia	int	número de día calendario

Tabla 4.4: Tabla Calendario

Tiempo

La tabla tiempo representa el tiempo dentro de un día, separado por horas, minutos y segundos. Semejantemente a la dimensión anterior, esta jerarquía puede ser útil para consultas por una hora del día en particular. Por ejemplo, gracias a esta granularidad se puede consultar la frecuencia de *tweeteo* a la hora de almuerzo. La tabla 4.5 muestra todos los atributos.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idtiempo	int	identificador de segundo único dentro de un día
hora	int	número de la hora, de 0 a 23
minutos	int	número de minutos de 0 a 59
segundos	int	número de segundo de 0 a 59

Tabla 4.5: Tabla Tiempo

4.3.2. Tablas de Hechos

Hay tres tablas de hechos que ocupan el resto de dimensiones a través de llaves foráneas. La primera en ser creada es la tabla 'tweet', que contiene los *tweets* y su polaridad. La siguiente es 'influencia' que permite ver la influencia de los usuarios en el tiempo. La última es la tabla 'rel_usuario_interes' que relaciona a los usuarios con sus intereses.

Tweet

La tabla 'tweet' contiene los *tweets* con los atributos útiles para los indicadores a obtener, para ordenarlos y para mostrarlos. Ocupa las dimensiones 'keyword', 'usuario', 'calendario' y 'tiempo'. La tabla 4.6 muestra todos los atributos.

Influencia

La tabla 'influencia' nace del propósito de ir registrando la influencia de los usuarios a través del tiempo. Nótese que si no se quisiera registrar la evolución histórica de este indicador, bastaría con ponerlo de atributo en la tabla de dimensión 'usuario' y actualizarlo periódicamente. La tabla 4.7 muestra todos los atributos.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idtweet	bigint	identificador de segundo único dentro de un día
text	varchar(150)	texto del tweet
polaridad	double	la orientación sentimental del *tweet*
indicador	double	polaridad por influencia
idtwittertweet	bigint	id del *tweet* según twitter
retweet	bigint	el número del *tweet* original. Es igual a idtwittertweet si no es retweet.
boolrt	tinyint	1 si es retweet, 0 si no
usuario_idusuario	double	id del usuario que emitió el *tweet*
calendario_idcalendario	double	id de la dimensión calendario
tiempo_idtiempo	double	id de la dimensión tiempo
keyword_idkeyword	double	id de la keyword por la que fue buscado el *tweet*

Tabla 4.6: Tabla Tweet

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idinfluencia	int	identificador de registro influencia
influencia_klout	double	influencia del usuario desde la API de Klout. Varía entre 0 y 100, donde 0 es para una persona no influyente y 100 para una persona muy influyente
influencia_oz	double	influencia del usuario desde la API de OpinionZoom, la cual aún no está construida
calendario_idcalendario	int	identificador del día calendario en el que fue obtenida la influencia
usuario_idusuario	int	identificador del usuario al que se le asocia la influencia

Tabla 4.7: Tabla Influencia

Relación Usuario-Interés

Finalmente se consignan las relaciones entre los usuarios y sus intereses en la tabla 'rel_user_interes', cuyos atributos se muestran en la tabla 4.8. Al igual que las tablas de hechos precedentes, se busca guardar los cambios históricos de intereses de los usuarios, sin embargo en este prototipo no se le da uso a esta posibilidad ni en la influencia ni en los intereses.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idrel_user_interes	int	identificador de la relación entre un usuario y uno de sus intereses
usuario_idusuario	int	identificador del usuario al que se le asocia un interés
interes_idinteres	int	identificador del interés
calendario_idcalendario	int	identificador del día calendario en el que fue obtenida la relación

Tabla 4.8: Tabla Relación Usuario-Interés

4.3.3. Vistas

Con las tablas de dimensiones y las tablas de hechos queda descrito completa y exhaustivamente el Data Warehouse. Sin embargo, con tablas de vistas definidas en la base de datos se pueden hacer consultas más simples, lo que hace más fácil el desarrollo de las vistas propias de la página Web. Estas tablas de vistas de la base de datos son una unión de varias tablas afines, ordenadas a consultas particulares que frecuentemente usan esas mismas tablas.

Vista Facttweet

La tabla vista 'facttweet' hace la unión entre la tabla de hechos tweet y sus dimensiones: keyword, usuario, calendario y tiempo. De esta forma se pueden ver todos los *tweets* en la misma fila con la keyword a la que está asociado, el usuario que lo emitió y el día y la hora en que fue creado. Además se le añaden a esta vista 4 columnas útiles para las consultas que se realizarán desde la página Web: datetime, datetimechile, ampm y date. Los atributos de esta tabla vista se encuentran en la tabla 4.9

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idtweet	bigint	identificador de segundo único dentro de un día
text	varchar(150)	texto del *tweet*
polaridad	double	la orientación sentimental del *tweet*
indicador	double	polaridad por influencia
idtwittertweet	bigint	id del *tweet* según twitter
retweet	bigint	el número del *tweet* original. Es igual a idtwittertweet si no es retweet.
boolrt	tinyint	1 si es retweet, 0 si no
usuario_idusuario	double	id del usuario que emitió el *tweet*
calendario_idcalendario	double	id de la dimensión calendario
tiempo_idtiempo	double	id de la dimensión tiempo
keyword_idkeyword	double	id de la keyword por la que fue buscado el *tweet*
idusuario	int	identificador de la keyword
name	varchar(45)	nombre del usuario
profile_img_url	varchar(60)	url de imagen de usuario en Twitter
screen_name	varchar(45)	nombre de la cuenta en Twitter
idtwitteruser	bigint	identificador original de Twitter
sexo	tinyint	número binario para determinar sexo del usuario
idcalendario	int	identificador de cada interés
year	int	número del año
mes	int	número del mes, de 1 a 12
dia	int	número de día calendario
idtiempo	int	identificador de segundo único dentro de un día
hora	int	número de la hora, de 0 a 23
minutos	int	número de minutos de 0 a 59
segundos	int	número de segundo de 0 a 59
datetime	datetime(6)	fecha en que fue emitido el *tweet* con uso horario UTC±00:00, que es la hora con la que viene el *tweet*
datetimechile	datetime(6)	fecha en que fue emitido el *tweet* con uso horario UTC+03:00, que es la hora chilena en la que fue emitido el *tweet*
ampm	varchar(5)	señala 'am' o 'pm' dependiendo si es mañana o tarde respectivamente para la fecha chilena
date	date	la fecha chilena en formato año-mes-día

Tabla 4.9: Vista facttweet

Vista Factinteres

Esta tabla une a las tablas interes, usuario y rel_usuario_interes. Los atributos se muestran en la tabla 4.10.

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idusuario	int	identificador de la keyword
name	varchar(45)	nombre del usuario
profile_img_url	varchar(60)	url de imagen de usuario en Twitter
screen_name	varchar(45)	nombre de la cuenta en Twitter
idtwitteruser	bigint	identificador original de Twitter
sexo	tinyint	número binario para determinar sexo del usuario
idcalendario	int	identificador de cada interés
year	int	número del año
mes	int	número del mes, de 1 a 12
dia	int	número de día calendario
idinteres	int	identificador de cada interés
nombre	varchar(45)	nombre del interés
categoría	varchar(45)	cada interés estará asociado a una categoría. Las categorías serán conjuntos disjuntos. Por ahora se comienza con este atributo en nulo.

Tabla 4.10: Vista factinteres

4.4. Construcción de tablas

Las tablas explicadas se dividen en 4 tipos a partir de la forma en que le son insertados sus datos, incluyendo la frecuencia de actualización.

- Iniciales: Son las primeras tablas completadas. De ellas dependen las demás. Las 4 tablas de este tipo son: calendario, tiempo, usuario e interés.
- Intermitentes: Son tablas que deben ser llenadas cada cierto tiempo para actualizar la información existente, pero cuya actualización infrecuente no va en desmedro significativamente de la calidad de los resultados del sistema. Esta actualización periódica se ocupa para las tablas de hechos 'influencia' y 'rel_usuario_interes', bajo el supuesto de que la influencia y los intereses de un usuario no varían significativamente en un período corto de tiempo.
- Manuales: Son las tablas que se modifican manualmente en phpmyadmin. Esto sucede cuando llega un cliente y cuando un cliente ya arribado modifica sus keywords. Las tablas de este tipo son 'cliente' y 'keyword'. En el caso de las keywords está habilitado un script en java que lee un archivo en formato *csv* que asigna keywords para un cliente ahí señalado. Las columnas del archivo son 3: 'trackeable', 'fecha_final' e 'idcliente'. Esto se usará cuando sean muchas las keywords a insertar. Para menos de 10 keywords la opción manual es más cómoda y rápida.

- Automática: Es la tabla 'tweet'. Ésta se actualiza cada 15 minutos a través de un script de java que corre desde el servidor en Amazon.

A continuación se detalla la construcción de las tablas Iniciales, Intermitentes y Automática.

4.4.1. Calendario

La tabla 'calendario' contiene todos los días del mes para todos los meses del año para todos los años desde el 2014 hasta el 2025. Los datos se crean y se insertan desde un archivo Java, cuyo pseudocódigo se muestra a continuación:

```
year = 2014;
while (year <= 2025) {
    mes = 1;
    while (mes <= 12) {
        dia = 1;
        while (dia <= 31) {
            if (dia <= Limite(mes,year)){
                Insertar(dia,mes,year);
            }
            dia += 1;
        }
        mes += 1;
    }
    year += 1;
}
```

La función 'Limite(mes,year)' arroja el último día del mes para cada año. Es necesario incluir el año, pues febrero cambia su número de días en los años biciestos.

4.4.2. Tiempo

A imagen de la tabla 'calendario', la tabla 'tiempo' contiene cada segundo, de cada minuto de cada hora de un día. Sin embargo la tabla 'tiempo' es más regular, lo que hace el código más sencillo como se puede apreciar en el pseudocódigo que se muestra a continuación:

```
hora = 0;
while (hora <= 23) {
    minutos = 0;
    while (minutos <= 59) {
        segundos = 0;
        while (segundos <= 59) {
            Insertar(hora,minutos,segundos);
            i+=1;
```

```
                segundos += 1;
            }
            minutos += 1;
        }
        hora += 1;
    }
```

4.4.3. Usuario

Los usuarios se obtienen de la base de datos en postgresql que, para los efectos del Data Warehouse, funciona como Data Staging Area. En PostgreSQL los datos están contenidos en formato JSON. Se quiere rescatar de ellos sólo la información importante para el Data Warehouse, como son las columnas 'id', 'name', 'screen_name' y 'profile_img_url'. Se aplica también la función que detecta el sexo de un usuario a partir de su nombre.

```
Data = ObtenerData(usuarios, PostgresqlDB);
while (JSON_usuario = Data.siguiente()) {
    id = JSON_usuario("id");
    name = JSON_usuario("name");
    screen_name = JSON_usuario("screen_name");
    imagen = JSON_usuario("profile_image_url");
    //1: hombre, 2: mujer
    sexo = Helpers.ObtenerSexo(name, screen_name);
    Insertar(id,name,screen_name,imagen,sexo);
}
```

4.4.4. Interés

Los datos de la tabla 'interes' se obtienen desde la aplicación de Klout. Se pretende tener la mayor cantidad de intereses posible, siendo ideal tenerlos todos. Para ello se pregunta por los intereses de todos los usuarios y se van insertando sin repetirlos.

```
IDs = ObtenerIDs();
while (id = IDs.siguiente()) {
    intereses = Klout.ObtenerIntereses(id);
    for (interes en intereses) {
        Insertar(interes);
    }
}
```

4.4.5. Influencia

Los datos de la tabla 'influencia' se obtienen también de la aplicación de Klout. Esta tabla fact está relacionada también con calendario, por lo que se pide para la inserción del nuevo registro, el identificador del día en que se ejecuta el script a la tabla calendario. En el caso en que la API de Klout no cuente a ese usuario de Twitter en sus registros, se le impone una influencia con valor 0.

```
idcalendario = ObtenerIDcalendario(fechaActual());
//tercer parámetro es la tabla
Resultado = Obtener(idusuario, idtwitteruser, usuario);
while (auxiliar = Resultado.siguiente()) {
    idusuario = auxiliar(1);
    idtwitteruser = auxiliar(2);

    try{ influencia = Klout.ObtenerInfluencia(idtwitteruser); }
    catch { influencia = 0; }

    Insertar(influencia, idcalendario, idusuario);
}
```

4.4.6. Rel_usuario_interes

La tabla 'rel_usuario_interes' se llena haciendo la conexión entre la tabla 'usuario' y la tabla 'interes', aunque debe considerar también la dimensión 'calendario'. El inicio del código es muy similar al anterior. Si no existe el interés, ese registro simplemente se deshecho y no es insertado.

```
idcalendario = ObtenerIDcalendario(fechaActual());
//tercer parámetro es la tabla
Resultado = Obtener(idusuario, idtwitteruser, usuario);
while (auxiliar = Resultado.siguiente()) {
    idusuario = auxiliar(1);
    idtwitteruser = auxiliar(2);

    topicos = Klout.ObtenerIntereses(idtwitteruser);
    for (topico en topicos) {
        if (topico == null) { break; }
        //segundo parámetro es la tabla
        try{ idinteres = ObtenerIdInteres(topico, interes); }
        catch{ idinteres = -1; }
        if (idinteres != -1) {
            Insertar(idusuario, idinteres, idcalendario);
        }
    }
}
```

4.4.7. Tweet

La tabla 'tweet' es la última tabla de hechos y por lejos la más transaccional. Cada 15 minutos comienza un nuevo proceso de inserción de registros, pero si los procesos demoraran más de un cuarto de hora, a la tabla tweet se le pueden insertar registros sin parar. El proceso de ETL para la carga de *tweet* es el más complejo, por la longitud que tiene y por los atributos y algoritmos que requiere.

```
 //Obtener idkeyword y trackeable de keywords a buscar
 //Los últimos 2 parámetros son tablas: keyword, cliente
Keywords = ObtenerKeywordsVigentes(idkeyword, trackeable,
    keyword, cliente);

 JsonNode = ObtenerTweets(Keywords.trackeables,
    minutos=15, limite=700);

 for (objNode en JsonNode) {
    id = objNode.Obtener("id");

    // Si el lenguaje no es español, escapar el proceso
    lang =  objNode.Obtener("lenguaje");
    if (!lang.equals("es")) { continue; }


    //Se pregunta si es un retweet o no. Si no, en el campo
    // retweet se pone la misma id (id de Twitter)
    if (objNode.has("retweeted_status_id")) {
        retweet = objNode.Obtener("retweeted_status_id");
        boolRT = 1;
    } else {
        retweet = id;
        boolRT = 0;
    }

    user_id = objNode.Obtener("user_id");
    text = objNode.Obtener("text");
    created_at = objNode.Obtener("created_at");

    //Encontrar idusuario de usuario con idtwitteruser = user_id
    idusuari = -1;
    idusuario = Obtener(idusuario, user_id, usuario);

    //Obtener polaridad. Se pregunta primero por si existe
    // la polaridad para ese texto en la base de datos.
    // Si no, se recurre a la PAPI, que es más lenta.
    if (boolRT == 1) {
```

```
            try{ polaridad = Obtener(polaridad, retweet, tweet); }
            catch {
                } else {
                    polaridad = PAPI.ObtenerPolaridad(texto);
                }
        } else {
            polaridad = PAPI.ObtenerPolaridad(texto);
        }

        //Encontrar las idkeyword  de las keywords relacionadas
        // al tweet y guardarlas en un arreglo
        idkeyword = -1;
        for (keyword en Keywords) {
            if (text.contiene(keyword.trackeable)) {
                idkeyword = keyword.idkeyword;
                arreglo_keywords.add(idkeyword);
            }
        }


        //Obtener influencia para cálculo de indicador
        influencia = 0.0;
        if (idusuario != -1) {
            influencia = Obtener(influencia, idusuario, influencia);
        }
        //Se normaliza
        influencia = (influencia^2)/1000;
        impacto = influencia * polaridad;


        //encontrar dimensiones calendario y tiempo
        year = ObtenerYear(created_at);
        mes = ObtenerMes(created_at);
        dia = ObtenerDia(created_at);
        hora = ObtenerHora(created_at);
        minutos = ObtenerMinutos(created_at);
        segundos = ObtenerSegundos(created_at);
        idcalendario = Obtener(idcalendario, year, mes, dia, calendario);
        idtiempo = Obtener(idtiempo, hora, minutos, segundos, tiempo);

        //Insertar para todas las keywords
        for (idkeyword en arreglo_keywords ) {
            Insertar(text, polaridad, impacto, idtwittertweet, teweet,
                boolRT, usuario_idusuario, calendario_idcalendario,
                tiempo_idtiempo, keyword_idkeyword);
        }
    }
```

Una vez finalizado el ETL, teniendo el Data Warehouse diseñado, construido y con datos en todas las tablas, sólo resta el análisis y la visualización ordenada a él en la página Web.

Capítulo 5

Servicios

El sistema OpinionZoom ofrece servicios de análisis de opiniones en Twitter. Hay cuatro grupos de servicios, que se explican en este capítulo de forma detallada. Primero se presenta brevemente a los cuatro grupos:

1. Artículos de interés general: es el primero de los servicios, es atómico y está abierto para todo usuario de la página Web. Consta de publicaciones periodicas sobre un tema de interés nacional, ya sea de contingencia política, deportiva o de alguna otra realidad que marque tendencia en Twitter.

2. Buscador de *tweet*: es un servicio para los clientes de OpinionZoom. Sirve para explorar *tweets* que contengan la palabra buscada.

3. Inteligencia de clientes: es el servicio brevemente introducido en el capítulo 1, y tiene que ver con una serie de mediciones sobre los usuarios de las redes sociales, futuros o actuales clientes de la organización que contrata este servicio, y sus opiniones.

4. Alertas: es un servicio que permite alertarle al cliente de OpinionZoom sobre la variación inusual de algún indicador en las keywords que le interese incluir.

Se incluye además una sección de Módulo de Administración, considerándolo como un servicio de uso interno para el administrador del sistema. A continuación, y en este mismo orden, se ofrece una explicación exhaustiva y detallada de cada uno de estos grupos de servicios.

5.1. Artículos de Interés General

OpinionZoom ofrece a la comunidad artículos de interés general obtenidos de la minería de opiniones en Twitter. Éstos pueden ser accesados gratuitamentes desde el Home de la página Web. Tienen como fin atraer la atención de prospectos a través de mostrar las capacidades del sistema. Estos artículos son construidos periódicamente y son estáticos, es decir, una vez publicados no varían ni en forma ni en contenido. A modo de ejemplo, la figura 5.1 muestra el primer artículo de

OpinionZoom, que da a conocer la popularidad en Twitter de la que goza el arquero y capitán de la selección chilena de fútbol, Claudio Bravo.

Figura 5.1: Artículo de interés general

Artículos / Keywords / Popular

Claudio Bravo, liderando también en Twitter

El portero titular de la selección chilena de fútbol, Claudio Bravo, no sólo lidera dentro de la cancha, sino que también supera a sus colegas en popularidad en Twitter. El capitán de La Roja, que 'twitea' varias veces por semana, supera los 1.190.000 seguidores, que siguen aumentando con sucriptores de todo el globo. OpinionZoom hizo un seguimiento por 1 mes a los futbolistas de nuestra selección, y el resultado fue concluyente: las apariciones de la cuenta @C1audioBravo en los tweets del territorio nacional superan por mucho a las apariciones de Alexis Sánchez (@Alexis_Sanchez) y Arturo Vidal (@kingarturo23). Sólo Gary 'Pitbull' Medell (@MedelPitbull) le muerde los talones.

Hay otras métricas interesantes en las que también destaca el connotado deportista. El portero es el primero también en polaridad, teniendo un promedio de 2,5 para este indicador. Le sigue justamente Gary Medel, con 2,3 puntos. Sin duda los jugadores más queridos por la 'Marea Roja'. La hinchada nacional también aprecia, en menor medida, al 'Rey Arturo' con 1,6 de polaridad. La siguiente tabla, liderada por el arquero del FC Barcelona, muestra el número de apariciones de cada jugador durante el mes de septiembre, junto con la polaridad promedio.

Futbolista	Cuenta	Menciones	Polaridad
Claudio Bravo	@C1audioBravo	8516	2,5
Gary Medel	@MedelPitbull	8005	2,3
Alexis Sánchez	@Alexis_Sanchez	3391	0,7
Arturo Vidal	@kingarturo23	2782	1,6

Fuente: Elaboración Propia

5.2. Buscador de Tweets

Con el fin de explorar preliminarmente lo que los usuarios de Twitter hablan sobre un tema, se dispone para todo usuario cliente de OpinionZoom un buscador de *tweets* por palabra. Se ingresa una palabra en el buscador y se muestran todos los *tweets* que contengan esa palabra. Se incluyen además dos columnas para enriquecer la exploración, una con la polaridad del *tweet* y la otra con el impacto del *tweet*. La figura 5.2 enseña los resultados de la búsqueda de la palabra 'marihuana'.

Figura 5.2: Búsqueda de *tweets*

Búsqueda de 'marihuana' ESTÁS AQUÍ: Servicios › Búsqueda de Tweets

Id	Texto	Polaridad	Impacto
1	soñe que tenia mucha marihuana y desperte u_u	0.00	0.00
2	Promulgan leyes en California que regulan el negocio de la marihuana medicinal http://t.co /iotxjr9qvv	0.00	0.00
3	Promulgan leyes en California que regulan el negocio de la marihuana medicinal http://t.co /9ah4Y3DJOt #FollowBackSeguro	0.00	0.00
4	RT @pongodetodo: 7 razones científicas por las que nunca deberías fumar marihuana http://t.co /zPxbdZb1cA http://t.co/60c3buiE12	3.00	0.00
5	RT @danialmond: Me cargan esas personas que andan con poleras de marihuana, onda a mi me gusta la chorillana y no me ven por ahí presumiénd...	2.00	1.46
6	RT @danialmond: Me cargan esas personas que andan con poleras de marihuana, onda a mi me gusta la chorillana y no me ven por ahí presumiénd...	2.00	0.00
7	@malandrinna fiel a la #marihuana...	2.00	0.00
8	#10CosasQueSoloPasanEnChile la marihuana es una droga dura....	-3.00	0.00

Fuente: Elaboración Propia

5.3. Inteligencia de Clientes

La inteligencia de clientes guarda relación con la escucha de los usuarios de las redes sociales con respecto a algún tema en particular, representándose éste con varias keywords. Esto incluye identificar a quienes están hablando de este tópico, cómo hablan de él y qué les interesa.

Se detallan a continuación todas las vistas correspondientes a Inteligencia de Clientes ordenadas en 3 conjuntos, tal como son presentadas e la página Web: Explora, Keywords e Indicadores. Los resultados mostrados en este capítulo corresponden a un cliente artificial creado en el WIC, que tiene 29 keywords y 8 alertas contratadas.

5.3.1. Explora

Este conjunto de gráficos y tablas muestra información agregada de todas las keywords contratadas por el usuario cliente. Se compone de un gráfico y 4 tablas. El objetivo es hacer un pequeño análisis exploratorio de lo que sucede en Twitter con respecto a las keywords contratadas.

El gráfico que se muestra en la fugura 5.3 muestra la variación de la frecuencia en el transcurso del día. Esto se obtiene contando los *tweets* emitidos por hora en los últimos 7 días. En la figura se puede ver cómo distribuye esta frecuencia dentro de un día para todas las keywords. En el eje de las ordenadas se muestra la frecuencia y en el de las abscisas la hora del día, desde las 21:00 horas hasta las 12 del día.

Figura 5.3: Frecuencia por hora

Fuente: Elaboración Propia

La tabla de la figura 5.4 lista los intereses complementarios de los usuarios de Twitter que han comentado al menos sobre alguna de las keywords contratadas. Esto sirve para conocer más a los prospectos o actuales clientes del cliente de OpinionZoom. La primera columna muestra el interés

y la columna de la izquierda, 'Adeptos', muestra el número de usuarios de Twitter que adhieren a ese interés.

Figura 5.4: Intereses Complementarios

INTERESES COMPLEMENTARIOS

Interés	Adeptos
Chile	138
Venezuela	42
Talca	39
Journalism	35
Spanish	19
WhatsApp	14
Antofagasta	13
Wine	12

Fuente: Elaboración Propia

La tabla de la figura 5.5 muestra a los usuarios de Twitter más influyentes que han opinado sobre alguna de las keywords en la última semana. Las columnas son 7: un identificador temporal, su imagen, el nombre de usuario, el *screen name*, la cantidad de veces que ha mencionado las keywords y su influencia.

Figura 5.5: Usuarios influyentes

USUARIOS INFLUYENTES

Mostrar 10 registros Buscar:

Nº	Imagen	Usuario	Screen Name	Frecuencia	Influencia
1		ZOUISPORRITOLOVERS	@sevastianvaldes	4	46.14
2		Luis Huacuja Acevedo	@Luis_Huacuja_A	3	39.99
3		El Boyaldía #Iquique	@ElBoyaldia	27	52.21
4		Radio Grodek Oficial	@RADIOGRODEKCOMP	11	44.98
5		Toto Stark ??	@ViejoZorroDeMar	2	41.17
6		Carlos Zárate V.	@CarlosZarateV	8	64.47
7		DIARIO DEPORTES	@DIARIODEPORTES	3	63.83

Fuente: Elaboración Propia

Por último hay 2 tablas, en las figuras 5.6 y 5.7, que muestran los 5 *tweets* más negativos y los 5 más positivos que hayan mencionado alguna de las keywords. Los *tweets* se ordenan por polaridad,

donde el más negativo es el primero en el caso de los más negativos y el más positivo es el primero para la tabla con los *tweets* más positivos.

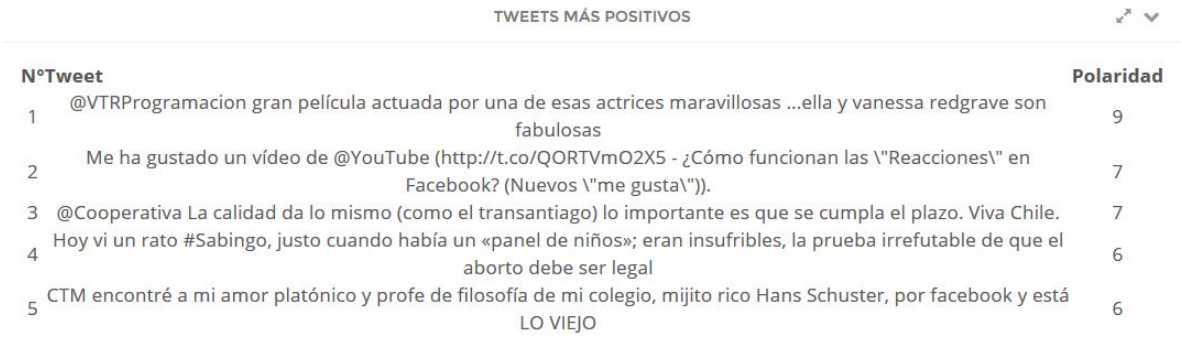
Figura 5.6: Los tweets más negativos

TWEETS MÁS NEGATIVOS

N°	Tweet	Polaridad
1	Terrible en la comuna de la Florida robaron camioneta Ford Explore con mucha violencia se detuvo en un semáforo en rojo delincuencia	-17
2	RT @alexis__1982: #10CosasQueSoloPasanEnChile Compran productos robados, pero se andan quejando de la delincuencia.	-11.4
3	RT @balenttina: Ay. Cuando Chile metió el gol, sentí como un leve terremoto. Pero fue un terremoto feliz ☺?.	-10
4	Gobierno condena y pide denunciar presuntos fraudes relacionados al terremoto #Ovalle #MtePatria #Punitaqui Coquimbo http://t.co/oOykanLwPj	-10
5	@MovistarChile es increíble la incompetencia del Call center, más de 5 llamadas y nadie sabe dar una respuesta, un asco de empresa	-10

Fuente: Elaboración Propia

Figura 5.7: Los tweets más positivos



TWEETS MÁS POSITIVOS

N°	Tweet	Polaridad
1	@VTRProgramacion gran película actuada por una de esas actrices maravillosas ...ella y vanessa redgrave son fabulosas	9
2	Me ha gustado un vídeo de @YouTube (http://t.co/QORTVmO2X5 - ¿Cómo funcionan las \"Reacciones\" en Facebook? (Nuevos \"me gusta\")).	7
3	@Cooperativa La calidad da lo mismo (como el transantiago) lo importante es que se cumpla el plazo. Viva Chile.	7
4	Hoy vi un rato #Sabingo, justo cuando había un «panel de niños»; eran insufribles, la prueba irrefutable de que el aborto debe ser legal	6
5	CTM encontré a mi amor platónico y profe de filosofía de mi colegio, mijito rico Hans Schuster, por facebook y está LO VIEJO	6

Fuente: Elaboración Propia

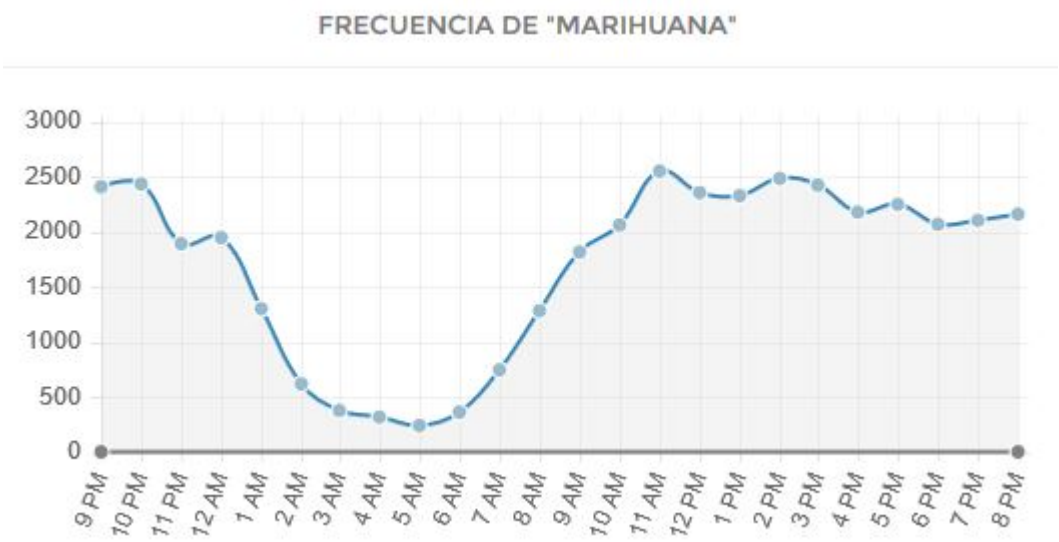
5.3.2. Keywords

Se pueden revisar las keywords contratadas, una por una, en esta sección. La información desplegada se divide en 5 gráficos y 3 tablas.

El primer gráfico, figura 5.8, muestra la frecuencia de *tweets* por hora. Esto se obtiene contando los *tweets* emitidos por hora en los últimos 7 días. En el eje de las ordenadas se muestra la frecuencia y en el de las abscisas la hora del día.

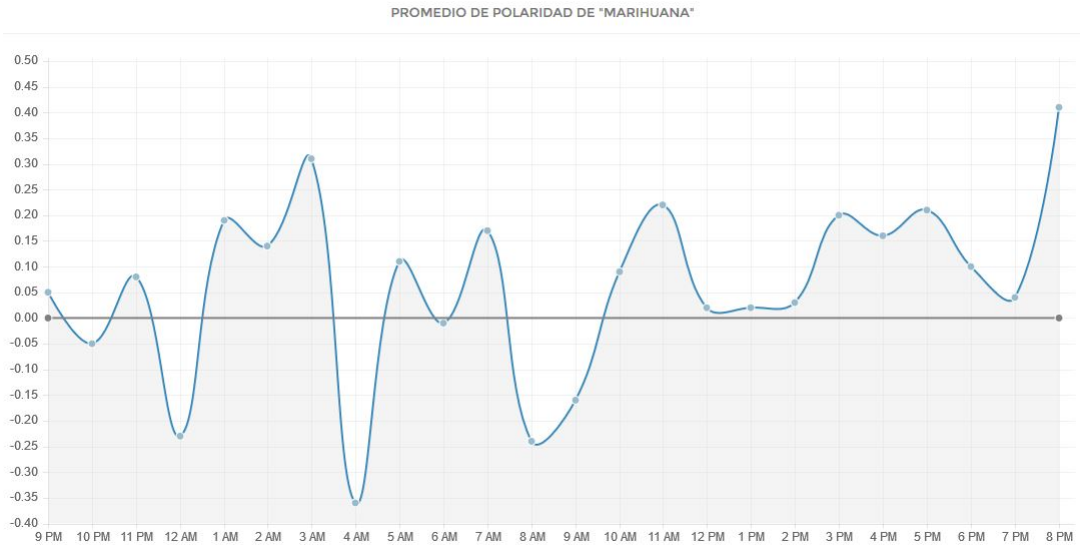
El segundo gráfico, figura 5.9, muestra el promedio de polaridad de los *tweets* por hora. Esto se obtiene promediando la polaridad de los *tweets* emitidos por hora en los últimos 7 días. En el eje de las ordenadas se muestra el promedio de la polaridad y en el de las abscisas la hora del día.

Figura 5.8: Frecuencia por hora para una keyword



Fuente: Elaboración Propia

Figura 5.9: Promedio de polaridad por hora para una keyword



Fuente: Elaboración Propia

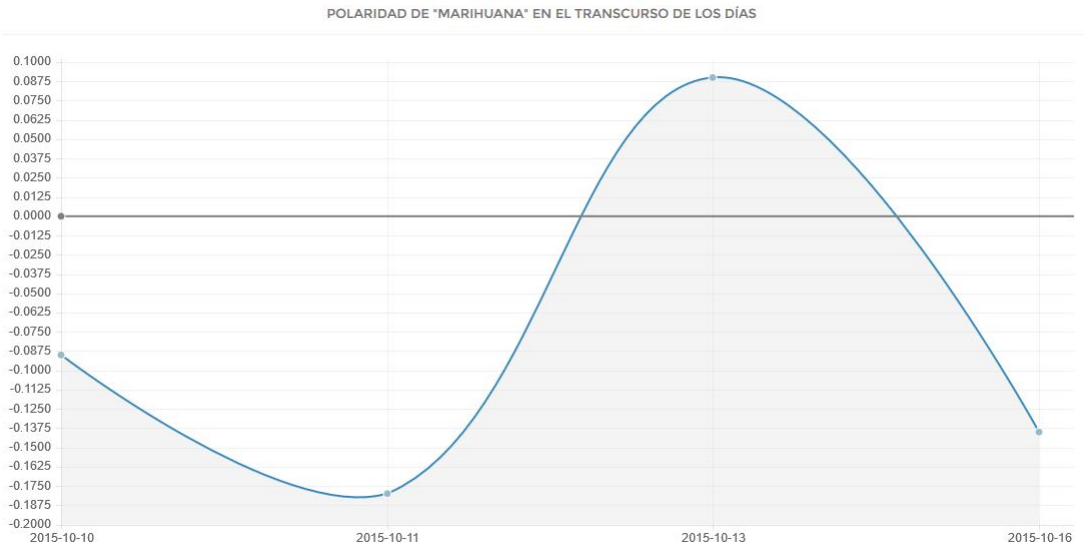
El tercer gráfico, figura 5.10, muestra la frecuencia de *tweets* en los últimos días. En el eje de las ordenadas se muestra la cantidad de *tweets* emitidos y en el de las abscisas el día calendario correspondiente.

Figura 5.10: Frecuencia en los últimos días para una keyword

Fuente: Elaboración Propia

El cuarto gráfico, figura 5.11, muestra la polaridad promedio de los *tweets* emitidos en los últimos días. En el eje de las ordenadas se muestra el promedio de polaridad y en el de las abscisas el día calendario correspondiente.

Figura 5.11: Promedio de polaridad en los últimos días para una keyword



Fuente: Elaboración Propia

Luego viene la tabla, en la figura 5.12, de usuarios influyentes que han estado hablando de la keyword durante los últimos 7 días. Las columnas son 6: un identificador temporal, su imagen,

el nombre de usuario, el *screen name*, la cantidad de veces que ha mencionado la keyword, su influencia y el promedio de polaridad de los *tweets* emitidos.

Figura 5.12: Usuarios influyentes para una keyword

USUARIOS INFLUYENTES QUE OPINAN DE DE "MARIHUANA"

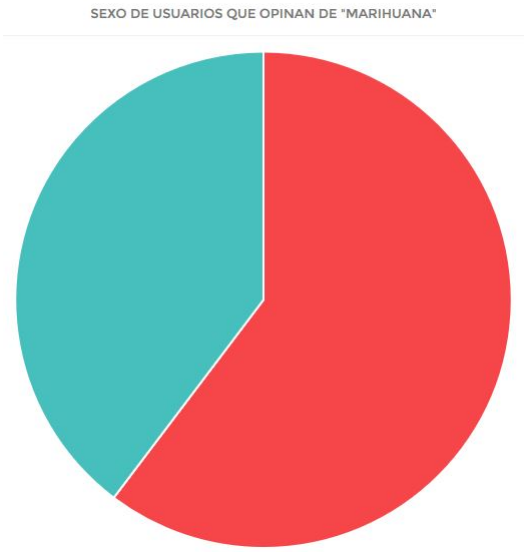
Mostrar 10 registros Buscar:

N°	Imagen	Usuario	Screen Name	Frecuencia	Influencia	Polaridad
1		Alejandro Almeida	@alealm1984	396	18.58	0.17
2		Ana María Gazmuri	@AnaMariaGazmuri	299	62.04	0.15
3		Rodrigo Cabrera	@Ro_Cbra	233	27.94	0.79
4		Fundación Daya	@FundacionDaya	153	53.87	0.26
5		Flor Farías	@Flor_Farii	108	36.37	0.76
6		paulina bobadilla	@paulibobadilla_	108	47.19	0.34
7		Psychocandy	@AlvaroGoo	80	36.94	0.24
8		Claudio Oyarzun™	@ClaudioOyarzunL	61	50.5	0.50

Fuente: Elaboración Propia

El último gráfico en mostrarse es el del sexo de los usuarios, en la figura 5.13. Es un gráfico de torta con dos variables: la cantidad de usuarios hombres y la cantidad de usuarios mujeres que han mencionado la keyword en la última semana.

Figura 5.13: Sexo de usuarios para una keyword



Fuente: Elaboración Propia

Por último, en el fondo de la vista, se despliegan 2 tablas. La primera tabla, figura 5.14, muestra los intereses complementarios. Ésta tiene 3 columnas: un identificador temporal, el interés y el número de usuarios que lo tiene.

Figura 5.14: Interés complementario para una keyword

INTERESES COMPLEMENTARIOS DE "MARIHUANA"

Mostrar 10 registros Buscar:

N°	Interés	Adeptos
1	Chile	6660
2	Journalism	1807
3	Spanish	1256
4	WhatsApp	1229
5	Talca	1009
6	Caracas	692
7	DJs	462
8	eReaders	460
9	Celebrities	419
10	Hillsong United	394

Mostrando registros del 1 al 10 de un total de 20 registros Anterior 1 2 Siguiente

Fuente: Elaboración Propia

Finalmente una pequeña tabla, figura 5.15, muestra los indicadores correspondientes a la keyword: promedio de polaridad, promedio de influencia y promedio de impacto.

Figura 5.15: Indicadores de una keyword

INDICADORES DE "MARIHUANA"

Polaridad	Influencia	Impacto
-0.51	38.90	-1.05

Fuente: Elaboración Propia

5.3.3. Indicadores

Para las keywords hay 4 indicadores, que ya fueron mostrados desagregadamente en el conjunto de la sección anterior. En esta sección, los indicadores se muestran todos juntos para todas las keywords listadas en una misma tabla, como se puede ver en la figura 5.16, facilitándole al usuario cliente la revisión y comparación de keywords. Los indicadores son: frecuencia, polaridad promedio, influencia promedio e impacto promedio, tal como se definen en el capítulo 4.

Figura 5.16: Indicadores para todas las keywords

PROMEDIOS

Mostrar 10 registros Buscar:

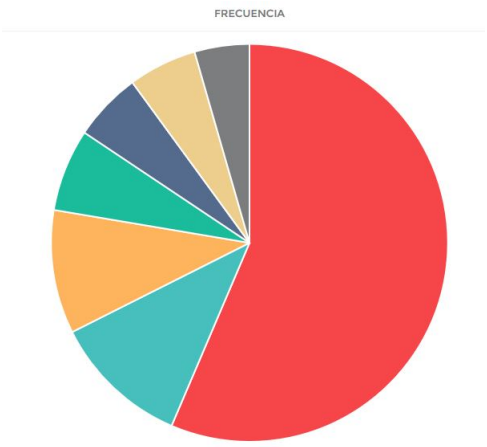
N°	Keyword	Frecuencia	Polaridad	Influencia	Impacto
1	Facebook	101	0.34	31.92	0.98
2	@AyudaMovistarCL	20	0.50	34.21	1.03
3	terremoto	18	-4.39	35.32	-16.42
4	aborto	12	0.50	39.82	-0.23
5	@entel	10	-1.95	33.35	-9.15
6	delincuencia	10	-4.20	39.58	-13.61
7	transantiago	8	-0.50	46.22	-2.15
8	temblor	7	0.29	36.95	-1.07
9	@zonaEntel	3	2.67	37.19	5.42
10	marihuana	3	-1.33	43.09	-1.94

Mostrando registros del 1 al 10 de un total de 18 registros Anterior 1 2 Siguiente

Fuente: Elaboración Propia

Además de la tabla, se muestra un gráfico de torta, figura 5.17, que muestra la frecuencia de las 7 keywords contratadas más populares. No existe leyenda, pues una de las características de este gráfico interactivo es que la leyenda aparece al colocar el mouse sobre algún color.

Figura 5.17: Frecuencia de principales keywords



Fuente: Elaboración Propia

5.4. Sistema de Alertas

La sección alertas tiene 3 vistas: (1) 'Alertas', donde se pueden ver las alertas contratadas; (2) 'Puntual', donde se pueden ver las alertas puntuales que se han producido; y (3) 'Generalizada', donde se pueden ver las alertas generalizadas que se han producido. Hay 2 tipos de alertas, las puntuales y las generalizadas. Las alertas puntuales refieren a aquellos *tweets* que por su polaridad, influencia o impacto deberían ser guardados. Las alertas generalizadas, por otro lado, apuntan a aquellos *tweets* que refieran a un conjunto de keywords que, superando ciertos umbrales de frecuencia, polaridad o promedio, requieran atención del cliente de OpinionZoom.

5.4.1. Alertas

En esta vista se despliegan 2 tablas, la tabla de alertas puntuales y la tabla de alertas generalizadas. Ambas difieren muy poco en sus parámetros. La tabla de alertas puntuales, en la figura 5.18, tiene 9 columnas: el identificador de la base de datos, el nombre de la alerta, un umbral negativo de polaridad, un umbral positivo de polaridad, un umbral de influencia, un umbral negativo de impacto, un umbral positivo de impacto, fecha de creación y fecha final.

Figura 5.18: Alertas Puntuales

ALERTAS PUNTUALES

Mostrar 10 registros Buscar:

ID	Nombre	Umbral negativo de polaridad	Umbral positivo de polaridad	Umbral de influencia	Umbral negativo de impacto	Umbral positivo de impacto	Fecha de Creación	Fecha Final
1	Aborto Positivo	-15	10	100	-150	150	2015-09-25 16:21:25	2015-11-27 00:00:00
2	Reclamos Entel	-3	15	100	-150	150	2015-09-25 17:20:35	2015-11-25 00:00:00
5	Marihuana buena	-15	7	100	-150	150	2015-09-25 17:38:40	2015-11-27 00:00:00

Mostrando registros del 1 al 3 de un total de 3 registros Anterior 1 Siguiente

Fuente: Elaboración Propia

La tabla de alertas generalizadas, en la figura 5.19, tiene a su vez también 9 columnas: el identificador de la base de datos, el nombre de la alerta, un umbral de frecuencia, un umbral negativo de polaridad, un umbral positivo de polaridad, un umbral negativo de impacto, un umbral positivo de impacto, fecha de creación y fecha final.

Figura 5.19: Alertas Generalizadas

ALERTAS GENERALIZADAS

Mostrar 10 registros Buscar:

ID	Nombre	Umbral de Frecuencia	Umbral negativo de polaridad	Umbral positivo de polaridad	Umbral negativo de impacto	Umbral positivo de impacto	Fecha de Creación	Fecha Final
1	Popular	30	-15	15	-150	150	2015-09-24 11:36:45	2015-12-21 00:00:00
2	Desprecio	100	-0.4	15	-150	150	2015-09-24 17:08:57	2015-11-26 00:00:00
3	Positiva	100	-15	0.1	-150	150	2015-09-24 17:26:22	2015-11-27 00:00:00
4	Frecuente	20	-15	15	-150	150	2015-09-24 18:20:29	2015-09-30 00:00:00
5	Frecuente2	15	-15	15	-150	150	2015-09-25 12:58:08	2015-12-25 00:00:00

Mostrando registros del 1 al 5 de un total de 5 registros Anterior 1 Siguiente

Fuente: Elaboración Propia

5.4.2. Puntuales

Cuando se produce una alerta puntual se crea un registro que luego se muestra en esta vista. La tabla tiene 8 columnas, como se ve en la figura 5.20: el identificador del registro, le fecha y hora del *tweet*, el texto del *tweet*, su polaridad, su influencia, su impacto, el identificador del *tweet* para Twitter y la id de la alerta a la que corresponde. El identificador del *tweet* tiene una característica atractiva: al presionar sobre él se despliega en una nueva pestaña del navegador el *tweet* en la página de Twitter.

Figura 5.20: Registro de alertas puntuales

ALERTAS PUNTUALES

Mostrar 10 registros Buscar:

ID	Fecha	Tweet	Polaridad	Influencia	Impacto	ID de Twitter	ID de Alerta
2	2015-09-29 11:15:59	Muere mi celular, voy a @entel me dan uno de repuesto y después de menos de un día el cargador no funciona.. Como los odio.	-11	41.64	-18.491	648853577646108672	2
5	2015-09-30 15:30:27	@_VjArt @MovistarChile @entel me saturo, es extremadamente caro renovar el equipo!	-3.05	41.82	-5.12705	649289917894410240	2
6	2015-10-01 17:45:14	Puta la wea @entel_ayuda @entel no me responden, mas de una hora tratando que los qls me contesten un tweet ¬¬ como la callampa	-6	18.35	-1.944	649686220696604672	2
7	2015-10-01 20:30:14	@entel no ofrece cambio de equipo a clientes, @MovistarChile mal servicio, @clarochile_cl mala señal. ¿Qué recomiendan?	-6	40.69	-9.600000000000001	649727761259606016	2

Fuente: Elaboración Propia

5.4.3. Generalizadas

Cuando se produce una alerta generalizada se crea un registro que luego se muestra en esta vista. Cuando la alerta se acaba se marca la fecha y la hora en que termina. La tabla tiene 7 columnas, como se ve en la figura 5.21: el identificador del registro, la fecha y hora de inicio de la alerta, la fecha y hora de finalización de la alerta, frecuencia, polaridad, impacto y la id de la alerta a la que corresponde.

Figura 5.21: Registro de alertas generalizadas


ALERTAS GENERALIZADAS

Mostrar 10 registros Buscar:

ID	Inicio	Término	Frecuencia	Polaridad	Impacto	ID de Alerta
81	2015-09-26 00:19:14	2015-09-26 00:31:17	16	-0.69	0	5
82	2015-09-26 00:26:04	2015-09-26 00:41:04	32	0.62	0.35	1
83	2015-09-26 00:32:14	2015-09-26 00:57:12	16	-1.25	0	5
84	2015-09-26 00:49:04	2015-09-26 00:51:04	31	0.1	-0.34	1
85	2015-09-26 00:58:05	2015-09-26 01:15:09	33	0.29	0	1
86	2015-09-26 01:31:11	2015-09-26 01:35:18	16	1.75	1.17	5
87	2015-09-26 02:27:15	2015-09-26 02:50:28	17	0.81	0.72	5
88	2015-09-26 07:47:12	2015-09-26 08:15:24	22	0.23	0.35	4
89	2015-09-26 07:47:18	2015-09-26 07:55:31	17	0.53	0.07	5
90	2015-09-26 08:19:10	2015-09-27 03:28:10	29	0.72	0.04	4

Mostrando registros del 81 al 90 de un total de 430 registros Anterior 1 ... 8 9 10 ... 43 Siguiente

Fuente: Elaboración Propia

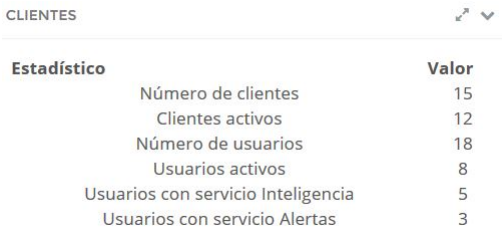
5.5. Módulo de Administración

Éste módulo es exclusivo para los usuarios administradores de la página, y en este sentido es un servicio interno de la página para el propio equipo de OpinionZoom. Éste módulo tiene 3 vistas: (1) 'Estadísticas', (2) 'Clientes' y (3) 'Usuarios'.

5.5.1. Estadísticas

Ésta es la vista inicial para el usuario administrador, la que aparece cuando inicia sesión. En ella puede ver las variables del sistema respecto a los clientes y a los usuarios. Como se puede ver en la figura 5.22 hay 6 variables que se muestran en una única tabla, que tiene 2 columnas: el estadístico y el valor. Las variables son: cantidad de clientes, cantidad de clientes activos (éstos son los cuya fecha de vencimiento aún no llega), cantidad de usuarios, cantidad de usuarios activos (los que pueden iniciar sesión), cantidad de usuarios con servicio Inteligencia de Clientes y cantidad de usuarios con servicio Alertas.

Figura 5.22: Estadísticas para el administrador



CLIENTES

Estadístico	Valor
Número de clientes	15
Clientes activos	12
Número de usuarios	18
Usuarios activos	8
Usuarios con servicio Inteligencia	5
Usuarios con servicio Alertas	3

Fuente: Elaboración Propia

5.5.2. Clientes

El administrador puede ver a todos los clientes de OpinionZoom, tanto si están activos como si no lo están. La tabla de la figura 5.23 tiene 5 columnas: el identificador del cliente, el nombre del cliente, cuando fue creado, su fecha de vencimiento y la cantidad de keywords a él asociados.

Figura 5.23: Clientes de OpinionZoom

CLIENTES

Mostrar 10 registros Buscar:

ID	Nombre	Creado	Vencimiento	Cantidad de Keywords
8	Politica	2015-07-14 00:00:00	2015-12-21	23
11	WIC	2015-07-15 00:00:00	2015-12-21	21
9	Copa América Chile 2015	2015-07-14 00:00:00	2015-12-21	9
10	Amigos	2015-07-14 00:00:00	2015-12-21	7
1	ENTEL	2015-07-14 00:00:00	2015-12-21	4
4	VTR	2015-07-14 00:00:00	2015-12-21	3
3	Movistar	2015-07-14 00:00:00	2015-12-21	2
2	Claro	2015-07-14 00:00:00	2015-12-21	1
15	Transantiago	2015-09-24 13:34:34	2016-04-14	1
5	Nextel	2015-07-14 00:00:00	2015-12-21	0

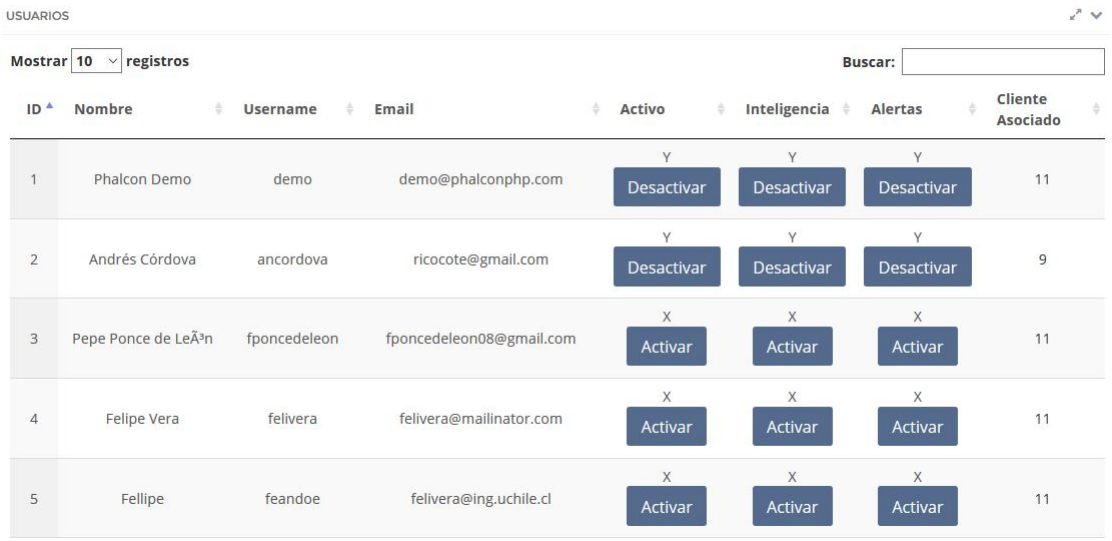
Mostrando registros del 1 al 10 de un total de 15 registros Anterior 1 2 Siguiente

Fuente: Elaboración Propia

5.5.3. Usuarios

El administrador puede ver a todos los usuarios OpinionZoom, tanto si están activos como si no lo están. La tabla de la figura 5.24 tiene 7 columnas: el identificador del usuario, el nombre de usuario, su 'username', su email, si está activo, si tiene contratado el servicio de inteligencia y si tiene contratado el sistema de alertas. Una muy buena funcionalidad del sistema es la opción de activar o desactivar a un usuario o a alguno de sus servicios con un sólo 'click' en un botón azul que se encuentra en la misma columna en la que se muestra el estado del usuario o sus servicios.

Figura 5.24: Usuarios del sistema



Fuente: Elaboración Propia

Capítulo 6

Página Web

La página Web es la forma de comunicación por excelencia de los usuarios con el sistema OpinonZoom, sin perjuicio de que los clientes de la aplicación y el equipo de OpinionZoom interactúen por otros medios, por ejemplo para la contratación de servicios. Su existencia es esencial al proyecto, no entendiéndose éste sin ella. La página Web arroja las salidas del sistema.

En este capítulo se mencionan la propuesta de valor que tiene el prototipo, los materiales de construcción, los requerimientos de usuario y los requisitos de software. Estos dos últimos muestran la construcción de la página según la convención de ingeniería de software.

6.1. Propuesta de valor

Existen otras plataformas Web que otorgan servicios similares a los de OpinionZoom, tales como Brandmetric [48] o Wholemeaning [49]. Existen similitudes tanto en los servicios como en las fuentes de datos, como Klout y Twitter. Brandmetric, por ejemplo, publica artículos periodicamente [1] con la información que obtiene de analizar las redes sociales. También ofrece un servicio de alertas, aunque distinto a OpinionZoom en sus funcionalidades [2]. La pregunta que surge entonces es por los elementos distintivos de OpinionZoom, que le agregan valor a la aplicación.

Los aspectos más importantes y ventajosos de esta plataforma son los siguientes:

- Usuarios chilenos. La base de datos 'La Gorda', que aloja a los usuarios a analizar, está en periódica revisión para tener sólo usuarios chilenos, de tal forma que no se contaminen los datos con opiniones del exterior del país.
- Excelencia. El algoritmo de polaridad, y los que prontamente se sumarán a la aplicación, están en el horizonte del conocimiento y son fruto de la investigación del WIC, la cual es de primer nivel, de clase mundial.
- Flexibilidad y Escalabilidad. Es fácil para el equipo agregar nuevas funcionalidades, a propó-

[1] http://blog.brandmetric.com/
[2] http://www.brandmetric.com/alertas.html

sito de nuevos requerimientos que los usuarios de la plataforma puedan agregar. La construcción modular de la página permite ser flexibles en fuentes de datos, nuevos procesamientos y nuevas vistas.

- Para la Gestión. Entregar datos no ayuda para la toma de decisiones, sino que es la información y el conocimiento los que sirven para gestionar mejor la entrega de productos y servicios. Es por ello que OpinionZoom entrega la información ya procesada, de acuerdo a lo que le interesa al cliente, de las formas más cómodas para el usuario de la página Web, de forma de acortar la brecha lo más posible entre los datos y las decisiones.

La propuesta de valor se irá cumpliendo a medida que avance el proyecto, por lo que el prototipo no cumple con todas estas ventajas en su máxima plenitud.

6.2. Materiales de Construcción

Para construir la página Web se utilizó el patrón de arquitectura de software MVC (Modelo-Vista-Controlador), que permite separar el trabajo para la reutilización de código y un mejor mantenimiento. De esta forma hay que trabajar la capa de interacción con el usuario (vista), la capa lógica (controlador) y la capa de los datos (modelo).

Sobre la componente vista cabe destacar, como se menciona en el capítulo 1, que un test de usabilidad del sitio está fuera de los alcances de este trabajo. Esto se debe a que este prototipo, como se menciona en el marco teórico, pretende responder principalmente inquietudes relacionadas a las herramientas y fines funcionales de la aplicación. Sin embargo, se ocupan *templates* probados para páginas Web siguiendo las orientaciones del modelo de negocios. De este modo esta página es funcional y a la vez entendible para los usuarios.

El *framework* escogido para desarrollar la página Web bajo la arquitectura MVC se llama Phalcon y se diferencia de los demás por ser el más rápido [50], algo que es importante considerar sobre todo para el sistema de alertas. Su velocidad superior se debe a que está construido en lenguaje C/C++. Esta y otras soluciones de bajo nivel le dan a Phalcon la rapidez necesaria para esta aplicación. Además tiene comandos de alto nivel y compactaciones de código que le hacen más sencilla la tarea al programador. Por último, este *framework* se programa en PHP, cuyas bondades son descritas en el capítulo 2.

El dominio ya está comprado en NIC Chile (NIC.cl) y se llama OpinionZoom.cl. La página se aloja en un servidor de Amazon®, especificado en la sección de Diseño Físico del capítulo 3.

6.3. Requerimientos de Usuarios

Se describen en esta sección los requerimientos de los distintos usuarios de la página `www.opinionzoom.cl`. Se describen primero las tareas por usuario y luego se representan los casos de uso por roles. La descripción de casos de uso por módulos se encuentran en el apéndice B, y la nomenclatura de los códigos allí establecidos son ocupados en el resto de este capítulo.

6.3.1. Tipos de Usuarios

Hay tres tipos de usuarios del sitio Web: visitante, cliente y administrador. A continuación se describe a cada usuario a partir de sus posibilidades dentro de la página.

- Usuario Visitante: Es quien por primera vez entra al sitio sin haber hecho un contrato e incluso sin tener conocimiento del proyecto. Este usuario puede:
  - Ver la página principal
  - Conocer al equipo que trabaja en el proyecto
  - Rellenar un formulario de contacto para enviar un mensaje
  - Revisar artículos públicos que emite OpinionZoom
  - Enterarse de los servicios que ofrece OpinionZoom
- Usuario Cliente: Es quien ya tiene un contrato con OpinionZoom y por lo tanto cuenta con a lo menos un servicio. Sus posibilidades dependerán de los servicios contratados, por lo tanto se lista primeramente los requisitos generales y luego separadamente las funcionalidades para el servicio inteligencia de clientes y el servicio alertas, pudiendo un cliente contar con sólo uno de ellos o con ambos. Este usuario puede:
  - Iniciar sesión con su cuenta de usuario
  - Modificar datos personales
  - Cerrar su sesión una vez iniciada
  - Rellenar un formulario de contacto para enviar un mensaje
  - Conocer al equipo que trabaja en el proyecto

  Para el servicio de inteligencia de clientes se despliegan tres opciones:
  - Explorar información general de Twitter como a qué hora del día está *twitteando* la gente con mayor frecuencia o cuáles son los Trendings Topics que detecta OpinionZoom a partir de los usuarios y *tweets* que registra.
  - Revisar la información por keyword, como la polaridad de los tweets en el transcurso de las horas y en el de los días, los usuarios más influyentes que hablan del tema o los intereses complementarios que tiene esa keyword.
  - Revisar la información agregada por keyword contratada, pudiendo comparar indicadores importantes como frecuencia, polaridad e impacto.

  Para el servicio de alertas se despliegan cuatro opciones:
  - Ver las alertas configuradas.
  - Revisar las alertas puntuales que se hayan generado.
  - Revisar las alertas generalizadas que se hayan generado.
- Usuario Administrador: Este rol puede ser ocupado por una o dos personas pertenecientes al proyecto OpinionZoom. Este usuario es el que administra las cuentas de los usuarios y las organizaciones clientes de OpinionZoom, pudiendo una misma organización tener varios Usuarios Clientes con los mismos privilegios. Este usuario puede:
  - Revisar las estadísticas del sistema.
  - Ver cada organización cliente: su ID, su fecha de creación, su fecha de vencimiento y la cantidad de keywords que tiene asociadas.

- Crear organizaciones clientes y modificar las existentes.
- Revisar las cuentas de usuarios y modificarlas.

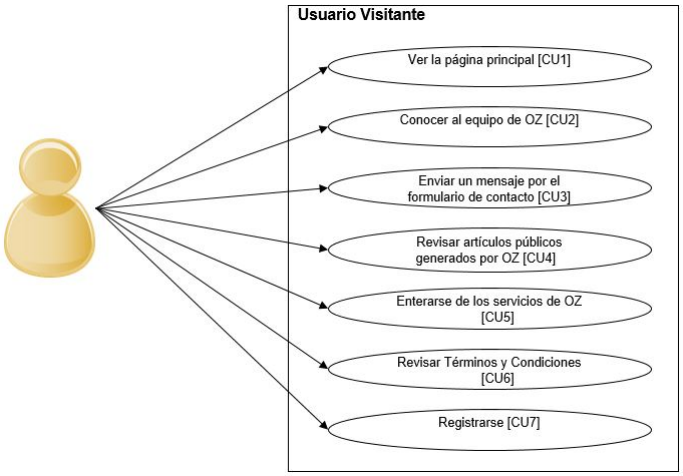
6.3.2. Descripción del Modelo según Roles

A continuación se presentan funcionalidades generales del sistema y sus actores. Para esto se exhiben los casos de uso separados según tipos de usuarios.

Usuario Visitante

Se pueden ver en la figura 6.1 los casos de uso del usuario visitante. Como se aprecia, son funciones básicas para cualquier persona que visite OpinionZoom.cl.

Figura 6.1: Caso de uso para el Usuario Visitante



Fuente: Elaboración Propia

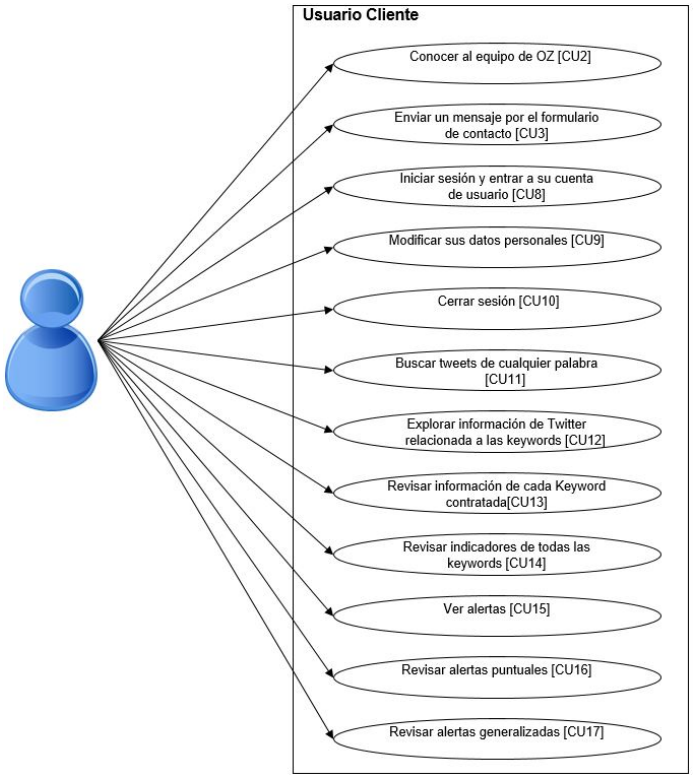
Usuario Cliente

El usuario cliente tiene doce casos de uso, mostrados en la figura 6.2, principalmente relacionados a los servicios ofrecidos por la plataforma.

Usuario Administrador

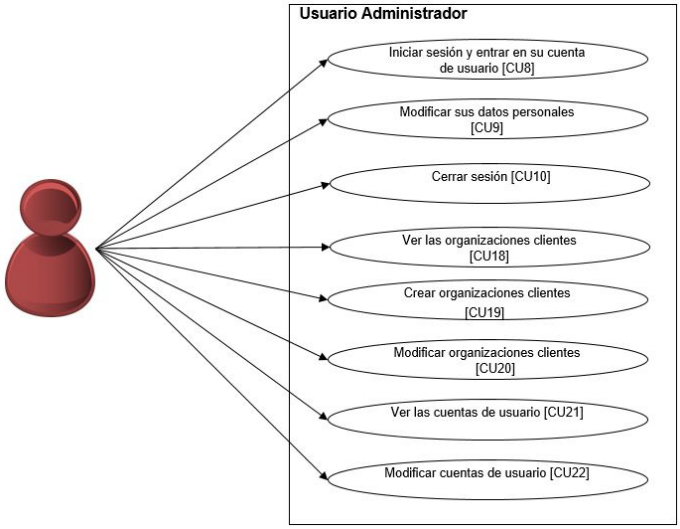
En la figura 6.3 se ve que el Administrador comparte algunos casos de uso con el usuario cliente, pero sus principales tareas serán naturalmente relacionadas a la administración del sistema.

Figura 6.2: Caso de uso para el Usuario Cliente



Fuente: Elaboración Propia

Figura 6.3: Caso de uso para el Usuario Administrador



Fuente: Elaboración Propia

6.4. Requisitos del sistema

Esta sección describe los requisitos de los Usuarios y los requisitos de Software con los que debe cumplir el sistema. Como se explica en el capítulo 1, los requisitos no fueron detallados desde un comienzo, implicando que los aquí presentados sean una construcción constante y aún no acabada. Sin embargo se considera este resultado estático como completo y cierto para el trabajo de título entregado.

6.4.1. Requisitos de Usuario

Los requisitos de los usuarios son aquellas condiciones especificadas por los usuarios, las cuales al ser cumplidas satisfacen sus necesidades. En este caso se plantean requisitos de datos, requisitos de calidad y requisitos de restricción.

Requisitos de datos

Esta subsección es la más extensa pues guarda relación con aquellos requisitos que implican principalmente transmisión de datos, desde el iniciar sesión hasta ver estáticamente al equipo que conforma el proyecto.

Fuente: Elaboración Propia

Identificador	RU01
Nombre	Control de Acceso
Descripción	Se controla el acceso al sistema mediante un nombre de usuario (o dirección de correo electónico) y una contraseña
Caso de Uso	CU8
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Cliente; Administrador

Tabla 6.1: RU01: Control de Acceso

Fuente: Elaboración Propia

Identificador	RU02
Nombre	Crear Cuenta
Descripción	El usuario debe ingresar su nombre completo, nombre de usuario, email, contraseña y confirmación de contraseña para crear una cuenta.
Caso de Uso	CU7, CM7, CM8
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Cliente; Administrador

Tabla 6.2: RU02: Crear Cuenta

Fuente: Elaboración Propia

Identificador	RU03
Nombre	Ingreso a 'Quienes Somos'
Descripción	El usuario podrá consultar en esta sección información del grupo y de sus miembros en el encabezado del Home.
Caso de Uso	CU2, CM3
Prioridad	Media
Estabilidad	Intransable
Usuarios	Visitante; Cliente

Tabla 6.3: RU03: Ingreso a 'Quienes Somos'

Fuente: Elaboración Propia

Identificador	RU04
Nombre	Revisar artículos
Descripción	Los usuarios podrán ver noticias y artículos sobre OpinionZoom y sus descubrimientos en la página de inicio o Home
Caso de Uso	CU4, CM4
Prioridad	Baja
Estabilidad	Deseable
Usuarios	Visitante

Tabla 6.4: RU04: Revisar artículos

Fuente: Elaboración Propia

Identificador	RU05
Nombre	Ver funcionalidades de la herramienta
Descripción	Los usuarios deben poder ver las funcionalidades de OpinionZoom en la página de inicio
Caso de Uso	CU5, CM2
Prioridad	Baja
Estabilidad	Transable
Usuarios	Visitante

Tabla 6.5: RU05: Ver funcionalidades

Fuente: Elaboración Propia

Identificador	RU06
Nombre	Términos y Condiciones
Descripción	Los usuarios siempre podrán acceder a conocer los términos y condiciones, además de las políticas de privacidad
Caso de Uso	CU6, CM8
Prioridad	Media
Estabilidad	Intransable
Usuarios	Visitante

Tabla 6.6: RU06: Revisión de Resultados

Fuente: Elaboración Propia

Identificador	RU07
Nombre	Revisión de Resultados en página de OpinionZoom
Descripción	Los usuarios visualizan gráficos e indicadores para los servicios contratados
Caso de Uso	CU11, CU12, CU13, CU14, CU16, CU17, CM11, CM12, CM13, CM15, CM16
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Cliente

Tabla 6.7: RU07: Revisión de Resultados

Fuente: Elaboración Propia

Identificador	RU08
Nombre	Configuración de Cuenta
Descripción	El usuario puede modificar información de su cuenta.
Caso de Uso	CU9, CM9
Prioridad	Media
Estabilidad	Intransable
Usuarios	Cliente; Administrador

Tabla 6.8: RU08: Configuración de Cuenta

Fuente: Elaboración Propia

Identificador	RU09
Nombre	Cerrar sesión
Descripción	El usuario puede cerrar su sesión.
Caso de Uso	CU9
Prioridad	Media
Estabilidad	Intransable
Usuarios	Cliente; Administrador

Tabla 6.9: RU09: Cerrar Sesión

Fuente: Elaboración Propia

Identificador	RU10
Nombre	Ver y dar permisos a usuarios clientes
Descripción	El Administrador puede consultar los perfiles de los usuarios clientes y puede modificar sus permisos.
Caso de Uso	CU21, CU22, CM20, CM21
Prioridad	Deseable
Estabilidad	Transable
Usuarios	Administrador

Tabla 6.10: RU10: Ver Usuarios

Fuente: Elaboración Propia

Identificador	RU11
Nombre	Ver organizaciones clientes
Descripción	El Administrador puede consultar los perfiles de los usuarios clientes y puede modificar sus permisos.
Caso de Uso	CU18, CU19, CM17, CM19
Prioridad	Deseable
Estabilidad	Transable
Usuarios	Administrador

Tabla 6.11: RU11: Ver organizaciones clientes

Fuente: Elaboración Propia

Identificador	RU12
Nombre	Acceso al sistema
Descripción	Los usuarios deben poder acceder al sistema desde cualquier punto de conexión a Internet.
Caso de Uso	TODOS
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Gratuito, Cliente, Administrador

Tabla 6.12: RU12: Ver Usuarios

Requisitos de Restricción

Los requisitos de restricción versan sobre la disponibilidad del sistema ya sea desde el servidor o desde el computador del cliente. De aquí en adelante las tablas que prescinden de las filas de Casos de Uso y de Listado de Usuarios indican requisitos transversales a todos los usuarios.

Fuente: Elaboración Propia

Identificador	RU13
Nombre	Disponibilidad de datos
Descripción	Los datos deben estar siempre disponibles
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.13: RU13: Disponibilidad de datos

Fuente: Elaboración Propia

Identificador	RU14
Nombre	Acceso via navegadores estándar.
Descripción	Se ingresa al sistema a través de navegadores estándar como Ïnternet Explorer", "Mozilla", “Safari” y “Chrome”.
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.14: RU14: Acceso via navegadores estándar

Requisitos de Calidad

La calidad debe ser aceptable en una variedad de ámbitos, desde la ortografía hasta la seguridad del servidor.

Fuente: Elaboración Propia

Identificador	RU15
Nombre	Sistema Operativo
Descripción	El sistema deberá funcionar bajo un ambiente superior a Windows XP.
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.15: RU15: Sistema Operativo

Fuente: Elaboración Propia

Identificador	RU16
Nombre	Ortografía
Descripción	No deben haber faltas ortográficas en las interfaces del sistema.
Prioridad	Deseable
Estabilidad	Transable

Tabla 6.16: RU16: Ortografía

Fuente: Elaboración Propia

Identificador	RU17
Nombre	Sistema extensible
Descripción	El sistema a desarrollar debe mantenerse extensible a futuro.
Prioridad	Deseable
Estabilidad	Transable

Tabla 6.17: RU17: Sistema extensible

Fuente: Elaboración Propia

Identificador	RU18
Nombre	Servidor
Descripción	El sistema debe funcionar en un servidor siempre disponible.
Prioridad	Deseable
Estabilidad	Transable

Tabla 6.18: RU18: Servidor

6.4.2. Requisitos de Software

Para los requisitos de software se utilizará el mismo formato de especificación que para los requisitos de usuario.

Requisitos Funcionales

Los requisitos funcionales definen una función del sistema de software o de alguna de sus componentes. Las funciones del sistema se describen como un conjunto de entradas, comportamientos y salidas.

Fuente: Elaboración Propia

Identificador	RS01
Nombre	Acceso a cuenta
Descripción	El sistema solicita el nombre de usuario, o email, y la contraseña al usuario para acceder a su cuenta. Como salida se tiene el acceso a los servicios contratados y a información de su cuenta.
Caso de Uso	CU8, CM6
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Cliente, Administrador

Tabla 6.19: RS01: Acceso a cuenta

Fuente: Elaboración Propia

Identificador	RS02
Nombre	Creación de cuenta
Descripción	El sistema generará un formulario para que el usuario ingrese su nombre completo, email y contraseña. El sistema comprobará si es la primera vez que el usuario ingresa, verificando si existe registro del email. Si no es así, procederá a crear la cuenta.
Caso de Uso	CU7, CM7
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Visitante

Tabla 6.20: RS02: Creación de cuenta

Fuente: Elaboración Propia

Identificador	RS03
Nombre	Cuenta creada exitosamente
Descripción	El sistema debe generar un aviso indicando que la cuenta se creó exitosamente.
Caso de Uso	CU7, CM7
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Visitante

Tabla 6.21: RS03: Cuenta creada exitosamente

Fuente: Elaboración Propia

Identificador	RS04
Nombre	Aviso de información insuficiente
Descripción	Si el usuario no completa todos los campos obligatorios para el registro en el sistema, este debe generar un aviso de “información insuficiente” para completar registro.
Caso de Uso	CU7, CM7
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Visitante

Tabla 6.22: RS04: Aviso de información insuficiente

Fuente: Elaboración Propia

Identificador	RS05
Nombre	Ingreso a "Quienes Somos"
Descripción	Al ingresar a la página de inicio, debe existir una pestaña que permita dar a conocer información sobre el equipo OpinionZoom y los miembros de éste.
Caso de Uso	CU2, CM6
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Visitante, Cliente

Tabla 6.23: RS05: Ingreso a "Quienes Somos"

Fuente: Elaboración Propia

Identificador	RS06
Nombre	Mostrar artículos
Descripción	Al ingresar a la página de inicio, debe existir una sección que permita ver artículos publicados por OpinionZoom.
Caso de Uso	CU4, CM4
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Visitante

Tabla 6.24: RS06: Mostrar artículos

Fuente: Elaboración Propia

Identificador	RS07
Nombre	Funcionalidades de la herramienta
Descripción	Al ingresar a la página de inicio, debe existir una sección que permita ver funcionalidades de OpinionZoom.
Caso de Uso	CU5, CM2
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Visitante

Tabla 6.25: RS07: Funcionalidades de la herramienta

Fuente: Elaboración Propia

Identificador	RS08
Nombre	Términos y Condiciones
Descripción	Debe existir un *link* de términos y condiciones que redirija a los usuarios a la información respectiva.
Caso de Uso	CU6, CM8
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Visitante

Tabla 6.26: RS08: Términos y Condiciones

Fuente: Elaboración Propia

Identificador	RS09
Nombre	Ver resultados de análisis
Descripción	Al ingresar a sus cuentas los usuarios pueden ver los gráficos e indicadores correspondientes a los servicios contratados
Caso de Uso	CU11, CU12, CU13, CU14, CU16, CU17, CM11, CM12, CM13, CM15, CM16
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Visitante

Tabla 6.27: RS09: Ver resultados de análisis

Fuente: Elaboración Propia

Identificador	RS10
Nombre	Configuración de cuenta
Descripción	El sistema debe permitir a los usuarios modificar sus datos personales. Esta posibilidad la debe disponer en un *link* del menú despegable del usuario.
Caso de Uso	CU9, CM9
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Cliente, Administrador

Tabla 6.28: RS10: Configuración de cuenta

Fuente: Elaboración Propia

Identificador	RS11
Nombre	Cerrar cuenta
Descripción	En el menú desplegable "Mi Cuenta", el sistema debe ofrecer al usuario una opción para cerrar su cuenta.
Caso de Uso	CU10
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Cliente, Administrador

Tabla 6.29: RS11: Cerrar cuenta

Fuente: Elaboración Propia

Identificador	RS12
Nombre	Ver y gestionar usuarios
Descripción	El sistema debe permitir al administrador consultar los perfiles de los usuarios y modificar sus permisos.
Caso de Uso	CU21, CU22, CM20, CM21
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Administrador

Tabla 6.30: RS12: Ver y gestionar usuarios

Fuente: Elaboración Propia

Identificador	RS13
Nombre	Ver y modificar clientes
Descripción	El sistema debe permitir al administrador consultar la información de las organizaciones clientes y modificar algunos de sus atributos, y junto con ello determinar si está activo o no.
Caso de Uso	CU18, CU20, CM18, CM19
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Administrador

Tabla 6.31: RS13: Ver y gestionar clientes

Fuente: Elaboración Propia

Identificador	RS14
Nombre	Crear clientes
Descripción	El sistema debe permitir al administrador crear clientes, insertando su nombre y su fecha de vencimiento
Caso de Uso	CU19
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Administrador

Tabla 6.32: RS14: Crear clientes

Fuente: Elaboración Propia

Identificador	RS15
Nombre	Estadísticas generales
Descripción	El sistema debe permitir que el administrador consulte la situación global del sistema: cantidad de clientes, cantidad de clientes activos, cantidad de usuarios, cantidad de usuarios activos y cantidad de usuarios por servicio.
Caso de Uso	CU23, CM22
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Administrador

Tabla 6.33: RS15: Estadísticas generales

Fuente: Elaboración Propia

Identificador	RS16
Nombre	Acceder al sistema
Descripción	El sistema debe permitir que desde la página de identificación y autentificación los usuarios accedan a él, navegando desde cualquier punto con conexión a Internet.
Caso de Uso	CU08, CM05
Prioridad	Crítica
Estabilidad	Intransable
Usuarios	Cliente, Administrador

Tabla 6.34: RS16: Acceder al sistema

Fuente: Elaboración Propia

Identificador	RS17
Nombre	Datos
Descripción	El sistema debe proveer un DBMS que mantenga siempre disponible los datos. En este caso MySQL en un primer momento y luego MariaDB.
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.35: RS17: Datos

Fuente: Elaboración Propia

Identificador	RS18
Nombre	Navegadores estándar
Descripción	El sistema debe funcionar en Internet Explorer 8 o superior, en Mozilla 4 o superior y en Chrome
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.36: RS18: Navegadores estándar

Fuente: Elaboración Propia

Identificador	RS19
Nombre	Sistema Operativo
Descripción	El sistema deberá funcionar en un ambiente con Windows 7 o superior
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.37: RS19: Navegadores estándar

Fuente: Elaboración Propia

Identificador	RS20
Nombre	Ortografía
Descripción	El sistema no debe presentar faltas ortográficas en ninguna de sus interfaces.
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.38: RS20: Ortografía

Requisitos de Mantenibilidad

El sistema debe ser mantenido en el futuro por el equipo de desarrollo. Hay que considerar que el equipo puede ir cambiando en el tiempo.

Fuente: Elaboración Propia

Identificador	RS21
Nombre	Extensible
Descripción	El sistema tiene que ser extensible por personas que en el futuro se unan al grupo de desarrollo. Para ello, el equipo debe hacer los documentos correspondientes a todas las etapas del desarrollo y entregarlas al jefe del equipo de desarrollo, seguir un diseño modular y hacer lo posible para disminuir la curva de aprendizaje de las herramientas utilizadas.
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.39: RS21: Extensible

Requisitos de Interfaz

Describen el formato con el que la aplicación se comunica con su entorno.

Fuente: Elaboración Propia

Identificador	RS22
Nombre	Interfaz de la página de identificación
Descripción	El sistema debe poseer una página de autentificación e identificación.
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.40: RS22: Extensible

Requisitos de Recursos

Especifican los recursos que el sistema requiere tales como uso de memoria, capacidad de tráfico o líneas de atención simultánea.

Fuente: Elaboración Propia

Identificador	RS23
Nombre	Servidor
Descripción	El sistema debe ser alojado en un servidor seguro, un procesador de un núcleo Intel Xeon o superior y al menos 4GB de RAM
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.41: RS23: Servidor

Requisitos de Documentación

Orienta sobre como y cuando debe ser documentado el sistema

Fuente: Elaboración Propia

Identificador	RS24
Nombre	Herramientas de diseño
Descripción	El sistema debe implementarse en PHP 5 o superior, MySQL 5 o superior y Apache.
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.42: RS24: Herramientas de diseño

Requisitos de Rendimiento

Los requisitos de rendimiento señalan el máximo tiempo que puede demorarse una petición a la página.

Fuente: Elaboración Propia

Identificador	RS25
Nombre	Respuesta
Descripción	Ninguna operación realizada por un usuario del sistema debe tardar más de 25 segundos.
Prioridad	Crítica
Estabilidad	Intransable

Tabla 6.43: RS25: Respuesta

Capítulo 7

Resultados

Los resultados de este trabajo se presentan en su mayoría en los capítulos precedentes. En este se hace un recuento de los resultados obtenidos, que responden a los resultados esperados, y se evalúa el rendimiento del sistema.

7.1. Sobre los Resultados Esperados

En el capítulo introductorio se definen los resultados esperados. a saber:

1. Selección de algoritmos a usar
2. Ubicación geográfica y descripción lógica del sistema y sus componentes
3. Data Warehouse
4. Página Web funcional
5. Evaluación de resultados de análisis para un caso ficticio
6. Evaluación de rendimiento del sistema

A más bajo nivel, el código alojado en `BitBucket.com` que acompaña varios de los resultados mencionados, es otro resultado de este trabajo. El WIC es dueño y administrador de todos estos resultados. A continuación se muestra cómo se obtiene cada resultado esperado, exceptuando los últimos, Evaluación de resultados de análisis para un caso ficticio y Evaluación de rendimiento del sistema, que ameritan cada uno una sección especial dentro de este capítulo.

7.1.1. R1: Selección de Agoritmos a Usar

En el capítulo 3 se muestran los algoritmos a usar, 2 de los cuales son desconocidos en cuanto a su funcionamiento. Los algoritmos que ocupa Klout para los intereses y la influencia son cajas negras en forma de API. Se ocupan estos algoritmos temporalmente a la espera de algoritmos propios del WIC que reemplacen en seguridad, estabilidad, confiabilidad y precisión a los ocupados actualmente. Por mientras es necesario ocupar los algoritmos de Klout para probar la funcionalidad del sistema. El algoritmo más importante que se seleccionó para su uso es el de polaridad, desarrollado por el centro y encapsulado en forma de API. Se sabe seguro, estable, confiable y preciso puesto que se conocen todos sus procesos internos en el WIC. El último algoritmo a utilizar es el que detecta el sexo de una persona por su nombre. Está en una primera etapa dentro del mismo código en Java a cargo del ETL pero sirve para probar la funcionalidad del sistema. Se seleccionaría un quinto algoritmo para integrar al sistema si es que estuviera listo: el algoritmo de ironía desarrollado por el WIC no está listo para su uso, pero debe estarlo antes de que venza el plazo del financiamiento con que INNOVA CORFO dotó al proyecto OpinionZoom.

7.1.2. R2: Ubicación Geográfica y Descripción Lógica del Sistema

También en el capítulo 3 se exhibe el sistema y sus componentes con alto nivel de granularidad. Se describen las entradas, el proceso y las salidas del sistema. Todo esto constituye a la vez la arquitectura del Data Warehouse, la cual es esquematizada en 5 capas, desde la capa de fuente de datos hasta la capa de presentación. Los subprocesos internos se abordan con mayor detalle en el capítulo 4 y finalmente la capa de presentación se abarca en los capítulos 5 y 6.

La ubicación geográfica está dada por la ubicación de los servidores dentro de internet. Ambos servidores están en Estados Unidos de América, específicamente en Oregon[1], donde AWS tiene una zona de disponibilidad de servidores desde el 2011.

7.1.3. R3: Data Warehouse

El Data Warehouse es transversal a todo el trabajo desarrollado, por lo que no es fácil circunscribirlo a sólo un capítulo. Su arquitectura gruesa se encuentra en el capítulo 3, los requisitos, el modelo de la base de datos y el ETL en el capítulo 4. Finalmente la capa de presentación es la misma página Web explicada en el capítulo de servicios y de página Web. Esta capa de presentación verifica en cierta forma la existencia de una base de datos real, modelada y con datos en su interior, que es la parte central del Data Warehouse.

[1] `https://aws.amazon.com/es/about-aws/global-infrastructure/`, 26 de octubre de 2015

7.1.4. R4: Página Web Funcional

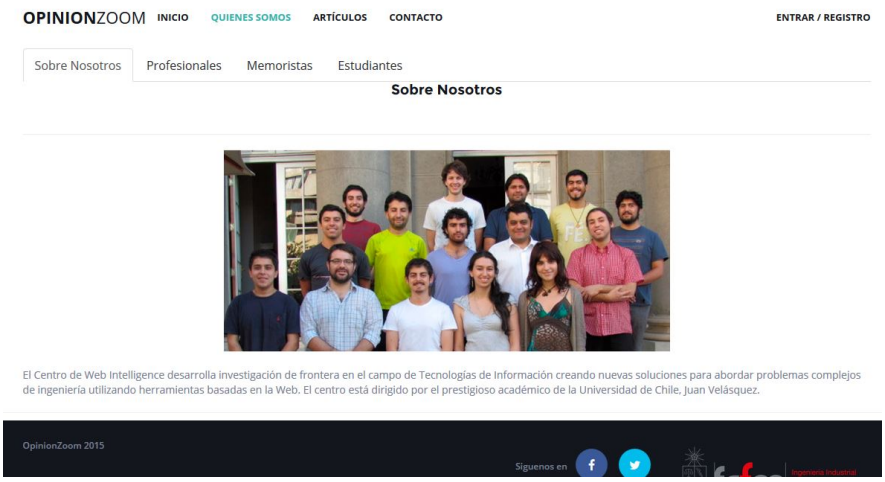
La página Web es el resultado más visible, a tal punto que cualquier persona del mundo con conexión a internet podría acceder a ella en este momento ingresando a `www.opinionzoom.cl`. A este usuario lo recibiría la página de inicio, que se muestra parcialmente en la figura 7.1.

Figura 7.1: Página de inicio de OpinionZoom.cl

Fuente: www.opinionzoom.cl

Desde la página de inicio el usuario visitante podría ver el equipo que conforma OpinionZoom (figura 7.2), los artículos publicados (figura 7.3) y podría acceder también al formulario de contacto (figura 7.4).

Figura 7.2: Equipo de OpinionZoom



Fuente: www.opinionzoom.cl

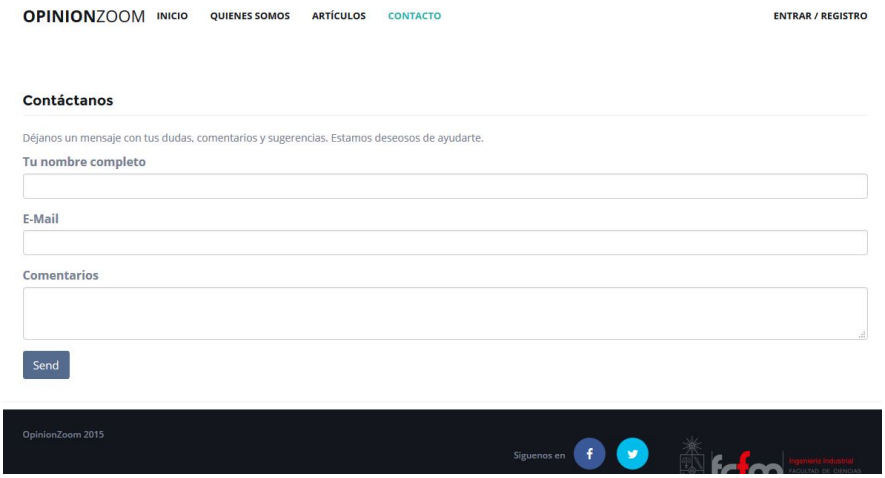
Si el usuario inicia sesión como cliente y tiene contratados los servicios de inteligencia y alertas podría ver vistas como las figura 7.5 y la figura 7.6 respectivamente.

Si se trata de un usuario administrador del sistema, podría entre otras cosas, ver las estadísticas del sistema, como se aprecia en la figura 7.7.

Figura 7.3: Artículos publicados por OpinionZoom

Fuente: www.opinionzoom.cl

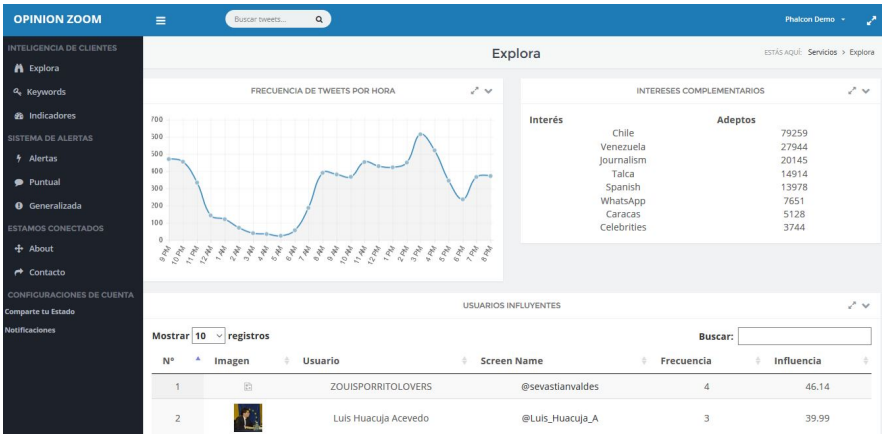
Figura 7.4: Formulario de contacto



Fuente: www.opinionzoom.cl

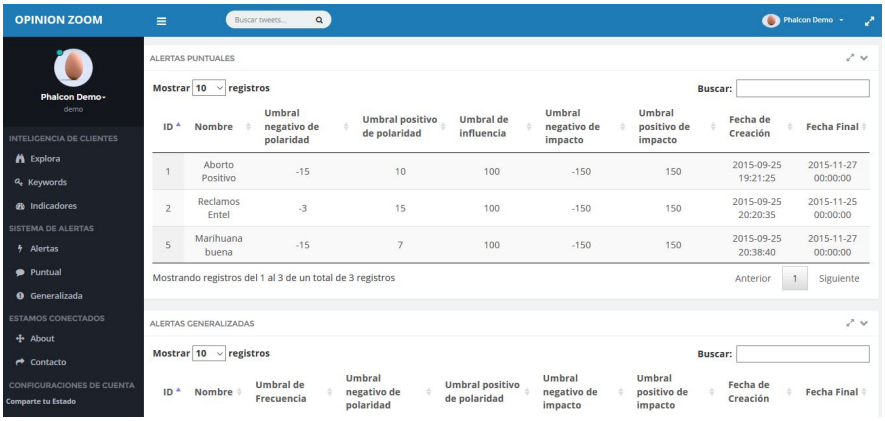
En el apéndice C se muestran más imágenes obtenidas de la página Web. Este resultado esperado está guiado por los requisitos de usuario especificados en el capítulo anterior, por lo que desde allí se pueden conocer todas las funcionalidades de la página Web.

Figura 7.5: Módulo de Inteligencia de Clientes para un usuario ficticio



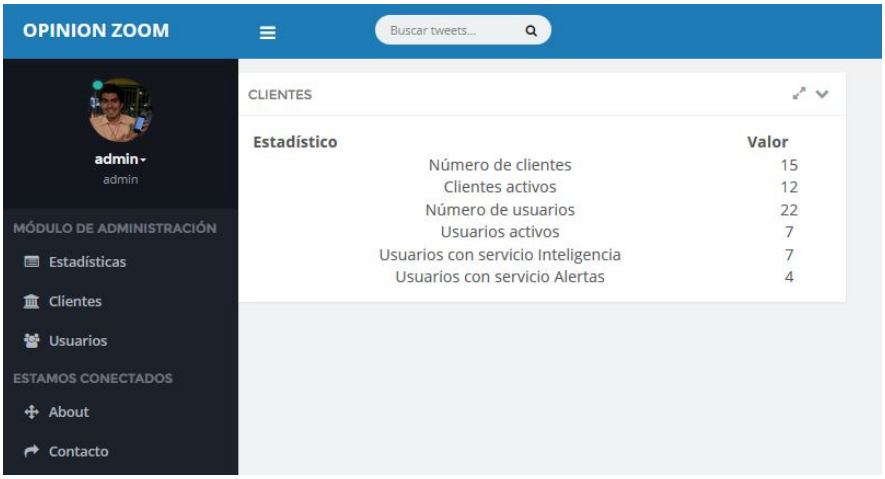
Fuente: www.opinionzoom.cl

Figura 7.6: Módulo de Alertas para un usuario ficticio



Fuente: www.opinionzoom.cl

Figura 7.7: Módulo de Administración



Fuente: www.opinionzoom.cl

7.2. Evaluación de Resultados de Análisis para un Caso Ficticio

Para evaluar la calidad de los resultados del análisis que es capaz de hacer el prototipo se considerará un cliente ficticio llamado "Phalcon". Este usuario tiene contratadas 22 keywords y cuenta con el servicio inteligencia de clientes y el servicio alertas. Las keywords contratadas se presentan en la tabla 7.1.

Fuente: Elaboración Propia

@NEXTEL_CHILE	Cruch	#FuerzaChile
@nextelcl	marihuana	Bonvallet
@NextelChile_	@entel	transantiago
@VirginMobile_cl	@womchile	Facebook
@AloVMCL	aborto	#Facebook
@MovilFalabella	delincuencia	#UnionCivil
@movilfalabellac	terremoto	
@RicardoLagos	temblor	

Tabla 7.1: Keywords contratadas por el cliente ficticio

El servicio de inteligencia de clientes tendrá tres vistas: Explora, Keywords e Indicadores. Los resultados de alertas, por otra parte, se dividen en Puntuales y en Generalizadas. Se analizan los gráficos y tablas correspondientes a cada vista.

7.2.1. Resultados de Explora

En la figura 7.8 se grafica la evolución de la frecuencia de emisión de *tweets* en 24 horas. Esto se calcula en promedio para las 22 keywords vigentes del usuario. Se puede ver que el valle global es largo, de aproximadamente 7 horas, desde las 00:00 hasta las 07:00 horas. Esto es esperable ya que la mayoría de los usuarios chilenos duermen durante esas horas. Por otra parte el horario peack es entre las 14:00 y 16:00 horas.

En la figura 7.9 se ven los intereses de los clientes que opinan de las keywords contratadas. Los únicos intereses que parecen tener algún sentido son Chile y Journalism. Quizás también podrían tener alguna relación con los temas estudiados los intereses WhatsApp, Spanish o Talca, pero luego se ve que estos 5 intereses frecuentes son constantes para casi todas las keywords y todos los clientes. La conclusión lógica entonces es la falta de precisión de Klout en la identificación de intereses y la necesidad de mejorar estos resultados.

En la figura 7.10 se ven los usuarios más influyentes que hablan de alguna de las keywords inscritas. Los usuarios más influyentes mostrados son tanto organizaciones como personas naturales. Los primeros hablan sobre equidad de género, probablemente por las keywords de aborto y #UnionCivil. Los siguientes usuarios son de Colombia y Venezuela respectivamente y recién el quinto vuelve a ser chileno. Se advierte aquí la necesidad de eliminar a usuarios extranjeros del sistema que ensucian los datos considerando que el fin de OpinionZoom es generar conocimiento sobre los usuarios chilenos.

Figura 7.8: Frecuencia de tweets por hora

Fuente: www.opinionzoom.cl

Figura 7.9: Intereses complementarios a todas las keywords

INTERESES COMPLEMENTARIOS

Interés	Adeptos
Chile	79259
Venezuela	27944
Journalism	20145
Talca	14914
Spanish	13978
WhatsApp	7651
Caracas	5128
Celebrities	3744

Fuente: www.opinionzoom.cl

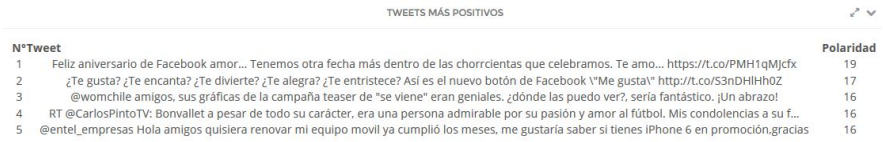
Figura 7.10: Usuarios influyentes

Usuario	Screen Name	Frecuencia	Influencia
Especialista.EVEFem	@generoenaccion	15	68.37
Carlos Zárate V.	@CarlosZarateV	8	64.47
DIARIO DEPORTES	@DIARIODEPORTES	3	63.83
marujatarre	@marujatarre	4	61.49
SOCIOP?DRO	@Sociopedro	23	52.99
Lady	@queordi	6	52.52
El Boyaldía #Iquique	@ElBoyaldia	27	52.21
#Cruzarock \m/	@Carlos_crxzado3	2	49.08
PRENSA ERCHILOE	@prensa_erchiloe	14	47.08
ZOUISPORRITOLOVERS	@sevastianvaldes	4	46.14

Fuente: www.opinionzoom.cl

Los *tweets* más positivos y los *tweets* más negativos se muestran en las figuras 7.11 y 7.12 respectivos. Se puede apreciar la buena calidad de PAPI. Los *tweets* con polaridad negativa son claramente negativos pues ocupan palabras como miseria, ignorancia y corrupción. Los *tweets* positivos por su parte hablan de amor, gusto, alegría y encanto. Se concluye que existe gran coherencia en los resultados.

Figura 7.11: Tweets positivos

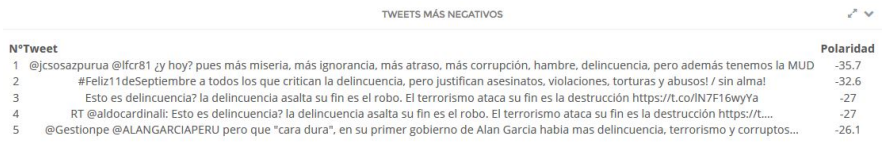


TWEETS MÁS POSITIVOS

N°	Tweet	Polaridad
1	Feliz aniversario de Facebook amor... Tenemos otra fecha más dentro de las chorrcientas que celebramos. Te amo... https://t.co/PMH1qMJcfx	19
2	¿Te gusta? ¿Te encanta? ¿Te divierte? ¿Te alegra? ¿Te entristece? Así es el nuevo botón de Facebook \"Me gusta\" http://t.co/S3nDHlHh0Z	17
3	@womchile amigos, sus gráficas de la campaña teaser de "se viene" eran geniales. ¿dónde las puedo ver?, sería fantástico. ¡Un abrazo!	16
4	RT @CarlosPintoTV: Bonvallet a pesar de todo su carácter, era una persona admirable por su pasión y amor al fútbol. Mis condolencias a su f...	16
5	@entel_empresas Hola amigos quisiera renovar mi equipo movil ya cumplió los meses, me gustaría saber si tienes iPhone 6 en promoción,gracias	16

Fuente: www.opinionzoom.cl

Figura 7.12: Tweets negativos



TWEETS MÁS NEGATIVOS

N°	Tweet	Polaridad
1	@jcsosazpurua @lfcr81 ¿y hoy? pues más miseria, más ignorancia, más atraso, más corrupción, hambre, delincuencia, pero además tenemos la MUD	-35.7
2	#Feliz11deSeptiembre a todos los que critican la delincuencia, pero justifican asesinatos, violaciones, torturas y abusos! / sin alma!	-32.6
3	Esto es delincuencia? la delincuencia asalta su fin es el robo. El terrorismo ataca su fin es la destrucción https://t.co/lN7F16wyYa	-27
4	RT @aldocardinali: Esto es delincuencia? la delincuencia asalta su fin es el robo. El terrorismo ataca su fin es la destrucción https://t....	-27
5	@Gestionpe @ALANGARCIAPERU pero que "cara dura", en su primer gobierno de Alan Garcia habia mas delincuencia, terrorismo y corruptos...	-26.1

Fuente: www.opinionzoom.cl

7.2.2. Resultados de Keywords

Para evaluar el rendimiento desplegado en esta vista se analizan los resultados obtenidos para la keyword "transantiago", por ser un tema popular que compete exclusivamente a usuarios chilenos. Se contrastarán estos resultados con la intuición y la realidad nacional en los días de la medición. El criterio de evaluación será dado por este contraste, siendo mejores los resultados entre mayor sea la coherencia con lo esperado.

En el gráfico de la figura 7.13 se puede ver, como en la vista Explora, la frecuencia de *tweeteo* por hora. Nuevamente resulta natural que el valle se encuentre en la noche, pero llama la atención que la actividad empieza una hora antes. Esto se debe que una gran cantidad de los usuarios del sistema de transportes capitalino se despierta temprano. Esto se comprueba con el peack de frecuencia que está marcado entre las 7 y las 9 de la mañana, coincidente con la hora peack del sistema transantiago.

En el segundo gráfico, mostrado en la figura 7.14, se ve la polaridad promedio durante el día. Ésta es neutral en general salvo a las 2 de la mañana y a las 5 de la mañana. Esto se puede atribuir a la baja frecuencia de tweeteo en esas horas. Los pocos usuarios que comentan a las 2 de la mañana pueden estar teniendo un muy buen servicio puesto que las calles están vacías y son pocos los paraderos en que los buses se detienen. Por otro lado pocos usuarios opinan negativamente del servicio a una hora en que la frecuencia puede ser baja, pero a diferencia de los usuarios de 3 horas antes, en este caso están presionados por la puntualidad. Estos son supuestos ulteriores que escapan de los gráficos y que necesitan de otras experiencias y estudios para generar mayor conocimiento.

Figura 7.13: Frecuencia por hora para keyword "transantiago"

Fuente: www.opinionzoom.cl

Figura 7.14: Polaridad por hora para keyword "transantiago"

Fuente: www.opinionzoom.cl

Se aprecia en la figura 7.15 el tercer gráfico, donde se ve la frecuencia en el transcurso de los días. Estos datos fueron tomados para la última semana de septiembre de 2015. Sólo el lunes 28 de septiembre presenta un alza significativa, pero se requiere más contexto para obtener información relevante de éste gráfico. El alza se explica por la huelga de los trabajadores de algunos servicios del transantiago que sucedió ese día y los aledaños. Se destaca así el buen rendimiento del sistema para decir algo de la realidad en tiempo real y se descubre un poco más la potencia de la herramienta.

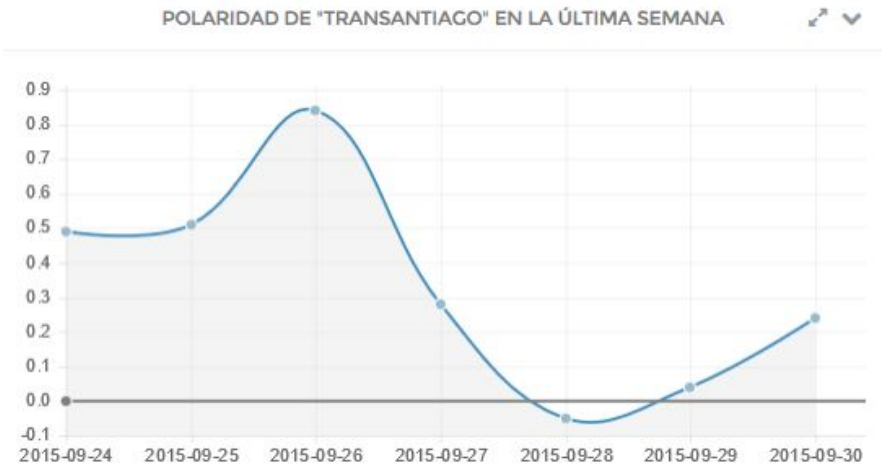
En el cuarto gráfico, mostrado en la figura 7.16, se ve la polaridad en el transcurso de los días. Esto fue tomado para la última semana de septiembre de 2015. Mientras que para los últimos días de la semana laboral los comentarios son positivos, al empezar la semana las opiniones presentan polaridad más negativa, siendo el domingo un día neutro. Esto puede explicarse por factores anímicos propios del fin de semana pero, nuevamente, se necesita más experiencia en el ámbito del sistema de transporte público capitalino para ser más concluyentes. Conociendo que en esos días

Figura 7.15: Frecuencia por día para keyword "transantiago"

Fuente: www.opinionzoom.cl

hubo un paro, se puede entender mejor el descontento de los usuarios con el sistema transantiago expresado a través de Twitter.

Figura 7.16: Polaridad por día para keyword "transantiago"



Fuente: www.opinionzoom.cl

La tabla central de esta vista, expuesta en la figura 7.17, muestra a los usuarios influyentes que hablan de transantiago y está ordenada de mayor a menor influencia. Los resultados resultan lógicos y muestran a usuarios como @Transantiago, que está obviamente relacionado con su keyword homónima; @louisdegrange, ingeniero experto en transporte y profesor del ramo en la Universidad Diego Portales; @AlsaciaExpress, empresa que opera buses al ganar la licitación del Transantiago; y @mtt_chile, ministerio precursor y fiscalizador del sistema Transantiago. Se ve que los resultados son muy coherentes y son un buen recurso para identificar actuales y potenciales líderes de opinión respecto del tema. La tabla incluye además la cantidad de veces que ha emitido un *tweet* con la keyword estudiada y el promedio de polaridad de sus opiniones al respecto.

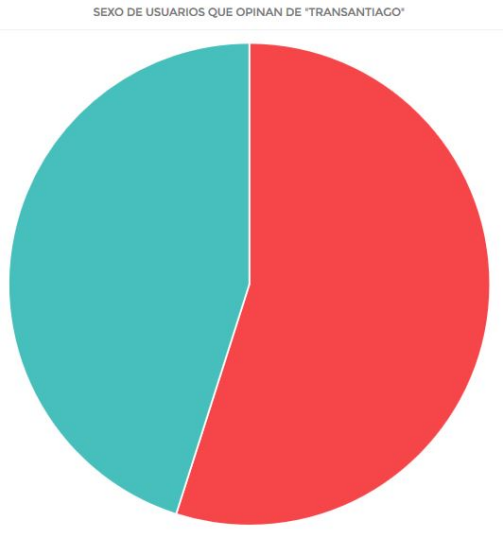
Figura 7.17: Usuarios influyentes para keyword "transantiago"

Usuario	Screen Name	Frecuencia	Influencia	Polaridad
Transantiago	@Transantiago	14	81.85	0.57
Louis de Grange	@louisdegrange	25	61.76	-1.43
Alsacia Express	@AlsaciaExpress	182	59.74	1.00
Min Trans y Telecom	@mtt_chile	25	59.19	0.42
Ivan Poduje	@ipoduje	12	55.15	-0.88
Jorge González #Jota	@JotaGonzalezGxD	18	52.17	0.00
Claudio Oyarzun™	@ClaudioOyarzunL	27	50.5	0.26
carolina brahim	@catibra	10	47.45	1.30
Roberto Contreras	@BobContreras	13	42.72	-0.38
rodrigo chiappe	@rodchiappe	10	42.48	0.80

Fuente: www.opinionzoom.cl

El gráfico de torta de la figura 7.18 muestra algo muy lógico: la proporción entre hombres y mujeres es significativamente la misma. En rojo se puede ver la participación de mujeres. Transantiago como concepto es un tema neutral que no puede ser caracterizado como masculino o femenino.

Figura 7.18: Sexo de usuarios para keyword "transantiago"



Fuente: www.opinionzoom.cl

La lista de intereses de la figura 7.19 es tan inútil como la de la vista Explora, al ser los resultados poco atingentes al tema en cuestión. Cuesta encontrar una relación verdaderamente evidente. Quizás la que más responde a la intuición es la presencia del interés *Bicycling*, debido a que es natural

que los usuarios que andan en bicicleta hablen del transantiago, o puede que haya una correlación entre opiniones sobre temas de transporte.

Figura 7.19: Intereses de usuarios para keyword "transantiago"

INTERESES COMPLEMENTARIOS DE "TRANSANTIAGO"

Mostrar 10 registros Buscar:

Nº	Interés	Adeptos
1	Chile	2314
2	Talca	622
3	Journalism	432
4	Spanish	272
5	WhatsApp	173
6	Antofagasta	89
7	Bicycling	84
8	Wine	80
9	Caracas	78
10	Burgos	71

Fuente: www.opinionzoom.cl

Por último en la figura 7.20 se ven los indicadores promedio para la keyword transantiago, desde donde se concluye que es un tema neutro, con influencia promedio y de bajo impacto.

Figura 7.20: Indicadores para keyword "transantiago"

INDICADORES DE "TRANSANTIAGO"

Polaridad	Influencia	Impacto
0.60	40.54	1.08

Fuente: www.opinionzoom.cl

7.2.3. Resultados de Indicadores

La información de la vista Indicadores de la figura 7.21 muestra simplemente a las keywords contratadas por frecuencia de *tweeteo*. Se ve que Facebook es por lejos la keyword más popular en esta muestra de Twitter. Le siguen temas controversiales y públicos como aborto, marihuana, delincuencia, terremoto y transantiago. El tema más negativo es delincuencia, lo que es completamente intuitivo. Mientras Facebook es el que cuenta con el impacto más alto, el impacto más negativo es de delincuencia.

Figura 7.21: Indicadores para todas las keywords

Nº	Keyword	Frecuencia	Polaridad	Influencia	Impacto
1	Facebook	2918	0.69	35.42	7.65
2	aborto	675	-0.87	39.87	-10.05
3	marihuana	544	-0.51	38.90	-6.63
4	delincuencia	492	-4.45	41.45	-48.51
5	terremoto	412	-2.65	38.56	-26.51
6	transantiago	408	0.60	40.54	6.49
7	@entel	293	1.06	34.48	6.81
8	temblor	193	-0.66	34.58	-5.91
9	#Facebook	157	0.69	45.42	6.75
10	@womchile	147	0.44	35.00	2.72

Fuente: www.opinionzoom.cl

7.2.4. Resultados de Alertas Puntuales

En la figura 7.22 se pueden ver las alertas puntuales generadas. Estos resultados obedecen a la configuración de una alerta que buscaba *tweets* con polaridad menor a -2. Se encuentran justamente opiniones negativas, que es lo que se andaba buscando.

Figura 7.22: Alertas para *tweets* puntuales

ALERTAS PUNTUALES

Mostrar 10 registros Buscar:

ID	Fecha	Tweet	Polaridad	Influencia	Impacto	ID de Twitter	ID de Alerta
2	2015-09-29 14:15:59	Muere mi celular, voy a @entel me dan uno de repuesto y después de menos de un día el cargador no funciona.. Como los odio.	-11	41.64	-18.491	648853577646108672	2
5	2015-09-30 18:30:27	@_VjArt @MovistarChile @entel me saturo, es extremadamente caro renovar el equipo!	-3.05	41.82	-5.12705	649289917894410240	2

Fuente: www.opinionzoom.cl

7.2.5. Resultados de Alertas Generalizadas

Las alertas generalizadas de la figura 7.23 obedecen a 3 alertas configuradas para encontrar eventos de alta frecuencia de *tweets*. Nótese que son distintas alertas, esto es porque son distintas keywords.

Cabe destacar la proximidad de dos alertas con la misma configuración, en el caso de la que tiene ID 4. Aquí naturalmente convendría unir ambos periodos de tiempo para avisar que durante un período de cierta prolongación significativa hubo muchos *tweets* sobre las keywords en cuestión. Para ello se recomienda analizar cada mayor tiempo las opiniones que van llegando. De todas formas, esta es una decisión que debe ser consensuada con el cliente de OpinionZoom a partir de sus requerimientos y necesidades.

Figura 7.23: Alertas provocadas por un conjunto de *tweets*

ALERTAS GENERALIZADAS

Mostrar 50 registros Buscar:

ID	Inicio	Término	Frecuencia	Polaridad	Impacto	ID de Alerta
51	2015-09-25 15:18:08	2015-09-25 15:19:03	92	-0.98	-0.89	4
52	2015-09-25 15:19:08	2015-09-25 16:16:12	98	-1.23	-0.9	4
53	2015-09-25 16:33:03	2015-09-25 17:30:09	31	-0.06	-0.25	1
54	2015-09-25 16:33:12	2015-09-25 16:59:11	17	-0.12	-0.46	5
55	2015-09-25 16:34:08	2015-09-25 17:41:08	23	-0.78	-0.34	4
56	2015-09-25 17:33:03	2015-09-25 18:30:05	32	-1.38	-1.05	1

Fuente: www.opinionzoom.cl

7.3. Evaluación de Rendimiento del Sistema

Los costos críticos del proyecto son temporales, y refieren principalmente a la capacidad de hacer ETL de *tweets* y al rendimiento de la página Web, específicamente al tiempo de respuesta que, para el usuario cliente, se ve dilatado por la complejidad de las consultas.

7.3.1. Rendimiento del ETL de *tweets*

Para medir el rendimiento de la extracción, transformación y carga de *tweets* en el sistema se creó la tabla 'logsrendimiento', la cual está descrita en la tabla 7.2.

Desde esta tabla, promediando los datos, se obtienen algunos ratios de rendimiento:

- Consulta: porcentaje del tiempo total del programa que es ocupado en consultar a la base de datos.

$$Consulta = \frac{tiempoconsultagorda}{tiempototal} \tag{7.1}$$

- Desechados: ratio de *tweets* desechados por lenguaje.

$$Desechados = \frac{desechadosLenguaje}{totaltweets} \tag{7.2}$$

- VelTweet: velocidad de procesamiento de un *tweet*. Se calcula como la división de la cantidad de *tweets* procesados por la diferencia en segundos entre el tiempo del programa y el tiempo de consulta, como se ilustra en la siguiente ecuación:

$$VelTweet = \frac{tweetsProcesados * 1000}{tiempoprograma - tiempoconsultagorda} \tag{7.3}$$

- Reinserción: muestra cuántas veces se inserta un mismo *tweet* en el sistema. No se consideran los *tweets* que fueron desechados por lenguaje, porque no tienen la oportunidad de ser insertados.

$$Reinsercion = \frac{tweetsInsertados}{tweetsProcesados} \tag{7.4}$$

Fuente: Elaboración Propia

Atributo	Tipo	Descripción
idlogsrendimiento	int	identificador del registro de rendimiento insertado
tiempoinicial	timestamp	momento exacto en que se inicia el programa
tiempoconsultagorda	bigint	el tiempo en milisegundos que demora la consulta a la base de datos 'La Gorda'
totalkeys	int	la cantidad total de keywords por las que se seleccionan *tweets*
tiempoprograma	bigint	el tiempo total que se demoró el programa en milisegundos
desechadosLenguaje	int	número de *tweets* que no venían en español
tweetsProcesados	int	cantidad de *tweets* que llegaron y no fueron desechados
tweetsInsertados	int	número de veces en que se insertó un *tweet*. Un *tweet* puede ser insertado más de una vez si contiene más de una keyword.
totaltweets	int	la cantidad total de *tweets* obtenidos en la consulta inicial
timestamp	timestamp	momento en el que se terminó el programa y se actualizaron los datos anteriores

Tabla 7.2: Tabla Logsrendimiento

Los resultados son útiles en sí al dar una idea de lo que sucede con el programa que alimenta el sistema. Sin embargo estas medidas son aún de mayor provecho cuando se compara el rendimiento del sistema, con estos ratios, entre la base de datos 'La Gorda' en PostgreSQL y SOLR. Los resultados se muestran así en la tabla 7.3. Todos los resultados están en promedio; las varianzas fueron excluidas porque, al ser todas cercanas al 0.01, no aportaban ningún valor al análisis.

Fuente: Elaboración Propia

Ratio	SOLR	PostgreSQL
Consulta	0,06 %	8,5 %
Desechados	3,9 %	2.6 %
VelTweet	0,13	0,35
Reinserción	1,1	1,18

Tabla 7.3: Ratios de rendimiento

De la tabla surgen naturalmente algunas observaciones. En primer lugar, al migrar la base de datos de PostgreSQL a SOLR se disminuyó el tiempo de consulta considerablemente. El tiempo de consulta para SOLR es aproximadamente en promedio 1,7 segundos mientras que para PostgreSQL era de 262 segundos. Esa diferencia abismante era la que se buscaba en la migración, pues SOLR está optimizada para consultas de lectura e inserción, que representan casi el total de las consultas que se realizan a 'La Gorda'.

En segundo lugar se ve que la cantidad de *tweets* desechados por lenguaje aumenta para SOLR.

Esto sólo tiene su explicación en la secuencialidad de SOLR con respecto a PostgreSQL. En efecto, 'La Gorda' está constantemente agregando nuevos usuarios, entre ellos una parcialidad de usuarios extranjeros, seguidores de algún usuario chileno, lo que aumenta la cantidad de *tweets* en otros idiomas recibidos. Siempre es desable que el número de *tweets* desechados por lenguaje disminuya. Para que ello ocurra es necesario que 'La Gorda' sea más fina al filtrar a los usuarios de Twitter que va a seguir, asegurándose que sean chilenos.

La tercera observación es sobre la diferencia en velocidad de procesamiento de un *tweet*. Cuando 'La Gorda' estaba PostgreSQL se procesaba uno cada 3 segundos aproximadamente. En cambio ahora el tiempo de procesamiento disminuye un tercio. Esto puede producirse por cambios en la escala temporal ya que la diferencia entre el tiempo total que demora el programa y el tiempo de consulta es menor para las consultas hechas en PostgreSQL que las hechas en SOLR, haciendo menor el denominador del ratio VelTweet.

Por último se nota que el ratio de reinserción no se altera por que se cambie o no el motor de base de datos. Simplemente depende de las keywords y la correlación textual que haya en los *tweets* que por ellas son seleccionados. Cada *tweet* está siendo insertado en promedio 1,1 veces. Esto quiere decir que cada 10 *tweets* hay uno más que se inserta para mantener la referencialidad a las Keywords. Si este número fuera mayor, por ejemplo cercano a 2, quizá valdría el esfuerzo revisar y modificar el modelo de tal forma que se inserte un *tweet* sólo una vez. Hay que advertir de todas maneras que esta normalización alejaría el modelo del paradigma de un Data Warehouse, con el consecuente empeoramiento del rendimiento de la página Web, que recopila las vistas que el cliente necesita para la gestión.

7.3.2. Rendimiento de las Consultas de la Página

Hay 16 consultas que hace el usuario cliente y 17 que hace el usuario administrador, con la diferencia que las ocupadas por el administrador son más repetitivas y mucho más simples. De éstas últimas consultas las respuestas vienen en un tiempo muy corto al ser sobre tablas pequeñas (menos de 100 registros), como lo son las tablas 'cliente' y 'users'. Las 4 consultas del módulo de alertas tampoco presentan problemas de rendimiento, pues son sencillas y sobre tablas también pequeñas. Los tiempos de respuesta críticos son los producidos por las 12 consultas del módulo de inteligencia, donde sus 3 vistas tardan en cargarse por esperar los datos de la base de datos. Éstas son las consultas que configuran la capa de análisis del sistema. Esta demora puede ser perjudicial tanto en la experiencia sensible del usuario con el sitio como en la funcionalidad propiamente tal, pues el servidor o bien puede terminar con la petición al superar el máximo tiempo de espera de un cliente, o se puede llenar de consultas concurrentes sin poder resolver ninguna. Por ello se analizan los tiempos de respuesta de las 3 vistas por separado.

Explora

La vista explora tiene 5 consultas, como se exhibe en el capítulo 5. Éstas consultas son la frecuencia de *tweeteo* sobre las keywords contratadas, los intereses de los usuarios que las emitieron, de ellos los usuarios más influyentes y los *tweets* más negativos y los más positivos que han emiti-

do. Todas estas consultas están hechas sobre la última semana. En la tabla se puede ver el tiempo promedio que demora cada consulta en segundos.

Fuente: Elaboración Propia

Consulta	Tiempo (segundos)
Frecuencia	2,57
Intereses	3,2
Usuarios	1,13
Tweets negativos	0,22
Tweets positivos	0,22
SUMA	7,34

Tabla 7.4: Tiempos de consulta para la vista Explora

Un resultado de interés para quien formule las consultas es la gran ventaja en eficiencia que tienen las consultas que ocupan la tabla vista frente a las que no. Todos los resultados de la tabla fueron obtenidos ocupando la tabla vista 'facttweet'. Se hizo la comparación con los resultados de las dos últimas tablas haciendo también la consulta sin ocupar la vista, uniendo las distintas tablas en la misma consulta. El resultado fue de 1,84 segundos promedio, es decir 9 veces más. Si no se ocupara la tabla vista 'facttweet', bajo las condiciones actuales, esta vista no podría ser desplegada con todas las consultas actuales.

Inteligencia

La vista inteligencia tiene 6 consultas, aunque éstas produzcan 8 salidas que componen la vista para el usuario final. La primera consulta es la frecuencia y polaridad promedio para cada hora del día. La segunda consulta lo mismo pero para cada día de los últimos 7. La tercera pregunta sobre los usuarios influyentes que han hablado de la keyword, la cuarta el sexo de los usuarios, la quinta los intereses y la sexta sobre el promedio de indicadores relacionados a la keyword.

Fuente: Elaboración Propia

Consulta	Tiempo (segundos)
Frec. y pol. por hora	1,88
Frec. y pol. por día	2,19
Usuarios	4,45
Sexo	2,76
Intereses	4,65
Indicadores	2,65
SUMA	18,58

Tabla 7.5: Tiempos de consulta para la vista Inteligencia

Indicadores

La vista indicadores sólo tiene una consulta, aunque ésta produzca 2 salidas que componen la vista para el usuario final. La consulta es por la frecuencia, la polaridad, la influencia y el impacto de casa keyword en los últimos 7 días. Esta consulta es pesada, y demora en promedio 5,2 segundos

7.4. Rendimiento Económico

Pese a que queda fuera del alcance de esta memoria, resulta natural en un trabajo de título para optar al grado de ingeniero civil industrial obtener algunos resultados económicos, sobre todo pudiendo acceder a los datos de primera mano.

Los primitivos cálculos que aquí se ocupan, más que los resultados en si, pueden ser un apoyo a la fijación de precios del producto final que será OpinionZoom.

Existen 2 costos fundamentales: el precio del servidor, pagado en dólares, y el tiempo de procesamiento. El precio del servidor es de 606 dólares anuales. Si un año (365 días) tiene 31.536.000 segundos, el precio que se paga por un segundo de servidor es de $1,92x10^{-5}$. Bajo el supuesto fuerte que el servidor esté funcionando todo el año a su máxima capacidad, la pregunta que surge es el tiempo que demora un *tweet* en procesarse. Este valor es el inverso del ratio VelTweet ya calculado, lo que da un valor de 7,69 segundos. Si el precio por segundo de servidor es de $1,92x10^{-5}$ dólares y el tiempo de procesar un *tweet* es de 7,69 segundos, el precio de procesar un *tweet* es de 0,0001478 dólares. Esto lleva inmediatamente a calcular el precio por keyword. Con un promedio de *tweets* diarios de 500 por cada Keyword, diariamente una keyword cuesta 0,0739 dólares y mensualmente 2,21. Con esta fijación de precios basada en costos, una cota inferior para el cobro de los servicios de OpinionZoom es de aproximadamente (con el dólar cercano a los 687 pesos) 1.500 pesos chilenos mensuales por keyword contratada.

Capítulo 8

Conclusiones

Se presentan en este capítulo dos secciones. Primeramente se muestran conclusiones generales y luego se proponen trabajos futuros, que son desafíos para las próximas iteraciones de la aplicación OpinionZoom.

8.1. Conclusiones Generales

Las conclusiones generales versan sobre el cumplimiento de los objetivos, la validación de la hipótesis, la metodología utilizada y los resultados obtenidos.

8.1.1. Sobre los Objetivos

Gracias a la estructura del presente trabajo se puede ver fácilmente que han sido cumplidos los objetivos planteados en el inicio del trabajo. Aún más, en el presente trabajo queda descrito todo el desarrollo que llevó a este logro.

El objetivo general de este trabajo de título es: **Diseñar y Construir un prototipo funcional de sistema de análisis de opiniones en Twitter integrando algoritmos de Data Mining**.

Este objetivo fue cumplido plenamente, pues luego de este trabajo de título el WIC cuenta con un prototipo funcional que recibe opiniones de Twitter, las procesa y analiza con algoritmos de Data Mining.

Objetivos Específicos

Los objetivos específicos también fueron cumplidos. A continuación se repite la lista de objetivos y se explica porqué están cumplidos:

1. Diseñar un sistema de análisis de opiniones en Twitter. El sistema fue diseñado tal como se plantea en el marco teórico, definiendo sus entradas, su proceso y sus salidas. En el capítulo 3 se definen las entradas del sistema y se esquematiza el proceso. En el capítulo 4 se explica la parte más importante del proceso, el ETL del Data Warehouse, y se finaliza con la base de datos ya diseñada y con datos, lista para ser consultada. Las salidas del sistema son abordadas ampliamente en los capítulos 5 y 6.

2. Construir un Data Warehouse que recopile datos y permita poner a disposición del cliente indicadores orientados a la gestión. Para construir un Data Warehouse hay que definir los indicadores, modelar la base de datos, planificar y construir un sistema ETL y mostrar los datos con vistas útiles para la gestión. Se realizaron todos los pasos y los resultados se muestran desde el capítulo 3 hasta el 6, donde se muestran las vistas resultantes, aunque es el capítulo 4 el que más detalla lo central del Data Warehouse.

3. Construir una plataforma Web que ofrezca un servicio de análisis de opiniones. La página Web está de hecho en uso, y tiene clientes ficticios y clientes internos reales. En los anexos se pueden ver vistas adicionales a las ya mostradas en los dos capítulos anteriores, pero se puede disfrutar del logro de este objetivo de forma más completa creando una cuenta, contratando los servicios y explorando todas las funcionalidades de OpinionZoom.cl

4. Evaluar el rendimiento del prototipo de sistema. En el capítulo de resultados se evalúan tanto los resultados del análisis de opiniones como el costo temporal de los análisis que se realizan. También, aunque fuera de los alcances, se hace una pequeña evaluación económica dado el costo del servidor y la capacidad de procesar *tweets* que tiene el sistema.

Hipótesis de Investigación

Se valida la hipótesis de investigación, especificada en el capítulo introductorio como

Es posible crear un sistema de análisis de opiniones en Twitter integrando algoritmos de Data Mining que por separado detecten entre otras cosas la orientación sentimental de una opinión, la influencia de los usuarios de Twitter y los intereses de estos usuarios.

Se ha creado el sistema de análisis, a nivel de prototipo, que integra algoritmos de Data Mining. Éstos algoritmos son los siguientes:

- Detección de polaridad, aportado por el WIC
- Detección de influencia, propio de Klout
- Detección de intereses, propio de Klout
- Detección de sexo, fruto de esta investigación

En el mismo sistema esta información se agrega para entregar servicios útiles para los usuarios de OpinionZoom, tales como los presentes en el capítulo 5. Esta hipótesis de investigación queda de esta forma completamente validada con el sistema creado.

8.1.2. Coherencia de los Resultados

Se puede ver en el capítulo anterior que los resultados son coherentes con la intuición y la realidad. Se verifica esto con la frecuencia de *tweeteo* que es menor durante la noche, con la polaridad de los *tweets* a los cuales un humano también clasificaría como negativo o positivo como fueron clasificados por PAPI y con los tipos de usuarios que opinan sobre una keyword entre otras coincidencias. Esta coherencia entre los resultados obtenidos por el sistema y la realidad, otorga al prototipo de OpinionZoom una confiabilidad suficiente pero que debe ser mejorada con vistas al producto final.

Los únicos resultados que no responden a la intuición son los de los intereses. Esto es debido a la API de Klout no tiene buenos resultados en el medio nacional, quizá por desconocer la cultura del país o porque sus algoritmos no son lo suficientemente buenos. De todas maneras, esta API debe ser reemplazada por otra, especialmente en cuanto a sus intereses, pues los resultados de influencia aun son útiles.

8.1.3. Sobre la Metodología

Para el desarrollo del prototipo se utilizó la metodología ágil de desarrollo llamada Srum. Ésta requirió de la participación de todo el equipo, el que se vio fortalecido por la forma de trabajar, respondiendo a los objetivos de la metodología.

Una de las ventajas que tuvo ocupar Scrum fue la flexibilidad, ya que permitió agregar al sistema funcionalidades imprevistas en un primer momento que resultaron muy útiles para el análisis de los datos. Dentro de estas innovaciones estuvo la creación del algoritmo de identificación de sexo de un usuario, la migración de la base de datos de PostgreSQL a Solr y el uso de la API de Klout. Esta última fue una solución muy buena frente a la falta inesperada de algoritmos de detección de influencia e identificación de intereses, que a principios de este trabajo se programaron para ser incluidas en el prototipo.

La desventaja mayor fue la constante suma de requerimientos y con ello la falta de limitación en los plazos y el mal control de los tiempos, problemas comunes en los equipos que usan esta forma de desarrollo pero tampoco ajena a algunos equipos que ocupan metodologías tradicionales.

La discusión teórica en cuanto a la metodología, enunciada en el capítulo 1, fue averiguar si era posible trabajar con una metodología ágil un proyecto que incluye la creación de un Data Warehouse. No existe en la literatura un caso en donde se haya construido un Data Warehouse usando Scrum. Esto se debe a que la construcción de un Data Warehouse, tal como lo plantean los autores Inmon y Kimball separadamente, tiene su propia metodología. En este caso fue la metodología del Data Warehouse la que se adaptó, siendo ésta una de las mayores innovaciones dentro de este trabajo. Para ello, se separaron las actividades propias de la metodología de un Data Warehouse y fueron evaluadas en cada Sprint de Scrum, adaptándolas a los nuevos requerimientos de los usuarios (que fueron usuarios internos del WIC).

La metodología Scrum logró su cometido, se pudo trabajar con ella, fue suficientemente flexible para acomodarse a las características dinámicas del equipo OpinionZoom y se logró desarrollar el

presente trabajo sin mayores dificultades.

8.1.4. Velocidad de Análisis

Lo primero que debe resaltarse es la consecuencia de la migración de DBMS a motor de búsqueda. No se puede soslayar la diferencia abismante en rendimiento entre el gestor de base de datos PostgreSQL con el motor de búsqueda Solr. La velocidad de Solr para las búsquedas es de más de 230 veces que la rapidez de PostgreSQL. Ésta diferencia es determinante para el buen funcionamiento de OpinionZoom y puede ser una variable clave de éxito cuando se sumen más clientes a OpinionZoom.

El siguiente cambio significativo en velocidad puede darse cambiando la PAPI de servidor. Cuando estaba en un servidor de propósito general (*General Purpose*) en AWS, se demoraba 7 veces más que en el servidor actual, el cual está optimizado para cómputo (*Compute Optimezed*). Aún hay servidores mejores que explorar, que pueden hacer más rápidos los complejos procesos internos de PAPI.

8.1.5. Importancia del Equipo

Este trabajo no pudo haber sido realizado por sólo una persona con los conocimientos de la carrera, ni siquiera en el doble del tiempo que este trabajo demoró. El equipo de OpinionZoom fue vital para la construcción de este prototipo. Esto se debe a que se aprovecharon en este trabajo los conocimientos que se consiguieron con mucho esfuerzo durante trabajos de título anteriores, capitalizando así el rico capital humano con el que cuenta el WIC.

La metodología de desarrollo aportó también en este aprendizaje colaborativo, en donde el equipo obtuvo durante este tiempo un *know-how* que puede ser aprovechado en futuros proyectos o para el producto final. El espacio físico fue otro factor que favoreció la buena comunicación, pues la cercanía espacial permitía que las dudas y reflexiones se pudieran plantear en el mismo momento en que surgian. La gestión del conocimiento dentro de esta pequeña organización no debe ser subvalorada, pues es necesaria una voluntad primera y un esfuerzo consecuente para obtener ideas, explorar caminos, sortear dificultades y mejorar el desarrollo a un nivel de excelencia, valor primero de la Universidad de Chile.

8.2. Trabajo Futuro

Definidos de antemano los alcances en el capítulo introductorio, y luego de exponer el trabajo desarrollado, se plantea una lista de futuros desafíos que varían tanto en magnitud como en áreas específicas del proyecto OpinionZoom, con vistaa a un segundo prototipo o directamente al producto final.

- Intentar poner importancia por tweet. Ahora un *tweet* tiene la influencia de quien lo emite,

pero lo ideal sería que cada *tweet* tenga su propia importancia de acuerdo no solamente a un usuario sino también a un contexto u otras variables a incluir en la tabla de hechos.

- Crear una nueva tabla de hechos para la información agregada de keywords. A fin de hacer más simples las consultas y sin miedo de hacer redundar información, según la filosofía de Data Warehouse, se puede incluir una nueva tabla de hechos para la información agregada de las keywords tales como frecuencia, polaridad, influencia e impacto.
- Personalizar las sesiones. Que los usuarios puedan modificar su foto de perfil, tener más información que compartir e incluso cambiar el color de fondo de la aplicación. Todas éstas funcionalidades más propias del diseño, pese a estar fuera del alcance de este prototipo, son necesarias para responder a las exigencias del mercado de hoy.
- Test de usabilidad. Hacer un test de usabilidad con el nuevo prototipo funcional para medir cinco variables claves: facilidad de aprender, eficiencia, facilidad de recordar, eficacia y minimización de errores, y satisfacción. Los resultados de este test de usabilidad deben ser integrados al sistema.
- Sumar más redes sociales al sistema. OpinionZoom declara en su objetivo general que ocupará información textual en las redes sociales. Dado que el prototipo está hecho sólo para la información textual de Twitter, se hace necesario para cumplir su objetivo incluir más redes sociales al sistema.
- Mejorar el rendimiento de la página. Buscar y aplicar maneras de disminuir el tiempo de respuesta en los análisis y su visualización en la página Web. Para ello, se hace imperativo disminuir los tiempos de consulta a la base de datos pero tabién se pueden explorar otras alternativas, como desagregar las vistas o cambiar las consultas por unas que produzcan menos demora.
- Actualizar imágenes de usuarios de Twitter. Los usuarios de Twitter actualizan periódicamente sus *avatar* o imágenes de cuenta. Esto hace que haya que actualizar también periódicamente la base de datos, en particular la tabla usuarios. Esto se puede hacer desde la Rest API de Twitter. El único problema es que las credenciales para acceder a ella tienen un límite de tiempo y de llamadas. Por esto se recomienda realizar la actualización en varios días, asi como también usar varias credenciales.
- Emitir reportes en PDF. Es un servicio que este prototipo no incluye pero que está dentro de los requisitos para OpinionZoom. Sin esta funcionalidad OpinionZoom quedaría relegada inexorablemente frente a sus competidores.
- Desarrollar una API de influencia y una API de intereses. Con mediana urgencia OpinionZoom necesita independizarse de Klout, pues con esa dependencia resulta demasiado inestable la comercialización y los resultados, especialmente los de intereses, pueden no tener la calidad adecuada para los usuarios de OpinionZoom. Con APIs propias del centro se podrían adaptar y mejorar los algoritmos periódicamente para responder a los desafíos siempre cambiantes del mercado.
- Agregar categorías a los intereses. Sería conveniente que en la nueva API de polaridad que desarrolle el centro se incluya un árbol de intereses de 3 o 4 pisos. De esta manera se podría dar una mejor información al usuario de OpinionZoom. Con este prototipo se puede acceder a una lista unidimensional de intereses y ver la cantidad de usuarios que a él adhieren. Al usuario de OpinionZoom podría interesarle, por ejemplo, saber que los usuarios de Twitter que hablan sobre sus keyword contratadas se dividen en 3 grupos: los que gustan de la música, los que están interesados en el deporte y aquellos ligados a la tecnología. Desagregadamente,

el usuario final vería una gran lista difícil de comprender, con instrumentos musicales, bandas de música, artes marciales, tipos de computadores, etcétera. El agregar esta información en categorías le acerca el conocimiento al usuario final, haciendo su análisis más rico. Esto podría hacerse con métodos de NLP o de *Machine Learning*.

- Definir automáticamente los umbrales de las alertas. Actualmente los umbrales son definidos por los mismos usuarios de OpinionZoom, pero el usuario desconoce *a priori* el nivel de frecuencia, polaridad, influencia o impacto que subjetivamente es alto o bajo. Siempre pensando en que OpinionZoom sea lo más posible una herramienta de análisis para la gestión, los umbrales se podrían definir por métodos estadísticos como medias móviles u otros que, considerando la media, la varianza o incluso el tipo de keyword contratada, recomiende niveles para configurar las alertas adecuadamente.
- Filtrar mejor a los usuarios de La Gorda y consecuentemente a los del Data Warehouse. Aún existe una cantidad significativa de usuarios extranjeros, principalmente venezolanos, que ensucian los resultados pensado en que el análisis está enfocado sobre usuarios chilenos. También podríanse eliminar a las cuentas de los medios de comunicación u otras organizaciones, que podrían no ser consideradas como clientes ni como prospectos por los usuarios de OpinionZoom. La sola identificación y categorización de distintos tipos de usuarios puede resultar en un interesante y útil trabajo futuro para OpinionZoom.
- Lograr incluir un algoritmo de detección de ironía. Es una promesa del proyecto OpinionZoom que debe cumplirse en el producto final. El WIC ha realizado investigación al respecto [51] pero falta mejorarla y construir una API para poder integrarla al sistema construido.
- Posicionar mejor la página `www.opinionzoom.cl` en los buscadores. Esta optimización para motores de búsquedas o SEO (*Search Engine Optimization* en inglés) mejora la visualización de un sitio Web en los resultados orgánicos de un buscador. Esto puede hacerse modificando el contenido de la página, cambiando su estructura, haciendo que otras páginas redireccionen a ésta y generando contenido interesante entre otras estrategias [52]. Aumentar el *ranking SEO* puede traer grandes beneficios económicos al aumentar el número de posibles clientes que se enteran de la existencia del sitio Web.

Bibliografía

[1] Web Intelligence Centre. (2015) Wic. Consulta: 24 julio 2015. [Online]. Available: http://wic.uchile.cl/

[2] E. Marrese-Taylor, J. D. Velásquez, and F. Bravo-Marquez, "A novel deterministic approach for aspect-based opinion mining in tourism products reviews," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7764–7775, 2014. [Online]. Available: http://wic.uchile.cl/wp-content/uploads/2015/09/A-novel-deterministic-approach-for-aspect-based-opinion-mining-in-tourism-products-reviews.pdf

[3] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.

[4] E. Marrese Taylor, "Diseño e implementación de una aplicación de web opinion mining para identificar preferencias de usuarios sobre productos turísticos de la x región de los lagos," 2013, Tesis de Pregrado, Universidad de Chile.

[5] E. Marrese-Taylor, J. Velasquez, and F. Bravo-Marquez, "Opinion zoom: A modular tool to explore tourism opinions on the web," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, vol. 3. IEEE, 2013, pp. 261–264.

[6] CORFO, "Proyecto final l2 opinionzoom, proyecto i+d aplicada," 2013.

[7] F. J. Ponce de León Pollman, "Uso de la ingeniería de negocios en diseño e implementación de negocios con tecnologías de información," 2015, Tesis de Magister, Universidad de Chile.

[8] H. Romero. (2015) Metodologías de desarrollo. Consulta: 16 julio 2015. [Online]. Available: http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo

[9] Kent Beck, Mike Beedle et al. . (2001) Manifiesto de metodologías Ágiles de desarrollo. Consulta: 17 julio 2015. [Online]. Available: http://www.agilemanifesto.org/iso/es/

[10] K. Schwaber, *Agile project management with Scrum*. Microsoft Press, 2004.

[11] F. A. Vera Cid, "Caracterización de perfiles influyentes en twitter de acuerdo a tópicos de opinión y la generación de contenido interesante," 2014, Tesis de Pregrado, Universidad de Chile.

[12] M. A. Russell, *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub and More*. O'Reilly Media, Inc., 2013.

[13] W. H. Inmon, *Building the data warehouse*. John wiley & sons, 2005.

[14] R. Kimball and J. Caserta, *The data warehouse ETL toolkit*. John Wiley & Sons, 2004.

[15] T. O. Davenport and L. Prusak, *Ecología de la Información*. Oxford, 1999.

[16] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The kdd process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[17] O. Maimon and L. Rokach, *Data mining and knowledge discovery handbook*. Springer, 2005, vol. 2.

[18] J. A. Balazs and J. D. Velásquez, "Opinion mining and information fusion: A survey," *Information Fusion*, vol. 27, pp. 95–110, 2016.

[19] J.-A. J.-M. Balazs Thenot, "Diseño, desarrollo e implementación de una aplicación de web opinion mining para identificar el sentimiento de usuarios de twitter con respecto a una compañia de retail," 2015, Tesis de Pregrado, Universidad de Chile.

[20] Oracle Corporation. (2015) Java. Consulta: 16 julio 2015. [Online]. Available: http://www.java.com/es/

[21] T. P. Group. (2015) Php. Consulta: 09 julio 2015. [Online]. Available: http://www.php.net/

[22] W3Schools. (2015) Sql. Consulta: 09 julio 2015. [Online]. Available: http://www.w3schools.com/sql/sql_intro.asp

[23] W3Schools. (2015) HTLM. Consulta: 09 julio 2015. [Online]. Available: http://www.w3schools.com/html/html_intro.asp

[24] V. Contributors. (2014) Volt. Consulta: 09 julio 2015. [Online]. Available: http://www.volt-lang.org/

[25] Oracle Corporation. (2015) Mysql. Consulta: 16 julio 2015. [Online]. Available: https://www.mysql.com/

[26] MariaDB Foundation. (2015) Mariadb. Consulta: 09 julio 2015. [Online]. Available: https://mariadb.org/

[27] The PostgreSQL Global Development Group. (2015) Postgresql. Consulta: 09 julio 2015. [Online]. Available: http://www.postgresql.org/

[28] The Apache Software Foundation. (2014) Solr. Consulta: 09 julio 2015. [Online]. Available: http://lucene.apache.org/solr/

[29] Oracle Corporation. (2015) Netbeans ide. Consulta: 10 julio 2015. [Online]. Available: https://netbeans.org/

[30] Apache Friends. (2015) Xampp. Consulta: 13 julio 2015. [Online]. Available: https://www.apachefriends.org/es/index.html

[31] Atlassian. (2015) Bitbucket. Consulta: 13 julio 2015. [Online]. Available: https://bitbucket.org/

[32] Amazon Web Services, Inc. (2015) Amazon web services. Consulta: 20 julio 2015. [Online]. Available: https://aws.amazon.com/es/

[33] Jon Skinner. (2015) Sublime text. Consulta: 14 julio 2015. [Online]. Available: http://www.sublimetext.com/

[34] Mozilla Foundation. (2015) Mozilla firefox. Consulta: 13 julio 2015. [Online]. Available: https://www.mozilla.org/

[35] Tom's Hardware Guide. (2015) The wbgp xvi winner's circle. Consulta: 13 julio 2015. [Online]. Available: http://www.tomshardware.com/reviews/chrome-27-firefox-21-opera-next,3534-12.html

[36] D. Moskovitz and J. Rosenstein. (2015) Asana. Consulta: 10 agosto 2015. [Online]. Available: https://asana.com/

[37] S. Houde and C. Hill, "What do prototypes prototype," *Handbook of human-computer interaction*, vol. 2, pp. 367–381, 1997.

[38] R. H. Saroka, *Sistemas de información en la era digital*. Fundación OSDE, 2002.

[39] C. G. Schoderbek, P. P. Schoderbek, and A. G. Kefalas, *Sistemas administrativos*. El Ateneo, 1984.

[40] K. C. Laudon and J. P. Laudon, "Management information systems: managing the digital firm," *New Jersey*, vol. 8, 2004.

[41] N. U. Rehman, S. Mansmann, A. Weiler, and M. H. Scholl, "Building a data warehouse for twitter stream exploration," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society, 2012, pp. 1341–1348.

[42] N. Cheng, R. Chandramouli, and K. Subbalakshmi, "Author gender identification from text," *Digital Investigation*, vol. 8, no. 1, pp. 78–88, 2011.

[43] J. D. Burger, J. Henderson, G. Kim, and G. Zarrella, "Discriminating gender on twitter," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1301–1309.

[44] Z. Miller, B. Dickinson, and W. Hu, "Gender prediction on twitter using stream algorithms with n-gram character features," 2012.

[45] D. Bamman, J. Eisenstein, and T. Schnoebelen, "Gender identity and lexical variation in social

media," *Journal of Sociolinguistics*, vol. 18, no. 2, pp. 135–160, 2014.

[46] M. Perl. (2015) Gender api. Consulta: 30 julio 2015. [Online]. Available: https://gender-api.com/

[47] B. Liu, "Sentiment analysis and subjectivity," *Handbook of natural language processing*, vol. 2, pp. 627–666, 2010.

[48] Brandmetric. (2015) Brandmetric.com. Consulta: 03 diciembre 2015. [Online]. Available: http://www.brandmetric.com/index.html

[49] W. Technologies. (2015) Wholemeaning. Consulta: 03 diciembre 2015. [Online]. Available: http://wholemeaning.com

[50] S. Architect. (2013) Performance benchmark of popular php frameworks. Consulta: 14 abril 2015. [Online]. Available: http://systemsarchitect.net/2013/04/23/performance-benchmark-of-popular-php-frameworks/

[51] V. A. Hernández Martínez, "Identificación de la presencia de ironía en el texto generado por usuarios de twitter utilizando técnicas de opinion mining y machine learning," 2015, Tesis de Pregrado, Universidad de Chile.

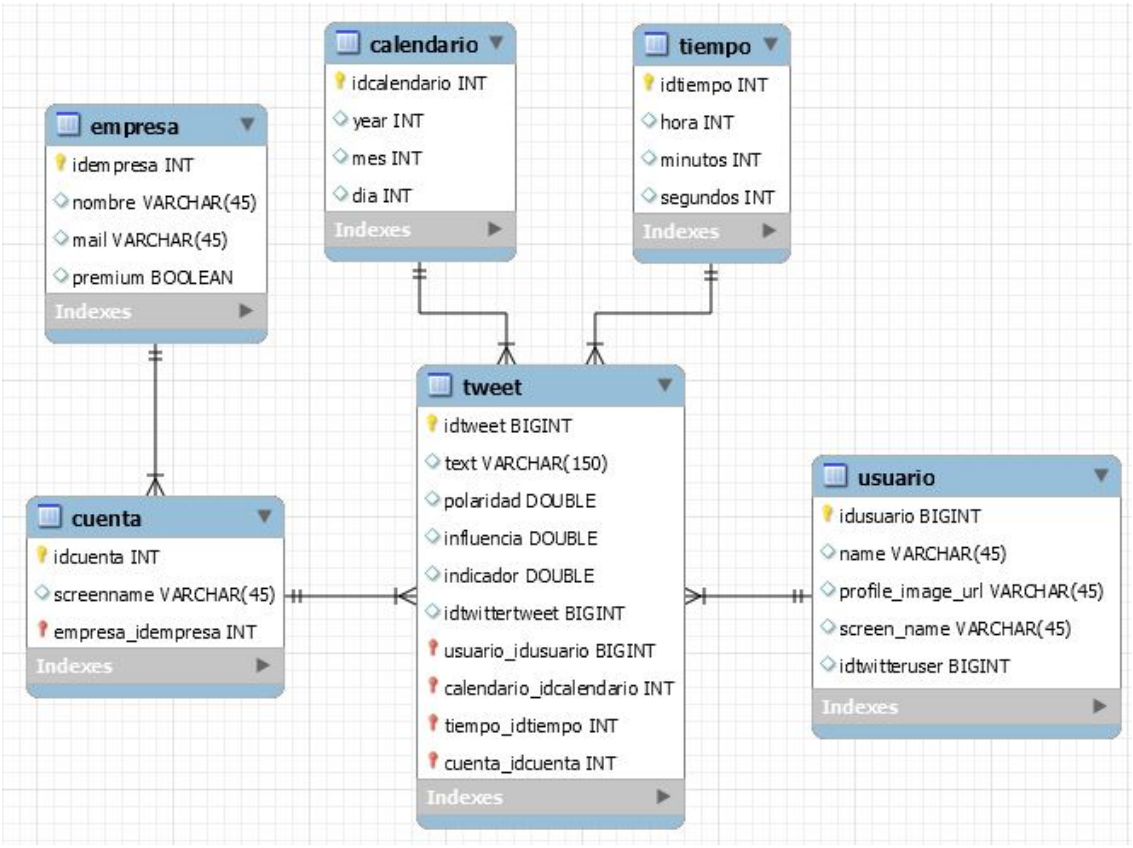
[52] Google. (2015) Optimización en buscadores. Consulta: 07 agosto 2015. [Online]. Available: https://support.google.com/webmasters/answer/35291?hl=es

Apéndice A

Modelo Estrella

En un comienzo se pensó un modelo Estrella. Este tipo de modelo tienen sólo una tabla de hechos, que en este caso es la tabla de *tweets*, como se puede ver en la figura A.1. A diferencia del modelo actual, la influencia en el modelo estrella era parte de los indicadores de la tabla de hechos.

Figura A.1: Modelo Estrella



Fuente: Elaboración Propia

Sin embargo, este modelo desconoce la importancia que tiene la dimensión 'usuario'. Al agregársele la dimensión 'interes', que no es dimensión de la tabla de hechos 'tweet', la tabla 'usuario'

pasó a ser una tabla de hechos, dejando el modelo como un modelo copo de nieve. Esta innovación duró poco, pues surgió el requerimiento de abrirse a consultas tales como la influencia de un usuario en el tiempo o la detección de múltiples intereses por usuario, lo que obligó a crear dos tablas de hechos más, de influencia (para analizarla en el tiempo) y de cruce entre la tabla 'usuario' e 'interes' (que resultó ser una normalización favorable a las consultas). Todas las tablas de hechos comparten las dimensiones 'calendario' y 'usuario'.

El modelo resultante es el llamado modelo constelación, de la figura 4.1 en el capítulo 4. En este prototipo no se realizan consultas por la evolución temporal de los intereses o de la influencia, dejando sin utilidad dichas relaciones. Sin embargo se tiene la posibilidad de hacerlo a futuro ya que los datos fueron cargados con este modelo. En este sentido, el actual prototipo tiene más potencial del que se aprovecha con las vistas explicadas en el capítulo 5.

Apéndice B

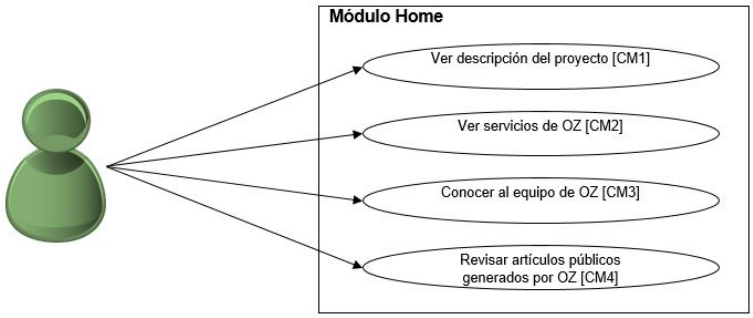
Casos de Uso por Módulo

A continuación se presentan las funcionalidades generales del sistema y como los actores intervienen en ellas. Para esto, se exhiben los casos de uso separados por módulos.

Módulo Home

El módulo Home tiene 4 casos de uso. Se de despliega al acceder a www.opinionzoom.cl.

Figura B.1: Caso de uso para el Módulo Home

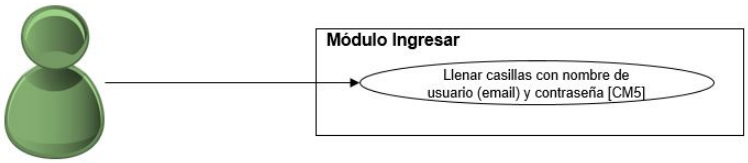


Fuente: Elaboración Propia

Módulo Ingresar

El módulo Ingresar tiene 1 caso de uso. Se ingresa a él oprimiendo el botón Entrar en la esquina superior derecha de la página Home.

Figura B.2: Caso de uso para el Módulo Ingresar

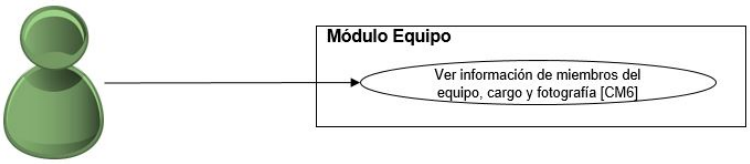


Fuente: Elaboración Propia

Módulo Equipo

El módulo Equipo tiene 1 caso de uso. Se ingresa a él desde la página principal, en la barra superior.

Figura B.3: Caso de uso para el Módulo Equipo

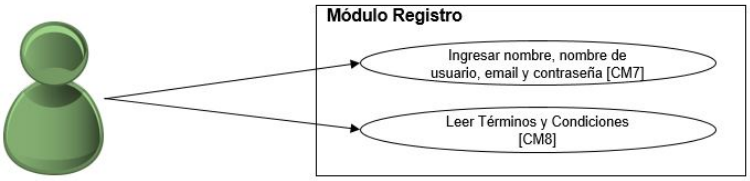


Fuente: Elaboración Propia

Módulo Registro

El módulo Equipo tiene 2 casos de uso. Este módulo se accede desde la vista del módulo Ingresar.

Figura B.4: Caso de uso para el Módulo Registro

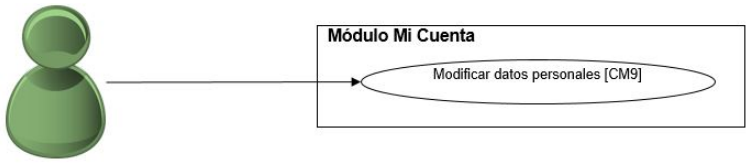


Fuente: Elaboración Propia

Módulo Mi Cuenta

El módulo Mi Cuenta tiene 1 caso de uso. Los usuarios clientes y administradores pueden acceder a este módulo desde el menú desplegable una vez iniciada la sesión.

Figura B.5: Caso de uso para el Módulo Mi Cuenta

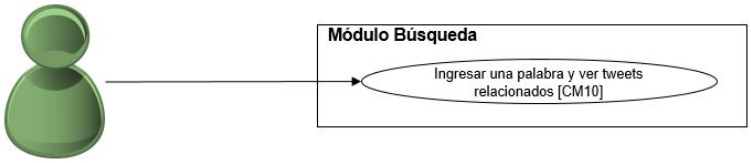


Fuente: Elaboración Propia

Módulo Búsqueda

El módulo Búsqueda tiene 1 caso de uso. Está ubicado en la barra superior que aparece luego de iniciar sesión.

Figura B.6: Caso de uso para el Módulo Búsqueda

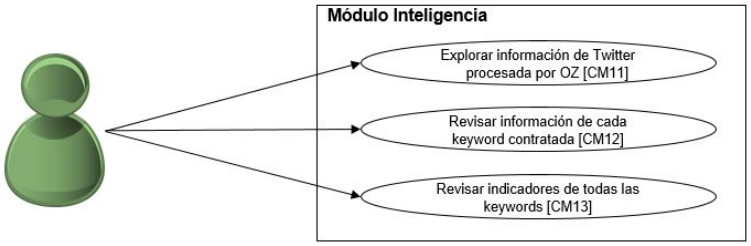


Fuente: Elaboración Propia

Módulo Inteligencia

El módulo Inteligencia tiene 3 casos de uso. Se accede a él con la barra lateral izquierda desplegada luego de iniciar sesión.

Figura B.7: Caso de uso para el Módulo Inteligencia

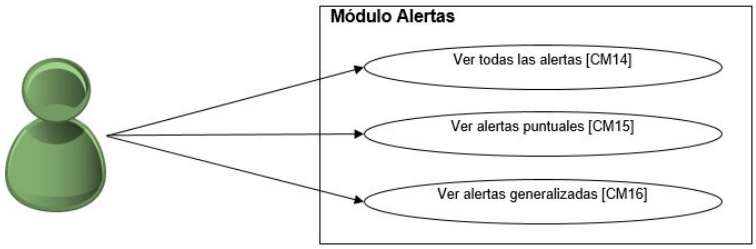


Fuente: Elaboración Propia

Módulo Alertas

El módulo Alertas tiene 3 alertas. Se accede a él con la barra lateral izquierda desplegada luego de iniciar sesión. En caso de que el usuario cliente tenga activado el servicio de inteligencia, este módulo se encuentra abajo del módulo de inteligencia.

Figura B.8: Caso de uso para el Módulo Alertas

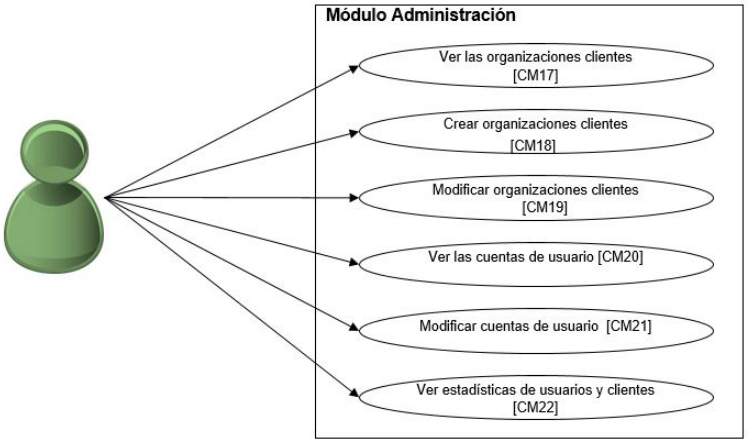


Fuente: Elaboración Propia

Módulo Administración

El módulo Administración tiene 5 casos de uso. El usuario administrador puede acceder a éste, luego de iniciar sesión, en la barra lateral izquierda.

Figura B.9: Caso de uso para el Módulo Administración



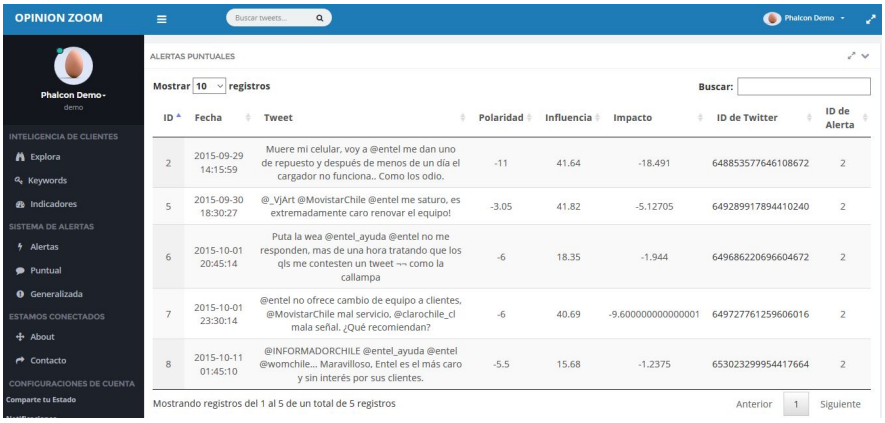
Fuente: Elaboración Propia

Apéndice C

Imágenes del sitio Web

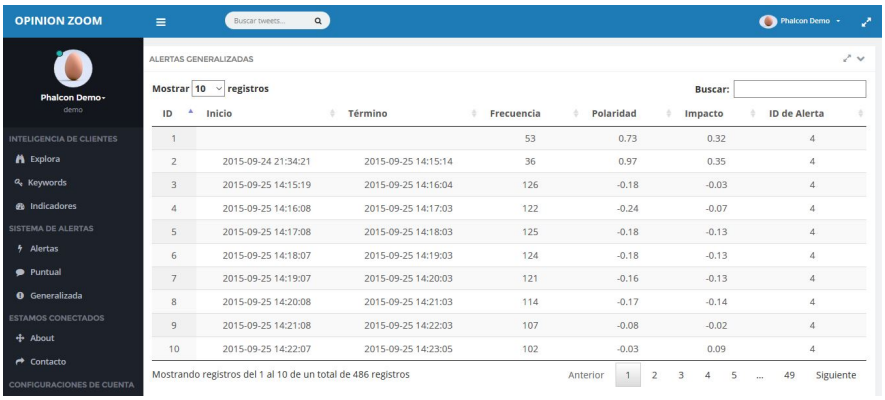
A continuación se pueden ver imágenes de funcionalidades claves de la página Web y otras imagenes sacadas de `www.opinionzoom.cl`

Figura C.1: Alertas Puntuales generadas



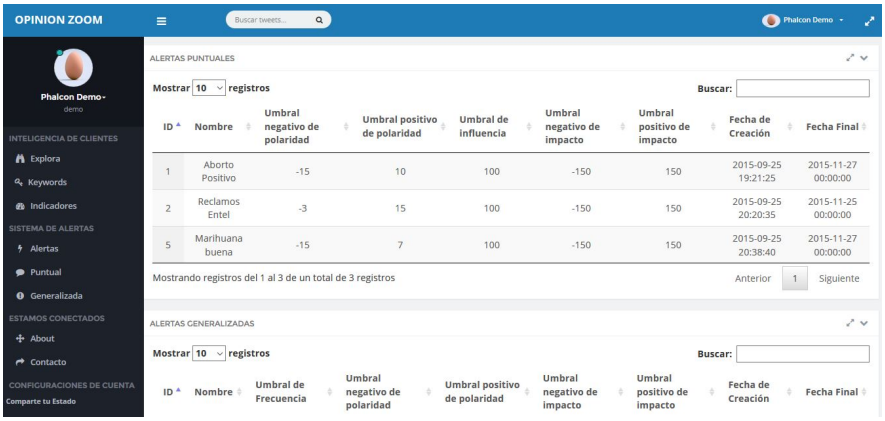
Fuente: www.opinionzoom.cl

Figura C.2: Alertas Generalizadas generadas



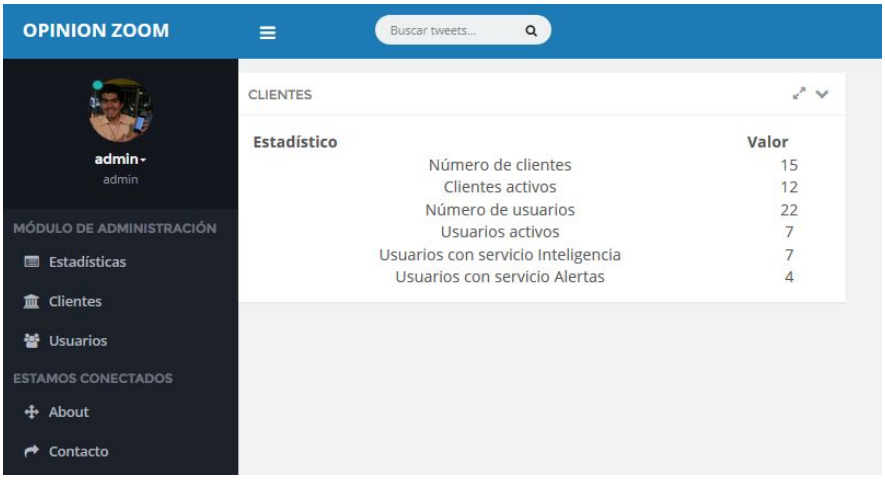
Fuente: www.opinionzoom.cl

Figura C.3: Alertas



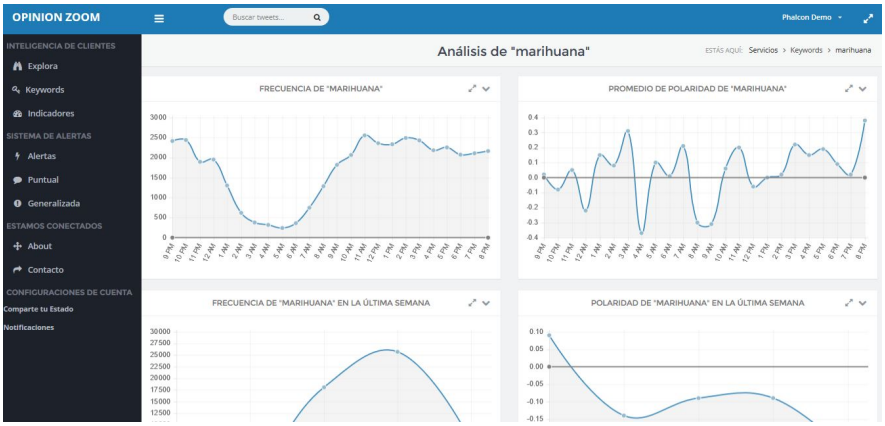
Fuente: www.opinionzoom.cl

Figura C.4: Estadísticas del Administrador



Fuente: www.opinionzoom.cl

Figura C.5: Inteligencia de Clientes - Keyword

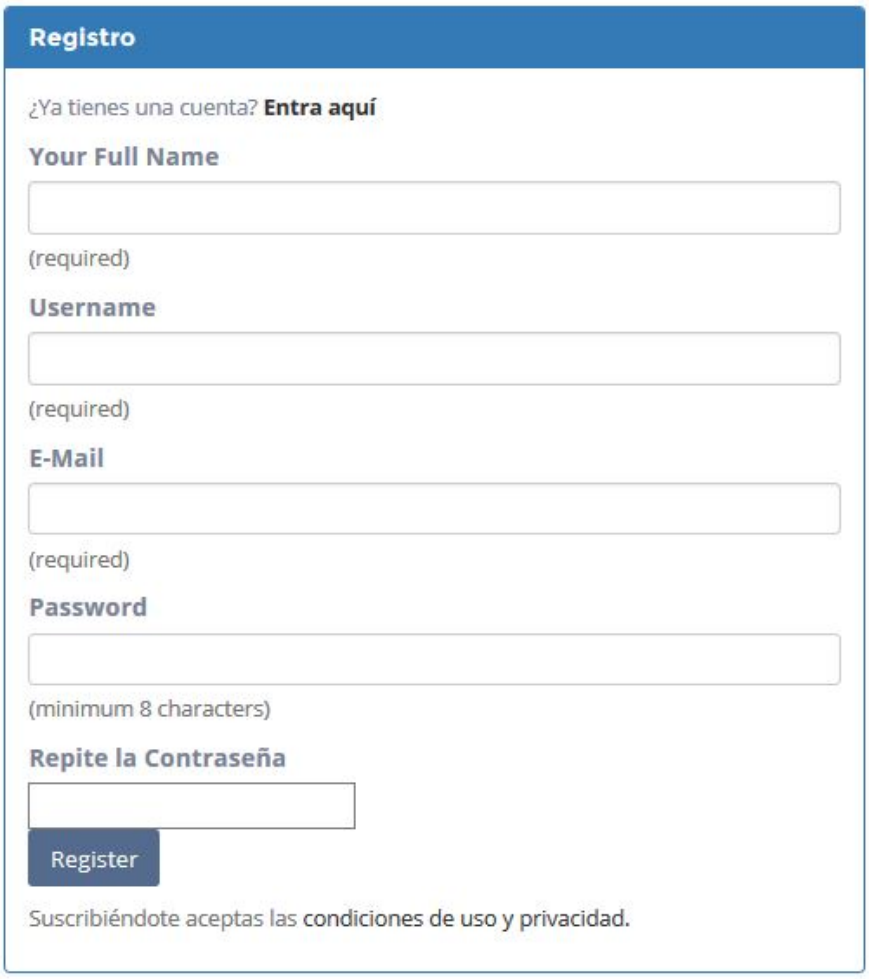


Fuente: www.opinionzoom.cl

Figura C.6: Inicio de sesión

Fuente: www.opinionzoom.cl

Figura C.7: Registro



Fuente: www.opinionzoom.cl

Respuesta a las Correcciones de la Comisión

1. Resumen Ejecutivo. Fue hecho nuevamente a partir de una nueva estructura:

    - Objetivo General
    - Contexto donde se desarrolla el problema
    - El problema
    - Hipotesis de investigación
    - La solución a implementar
    - Principal conclusión del trabajo desarrollado

2. Formulación de objetivos específicos. En el capítulo 1 los primeros 2 objetivos específicos fueron completados, para dar una mejor idea del trabajo a realizar.

3. Hipótesis de Investigación. No existía como tal, por lo que se incorporó al resumen ejecutivo, en la introducción y también fue considerada en conclusiones.

4. Indicadores. En el capítulo 4 fue agregado un párrafo que muestra el origen de los indicadores antes de explicarlos detalladamente.

5. Modelo Estrella. Fue agregado en los anexos para dar cuenta del proceso, de tal forma de entender mejor el camino recorrido para llegar a la solución final.

6. Propuesta de Valor. Se agrega un párrafo que explica el valor distintivo de la plataforma Web construida en el capítulo 6.

7. Casos de uso por módulo a anexos. Mientras el capítulo 6 mantiene los casos de uso por usuario, los casos de uso por módulo, que siguen siendo citados en el mencionado capítulo, son trasladados a anexos.

8. Corrección de errores de bibliografía. Algunas referencias estaban mal citadas, por configuraciones por *default* del archivo bibtex correspondiente. Otras estaban incompletas. Se corrigieron esos errores.

9. Revisión de errores. El escrito presentaba problemas de tipeo y pocos errores de ortografía, los que fueron revisados y eliminados.

10. Otros. El segundo capítulo se renombra como Marco Referencial, y en él se distingue el marco conceptual del teórico. También se hace referencia a todas las imágenes en el texto, lo que facilita su lectura y su visión en su versión digital. Por último, en el capítulo 7 se agregaron un par de sub-secciones para evaluar los resultados de las alertas generadas para un caso ficticio.