

PROTOTIPO DE UN CENTRO DE FUSIÓN DE DATOS DE REDES DE SENSORES
PARA LA AUTOMATIZACIÓN DE TOMA DE DECISIONES EN LA GESTIÓN DEL
RIESGO

Juan Andrés González Valderrama



UNIVERSIDAD PONTIFICIA BOLIVARIANA
POSTGRADOS ESCUELA DE INGENIERÍAS
MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN
MEDELLÍN
2015

PROTOTIPO DE UN CENTRO DE FUSIÓN DE DATOS DE REDES DE SENSORES
PARA LA AUTOMATIZACIÓN DE TOMA DE DECISIONES EN LA GESTIÓN DEL
RIESGO

JUAN ANDRÉS GONZÁLEZ VALDERRAMA

Trabajo de grado para optar al título de Magister en Tecnologías de la Información y la
Comunicación

Director

LEONARDO BETANCUR AGUDELO

PhD Ingeniería énfasis Telecomunicaciones



UNIVERSIDAD PONTIFICIA BOLIVARIANA

POSTGRADOS ESCUELA DE INGENIERÍAS

MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

MEDELLÍN

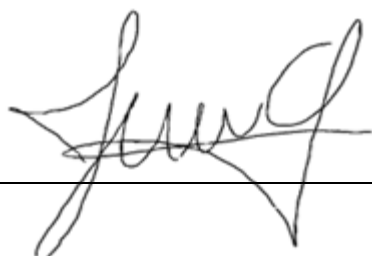
2015

21/05/2015

Juan Andrés González Valderrama

“Declaro que esta tesis (o trabajo de grado) no ha sido presentada para optar a un título, ya sea en igual forma o con variaciones, en esta o cualquier otra universidad” Art 82 Régimen Discente de Formación Avanzada.

Firma



NOTA DE ACEPTACION

Firma
Nombre
Presidente del jurado

Firma
Nombre
Presidente del jurado

Firma
Nombre
Presidente del jurado

Medellín, 21 de Mayo de 2015

A la memoria de mis padres y hermanos

AGRADECIMIENTOS

Deseo expresar un especial agradecimiento a mis padres y hermanos por el soporte y acompañamiento continuo a lo largo de mi vida académica. Con esfuerzo, disciplina y dedicación eventualmente habré de retribuírselos. Por el acompañamiento, orientación y permanente disposición durante la realización del presente, deseo manifestar mi agradecimiento al Doctor Leonardo Betancur.

Medellín 21 de Mayo de 2015

Juan Andrés González Valderrama

CONTENIDO

INTRODUCCIÓN	15
PLANTEAMIENTO DEL PROBLEMA	17
OBJETIVOS	19
Objetivo General	19
Objetivos Específicos	19
1. CONSIDERACIONES DE ARQUITECTURA	20
1.1. Redes de sensores	20
1.2. Computación en la nube	23
1.3. Big data	32
1.4. Inteligencia del negocio	36
2. DISEÑO E IMPLEMENTACIÓN	50
2.1. Consolidación de la arquitectura	50
2.2. Modelo transaccional	57
2.3. Modelo analítico	59
2.4. Diseño de ETL	61
2.5. Implementación en la nube	63
2.6. Elección de Explotación	66
3. PROPUESTA DE ANÁLISIS	71
3.1. Minería de datos	71
3.2. Parametrización de reglas de negocio	79
3.3. Visualización de datos	82
4. DISCUSIÓN DE RESULTADOS	85

4.1. Descripción del escenario	85
4.2. Lanzamiento de prueba	88
4.3. Análisis del escenario	91
4.4. Recomendaciones y Conclusiones	94
REFERENCIAS.....	98
ANEXOS	103

LISTA DE FIGURAS

Figura 1 Abstracción de funcionamiento	16
Figura 2 Representación de capa física	22
Figura 3 Disponibilidad por proveedor de servicios en la nube en 2013	26
Figura 4 Precios (US \$) por hora por, tipo de servicio de cómputo y plazo.....	29
Figura 5 Precios (US \$) por proveedor por Gb de almacenamiento mensual	30
Figura 6 Retos de Big Data	32
Figura 7 Funcionamiento MapReduce	35
Figura 8 Aproximación Top - Down	38
Figura 9 Aproximación Bottom -Up	39
Figura 10 Cuadrante Mágico de Gartner para BI	46
Figura 11 Ciclo de vida SCRUM.....	53
Figura 12 Planteamiento de arquitectura	55
Figura 13 Modelo transaccional 3NF.....	59
Figura 14 Modelo de estrella para el análisis	60
Figura 15 Implementación en la nube	65
Figura 16 Orquestación de servicios en la nube	66
Figura 17 Cubo creado para Pentaho	68
Figura 18 Reporte de cubo mondrian.....	68
Figura 19 Visualización gráfica de datos	69
Figura 20 Metodología CRISP-DM.....	72
Figura 21 Vista preparada de datos	73
Figura 22 Actividad de lanzamiento de alertas.....	80
Figura 23 Mapa de calor basado en muestras de sensores.....	83
Figura 24 Mapa de bolas usando muestras de sensores	83
Figura 25 Zona de riesgo implantada.....	86
Figura 26 Composición de muestras por tipo de variable	86
Figura 27 Distribución de muestras por sensor	87
Figura 28 Composición de grupos en la implantación de prueba.....	87
Figura 29 Reporte de alertas vía correo electrónico.....	90

Figura 30 Estadísticas por elección de sensores	91
Figura 31 Estadísticas por elección de sensores en el tiempo.....	92
Figura 32 Reporte resumido de mediciones en el tiempo	92
Figura 33 Análisis gráfico del escenario de prueba	93

LISTA DE TABLAS

Tabla 1 Descripción de proveedores en la nube	25
Tabla 2 Cantidad de instancias de computo por proveedor de servicios	25
Tabla 3 Clasificación de servidores web en la nube.....	27
Tabla 4 Clasificación de servidores de bases de datos en la nube	27
Tabla 5 Tecnologías de cómputo disponibles por proveedor	27
Tabla 6 Estado de la tecnología ofrecida por proveedor	27
Tabla 7 Características generales de servidores de cómputo en la nube	28
Tabla 8 Comparativo Bill Inmon y Ralph Kimball.....	40
Tabla 9 Comparativo entre tecnologías para administración de bases de datos ..	41
Tabla 10 Comparación de métodos para construcción de ETL.....	44
Tabla 11 Requerimientos por iteraciones	55
Tabla 12 Estructura de datos para la generación de muestras	88

GLOSARIO

Crowd sensing: Un nuevo paradigma de recolección de datos que aprovecha la información medida o generada por dispositivos móviles utilizados por personas comunes. La información generada es agregada y fusionada en la nube para extraer conocimiento que beneficie la prestación de servicios a la población [1].

Software as a Service (SaaS): Aplicaciones diseñadas para usuarios finales ofrecidas a través de la web. Su estructura de pago corresponde a la de un servicio, pago por uso [2].

Platform as a Service (PaaS): Conjunto de herramientas y servicios diseñados para facilitar la creación y distribución de aplicaciones como servicio de forma ágil y eficiente [2].

Infrastructure as a Service (IaaS): Son los componentes físicos que permiten el funcionamiento de todo lo demás. Servidores, almacenamiento, redes bajo una estructura de costos basado en el uso [2].

RESUMEN

El presente trabajo de grado de maestría exhibe el proceso de diseño e implementación de un prototipo funcional de un centro de fusión de datos en la nube para redes de sensores. La red de sensores, hace parte de un proyecto de mayor envergadura orientado a la gestión del riesgo sobre la red vial de Colombia llamado Prototipo TICS para el Monitoreo, Prevención y Atención de Deslizamientos de Tierra en la Red Vial de Colombia. Las teorías presentadas a lo largo del texto, revelan las ventajas y desventajas de las arquitecturas de diseño para bases de datos que han venido siendo implementadas en los últimos años, NoSQL y SQL. En este proyecto, la infraestructura para el desarrollo de la solución completa ha sido adquirida como servicio en la nube de uno de los proveedores más reconocidos en este campo. Asimismo, en cada etapa, se dan a conocer los criterios de selección del sistema implementado como prueba de concepto en términos de escalabilidad, rendimiento y costo como los principales de determinantes en tal decisión. El diseño resultante al igual que su implementación en uno de los proveedores de infraestructura disponibles en la nube, es entregado como una muestra a escala del funcionamiento real del sistema en las futuras condiciones reales, acompañado también de recomendaciones y discusiones acerca de posibles mejoras posteriores. Finalmente, se describe el diseño de una bodega de datos donde se aplican técnicas de inteligencia de negocios para la visualización, análisis y descubrimiento de patrones de alerta.

Palabras clave: Arquitectura de base de datos, Internet de las Cosas, SQL, NoSQL, Redes de Sensores.

ABSTRACT

This Master's degree work describes the process of designing and implementing a cloud-based functional prototype of a data fusion center for sensor networks. The sensor network itself is part of a larger project called "*Prototipo TICS para el Monitoreo, Prevención y Atención de Deslizamientos de Tierra en la Red Vial de Colombia*" aimed at managing risk on the road network in Colombia. The theories presented throughout the text reveal the advantages and disadvantages of design architectures for databases that have been implemented in recent years, NoSQL and SQL. The infrastructure for developing the complete solution shown in this work has been acquired as a cloud service from one of the most popular suppliers in this field. Also, at each step, the selection criteria is discussed in terms of scalability, performance and cost as major determinants in the decision. The resulting design as well as its implementation in one of the available infrastructure providers in the cloud is delivered as a scaled prototype of the real-operating-conditions system. Some discussions and recommendations about possible further improvements are stated. Finally, the design process of a data warehouse is described and an application of business intelligence techniques for visualization, analysis and discovery of patterns is exhibited.

Key Words: Cloud; Internet of Things; Database architecture; Crowd Sourcing; Sensor Networks.

INTRODUCCIÓN

Hoy en día el ser humano se ve expuesto a diversas situaciones inesperadas provenientes en ocasiones del hombre mismo y en otras de la naturaleza. En diferentes escenarios, la vida de algunos individuos se ha visto en peligro y en algunas otras situaciones, dependiendo de la magnitud de los eventos, incluso la de poblaciones enteras. Algunas emergencias se presentan de forma completamente aleatoria y pueden ser consideradas imposibles de prevenir, aunque en muchos casos tales situaciones podrían llegar a ser mejor tratadas [3]. La gestión del riesgo enfoca su esfuerzo en coordinar e integrar actores y herramientas en eventos de riesgo con el fin de mitigar consecuencias en hechos impredecibles y eliminar secuelas de situaciones predecibles en diversos contextos [4].

Las tecnologías de información han avanzado rápidamente durante los últimos años y han establecido soluciones de comunicación mundial mediante potentes plataformas de infraestructura. La nube es un concepto moderno que ha sido ampliamente difundido como una infraestructura extendida a nivel mundial que ha facilitado el acceso a servicios tecnológicos por costos acordes al uso. Lo anterior a su vez, ha generado nuevas tendencias tecnológicas mundiales como por ejemplo el termino Internet de las Cosas (IoT). Dicha idea evolucionó hasta hoy como un concepto que integra objetos e individuos del día a día mediante el uso de equipos especializados que permiten la medición y transmisión de muestras facilitando así el análisis remoto o local del estado físico de las cosas [5]. Tales aplicaciones, constituyen una gran ventaja en el campo de gestión del riesgo y es por esta razón un campo abierto de investigación y desarrollo a nivel mundial. Sin embargo, los volúmenes de información obtenidos a partir de aplicaciones de Internet de las Cosas como las redes de sensores, representan un reto enorme en tareas de diseño de bases de datos teniendo en cuenta las necesidades de disponibilidad, rendimiento, escalabilidad y seguridad de la información.

Este trabajo, hace parte de un proyecto de gestión del riesgo llamado “*Prototipo TICS para el Monitoreo, Prevención y Atención de Deslizamientos de Tierra en la Red Vial de Colombia*” que está siendo dirigido actualmente desde la ciudad de Medellín. Dicho proyecto está dividido en varias etapas que integran desde la recolección de muestras en campos implantados con sensores, hasta la visualización y análisis automatizado de la información en bases de datos

distribuidas en la nube. El presente trabajo se centra en el diseño e implementación de un prototipo de centro de fusión de datos en la nube para el análisis automático de muestras y el posterior lanzamiento de alertas como una etapa crítica dentro del proyecto de mayor envergadura mencionado líneas atrás.

Este trabajo de grado de maestría presenta los retos existentes en el proceso de diseño e implementación de una solución de almacenamiento de datos provenientes de redes de sensores distribuidas en áreas de estudio. Los requisitos de rendimiento y escalabilidad son el determinante principal en la toma de decisiones en cuanto a herramientas y tecnologías a utilizar así como los costos que tales elecciones representan en los proveedores de servicios existentes. Las comparaciones realizadas a lo largo del documento, involucran las tecnologías recientes de almacenamiento como SQL y NoSQL. Así mismo, se evalúan los costos asociados a la implementación del sistema en algunos de los proveedores más conocidos de servicios en la nube. La presentación abstracta del proyecto se presenta en la Figura 1.

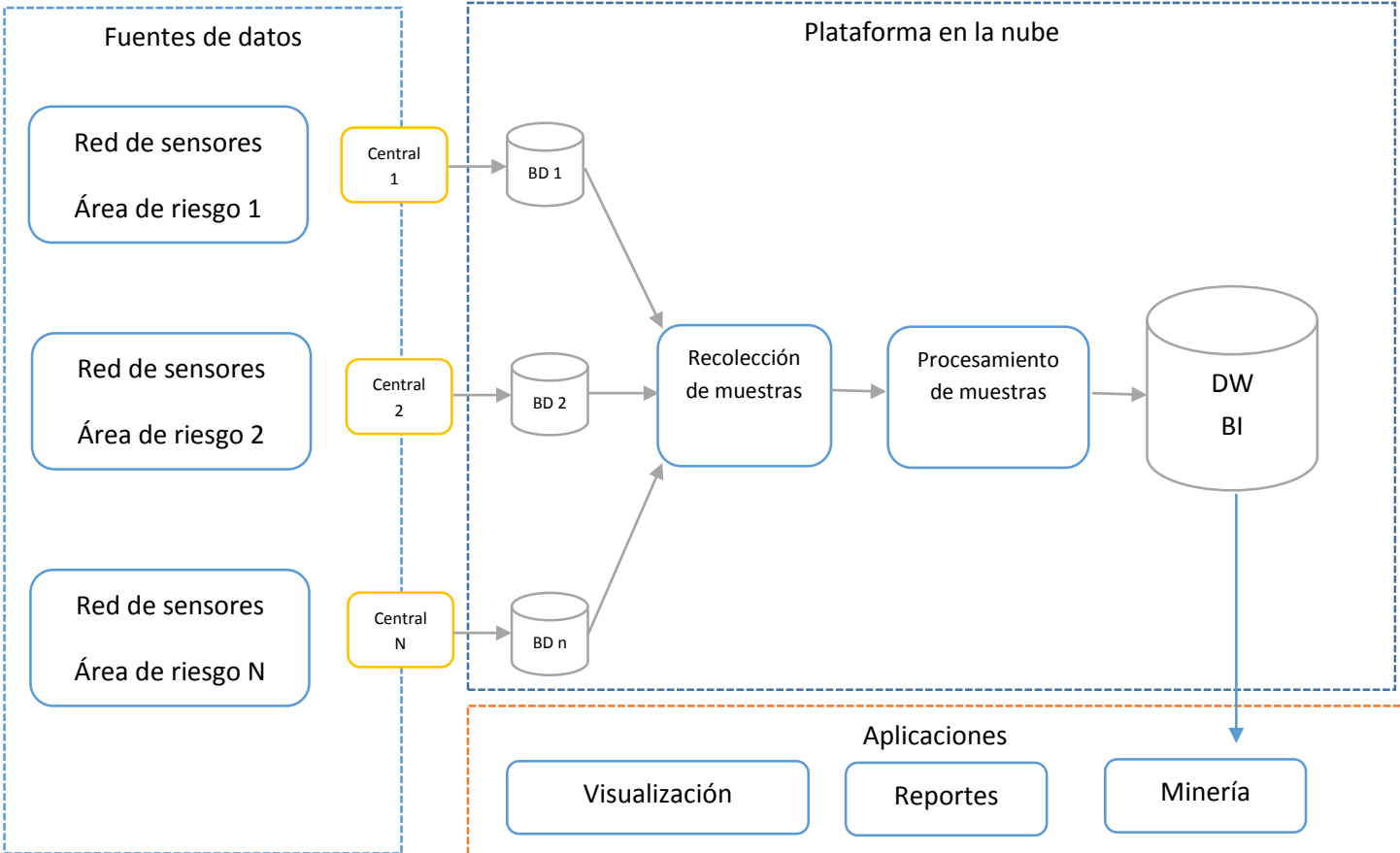


Figura 1 Abstracción de funcionamiento

PLANTEAMIENTO DEL PROBLEMA

En la actualidad, el ser humano se ve enfrentado frecuentemente a circunstancias imprevistas causadas en algunas ocasiones por la naturaleza y en otras por el hombre mismo. Sin bien es cierto que un sinnúmero de emergencias no pueden ser predichas por la gran cantidad de eventos fortuitos que las desencadenan, muchas de estas sí podrían ser mejor atendidas y tratadas. Sin embargo, la buena gestión de cada situación depende en gran medida de la correcta planeación y la ajustada sincronización de recursos que se inviertan durante el evento [3].

Para algunos proyectos orientados a la gestión del riesgo, la sincronización de esfuerzos que resulten eficaces tanto en la prevención como en la gestión de emergencias es el principal reto dentro del ciclo de vida del evento [4]. En dicho reto actualmente, la tecnología juega un papel trascendental dado las grandes prestaciones que otorga en temas de transmisión, almacenamiento y presentación de información. Sin embargo, en técnicas emergentes de recolección de datos para la gestión del riesgo como *Crowd Sensing*, el reto de transmisión y centralización de las muestras recolectadas para análisis subsecuentes en visualizaciones complejas resulta ser un proceso arduo teniendo en cuenta los altos requisitos de confiabilidad, credibilidad, pertinencia y oportunidad en la información que una situación catastrófica exige.

En el proyecto *Prototipo TICS para el Monitoreo, Prevención y Atención de Deslizamientos de Tierra en la Red Vial de Colombia* ejecutado en la ciudad de Medellín – Colombia, se pretende implementar una solución que incluye la implantación de redes de sensores en áreas de supervisión, con el fin de recolectar muestras sobre variables relevantes que puedan aportar elementos clave en predicciones adecuadas de incidentes probables. No obstante, de forma similar a los problemas percibidos en *Crowd Sensing*, dentro de la ejecución actual existe aún el reto de transmisión de los datos hacia un punto central de almacenamiento sobre el cual se puedan realizar análisis y visualizaciones claras que faciliten el proceso de toma de decisiones frente a un hecho [6]. Adicionalmente, dentro de los requerimientos de dicho proyecto se enfrentan un requisito no funcional que resulta ser altamente significativo para la viabilidad del mismo, la escalabilidad en la solución.

Este trabajo, mediante la aplicación de conceptos y criterios propios de las tecnologías de información y comunicaciones, pretende desarrollar e implementar un prototipo funcional de un sistema de recolección y almacenamiento estructurado

de datos llamado centro de fusión de datos. El producto final permitirá completar satisfactoriamente transmisiones de datos provenientes de las fuentes instaladas hacia un sistema central que contará con el diseño adecuado para cumplir tanto con los requisitos funcionales como los no funcionales.

OBJETIVOS

Objetivo General

Desarrollar un prototipo funcional de un centro de fusión de datos para la recolección, almacenamiento, visualización y análisis de la información capturada en una o múltiples redes de sensores implantadas para la gestión del riesgo.

Objetivos Específicos

- Diseñar la arquitectura de un centro de fusión de datos que permita la realización de análisis de información para la toma de decisiones automatizadas en la gestión del riesgo.
- Caracterizar el procedimiento necesario para completar la implantación de la arquitectura de un centro de fusión de datos en un ambiente funcional de muestra.
- Implementar la arquitectura diseñada a nivel de prototipo de prueba de concepto los procesos de adquisición y visualización de datos en un centro de fusión.
- Proponer técnicas básicas de análisis de información para la automatización de toma de decisiones a partir de las observaciones inferidas sobre el centro de fusión.

CAPITULO 1

1. CONSIDERACIONES DE ARQUITECTURA

1.1. Redes de sensores

Las redes de sensores, conocidas como y nombradas a partir de este punto *WSN* por sus siglas en inglés (Wireless Sensor Networks), son agrupaciones de sensores que permiten la extracción de datos del ambiente donde se encuentran ubicadas. Sus aplicaciones son múltiples y varían desde el control de procesos productivos, el control de ciudades inteligentes y muchas otras [7], entre las cuales se resalta la gestión del riesgo dado que representa un interés especial en este trabajo.

Los dispositivos electrónicos que hacen posibles las aplicaciones en diferentes ambientes, varían en diversos aspectos tales como en su tipo, construcción, capacidades, variables de estudio entre muchas otras. Sin embargo, una característica común entre aquellos usados en aplicaciones de *WSN* es la capacidad de transmisión de datos inalámbricamente. Como consecuencia, las redes inalámbricas de sensores han revolucionado la forma en que se recolectan datos de lugares remotos y ha disparado el volumen de información almacenado posterior a su implantación. Tal recolección, es un vínculo que se establece entre ambientes de interés e internet y que ha dado origen a un término conocido como *IoT* por sus siglas en inglés (Internet of Things).

El concepto *IoT* ha sido ampliamente difundido a través de la red mundial y es un campo de investigación actual bajo el cual un sinnúmero de aplicaciones han sido desarrolladas y continúan avanzando de la mano de otras concepciones emergentes. Una aplicación que ha tomado fuerza en una parte importante del primer mundo y que incursiona fuertemente en países en vía de desarrollo es el de ciudades inteligentes [8]. En esta aplicación, se hace uso extensivo de *WSN* en conjunto con nuevas plataformas en internet que hacen posible la recolección, almacenamiento,

transformación y todos los desarrollos que sobre esta información puedan ser concebidas.

Este trabajo como etapa del proyecto “*Prototipo TICS para el Monitoreo, Prevención y Atención de Deslizamientos de Tierra en la Red Vial de Colombia*” no pretende ahondar en los aspectos de diseño, ni implantación de las redes sensoriales así como tampoco pretende explicar los criterios de selección de dispositivos. Dichas tareas son el alcance de otros sub proyectos que hacen las veces de actividades predecesoras de este trabajo. No obstante, en el alcance actual si se pretende consolidar las herramientas necesarias para el envío de las mediciones recogidas en sitios remotos hacia la plataforma en la nube. Un esquema general de implementación en la capa física es ilustrado en la Figura 2.

Dadas las pretensiones de este trabajo en la tarea de hacer posible el envío de muestras recolectadas, se tienen en cuenta algunos aspectos y supuestos relevantes que deben ser considerados en la definición de la arquitectura:

- Cada WSN está constituida por un número finito de sensores implantados en zonas de interés para la gestión del riesgo.
- Cada sensor puede transmitir sus mediciones hacia un sistema de almacenamiento local por algún protocolo de comunicaciones que podría ser o no ser internet (IP).
- El sistema de almacenamiento local o también conocido como *Gateway* no hace parte del diseño de la arquitectura de este trabajo, es parte de otra etapa previa donde se define la forma de almacenamiento y los mecanismos de envío entre los sensores y dicho sistema.
- La etapa de muestreo y envío hacia el *Gateway* es referido en este trabajo de forma general como capa física.
- El *Gateway* permite establecer conexiones a internet y manejar solicitudes en el protocolo HTTP.
- El tipo de variables medidas y los dispositivos usados son independientes del centro de fusión de datos. Lo anterior implica que la arquitectura debe tener flexibilidad en la configuración de los diferentes escenarios que se deseen medir.

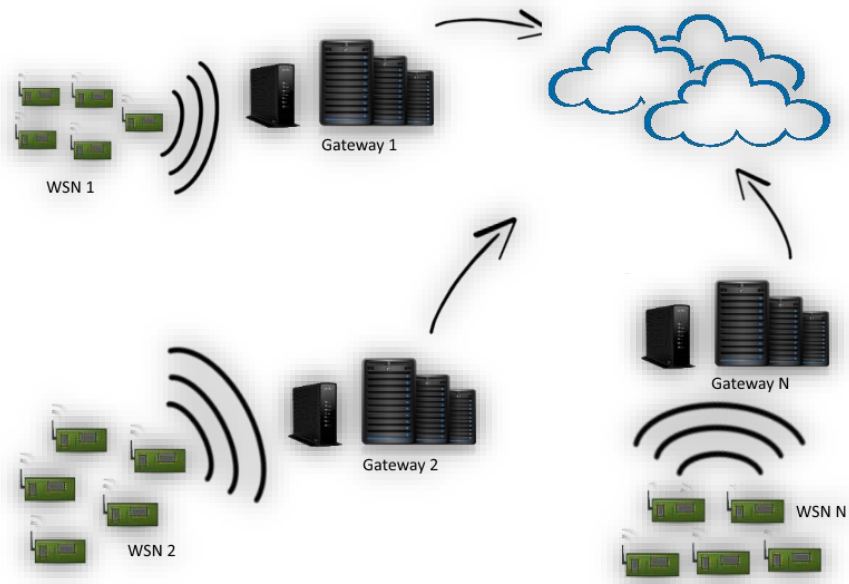


Figura 2 Representación de capa física

1.2. Computación en la nube

Las redes de sensores, son fuentes de datos que implican grandes retos dados los volúmenes de generación que pueden llegar a alcanzar [5]. Dependiendo de la frecuencia con que se generan las mediciones, una WSN puede requerir sistemas transaccionales de alto rendimiento diseñados para soportar alta concurrencia, disponibilidad cercana al cien por ciento del tiempo, y escalabilidad económicamente sostenible que permita en el transcurso de los años mantener repositorios históricos [9]. Las características mencionadas, son requerimientos hoy en día comunes que podrían ser solucionados de forma efectiva en términos de costos con las herramientas ofrecidas por algunos proveedores de servicios en la nube [10].

La computación en la nube es un concepto que hace posible la adquisición de herramientas de cómputo, almacenamiento, redes y aplicaciones entre otros, pagando de acuerdo a su uso. Adicionalmente, todos los servicios que puedan ser adquiridos son administrados completamente por los proveedores haciendo posible la eliminación de costos fijos. En consecuencia, la implementación y gestión de tales elementos como servicios son de fácil acceso teniendo en cuenta que la mayoría de los proveedores cuenta con documentación extensiva e interfaces de usuario que disminuyen la dificultad de interacción.

Actualmente, los servicios que pueden obtenerse en la nube son de plataforma PaaS por sus siglas en inglés, de infraestructura IaaS y de software SaaS [2]. En conjunto, los servicios mencionados ofrecen herramientas para la creación de sistemas que deben ser pagados de acuerdo a su uso, que pueden crecer de acuerdo a las necesidades que surgen con el tiempo y que tienen alta disponibilidad dado que todos se encuentran soportados por una extensa infraestructura compartida que se distribuye globalmente. Otro aspecto que resulta atractivo del tipo de servicios en la nube está relacionado con el licenciamiento y el mantenimiento; Ambas implicaciones conllevan costos que tradicionalmente deshabilitan el desarrollo de proyectos como pruebas de concepto, los cuales en sus etapas iniciales de evaluación financiera resultarían no ser viables si se hicieran en un entorno propio. Sin embargo, los proveedores de servicios en la nube han logrado eliminar las barreras de entrada a infraestructura de alto rendimiento y disponibilidad a la vez que excluyen la

necesidad de mantenimiento y licenciamiento que previo a su existencia eran requeridos [11].

Algunos proveedores ya posicionados y reconocidos a nivel mundial por sus servicios de computación en la nube como Amazon Web Services [12], Microsoft Azure [13], Google Cloud Platform [14] y Rackspace [15] entre algunos otros ofrecen elementos comunes que pueden ser efectivos en términos de costos dado su cobro por uso [5]. Teniendo en cuenta que el presente trabajo busca alcanzar una prueba de concepto que garantice la escalabilidad y óptimo rendimiento de la solución, se consideran importantes las características ofrecidas por dichos proveedores en estos términos. Entre los aspectos relevantes para la comparación de los proveedores disponibles en el mercado, se destaca la importancia de las prestaciones obtenidas en los siguientes rasgos a la luz de escalabilidad, rendimiento y costo: Computación, almacenamiento, disponibilidad, experiencia de usuario. Previo a la comparación, se hace una breve descripción de los proveedores de acuerdo a lo presentado en [16]:

Nombre	Descripción
<p>Amazon Web Services (AWS)</p> 	<p>AWS es una plataforma de servicios en la nube lanzado al mercado en 2006. Ofrece servicios de cómputo, almacenamiento, seguridad, redes, distribución de contenido, bases de datos (SQL y NoSQL), Análisis en Hadoop [17], Virtualización de máquinas y otros.</p>
<p>Google Cloud Platform</p> 	<p>Google ingresó a la prestación de servicios en la nube en 2008. Ofrece servicios de cómputo, bases de datos no relacionales, plataforma de desarrollo de aplicaciones, y análisis de datos.</p>



<p>Microsoft Azure</p> 	<p>Microsoft presta servicios en la nube desde 2008. Inicialmente se prestaban servicios de plataforma como servicio para desarrollo .Net. Sin embargo, actualmente soporta bases de datos, redes, cómputo entre otros.</p>
<p>Rackspace Cloud</p> 	<p>Rackspace presta servicios en la nube desde 2008. Ofrece herramientas de cómputo, almacenamiento, redes entre otros.</p>

Tabla 1 Descripción de proveedores en la nube

Es de significar que las comparaciones realizadas a continuación, son solamente de acuerdo a algunos detalles representativos para la creación de la prueba de concepto. Este documento no pretende ahondar en los procesos comparativos ni pretende abarcar todos los ámbitos que entre estos proveedores puedan existir. Se soporta la comparación de acuerdo con [16], [11], [13]–[15].

En cuanto a escalabilidad, tanto horizontal como vertical, inicialmente se puede decir que todos los proveedores de servicios ofrecen condiciones suficientes para el incremento de infraestructura de cómputo y almacenamiento para concebir la prueba de concepto y en un futuro la solución real misma. La tabla construida a partir de datos en [16], presenta la cantidad de instancias de cómputo y su tipo que pueden tenerse simultáneamente por cada proveedor.

Tamaño de Instancia	Amazon EC2	Google	Microsoft Azure	Rackspace
1 core / 2 GB	20 / región	24 / región	20	20 diariamente
2 cores / 4 GB	20 / región	12 / región	10	21 diariamente
4 cores / 8 GB	20 / región	6 / región	5	22 diariamente
8 cores / 16 GB	20 / región	3 / región	2	23 diariamente
16 cores / 32 GB	20 / región	1 / región	1	24 diariamente

Tabla 2 Cantidad de instancias de computo por proveedor de servicios

Por otra parte, en términos de disponibilidad, los proveedores de servicios analizados en este trabajo, cumplen con los requerimientos no funcionales ya que en todos los casos se evidencia un porcentaje de tiempo en línea superior al 99%. No obstante, de acuerdo con [18] se destaca el rendimiento de funcionamiento en las opciones prestadas por Amazon y Google con cien por ciento del tiempo en servicio en 2013 de acuerdo con la Figura 3.

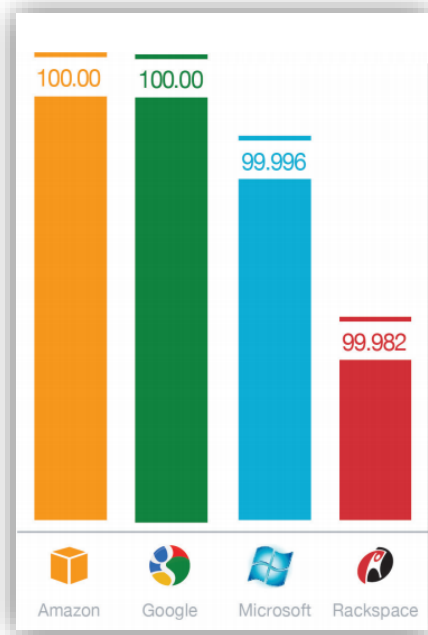


Figura 3 Disponibilidad por proveedor de servicios en la nube en 2013. Imagen modificada de <http://www6.nasuni.com/rs/nasuni/images/Nasuni-White-Paper-State-of-Cloud-Storage-2013.pdf>

Habiendo analizado los criterios de escalabilidad y disponibilidad para los proveedores en la nube estudiados, se presentan a continuación las variables de decisión restantes para la selección.

- **Cómputo**

La comparación de los servicios de cómputo en la nube ofrecidos por los proveedores analizados se lleva a cabo de acuerdo al análisis de instancias de tamaños comparables presentados en [16]. Los tipos de servidor de interés en este trabajo son de tipo de base de datos y de aplicaciones web. En la Tabla 3 y Tabla 4 presentan una clasificación general de las máquinas comúnmente encontradas en estos proveedores y que pueden ser compradas entre sí.

Tipo de servidor Web	Núcleos	Memoria	Almacenamiento
Pequeño	2	2-4 GB	Local
Medio	4	4-8 GB	Local
Grande	8	8-16 GB	Local

Tabla 3 Clasificación de servidores web en la nube

Tipo de servidor Base de datos	Núcleos	Memoria	Almacenamiento
Pequeño	4	8-16 GB	50 GB
Medio	8	16-32 GB	100 GB
Grande	16	32-64 GB	200 GB

Tabla 4 Clasificación de servidores de bases de datos en la nube

Así mismo a manera de contexto, la Tabla 5 presenta los tipos de tecnología ofrecidos por cada proveedor y las posibles configuraciones que pueden lograrse en cada uno, en tanto que la Tabla 6 presenta brevemente el estado de la tecnología de algunos tipos de instancia.

Proveedor	Tecnología	Variedad
Amazon Web Services	EC2 - Xeon	13 clases
Google Cloud Platform	Sandy Bridge	Standard, high memory, high CPU
Microsoft Azure	AMD	Standard, performance
Rackspace	AMD	Standard, memory intensive, CPU Intensive

Tabla 5 Tecnologías de cómputo disponibles por proveedor

Proveedor servidor base de datos	Tipo de instancia	Procesador	Fecha Tecnología
Aamazon EC2	T2.medium	Intel Xeon E5-2670 v2 2.50GHz	Q3 2013
Aamazon EC2	c3.xlarge	Intel Xeon E5-2680 v2 2.80GHz	Q3 2013
Google	n1-standard-4	Intel Xeon 2.60GHz	Q1 2012
Microsoft Azure	Medium A2	AMD Opteron 4171 HE	Q2 2010
Rackspace	Performance 2 15 GB	Intel Xeon E5-2670 2.60GHz	Q1 2012

Tabla 6 Estado de la tecnología ofrecida por proveedor

Dado que el propósito de este proyecto es consolidar una aplicación escalable que pueda adaptarse apropiadamente a las variaciones en los volúmenes de datos provenientes de las redes de sensores que sean configuradas, es necesario resaltar que el panorama actual según lo

revisado hasta este punto indica que verticalmente se podría lograr un crecimiento en tres etapas, pequeño, mediano y grande según la Tabla 3 y Tabla 4. En tanto que horizontalmente se podrían lograr los incrementos en número de instancias presentados en la Tabla 2.

No obstante, teniendo en cuenta que el alcance propuesto se limita a la construcción de una prueba de concepto, se tiene especial interés por los resultados asociados a pequeñas instancias. Del mismo modo, se valoran aquellas opciones sin costo disponibles en algunos proveedores que podrían hacer posible la realización de todo el sistema libre de gastos.

A continuación, en la Tabla 7 se presentan las características básicas de los servidores pertenecientes a la categoría “pequeño” ofrecidos en la nube por los diferentes proveedores.

Proveedor servidor Web	Tipo de instancia	CPU	Memoria	Almacenamiento
Aamazon EC2	t2.small	1 vCPU	2 GB	Fuera de la máquina
Aamazon EC2	t2.medium	2 vCPU	4 GB	Fuera de la máquina
Google	n1-highcpu-2	2 Cores	1.8 GB	Fuera de la máquina
Microsoft Azure	Medium A2	2 Cores	3.5 GB	Fuera de la máquina
Rackspace	Performance 1	2 vCPU	2 GB	20 GB SSD interno

Tabla 7 Características generales de servidores de cómputo en la nube

La Figura 4 evidencia el comparativo en precios de los diferentes tipos de instancia ofrecidos por los proveedores de servicios en la nube. Se debe resaltar que las ofertas de AWS varían de acuerdo a los plazos pactados para el uso de las instancias. Por ejemplo, en su tipo de plan “Por demanda” se encuentran los precios más altos dado que el uso de las maquinas puede interrumpirse en cualquier momento y de esta manera sus costos también se detienen. Por otra parte, AWS ofrece instancias reservadas a 3 años disminuyendo el costo por hora en un 50% con respecto a su contraparte por demanda.

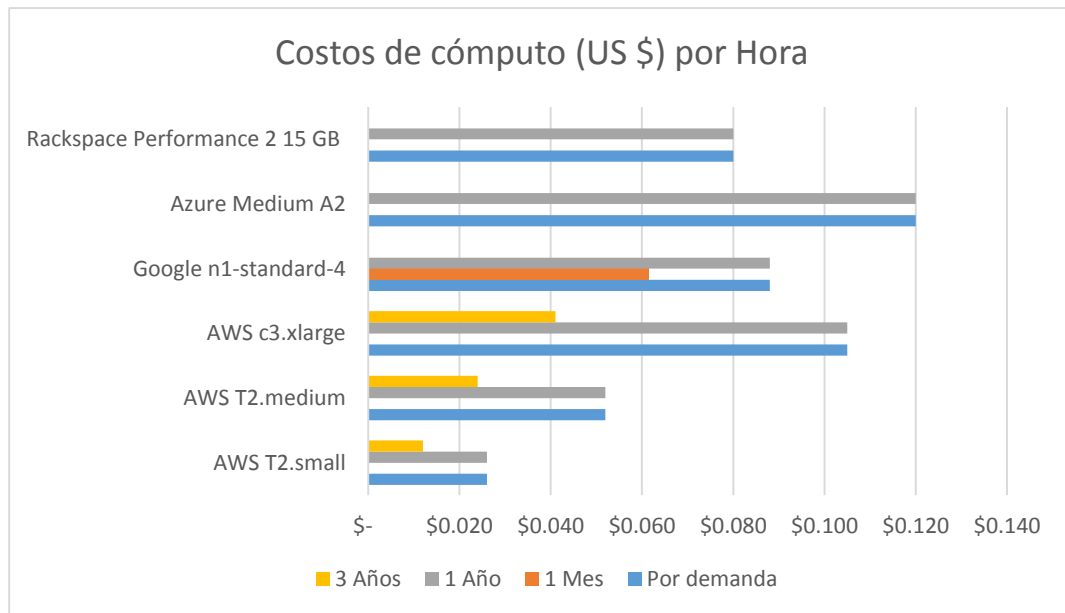


Figura 4 Precios (US \$) por hora por, tipo de servicio de cómputo y plazo. Imagen modificada de <http://www6.nasuni.com/rs/nasuni/images/Nasuni-White-Paper-State-of-Cloud-Storage-2013.pdf>

- Almacenamiento

Los servicios de almacenamiento encontrados en los proveedores de interés presentan similitud en su forma de funcionamiento y en su capacidad de escalamiento. El servicio de almacenamiento que se analiza está relacionado con la persistencia de objetos de diferentes tipos, es decir, son herramientas integradas a otros servicios y permiten la gestión de información de diversas tipologías en diversos formatos, desde archivos planos CSV hasta imágenes y videos.

Este servicio es vital en la construcción de la prueba de concepto teniendo en cuenta que su presencia habilita la existencia de datos semiestructurados y no estructurados así como el procesamiento en paralelo de los mismos. Se considera dentro de los criterios de diseño ya que mediante este elemento se desacoplan los sistemas de almacenamiento tradicionales y se da flexibilidad o lo que podría llegar a ser la implementación futura del centro de fusión de datos.

Luego de revisar los servicios de almacenamiento de objetos en la nube ofrecidos por cada uno de los proveedores y teniendo en cuenta que los detalles de evaluación del rendimiento de lectura, edición y eliminación están por fuera del alcance, se observan diferencias leves entre los servicios Microsoft Azure Blob, Amazon Web Services S3 y Google Cloud Storage,

dejando al proveedor Rackspace como el más costoso en este ámbito. Los resultados se presentan en la Figura 5

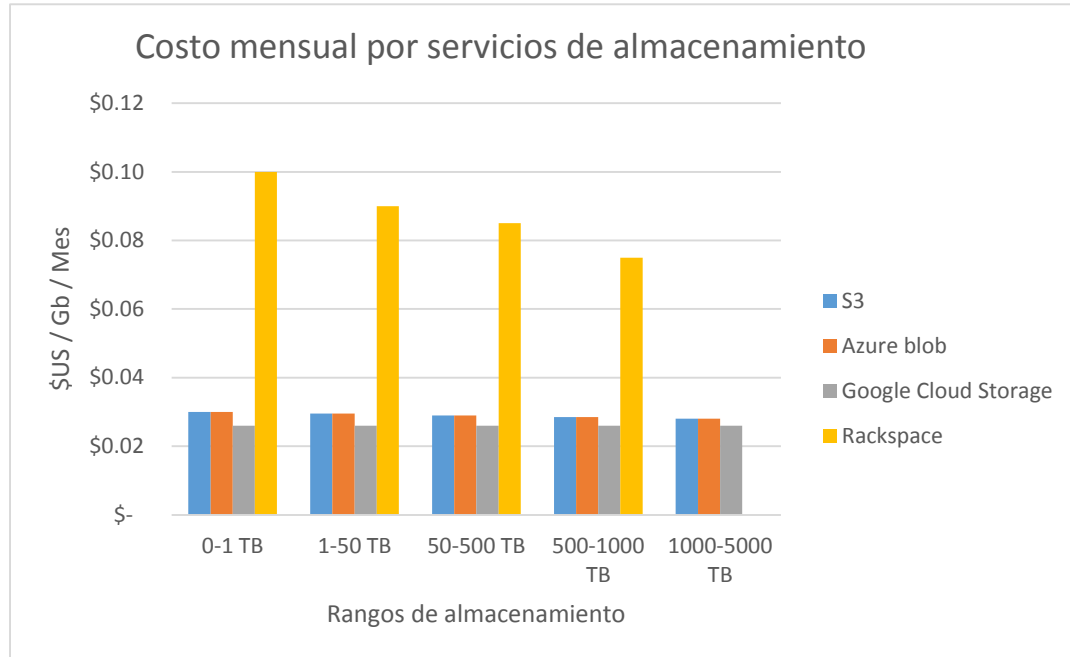


Figura 5 Precios (US \$) por proveedor por Gb de almacenamiento mensual. Imagen modificada de <http://www6.nasuni.com/rs/nasuni/images/Nasuni-White-Paper-State-of-Cloud-Storage-2013.pdf>

- Experiencia de usuario

Al presentar la información comparativa entre los proveedores de servicios en la nube actualmente disponibles, se deben aclarar algunas condiciones. No existen medidas absolutas entre los tipos de servicios dada la alta variabilidad presente entre las tecnologías utilizadas, tipo de virtualización y ubicaciones físicas entre algunos otros. Por tal motivo, este trabajo solo presenta un paralelo entre aquellas tecnologías que resultan similares en su composición y que de alguna manera pueden ser medidas entre sí [16].

Un aspecto que resulta ser significativo en la decisión del proveedor de servicios a elegir para llevar a cabo la etapa subsecuente de diseño e implementación de la arquitectura apropiada para el proyecto, tiene que ver con las promociones de inicio para nuevos usuarios. Por ejemplo, Amazon ofrece muchos de sus servicios, con características básicas, gratis por un periodo de 12 meses. Lo anterior implica que la mayoría de los servicios requeridos para implementar la prueba de concepto podrían ser adquiridos y evaluados sin costo alguno usando dicho proveedor.

Adicionalmente, otro criterio de selección que resalta por su importancia tiene que ver con la facilidad de uso para la implementación de la solución que debe ser creada para cumplir con los requerimientos funcionales y no funcionales. En este ámbito, Amazon presenta una ventaja relevante dado que sus opciones de interacción son múltiples, desde integraciones con IDE de programación como Visual Studio, Eclipse y PyDev hasta interfaces de usuario en la web que hacen posible la construcción de sistemas de forma intuitiva con componentes modernos.

Finalmente, la elección del proveedor de servicios en la nube se inclina por Amazon Web Services a la luz de los conceptos previamente expuestos y tomando en consideración otro argumento destacable. Los servicios disponibles en AWS actualmente, incluyen herramientas de integración entre Hadoop y motores bases de datos relacionales de varios tipos tales como Oracle, Microsoft SQL Server, MySQL y PostgreSQL. Este hecho, es de alta relevancia dado que se ajusta adecuadamente a los requerimientos de la solución que busca implementarse. En adición, la flexibilidad percibida en dicho proveedor para lograr trabajos de automatización centrados en datos que hacen posible las tareas de extracción, transformación y carga es pues el aspecto con mayor peso para su elección.

1.3. Big data

El término *Big Data* ha sido difundido globalmente como una de las tendencias más importantes en el campo de tecnologías de información, IT, por sus siglas en inglés. Su definición ha sido asociada a grandes y complejas colecciones de datos que podrían contener sets estructurados, semiestructurados y no estructurados que resultan difíciles de almacenar y analizar en sistemas de almacenamiento tradicionales como las bases de datos relacionales [19]. Los retos que actualmente se enfrentan en este campo tienen que ver con las técnicas utilizadas para capturar, almacenar, buscar, analizar y visualizar la información de forma eficiente y escalable. De igual manera, la representación de tales retos se ha materializado en un conjunto de términos que describen el contexto. Actualmente, los términos asociados a *Big Data*, las 5 V, que mejor definen el entorno se conocen como Velocidad, Variedad, Volumen, Veracidad y Variabilidad. No obstante, se resalta que fundamentalmente las primeras tres fueron la definición asociada con los retos referentes a *Big Data* y a partir de allí nuevos términos fueron añadidos de acuerdo con las necesidades emergentes. Una descripción gráfica aproximada de los fundamentos de este campo se muestra en la Figura 6.



Figura 6 Retos de Big Data. Imagen tomada del sitio <http://sg.com.mx/images/stories/sg35/BigDataimage01.png>

El tema de esta sección se encuentra en desarrollo actualmente a nivel mundial. Su evolución ha sido impulsada por grandes compañías como Facebook, Yahoo, Google entre muchas otras que han enfrentado las dificultades de procesamiento y almacenamiento de volúmenes altos de información [20], [21], [22]. No obstante, es una gran oportunidad y un gran

reto considerar el uso de últimas tendencias tecnológicas en un proyecto que puede llegar a tener un uso extensivo a nivel nacional como lo es “Prototipo TICS para el Monitoreo, Prevención y Atención de Deslizamientos de Tierra en la Red Vial de Colombia”.

Para abordar el concepto de Big Data y sus implicaciones, se debe empezar por resumir los componentes del ecosistema llamado *Hadoop*. Para esto, se soportan los siguientes puntos con lo explicados en [23],[24]. *Hadoop* es un proyecto de código abierto originado por Apache y está compuesto por otros sub proyectos que en conjunto dan origen a lo que se conoce como el ecosistema. Los componentes presentes se describen brevemente a continuación:

- *Common:*
Componentes e interfaces comunes para entrada y salida en el sistema de archivos distribuido HDFS [23].
- *Avro:*
Sistema de serialización para el almacenamiento eficiente de información [23].
- *MapReduce:*
Modelo de procesamiento de datos distribuido que hace posible la ejecución de algoritmos en un clúster de computadores [23].
- *HDFS:*
Sistema de archivos que se ejecuta sobre un clúster grande de computadores. Sus siglas en inglés significan Hadoop Distributed File System [23].
- *Pig:*
Lenguaje de programación como flujo de datos para explorar grandes conjuntos de datos. Corre sobre HDFS como abstracción de MapReduce [23].
- *Hive:*
Bodega de datos distribuida. Permite la administración de datos almacenados en HDFS y permite la ejecución de comandos similares a SQL en un lenguaje llamado HQL por sus siglas en inglés (Hive Query Language) [23].
- *HBase:*

Base de datos distribuida, orientada en columnas. Utiliza HDFS como el sistema de almacenamiento [23].

- *ZooKeeper*:
Servicio de coordinación que hace posible la sincronización de tareas en las aplicaciones construidas sobre HDFS [23].
- *Sqoop*:
Herramienta de movimiento de datos entre bases de datos relacionales y HDFS [23].

La característica más importante que puede resaltarse del entorno *Hadoop* para este proyecto, es la capacidad de procesamiento paralelo de información en múltiples máquinas de bajo costo distribuidas. Esto significa que una solución de este tipo podría tener escalabilidad horizontal, es decir, nuevas máquinas pueden ser agregadas a los clúster de procesamiento si nuevos requerimientos o mayor demanda de recursos así lo indicaran. Es de significar, que la escalabilidad horizontal mencionada líneas atrás, es de menor costo que su contraparte de tipo vertical. En el segundo caso, el aumento en las condiciones de procesamiento o almacenamiento se logra invirtiendo dinero en la expansión de la infraestructura existente, por ejemplo, añadiendo unidades de procesamiento, discos de almacenamiento o unidades de memoria [25].

El procesamiento en paralelo, es pues una tarea que puede ser lograda en el ecosistema *Hadoop* utilizando las herramientas dispuestas como *MapReduce* o alguna de sus abstracciones, *Pig* o *Hive* las cuales operan sobre el sistema de archivos distribuido HDFS. La Figura 7 presenta un diagrama ilustrativo del funcionamiento general de los programas *MapReduce*. Vale la pena resaltar, que las abstracciones mencionadas previamente, son interpretadas finalmente como operaciones *MapReduce*, es decir, todo tipo de procesamiento de datos en *Hadoop* es reducido finalmente a este último lenguaje.

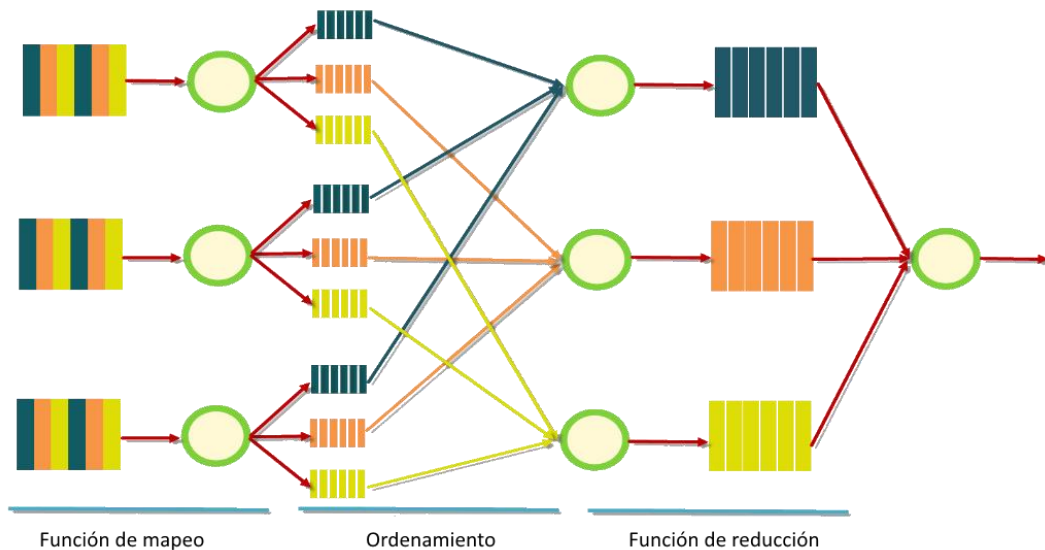


Figura 7 Funcionamiento MapReduce. Imagen original tomada de <http://blog.sqlauthority.com/2013/10/09/big-data-buzz-words-what-is-mapreduce-day-7-of-21/>

La conexión entre los conceptos revisados en este trabajo hasta este punto; Big Data, WSN y computación en la nube, constituyen el valor de la prueba de concepto para el proyecto padre que ha sido previamente referenciado. La propuesta de integración entre las mediciones obtenidas en la capa física, los servicios ofrecidos en la nube en términos de computación y almacenamiento y la aplicación de métodos de análisis entre otros componentes propios del ecosistema de Big Data, son los fundamentos de justificación para el diseño de la arquitectura.

1.4. Inteligencia del negocio

La Inteligencia de Negocios o *BI* por sus siglas en inglés, es un concepto que abarca a un conjunto de técnicas desarrolladas en el campo de las tecnologías de información. Su propósito de desarrollo consiste en posibilitar transformaciones en conjuntos de datos de diferentes tipos, orígenes, formas y tamaños que permitan obtener puntos de vista aventajados estratégicamente para la toma de decisiones. En una analogía con la inteligencia militar, la inteligencia de negocios aplicada, tiene como objetivo principal otorgar a los comandantes militares de cada rango una visión clara del campo de batalla, con detalles acerca de las ventajas y desventajas de sus opciones [26]. En el contexto de gestión del riesgo, el significado del objetivo es dar a conocer el comportamiento de una zona de análisis a quienes pudiera interesarles en su proceso de toma de decisiones, es decir, en un escenario de minimización de riesgos la inteligencia del negocio debe presentar a sus usuarios aquellas ocurrencias que pudieran influir en el estado futuro del objeto de estudio.

En una definición formal, *BI* comprende los procesos, tecnologías y herramientas necesarias para transformar datos en información y tal información en conocimiento que pueda guiar favorablemente la toma de decisiones [27]. En cuanto a tecnologías y herramientas, actualmente existen en el mercado una gran variedad de opciones adaptables a diversos contextos, es decir, que no existe un solo tipo de tecnología con una herramienta exclusiva que responda exitosamente al universo de aplicaciones posibles. En contraste, los procesos de generación de información y conocimiento a través de tecnologías de información han demostrado una tendencia en su trayectoria de maduración que apuntan hacia contados conceptos aplicables de forma generalizada. De forma específica, se enuncian algunos de los procedimientos mencionados que comúnmente intervienen:

- Diseño de bodegas de datos o *Warehousing* en inglés
- Diseño de extracción, transformación y carga o ETL
- Explotación
 - ✓ Informes y análisis
 - ✓ Minería de datos.
 - ✓ Visualización

Cada punto anterior representa una rama de estudio dentro de la Inteligencia del Negocio y en esencia todas atienden a necesidades específicas en diferentes grados de complejidad. Sin embargo, es común entre las tecnologías y herramientas de *BI* disponibles en el mercado encontrar elementos de muchas o todas las ramas enunciadas al servicio del usuario final.

Con el propósito de soportar teóricamente la solución a plantear en este trabajo, se presentan a continuación de forma descriptiva algunos criterios generales dentro de las ramas descritas previamente. No se pretende introducir detalladamente cada nivel dado que abarcar toda su complejidad está por fuera del alcance del presente documento. No obstante, se pretende aquí resaltar los conceptos de selección a emplear de acuerdo a lo investigado en otras fuentes, a la vez que se describen posibles tecnologías y herramientas disponibles.

Diseño de bodega de datos

Este campo de la Inteligencia de Negocios constituye uno de los conceptos más debatidos y frecuentemente cuestionados entre los grandes ponentes de esta rama de la ciencia de computación. A pesar de haber madurado a un ritmo constante a lo largo de la pasada década, hoy en día aún se encuentran en oposición dos postulados de los más reconocidos exponentes Bill Inmon y Ralph Kimball [28].

- **Aproximación de Bill Inmon**

Este autor describe la bodega de datos o *WH* por sus siglas en inglés, como un repositorio centralizado para toda la empresa o en otros contextos para todos los grupos de interés. Su propuesta se conoce como *Top-Down* y consiste en un diseño normalizado que estructura de forma atómica desde su concepción todos aquellos hechos y dimensiones de posible interés para toda la compañía. Solo después de consolidar un diseño que responda a cualquier departamento interesado en la información al mínimo nivel de detalle, se propone la construcción de *Data Marts* dimensionales para la explotación en reportes, análisis, visualización o minería [28]. La Figura 8 representa este enfoque.

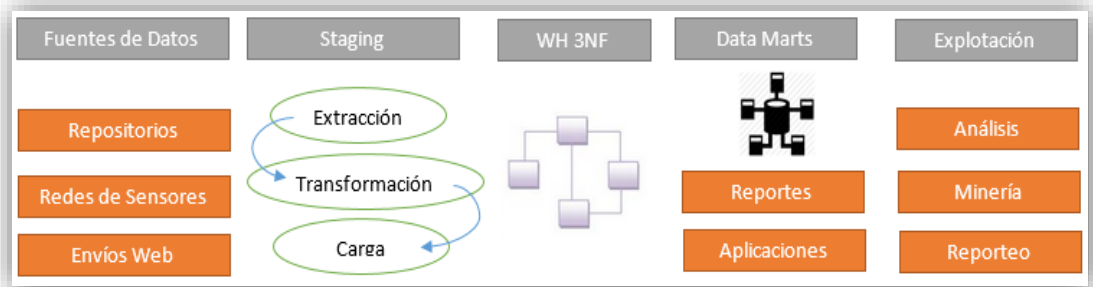


Figura 8 Aproximación Top – Down. Construida a partir de <http://cdn.ttgmedia.com/rms/enterpriseApplications/Inmon's%20Approach.png>

Las implicaciones del enfoque de este autor con respecto a la bodega de datos se describen a continuación de acuerdo con [29]:

- ✓ **Orientada al tema:** Los datos en la base de datos son organizados de tal forma que todos los elementos relacionados con el mismo evento del mundo real estarán conectados.
 - ✓ **Cambiante en el tiempo:** Los cambios en los datos de la base de datos son almacenados y observados de tal forma que se podrán generar reportes que presenten los cambios en el tiempo.
 - ✓ **No volátil:** Los datos en la base de datos no se sobre escriben ni se eliminan. Una vez almacenados, los datos permanecen estáticos de solo lectura.
 - ✓ **Integrada:** La base de datos contiene data de la mayoría de las aplicaciones de la organización de forma consistente.
- **Aproximación de Ralph Kimball**

En su perspectiva conocida como *Bottom - Up*, este autor postula que los aspectos más importantes de cada contexto deben ser creados primero, es decir, los *Data Mart*. De acuerdo a los requerimientos emergentes, los *Data Mart* podrán ser combinados para conformar la bodega de datos. Kimball define la bodega de datos como una copia de los datos transaccionales estructurados específicamente para consultas y análisis [28].

La Figura 9 representa los postulados mencionados adicionando la premisa de que el modelado dimensional se enfoca en la facilidad de acceso para el usuario y el rendimiento en el funcionamiento.

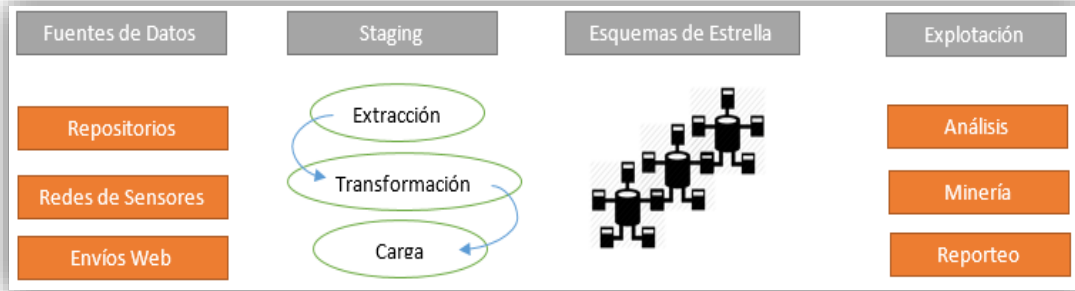


Figura 9 Aproximación Bottom –Up. Construida a partir de <http://cdn.ttgtmedia.com/rms/enterpriseApplications/Kimball's%20Approach.png>

Se utilizará la Tabla 8, construida con base en [28], [29], para comparar los principales aspectos de ambos postulados y así visualizar las ventajas y desventajas de los mismos.

Aspecto / Autor	Inmon	Kimball
Tiempo de diseño y construcción	Alto	Medio
Dificultad para el mantenimiento	Bajo	Alta
Costo al inicio de la implementación	Alto: Requiera alta especialización en herramientas y recursos humanos.	Medio: No se requieren especialistas para el diseño.
Costo de modificaciones y adiciones	Bajo: Las adiciones luego de la implementación son de bajo costo.	Medio: Continúan siendo el mismo costo de la etapa de inicio.
Habilidades requeridas para la implementación	Alto: Equipos especializados.	Medio: Equipos con nivel general.
Requerimientos de integración	Alto: Se debe conocer y manipular todas las fuentes de datos al momento de la implementación.	Bajo: Solo se requieren diseños parciales hacia las fuentes de interés.

Clientes finales	Equipos técnicos	Usuario final
Objetivo	Prestar una robusta solución técnica basada en métodos relacionales ya probados.	Prestar una solución que facilite el acceso a usuarios finales.
Tipo de decisiones que permite tomar	Estratégicas.	Tácticas.
Generación de métricas	Previo a la construcción de data marts no se tienen métricas de un contexto dado que el diseño se realiza para cumplir múltiples necesidades de información.	Las medidas de cada contexto son fácilmente extraíbles para la construcción de tableros de control dada la disposición de diseño por área o departamento.

Tabla 8 Comparativo Bill Inmon y Ralph Kimball

Los diseños de las bodegas de datos mencionadas previamente pueden ser implementados en sistemas relacionales de diferentes tecnologías. Listar todas las posibles opciones de sistemas de administración de bases de datos relacionales en el mercado se encuentra por fuera del alcance y propósito de este documento. Sin embargo, en la Tabla 9 se mencionan algunas propuestas como posibles candidatos a la solución de bajo costo en la nube que se pretende lograr.

	SQL Server	MySQL	Oracle	PostgreSQL
Creador	Microsoft	Oracle	Oracle	PostgreSQL
Licencia	Comercial	Código Abierto	Comercial	Código Abierto
Base de datos como servicio (DBaaS)	Amazon Web Services. Microsoft Azure.	Amazon Web Services.	Amazon Web Services.	Amazon Web Services.

Sistema Operative	Windows	FreeBSD Linux Solaris Windows	FreeBSD Linux Solaris Windows	FreeBSD Linux Unix OS X Solaris Windows
APIs y métodos de acceso	OLE DB ADO.NET JDBC ODBC	ADO.NET JDBC ODBC	ODP.NET OCI JDBC ODBC	ADO.NET JDBC ODBC
Concurrencia	Si	Si	Si	SI
Disparadores	Si	Si	Si	Si
Particionamiento	Archivos	Horizontal	Horizontal	No
Seguridad	Si en varios niveles	Si en varios niveles	Si en varios niveles	Si en varios niveles
Creación de replicas	Depende de cada version	Si en varios niveles	Si en varios niveles	Solo Maestro - Esclavo

Tabla 9 Comparativo entre tecnologías para administración de bases de datos

Se debe resaltar que sobre estas herramientas pueden ser construidos los esquemas de estrella propuestos por Ralph Kimball y la bodega de datos por Bill Inmon. Sin embargo, la explotación para el análisis y visualización debe ser realizada sobre motores de análisis donde puedan ser creados cubos calculados para optimizar el rendimiento de las consultas. En la próxima sección, donde se describen los métodos, herramientas de análisis y creación de informes, se describirán algunas opciones disponibles.

Diseño de Extracción, Transformación y Carga (ETL)

La extracción, transformación y carga son procesos fundamentales en cualquier sistema de consolidación de datos como una bodega de datos. Del buen diseño de este sistema depende en gran medida la gobernabilidad de los datos dado que en estas fases se configuran reglas de calidad en términos de consistencia a lo largo y ancho de todas las fuentes [30].

Su diseño e implementación es una tarea poco visible para los usuarios finales dado que su objetivo se centra en el cuarto trasero de la solución y debe en esencia ser transparente para quien explota los datos en búsqueda de información y conocimiento. Dentro de su construcción no solo se busca re estructurar los datos provenientes de diversas fuentes y plasmar los resultados finales en la bodega de datos, sino también añadir valor a los datos mediante trabajos adicionales sobre estos según [30] como:

- Eliminar errores y reemplazar datos faltantes según sea posible.
- Otorgar mediciones de confiabilidad de los datos.
- Ajustar y combinar datos de diversas fuentes.
- Estructurar y dar formato a los datos para hacerlos consumibles.

El diseño de la ETL es pues una labor que requiere esfuerzos considerables a pesar de ser un concepto simple si al analizar su propósito principal. Lo anterior se justifica teniendo en cuenta que su correcto funcionamiento determina la medida del valor percibido por los usuarios finales, aun cuando tal funcionamiento es invisible para los mismos. Adicionalmente, su construcción para cada aplicación debe ser planeada y diseñada a la medida dado que todos los contextos de análisis son diferentes y las fuentes transaccionales de la información los son también. Por tales razones, es necesario que los diseños de las ETL en las soluciones de inteligencia de negocios tengan un alcance definido y de forma transversal al contexto consideren en detalle los siguientes ámbitos:

- Necesidades del contexto de análisis.
- Requerimientos de conformidad con estándares.
- Perfil de los datos.
- Requerimientos de seguridad.
- Horizonte de tiempo de almacenamiento de los datos.
- Interfaces de acceso al usuario final.
- Recursos humanos capacitados necesarios.

El análisis de los factores mencionados previamente cubre la mayor parte de los retos que se deben enfrentar al desarrollar esta etapa clave de la solución que se pretender concebir. No obstante, antes de abordar los ámbitos descritos, hay ciertas decisiones tempranas que deben ser tomadas con base en estudios preliminares sobre los requerimientos; La primera decisión importante tiene que ver con las herramientas a utilizar, es decir,

se debe contemplar si el desarrollo será una labor manual de codificación o si se utilizara un paquete de herramientas disponible en el mercado para llevar a cabo el proceso. La segunda decisión, es el tipo de tecnología de administración de bases de datos relacionales que se utilizara y la infraestructura que alcanzará el proyecto. Ambas elecciones determinan en gran medida el presupuesto y los tiempos requeridos para consolidar el diseño.

Con respecto a la primera decisión, se debe comentar que en ambos escenarios, codificación manual o uso de paquetes de herramientas, son mutuamente excluyentes y cada uno tiene sus ventajas que determinan la selección según el contexto. Algunas ventajas se presentan en la Tabla 9 construida a partir de [30].

Aspecto / Técnica	Codificación Manual	Paquete de Herramientas
Nivel de conocimiento de desarrollo de Software	Alto: Se requieren equipos técnicos para lograr una solución mantenible	Bajo: Profesionales con pocos conocimientos técnicos pueden lograr la implementación.
Dificultad para probar la solución	Bajo: Dado que la construcción se hace a bajo nivel, los desarrolladores pueden lograr sistemas probados en todos sus niveles de forma automática.	Medio: Para lograr pruebas unitarias o de integración se requieren equipos especializados en la tecnología y paquete elegido para la solución.
Dificultad para el mantenimiento y realización de cambios	Alta: Se debe contar con un equipo de desarrollo de software para hacer cambios y analizar impactos al igual que para el mantenimiento	Medio: A pesar de que los análisis de impactos pueden ser arrojados por la herramienta, el mantenimiento y la adición de módulos puede requerir conocimiento especializado.

Dificultad para creación y manejo de meta data	Media: El desarrollo orientado a objetos permite fácilmente generar y actualizar meta data. Sin embargo, siempre se debe crear la meta data manualmente.	Baja: La creación y actualización de meta data generalmente es lograda directamente por la herramienta sin tener intervención manual.
Restricciones de diseño o implementación	Bajo: No existe restricción de lenguaje ni de infraestructura.	Alto: Dependencia alta al lenguaje y requisitos del proveedor de la herramienta.

Tabla 10 Comparación de métodos para construcción de ETL

A pesar de que existen muchos criterios de comparación adicionales a los presentados, las condiciones expuestas son relevantes para el prototipo que busca este trabajo como producto final. En adición, la rigurosidad del diseño y la implementación en esta prueba de concepto no pretende abordar todos los detalles y condiciones establecidas en la documentación estudiada sino tal vez, dar un conjunto de criterios base que sirvan de línea de estudio futuro en una implementación real del sistema propuesto.

Con respecto a la segunda decisión indicada previamente, es importante considerar que las tendencias actuales en el campo de las tecnologías de información han cambiado radicalmente apuntando hoy hacia entornos automáticos implementados en la nube en un contexto donde cada vez se vuelve más influyente el concepto de Big Data. Lo anterior, implica que los sistemas de información sean diseñados para soportar mayor dinamismo, es decir, infraestructuras flexibles y escalables de forma horizontal y vertical, volúmenes de datos del orden de tera bytes y peta bytes, fuentes de datos sin estructura, semiestructurados y estructurados, alta concurrencia, alta velocidad y poca latencia etc. Todo lo anterior se menciona con el fin de dar una evidencia del panorama en el que se desarrolla este trabajo, el cual, siendo un prototipo aplicado en el campo de Internet de las cosas debe tener en cuenta estas características para lograr una propuesta base sobre la que se pueda avanzar posteriormente bajo las más recientes directrices. Todo lo mencionado en este párrafo influye directamente en la concepción de los métodos adecuados para la extracción, transformación y carga de las

muestras recolectadas en las redes de sensores que puedan implantarse, dado que algunos de los componentes enunciados en el escenario contemplado, se encuentran actualmente en investigación y maduración.

Algunas técnicas de ETL aplicables en el contexto de Big Data y computación en la nube han sido desarrolladas por proveedores de código abierto como Apache [23]. Su alcance se ha expandido rápidamente hacia grandes compañías como Microsoft, Google, Amazon entre otros, los cuales han puesto el entorno Hadoop a disposición del público en una plataforma como servicio. Una de las herramientas disponibles y comúnmente utilizadas por grandes compañías como Facebook y Twitter para sus procesos de ETL es Hive [22]. Sobre esta herramienta se hizo una descripción en la sección anterior, sección en la cual se presentó como un componente del ecosistema Hadoop que hace posible el procesamiento en paralelo de altos volúmenes de datos provenientes de diversas fuentes. Su aplicación puede ser en si una bodega de datos distribuida sobre el sistema de archivos HDFS pero también Hive puede ser utilizado como una potente herramienta de transformación de datos en una infraestructura distribuida altamente flexible y escalable. La transformación se hace posible a través de consultas escritas en una sintaxis SQL familiar usando el lenguaje HiveQL o HQL, sentencias que son finalmente traducidas a operaciones *Map Reduce* sobre los conjuntos de muestras distribuidas a lo largo y ancho del cluster Hadoop que se esté utilizando. Esto representa una gran ventaja para el diseño de ETL dado que aporta la escalabilidad y flexibilidad buscada en el procesamiento del volumen muestras provenientes de las diversas redes de sensores.

Explotación

Esta etapa constituye el punto de interacción con el usuario final que, en una descripción metafórica, representa la punta del Iceberg de la solución de *BI*. Su implementación se planea y diseña acorde con los requerimientos de análisis que estén estipulados y generalmente se consolida a través de paquetes de herramientas ya posicionadas en el mercado global. De la misma forma que ocurre con los sistemas de administración de bases de datos descritos en apartados previos, la cantidad de opciones disponibles para la explotación hace que este trabajo deje por fuera del alcance la especificación detallada de las mismas. Sin embargo, para esbozar el

contexto actual se acude a una representación breve de las tendencias establecidas, presentada en la Figura 10 tomada de [31].



Figura 10 Cuadrante Mágico de Gartner para BI. Tomada de <http://goo.gl/fz57IU>

Las herramientas presentadas responden a las necesidades recientes de un mercado que es cambiante y cada vez más exigente en términos de análisis avanzados, predicciones, simulaciones y optimizaciones [31]. En consecuencia la mayoría de estas comparten, en cierta medida, una visión similar en cuanto a las prestaciones que brindan a tales necesidades determinadas por el mercado. Un análisis general de lo que puede encontrarse en la mayor parte de los paquetes de explotación se describen a continuación.

- **Informes y Análisis**

Informes: proporcionan la capacidad de crear reportes altamente formateados, interactivos, imprimibles, con o sin parámetros etc [31].

Consultas: Se permite a los usuarios hacer sus propias consultas sobre los datos, sin depender de TI para obtenerlas. En particular, las herramientas cuentan con una capa semántica reutilizable para permitir a los usuarios

navegar por las fuentes disponibles de datos, métricas predefinidas, jerarquías entre otros [31].

Integración con Microsoft Office [32]: A veces, Microsoft Office (especialmente Excel) actúa como el cliente de informes o análisis. En general, se encuentra que las herramientas proporcionan integración con Microsoft Office, incluyendo soporte para formatos nativos de documentos y presentación, fórmulas, gráficos, tablas de datos y tablas dinámicas [31].

BI Móvil: Se encuentran herramientas que facilitan la distribución de contenido a los dispositivos móviles a través de publicaciones estáticas o de modo interactivo, aprovechando muchas de ellas las capacidades nativas de los dispositivos móviles, tales como pantalla táctil, cámara, conocimiento de la ubicación y consulta en lenguaje natural [31].

Descubrimiento de datos: Se usan búsquedas de fuentes de datos estructurados y no estructurados y los asigna a una estructura de clasificación en las dimensiones y hechos de forma que los usuarios pueden navegar y explorar fácilmente. Esto es una característica básica de una plataforma de BI [31].

Procesamiento analítico en línea (OLAP): Permite a los usuarios analizar los datos con la consulta y sus métricas, lo que permite un estilo de análisis conocido como "reordenamiento" dado que los usuarios pueden navegar por rutas multidimensionales. También tienen la capacidad de escribir y devolver valores de una base de datos para la planificación y el modelado "¿y si?". Esta capacidad en algunas herramientas podría abarcar una variedad de arquitecturas de datos tales como relacional, multidimensional o híbrido y arquitecturas de almacenamiento tales como en memoria o en disco [31].

- **Minería de Datos**

El término minería de datos describe el proceso diseñado para explorar datos con el propósito de encontrar patrones consistentes y/o relaciones semánticas entre las variables de estudio que sean verificables en conjuntos de datos nuevos para los modelos diseñados. El objetivo final del proceso es realizar predicciones utilizando dichas relaciones encontradas entre los datos estudiados [33]. El análisis avanzado de los conjuntos de datos se describe brevemente a continuación.

Análisis avanzado: Permite a los usuarios aprovechar una biblioteca de funciones estadísticas embebidas en un servidor de BI. También, algunas tienen incluidas las habilidades para consumir métodos comunes de análisis, tales como *Predictive Model Markup Language* (PMML) y modelos basados en R sobre la capa de metadatos. Adicionalmente, en general se facilitan herramientas para crear visualizaciones analíticas avanzadas como informes de correlaciones, realizar segmentación de datos, obtener predicciones y tendencias [31]. Algunas incluyen paquetes específicos de algoritmos para aplicación de técnicas de aprendizaje supervisadas y no supervisadas.

- **Visualización**

Tableros de control: Se incluyen para la generación de informes que representan gráficamente las medidas de rendimiento. Incluye la capacidad de publicar múltiples objetos, informes vinculados, uso de parámetros con pantallas intuitivas e interactivas; Los *Dashboard* como se conocen en inglés a menudo emplean componentes de visualización tales como medidores, deslizadores, casillas de verificación y mapas. Se utilizan para mostrar el valor real de la medida en comparación con una meta o valor objetivo [31].

Inteligencia Geoespacial y ubicación: Las últimas tendencias apuntan a la inclusión de herramientas para análisis especializados y visualizaciones que proporcionan un contexto geográfico, espacial y temporal. Otorgan la capacidad de representar las características físicas de los datos a la vez que involucran información de otras fuentes como referencias geográficas, incluyendo mapas aéreos, los SIG y demografía de los consumidores, data de la empresa entre otros posibles. Algunas, permiten generar relaciones básicas mediante la superposición de datos en mapas interactivos. En tendencias más recientes, se emplean capacidades más avanzadas apoyadas en algoritmos geoespaciales especializados, por ejemplo, para la distancia y cálculo de rutas. Finalmente, en la mayoría de paquetes se permite realizar la estratificación de los datos geoespaciales en los mapas, incluir marcadores, producir mapas de calor y mapas temporales, visualizaciones 3D etc. [31]

Visualización interactiva: Las herramientas en general permiten la exploración de datos a través de la manipulación de las imágenes de gráfico, con el color, el brillo, el tamaño, la forma y el movimiento de los objetos visuales que representan los aspectos del conjunto de datos que se analiza.

Esto incluye una variedad de opciones de visualización que van más allá de los gráficos de torta, barras y líneas, incluyendo mapas de calor, árboles, mapas geográficos, gráficos de dispersión y otros elementos visuales para fines especiales. [31].

CAPITULO 2

2. DISEÑO E IMPLEMENTACIÓN

Luego de realizar un ambicioso intento en el capítulo 1 por contextualizar las tendencias de amplios temas de investigación en el campo de tecnologías de la información, en este apartado se pretende sustentar una propuesta de diseño e implementación que dé cumplimiento a los objetivos trazados y que en consecuencia atienda el problema planteado inicialmente. El capítulo anterior servirá pues como contexto base para enlazar los temas intervinientes en esta solución: Redes de Sensores, Computación en la Nube, *Big Data* e Inteligencia de Negocios.

Dado que el presente trabajo pretende establecer una solución al problema de almacenamiento de muestras, procesamiento y disposición de los datos para análisis y visualización como etapa intermedia de un proyecto llamado “*Prototipo TICS para el Monitoreo, Prevención y Atención de Deslizamientos de Tierra en la Red Vial de Colombia*”, se busca consolidar una propuesta de arquitectura flexible, escalable y de bajo costo que soporte la recepción, almacenamiento, procesamiento y disposición de los datos provenientes de diversas fuentes. Adicionalmente, el planteamiento pretende exhibir un prototipo funcional sobre el cual puedan ser ejecutados escenarios de prueba que demuestren algunos casos de uso posibles. Para esto, se asume la etapa de recolección física de muestras como parte de otro proyecto y se busca entonces demostrar los procedimientos de procesamiento y consolidación de los datos ya persistentes mediante una simulación que deja como resultado la posibilidad de visualización de los datos así como la aplicación de técnicas de análisis sobre estos para la toma de decisiones en eventos de riesgo.

2.1. Consolidación de la arquitectura

La consolidación de la arquitectura planteada es un proceso de construcción de artefactos de software que al ser integrados cumplirían con los requerimientos funcionales del proyecto. Por tal razón dentro de las etapas de análisis y de diseño, se identificó que existe una similitud entre el presente

trabajo y el proceso de desarrollo de software convencional. Lo anterior dado que partiendo de un conjunto de necesidades de usuario se pretende en principio consolidar un sistema informático que dé solución a tales requisitos. No obstante, se tuvo en cuenta que la solución buscada no era una aplicación convencional sino una herramienta propia de inteligencia de negocios y en ese sentido se evaluaron las metodologías de desarrollo de software comúnmente utilizadas pero haciendo un enfoque especial hacia BI. Habiendo dicho lo anterior, con base en información estudiada en [34]–[36] se concluyó que una aproximación adecuada podía ser encontrada en metodologías ágiles como SCRUM y Kanban [37]. De las cuales se eligió SCRUM como la opción adecuada para enmarcar la implementación del presente proyecto teniendo en cuenta los conceptos que se describen a continuación.

Como se mencionó previamente, el proceso fue llevado a cabo haciendo uso de la aproximación ágil de desarrollo de software llamada SCRUM [36]. En las próximas líneas se hará una breve contextualización de algunos conceptos claves de tal aproximación metodológica a la vez que se orientan tales nociones al desarrollo actual. Vale la pena resaltar que a pesar de que el presente trabajo se considera una solución de Inteligencia de Negocios más que un proyecto de desarrollo de software, las metodologías ágiles pueden ser aplicadas al contexto actual dado su versatilidad y de esta manera permitirán un avance continuo y controlado a lo largo del proyecto.

La aproximación metodológica llamada SCRUM es un marco de trabajo ágil para el desarrollo de software. Su descripción detallada es presentada en trabajo conocido como manifiesto ágil. Sus principios filosóficos de trabajo son los siguientes [36]:

- Individuos e interacciones prevalecen sobre procesos y herramientas.
- Software funcional prevalece sobre documentación extensiva.
- Colaboración del cliente prevalece sobre contratación detallada.
- Respuestas al cambio prevalecen sobre seguimiento a planes.

Sobre las premisas mencionadas se fundamenta el funcionamiento de esta aproximación metodológica iterativa que tiene como propósito guiar a equipos de trabajo auto-organizados hacia la producción de artefactos funcionales en el mejor tiempo posible. Los roles que intervienen durante una iteración o Sprint son:

- **Responsable del Producto:**
Estipula los requerimientos y deseos para la producción. Dicta los criterios de aceptación y vigila el funcionamiento de los entregables[36].
- **Equipo de desarrollo:**
Encargado de llevar a cabo las tareas de producción y de calidad de cada uno de los entregables requeridos[36].
- **Scrum Master**
Coordina el equipo de desarrollo hacia la obtención de entregables en el mejor tiempo posible de acuerdo a los requerimientos. Busca disminuir barreras que obstaculizan al equipo de desarrollo[36].

Los artefactos que se involucran dentro del proceso iterativo para describir las unidades de trabajo son:

- **Visión:**
Idea fundamental del producto.
- **Backlog** o requerimientos del producto:
Listado de requerimientos descriptivos para lograr el producto.
- **Backlog** o requerimientos del *Sprint*:
Listado de requerimientos para la iteración estimado entre 2 y 4 semanas.
- **Entregable:**
Pieza de software funcional que atiende los requerimientos de la iteración.

A lo largo del proceso iterativo se llevan a cabo un conjunto de actividades que permiten hacer seguimiento al desarrollo del producto. La Figura 11 evidencia el ciclo de vida del proceso indicando los artefactos involucrados y las actividades ejecutadas en su transcurso [35]. Esta breve descripción bosqueja someramente la aproximación utilizadas para consolidar el producto del presente trabajo. Pese a que su aplicación no se toma de forma estricta, si se emplean sus principios metodológicos al igual que la mayor parte de sus actividades.

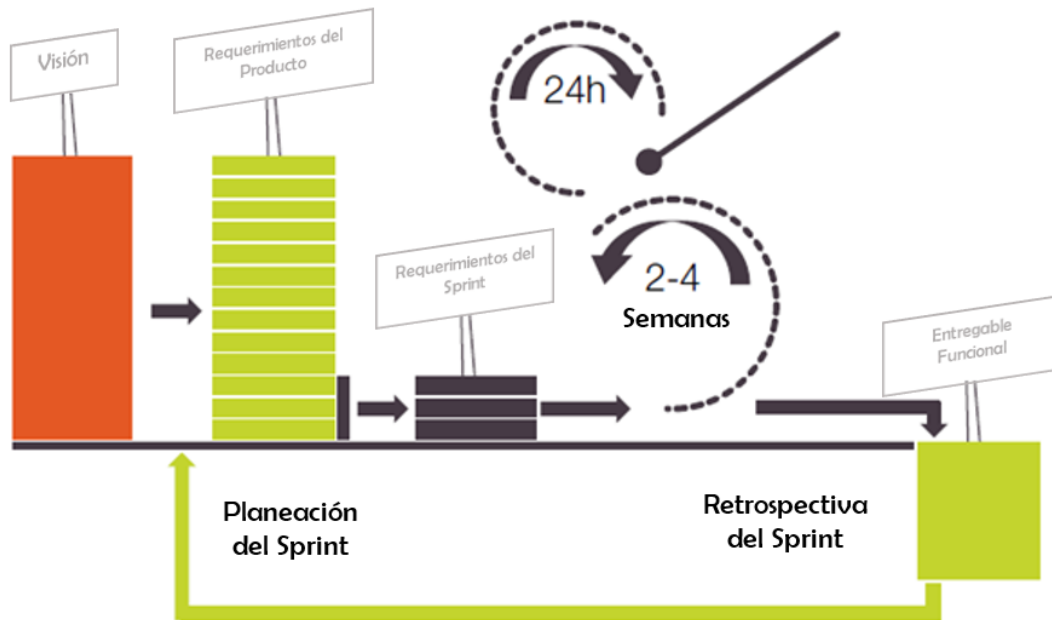


Figura 11 Ciclo de vida SCRUM. Figura tomada y adaptada a partir de [35]

emplear, se presenta a continuación la Tabla 11 resumen que evidencia el marco de desarrollo de este trabajo.

Requerimiento	Sprint	Dueño	Desarrollador	Duración
Definir los requisitos funcionales y no funcionales de la solución.	1	Leonardo Betancur	Juan González	2 semanas
Realizar una búsqueda bibliográfica sobre herramientas comúnmente utilizadas. Spike	1, 2	Leonardo Betancur	Juan González	3 semanas
Esbozar arquitectura apropiada para dar cumplimiento a los requisitos no funcionales.	2,3	Leonardo Betancur	Juan González	2 semanas
Seleccionar herramientas óptimas	2,3	Leonardo Betancur	Juan González	2 semanas

para dar cumplimiento a los requisitos funcionales.				
Evidenciar el diseño planeado para la solución incluyendo detalles del ambiente de implementación.	3,4	Leonardo Betancur	Juan González	5 semanas
Adecuar el ambiente de implementación para la implantación de la arquitectura.	5	Leonardo Betancur	Juan González	2 semanas
Documentar el procedimiento empleado para realizar la implantación de la arquitectura diseñada.	5	Leonardo Betancur	Juan González	2 semanas
Ejecutar pruebas iniciales de funcionamiento sobre los subsistemas empleados en la solución.	5,6	Leonardo Betancur	Juan González	2 semanas
Aplicar un caso de prueba usando dispositivos periféricos que interactúen con el centro de fusión de datos.	6	Leonardo Betancur	Juan González	2 semanas
Analizar los resultados obtenidos en las pruebas de funcionamiento.	6,7	Leonardo Betancur	Juan González	2 semanas
Proponer mejoras sobre las observaciones realizadas.	7	Leonardo Betancur	Juan González	2 semanas
Emplear herramientas de análisis y visualización sobre	7,8	Leonardo Betancur	Juan González	2 semanas

conjuntos de datos de prueba.				
Documentar las recomendaciones sobre técnicas de análisis apropiadas para la automatización de toma de decisiones.	8,9	Leonardo Betancur	Juan González	6 semanas

Tabla 11 Requerimientos por iteraciones

Una vez dado dar inicio al proceso iterativo, los primeros resultados se vieron reflejados a partir del tercer *Sprint* luego de haber consolidado los requerimientos funcionales de la solución deseada. La tarea de esbozar la arquitectura del sistema a desarrollar dejó como resultado el diagrama mostrado en la Figura 12.

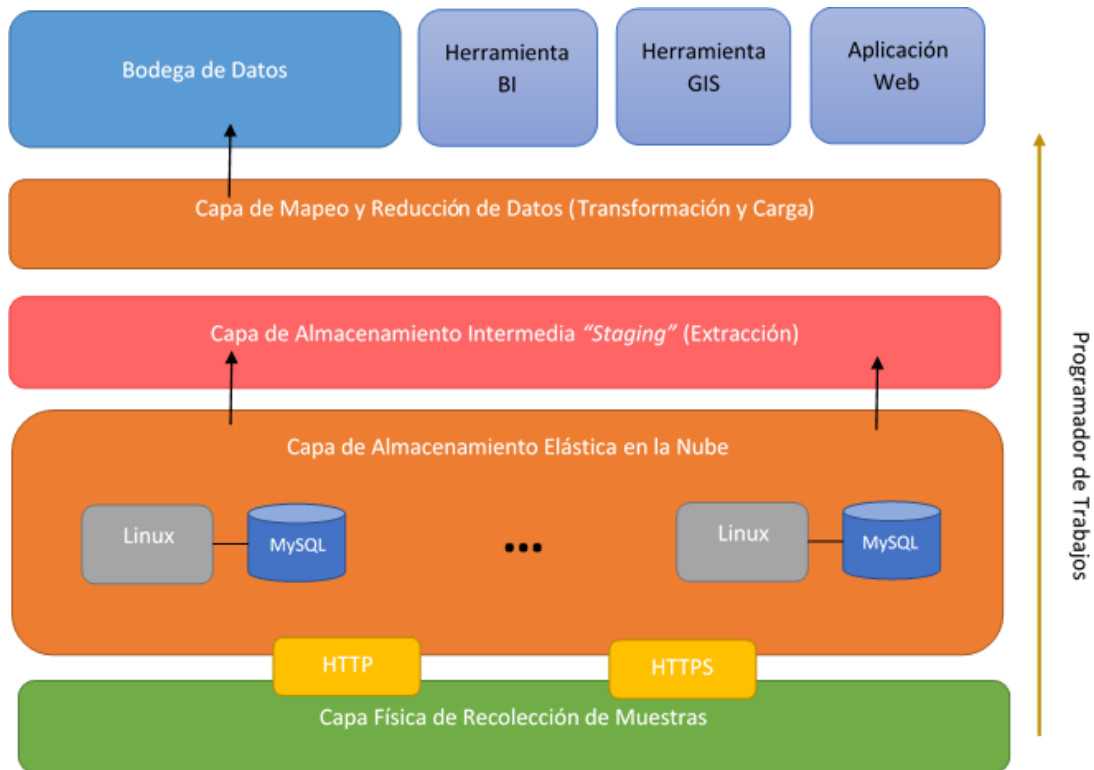


Figura 12 Planteamiento de arquitectura

El resultado obtenido de las primeras tres iteraciones dio como resultado un entendimiento detallado de los requerimientos, entendimiento que se ve reflejado las capas presentes en la Figura 12.

Partiendo de una capa física, los requerimientos dictan que se deben exponer terminales para la transferencia de muestras a través de los protocolos Http o Https hacia una capa de almacenamiento. Dicha capa de almacenamiento contiene maquinas con sistemas operativos Linux y bases de datos MySQL donde persistirán los datos provenientes de los sensores implantados en la capa física. La exposición de los terminales en la capa de almacenamiento se hace posible haciendo uso de servidores web Tomcat. Estos servidores de almacenamiento deben ser escalables tanto de forma vertical como horizontal, es decir, debe permitirse la adición de componentes de hardware para mejorar el rendimiento de los ya existentes así como el suplemento de nuevas máquinas. Adicionalmente, las máquinas de almacenamiento deben encontrarse en funcionamiento 24 horas al día 365 días al año y tolerar accesos concurrentes. Finalmente para esta etapa se deben poder incluir mecanismos de seguridad que protejan la escritura en las bases de datos.

Una vez la capa de almacenamiento se encuentre en funcionamiento continuo, los datos de todas las bases de datos allí presentes deben ser extraídos hacia una capa de almacenamiento adicional, llamada intermedia o *Staging* en inglés. Esta capa debe contener la extracción periódica de las muestras persistentes en las bases de datos expuestas para recolección. Es importante resaltar que la extracción debe contener solo datos nuevos, es decir, que se debe identificar el delta en cada una de las bases de datos y posteriormente copiar tales muestras hacia un lugar adicional donde puedan ser llevados a cabo análisis y manipulaciones subsecuentes. El tiempo de permanencia del contenido en este lugar intermedio de almacenamiento no está determinado por los requerimientos. Un requerimiento funcional indica que la extracción debe poder lograrse de forma automática con una frecuencia determinada.

El siguiente paso, la capa de transformación, surge del requerimiento de resumen y totalización de los datos de acuerdo a la granularidad estipulada para la bodega de datos. Para este trabajo se estipula como criterio clave de diseño una granularidad de un registro en la bodega de datos por minuto. Con esto dicho, la capa de transformación debe permitir la reducción de

datos a esta granularidad y a la estructura adecuada para la carga posterior en la bodega de datos. Es importante incluir en este apartado que la transformación de datos, al igual que las demás capas, debe contener mecanismos escalables para soportar volúmenes variables de entrada. Una buena aproximación es el procesamiento paralelo que se ofrece en las plataformas de Big Data, el cual haciendo uso de sistemas distribuidos permite amplia flexibilidad y escalabilidad en eventos pico de estrés. Las tareas de procesamiento en esta capa son dependientes de las actividades de extracción y debe ser ejecutadas de forma automática periódicamente.

La etapa final del diseño contiene la bodega de datos como componente de almacenamiento final sobre el cual podrán ser conducidos análisis y explotaciones en general que permitan obtener información y conocimiento sobre los eventos que en la capa física sucedan. Entre los componentes de esta capa expuesta al usuario se incluyen elementos adicionales opcionales dentro de los requerimientos a parte de la bodega de datos, estos son: Herramientas BI, Herramientas GIS y Aplicaciones Web. Todas las anteriores podrán consumir los datos persistentes en la bodega de datos o en la capa de transformación misma. La posibilidad de acceso tanto a la bodega de datos como a la capa de transformación es la razón por la cual no se plantean relaciones entre los bloques.

Hasta este punto se ha descrito el transcurso de las primeras iteraciones que deja como resultado la representación de los requerimientos funcionales y no funcionales en un modelo de bloques que servirá para procesos siguientes. A continuación se expondrán los detalles de iteraciones subsecuentes desglosando el trabajo realizado sobre cada una de las capas.

2.2. Modelo transaccional

La capa de almacenamiento en la arquitectura planteada requiere el uso de un modelo transaccional para el soporte eficiente de transacciones de escritura concurrente. Los datos enviados desde los sensores implantados en la capa física son transmitidos vía protocolo Http hacia bases de datos transaccionales según los requerimientos.

Durante el desarrollo de esta etapa del proyecto se plantea la cuestión de diseñar e implementar un sistema que atienda estos requerimientos de

forma adecuada y que adicionalmente sea de bajo costo para efectos de obtener el beneficio de una prueba de concepto bajo un prototipo. En este sentido, se analiza el tipo de tecnología recomendado y el esfuerzo que implica el desarrollo y puesta en marcha de un sistema de este tipo.

Con base en lo tratado en el capítulo 1, puntualmente en la Tabla 9, las comparaciones entre las tecnologías comúnmente utilizadas para sistemas transaccionales arrojan que una solución de bajo costo acorde con los requerimientos funcionales y los no funcionales como escalabilidad es la tecnología MySQL. Una ventaja determinante en este contexto con respecto a las tecnologías Microsoft SQL Server y Oracle es la posibilidad de uso sin licenciamiento comercial. En cuanto a la comparación con su contra parte libre de licencia, PostgreSQL, se ve una ventaja en cuanto a la posibilidad de particionamiento en la perspectiva de escalabilidad.

Dentro del proceso investigativo de la iteración 3 se concluyó que el proceso de desarrollo de un sistema transaccional que cumpliera satisfactoriamente con lo requerido implicaba un esfuerzo por fuera del alcance del presente trabajo. Sin embargo, se buscó y encontró una solución apta para la recreación del sistema y que resulta totalmente acorde con los requerimientos de bajo presupuesto además de los funcionales y no funcionales del sistema. La respuesta fue la plataforma *Open Data Kit (ODK)* [38], [39], La cual permite despliegues rápidos de sistemas web soportados en bases de datos MySQL listos para la recepción concurrente de datos. Su implementación detallada y puesta en marcha se encuentra en los Anexos 1 y 2 de este mismo trabajo.

Aun cuando la solución encontrada en la plataforma *ODK* es amplia y suficiente para esta prueba de concepto, en la Figura 13 se presenta una propuesta de diseño del modelo de datos transaccional optimo que de implementarse en el sistema real, cumpliría con buenas prácticas de diseño normalizado. El modelo está consolidado en tercera forma normal y es apto para el almacenamiento de datos provenientes de diversas fuentes. No obstante, se opta por la solución *ODK* teniendo en cuenta que se requeriría un esfuerzo por fuera del alcance actual para desarrollar una interface Http o Https para cumplir fielmente con los requerimientos del prototipo.

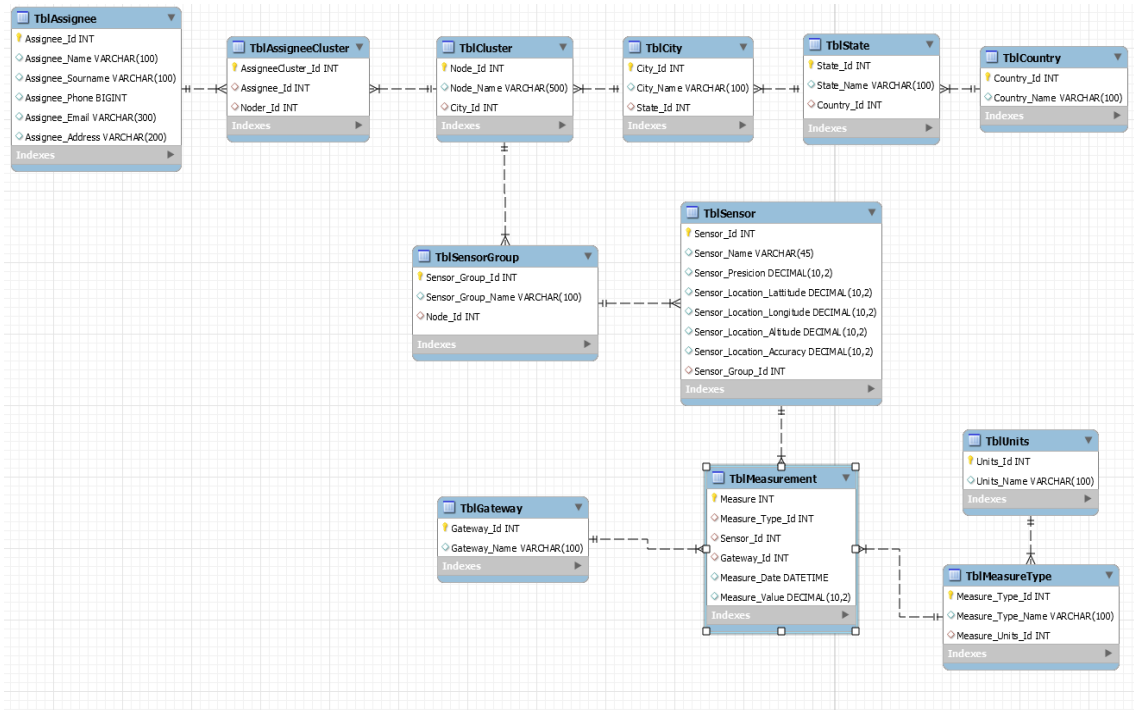


Figura 13 Modelo transaccional 3NF

2.3. Modelo analítico

A pesar de que el modelo analítico se debe implementar en el extremo de explotación, o parte final, su diseño se debe consolidar en una etapa temprana dada la dependencia de la construcción de ETL por este.

El modelo transaccional obedece a reglas de diseño normalizados que buscan optimizar el espacio de almacenamiento de los datos evitando duplicidad y redundancia. En contraste, el funcionamiento de modelos analíticos se diseña a partir de reglas de des normalización que a pesar de romper los criterios de optimización de espacio, facilitan la construcción de análisis con altos rendimientos de ejecución. Para el diseño, se utilizan los conceptos de modelos dimensionales que proponen esquemas de estrella donde su centro lo constituye una tabla de hecho y en sus extremos se ubican tablas de dimensiones relacionadas con el hecho. La tabla de hecho contiene las medidas del evento que se pretende analizar, en tanto que las dimensiones son tablas que almacenan la información detallada de cada una de las ópticas o dimensiones desde las cuales puede ser visto el hecho. Un ejemplo de este modelo se puede ver de la siguiente forma; La tabla de hecho llamada “Medida” representa el acontecimiento o evento susceptible

de análisis que para este contexto es la sucesión de muestras físicas extraídas de los sistemas transaccionales [40]. Tales muestras podrían ser analizadas desde diferentes puntos de vista tales como su comportamiento en el tiempo, por tipo, por fuente, por sensor. Dichos puntos de vista constituyen las dimensiones por las cuales podrán agruparse, totalizarse y en general navegar por las muestras existentes en la tabla de hecho. El termino navegar, se refiere a la posibilidad que otorga la constitución de jerarquías dentro de las tablas de dimensión, por ejemplo la relación día – mes, mes- año permite a los usuarios agrupar y desagrupar los conjuntos de datos analizados de acuerdo a las necesidades.

A continuación en la Figura 14 se presenta la aproximación de diseño lograda para el análisis de muestras que se plantea en el objetivo de este proyecto. Para sustento de este esquema se utilizó la propuesta de Ralph Kimball para la creación de bodegas de datos expuesta en la sección anterior.

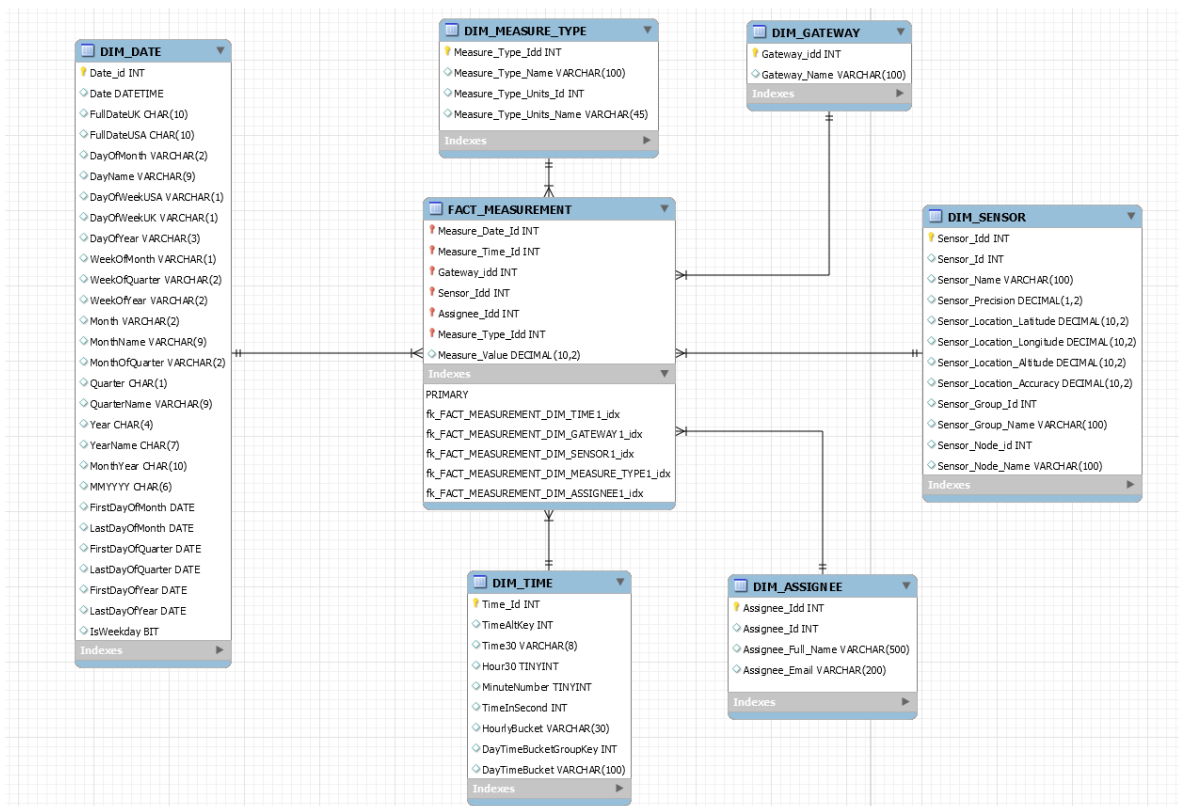


Figura 14 Modelo de estrella para el análisis

El modelo constituye la estructura de un *Data mart* como respuesta al problema de análisis de datos proveniente de diversos sensores. Las ventajas de la aproximación Kimball se ven reflejadas en la facilidad de concepción y las posibilidades de crecimiento de la bodega mediante el enlace dimensional de futuras estructuras. Las propiedades del esquema de estrella se modelan con base en los conceptos exhibidos por [40].

Una vez completado este diseño, se abre la puerta para la concepción de otros elementos incluidos en la arquitectura planteada tales como los extractores de las bases transaccionales y los algoritmos de transformación y carga. Es fundamental conocer los modelos de origen y fin al momento de construir los artefactos necesarios para el procesamiento de los datos dado que con base en estos se origina la lógica de transformación y reducción que hace posible el reporte y análisis posterior.

2.4. Diseño de ETL

El diseño y construcción de los procesos de extracción, transformación y carga representan uno de los retos más importantes dentro de la constitución de las soluciones de inteligencia de negocios como se ha mencionado previamente en este trabajo. Para lograr el objetivo de consolidar los datos provenientes de varias redes de sensores, de forma eficiente, escalable, flexible y que finalmente como resultado puedan ser estructurados de una manera que puedan ser almacenados en la bodega de datos para análisis posteriores, varios elementos se deben tener en cuenta [30].

- En primera instancia, el proceso de extracción de datos debe ser basado en la identificación de variaciones presentes en las fuentes en vez de realizar una extracción total cada periodo de ejecución.
- La programación de las tareas de extracción, transformación y carga deben ser sincronizadas de forma automática según una frecuencia estipulada.
- El sistema de almacenamiento intermedio o también conocido como *staging* debe ser escalable en la medida en que se agregan nuevas

bases de datos relacionales encargadas de retener las muestras recolectadas en la capa física.

- El proceso de transformación de datos debe ser igualmente escalable y flexible con el fin de soportar altos volúmenes de información entrante y ofrecer tiempos de respuesta aceptables de acuerdo a los análisis de riesgo que se deseen implementar.

Bajo estas premisas se plantea un sistema que da respuesta de forma efectiva al problema:

Un sistema de extracción automatizado que utilice periodos de ejecución parametrizables. Los resultados de las diversas fuentes deben ser depositados de forma asíncrona en un componente de almacenamiento que tolere concurrencia. Posteriormente, aprovechando las nuevas tecnologías de *Big Data* para el procesamiento paralelo de datos, un cluster compuesto por un numero configurable de máquinas debe repartir la información entrante entre los n componentes del cluster de forma que el procesamiento, agrupación y posterior estructuración de los datos ocurra de forma paralela garantizando la escalabilidad horizontal y vertical del sistema. Finalmente, el proceso de carga debe garantizar el traslado de los datos ya estructurados de forma adecuada para el análisis hacia una el modelo dimensional descrito en el apartado anterior. Una opción viable para lograr el retante proceso de transformación se encuentra en el uso del ecosistema *hadoop* descrito en 1.3, el cual hace posible la constitución de un set de equipos dispuestos para el procesamiento paralelo de altos volúmenes de datos haciendo posible alcanzar la escala de operación de grandes exponentes como Facebook [41] Yahoo [42] y Twitter [43], empresas que han venido usando masivamente el uso de estas tecnologías [23].

En resumen, el proceso de extracción, transformación y carga debe ser orquestado de forma automática por un mecanismo en la nube que integre de forma efectiva las tecnologías relacionales tradicionales con aquellas nuevas tendencias de *Big Data* como *Map Reduce* o *Hive* para lograr la flexibilidad y escalabilidad propuestas en los requerimientos no funcionales. Es importante resaltar que la estructuración de los datos desde su forma original en el sistema transaccional hasta el modelo dimensional en la bodega de datos se logra fácilmente mediante consultas SQL en las etapas de extracción y transformación. Los detalles de construcción teniendo en

cuenta los modelos presentados en las secciones 2.2 y 2.3 se evidencian en el anexo 3.

2.5. Implementación en la nube

Uno de los requerimientos clave de implementación indica que el diseño propuesto sea una solución en la nube con una estructura de costos de plataforma como servicio. Inicialmente, el presente prototipo busca ser desplegado en un proveedor de servicios en la nube que facilite la verificación de la prueba de concepto en un presupuesto bajo, pero que de ser verificada la validez de la misma, pueda ser escalada hacia la versión real.

Teniendo en cuenta el análisis completado en el capítulo anterior con respecto al contexto actual de la nube, sus proveedores y costos por servicio (ver sección 1.2), se concluyó que el modelo que se busca implementar se ajusta adecuadamente a las ofertas del proveedor Amazon Web Services. Una de sus grandes ventajas, además de ser el líder actual en prestación de servicios en la nube a nivel mundial, es la facilidad de contar con servicios de prueba sin costo por un periodo de 1 año. Este servicio, es un beneficio para nuevos usuarios que desean comprobar el funcionamiento de sus soluciones a escala mínima sin incurrir en costos de infraestructura como ocurriría en otros proveedores de estos mismos servicios.

Habiendo mencionado lo anterior, en la Figura 15 se ilustra la implementación tentativa de la arquitectura mostrada en la Figura 12 Esta propuesta, resolvería satisfactoriamente los requerimientos funcionales y no funcionales de los que se trata la prueba de concepto. Sus elementos serán descritos de forma detalla en los próximos párrafos en tanto que su implementación y puesta en marcha se verán reflejados en el transcurso de los anexos a este documento. Vale la pena resaltar que a pesar de que esta implementación se establece sobre las características de funcionamiento del proveedor AWS, el modelo planteado en la Figura 12 es una estructura diseñada de forma genérica y presenta la solución conceptual al problema siendo agnóstica a la implementación que se elija en uno u otro proveedor.

El ciclo de vida de la implementación discutida transcurre como se exhibe a continuación; En la capa física, extremo izquierdo de la figura, se encuentran implantados n *clusters* de sensores cada uno con n sensores en su interior.

Cada *cluster* cuenta con un colector o *Gateway*, que recibe las muestras tomadas en cada uno de los sensores y además cuenta con acceso a internet para el envío de datos a través de los protocolos http o https. Por cada cluster se tiene un servidor tipo EC2 disponible cien por ciento del tiempo, cada uno de los cuales tiene expuestos servicios web para la recepción de muestras y almacenamiento de las mismas en bases de datos MySQL a las que cada una tiene acceso. Estas bases de datos MySQL contendrían la estructura transaccional propuesta en la sección 2.3. Las instancias EC2 tienen la característica de ser escalables de forma vertical bajo demanda y el diseño permite la inclusión de n instancias de este tipo para la recepción de datos, lo cual es traducido como una posibilidad de escalamiento ilimitada para este proceso (ver Anexos 1, 2 y 3 para implementación). En la parte inferior se presenta un servicio de automatización de tareas llamado Amazon Data Pipeline, el cual es un coordinador de tareas que puede ejecutar trabajos de forma automática según una frecuencia establecida. En este sentido, se dice que la orquestación de toda la solución se logra mediante este componente. El primer trabajo que se ejecuta en el servicio descrito previamente realiza la extracción de todas las muestras recolectadas en las bases de datos MySQL y las deposita en un servicio de almacenamiento intermedio o *staging* llamado Amazon S3. Es importante anotar que la extracción solo contempla las nuevas muestras incluidas en las bases de datos, identificando el delta en cada una de ellas de acuerdo con la frecuencia de ejecución establecida. Una vez realizada la extracción de forma satisfactoria, el programador de trabajos procede con el lanzamiento de un *cluster hadoop* a donde son trasladadas de forma paralela todas las muestras ubicadas en el servicio S3. Las ventajas y descripción general del uso de *hadoop* para el procesamiento de datos son discutidas en la sección 1.3. Una vez el cluster conocido como Amazon EMR ha sido configurado, de forma automática se ejecuta un algoritmo de transformación de los datos escrito en el lenguaje HQL, el cual hace una agrupación de todos los registros según la granularidad establecida para la bodega de datos y completa su tarea al enviar los datos resultantes en la estructura correcta hacia la bodega de datos. La bodega se implementa de acuerdo al modelo analítico descrito en la sección 2.3 en un servicio de base de datos conocida como Amazon RDS. Este último utiliza un motor MySQL para efectos de bajo costo, aunque podría ser elegido cualquier otro motor licenciado en el evento de un presupuesto mayor. Como anotación, el diseño evidencia en su extremo derecho un conjunto de herramientas de explotación que podrán ser usadas sobre los

componentes finales. Por ejemplo, sobre la bodega RDS podrán ser usadas herramientas tradicionales de explotación BI como Microstrategy, Tableau, Pentaho, entre muchas otras presentadas en el cuadrante mágico de la Figura 10. En tanto que sobre el cluster EMR, de ser dejado todo el tiempo en ejecución, podrían utilizarse herramientas de BI y de geolocalización dado que *hive* es en esencia una bodega de datos distribuida que admite conexiones JDBC. No obstante, sus iconos se dejan en gris con el fin de exhibir su posible uso pero no son incluidas en el alcance del presente texto.

Esta implementación es concebida con los servicios encontrados en el proveedor AWS. Sin embargo, otros proveedores de servicios en la nube prestan herramientas similares que harían posible la migración de este diseño hacia tales servicios sin incurrir en una dificultad representativa.

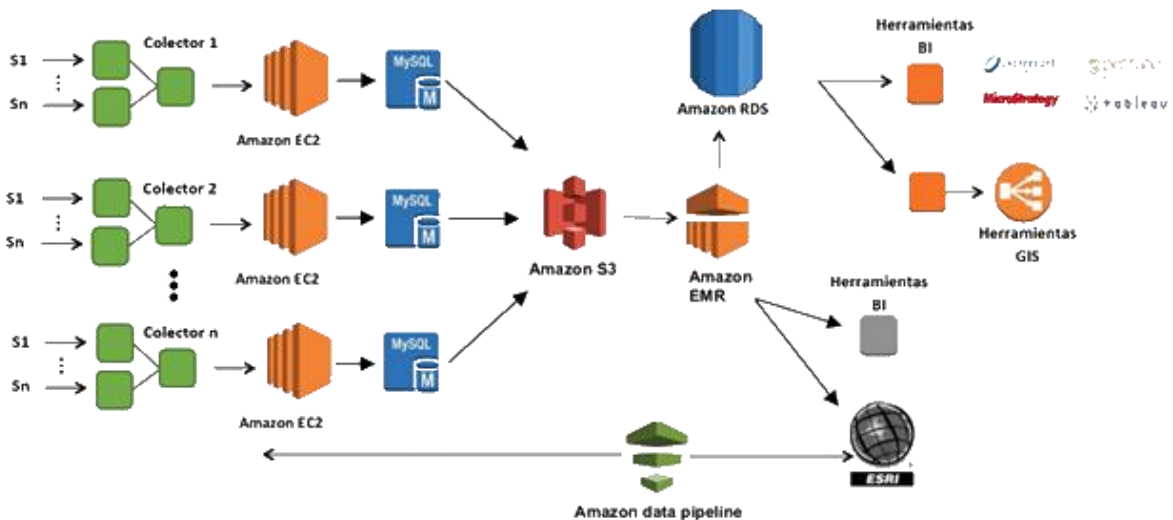


Figura 15 Implementación en la nube

Es vital presentar en este punto la forma como se logra la orquestación entre todos los servicios en la nube que hacen posible la ejecución programada de trabajos en los diferentes componentes del sistema y que en síntesis dan cumplimiento a los procesos de ETL. El servicio encargado de las labores de automatización es llamado *Data Pipeline* y su constitución es en consecuencia el corazón del sistema que hace posible el logro de los

requerimientos funcionales y no funcionales del trabajo. Su esquema real se presenta en la Figura 16

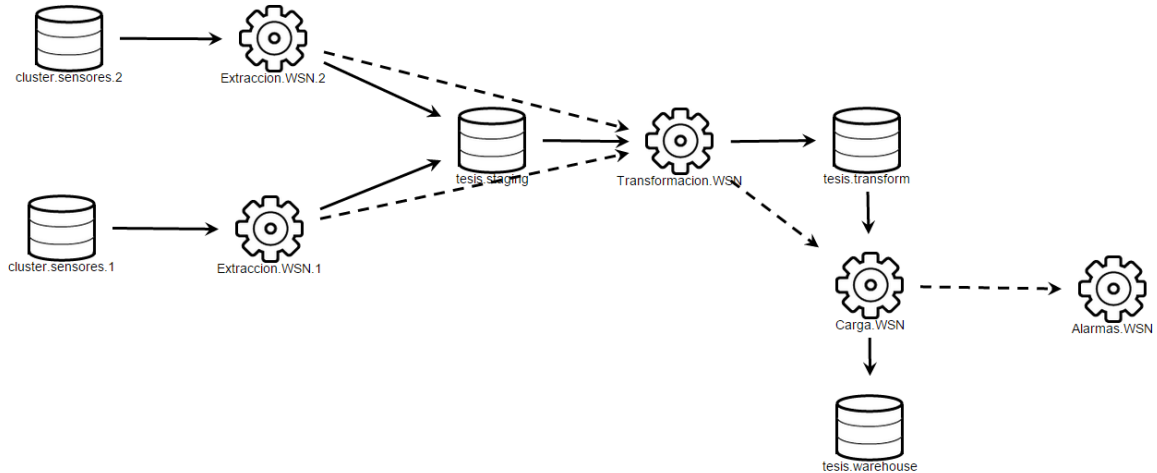


Figura 16 Orquestación de servicios en la nube

2.6. Elección de Explotación

Una vez consolidadas las etapas de extracción, transformación y carga de los datos de forma automática, la fase final representa la disposición del contenido para consumo del cliente. Esta fase de explotación, como se ha indicado previamente en este texto raras veces es desarrollada a la medida para la solución, por el contrario es comúnmente lograda a través de herramientas de inteligencia de negocios ya existentes en el mercado que permiten crear meta datos de la bodega existente y previamente poblada.

El esquema de funcionamiento de las herramientas existentes en el mercado es similar en cuanto a la configuración inicial y algunas funcionalidades. Sin embargo, tal y como se muestra en [31] existen características entre algunos proveedores de soluciones que hacen la diferencia en el mercado. Dado que no se encuentra en el alcance de este proyecto consolidar una evaluación de las posibles soluciones sino llevar a cabo la implementación de alguna de las alternativas disponibles, la elección de la herramienta a utilizar se basa tomando en cuenta los exhaustivos análisis realizados por [31] y la pertinencia de tales con los requerimientos del proyecto actual.

Teniendo en cuenta que uno de los requerimientos clave del presente prototipo es conseguir una implementación funcional de bajo costo, se tuvieron en cuenta aquellas soluciones mostradas en la Figura 10 que disponen de alguna opción de uso libre no expirable. Se encontró, que dentro de las herramientas líderes del cuadrante, solo Tableau presenta una edición de uso libre para la comunidad con la obligación de uso compartido en internet en adición a un máximo de 10 Gb de almacenamiento. A pesar de que esta herramienta es atractiva por su posición en el mercado y su capacidad de almacenamiento es suficiente para el prototipo, la condición de uso compartida no es viable dado que la infraestructura en la nube tiene un costo bajo demanda que en un escenario público generaría costos no aceptables en el requerimiento económico.

Al investigar más acerca de las herramientas cercanas al centro del cuadrante, se encontró que una solución bien reconocida en el medio llamada Pentaho [44] ofrece una versión a la comunidad con licencia abierta no comercializable que cumple con todas las expectativas de explotación identificadas en los requerimientos del proyecto. Esta solución, permite configurar cubos de explotación sobre bodegas de datos de diferentes tipos, entre ellas MySQL motor sobre el cual fue implementada la de este prototipo. El proceso de configuración de los cubos se realiza haciendo uso de la herramienta *Schema Workbench* provista por Pentaho. Dicha herramienta, es una interfaz de creación de archivos XML aptos para la carga de meta datos al motor de explotación usado por Pentaho llamado *Mondrian*. Una vez un cubo ha sido configurado en *Schema Workbench*, podrá ser publicado en la herramienta *Business Analytics Platform* implementada sobre *Mondrian* como paquete de explotación. La *Business Analytics Platform* es un sistema con interfaz web donde podrán crearse reportes, tableros de control y otras aplicaciones de forma rápida e interactiva. Su implementación es lograda sobre el servidor web Tomcat y su instalación es gratis para sistemas no comercializados.

Con las dos herramientas descritas en el párrafo anterior, se logró construir y publicar un cubo OLAP sobre el modelo analítico implementado en la sección 2.3 para la explotación. La Figura 17 evidencia el resultado de la construcción.

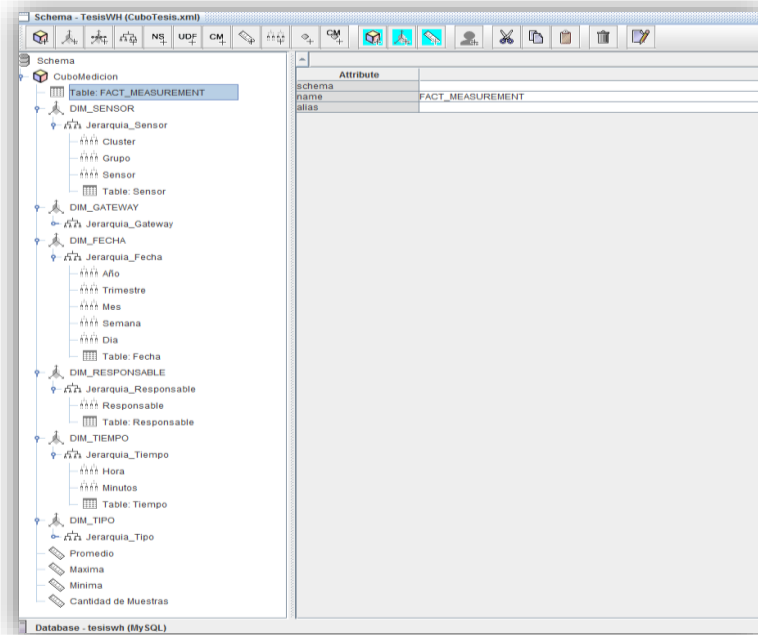


Figura 17 Cubo creado para Pentaho

El prototipo cuenta a partir de este punto con una interfaz web de explotación que puede servir para realizar pruebas de funcionamiento que validen los criterios de aceptación. El proceso de creación y publicación se logró siguiendo la extensiva documentación con la que cuenta la aplicación en [44]. Adicionalmente, la contextualización sobre el uso y la aplicación del motor *Mondrian* fue obtenida a partir de [45].

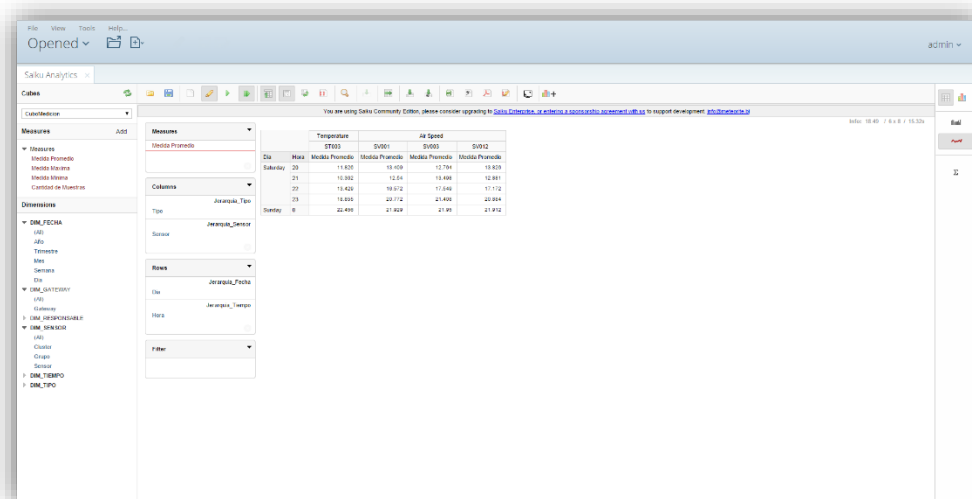


Figura 18 Reporte de cubo mondrian

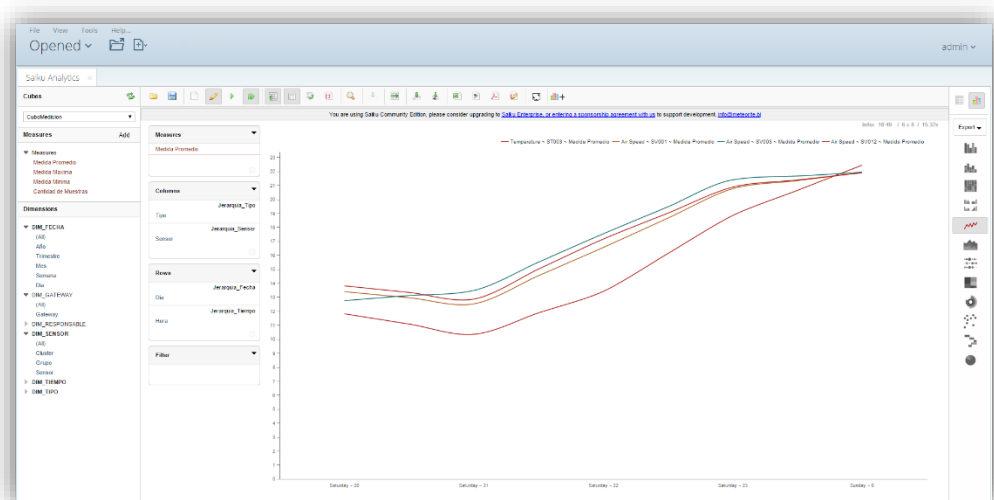


Figura 19 Visualización gráfica de datos

En cuanto a la configuración y uso de *Business Analytics Platform*, la documentación utilizada fue nuevamente [44]. Su uso es bastante intuitivo y en comparación a otras herramientas de inteligencia de negocios, se encontró una ventaja representativa en términos de extensibilidad gracias a la existencia de un *Market Place* para la descarga e instalación de funcionalidades desarrolladas por la comunidad para tareas que la herramienta en su presentación nativa no podría completar. Lo anterior implica también que hay una posibilidad de desarrollar nuevas funcionalidades adaptables al sistema en caso de ser requerido y de extender algunas ya existentes en el evento de no solucionar un caso de uso determinado.

La Figura 18 y Figura 19 presentan la evidencia del uso de la herramienta Pentaho *Business Analytics Platform* en su versión para la comunidad. En ambas se puede apreciar la disponibilidad del cubo creado y publicado descrito en los párrafos anteriores. Su uso es intuitivo y permite sacar un óptimo provecho de los datos de hecho disponibles en contraste con todas sus dimensiones. Del mismo modo puede crearse gráficos dinámicos dependientes de filtros y eventos de interfaz de usuario, lo cual es fundamental para lograr efectivamente la toma de decisiones en un contexto dado.

Finalmente, antes de terminar la descripción de la implementación, se deja como comentario que esta fase es la más variable de todo el proceso dado el universo de posibilidades que existe para lograrlo. Sin embargo, el determinante de la mejor opción es en cualquier caso el presupuesto disponible ligado al conjunto de requerimientos funcionales en temas de visualización, análisis y reportes. Pentaho, para este prototipo específico cumple con todos los requisitos contemplados y es apto para dar solución al problema planteado en este trabajo.

CAPITULO 3

3. PROPUESTA DE ANÁLISIS

Una vez configuradas e implementadas las herramientas requeridas para la consolidación del trabajo, es necesario describir la propuesta de análisis que den solución al problema planteado. El propósito de este capítulo es puntualizar el procedimiento sugerido para el análisis de los datos que puedan consolidarse sobre los modelos diseñados a lo largo del capítulo anterior.

Para iniciar, es importante resaltar que el contexto de análisis para el presente proyecto se determina por grandes series de tiempo provenientes de diversas fuentes que deben ser interpretadas en cortos periodos de tiempo para arrojar resultados que puedan servir en la gestión de eventos de riesgo. Lo anterior es importante dado que este tipo de series de tiempo son por una parte susceptibles de análisis profundos utilizando técnicas de minería de datos para la identificación de patrones [46] pero también podrían aplicarse reglas parametrizadas de lanzamiento de alertas en adición a la disposición de herramientas de visualización continua para la identificación de eventos. Con base en lo anterior, se realizan propuestas de análisis e interpretación de los datos en los tres puntos que resultan de especial interés para el presente trabajo, esto es; Minería de datos, Parametrización de reglas y visualización de datos.

3.1. Minería de datos

La minería de datos es un proceso analítico diseñado para explorar, usualmente, grandes volúmenes de datos en busca de patrones y/o relaciones sistemáticas entre variables, teniendo como último fin permitir realizar predicciones verificables sobre nuevos conjuntos de datos [33]. La minería de datos es un campo multidisciplinario que incluye técnicas de aprendizaje de máquinas, estadística, inteligencia artificial, teorías de información y visualización entre otros para descubrir patrones estructurales que permitan construir modelos predictivos [47].

Lo anterior implica que la aplicación de las técnicas de minería de datos contiene un extenso fundamento técnico que se encuentra por fuera del alcance de este proyecto explorar a profundidad. No obstante, se pretende hacer una propuesta aplicable a futuros desarrollos de este prototipo utilizando algunos conceptos relevantes enmarcados en una metodología de exploración frecuentemente utilizada para la minería de datos.

A manera de propuesta, se pretende esbozar el contexto de la metodología CRISP-DM con base en [48] para el desarrollo de modelos predictivos que puedan ser útiles en el sistema de gestión de riesgos. Algunas herramientas gratuitas son igualmente comentadas con el fin de dar un contexto a los comentarios realizados sobre los algoritmos sugeridos.

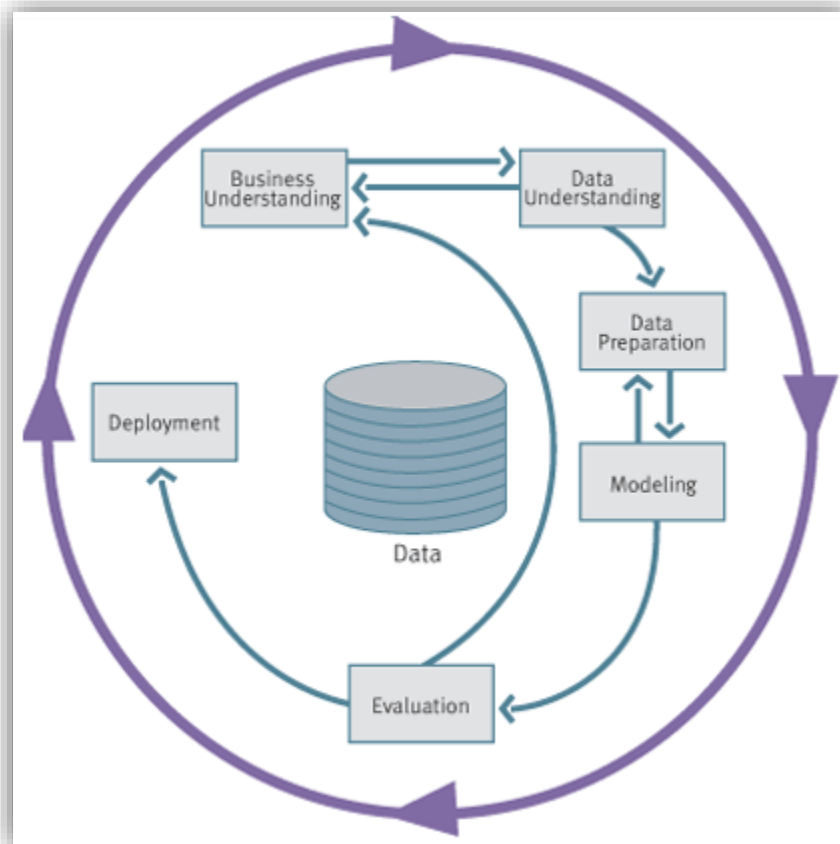


Figura 20 Metodología CRISP-DM. Figura tomada de <http://goo.gl/SMEI7L>

Como se aprecia en la Figura 20, el ciclo de vida de la propuesta CRISP-DM consta de 6 fases relacionadas entre sí de una manera retroalimentada durante su ejecución. El proceso de exploración se inicia en una fase de entendimiento

del negocio donde deben ser completadas una serie de tareas orientadas a la comprensión del contexto general de funcionamiento así como el objetivo general del escenario y el objetivo de la minería. Para este caso puntual, esta fase es resuelta a lo largo del planteamiento del problema y el cuarto objetivo de este trabajo.

La segunda fase de este proceso plantea el entendimiento de los datos desde una recolección preliminar hasta su descripción detallada y análisis de calidad. Para este proyecto, esta fase puede ser obtenida de las etapas de modelamiento relacional y analítico presentadas en secciones anteriores, en tanto que la calidad puede ser recomendada como criterio de aceptación futura aunque no aplica para este alcance dada la condición de prototipo al nivel de prueba de concepto.

A lo largo de la tercera fase, se propone una estructura de los datos disponibles que sea apta para el análisis, incluyendo formatos adecuados, mezcla de varias fuentes, limpieza de condiciones inadecuadas y construcción de atributos adicionales. Esta etapa deja como resultado una vista minable que puede ser luego analizada para la construcción de modelos. Para este proyecto, se propone una vista consolidada y des normalizada de las muestras cargadas a la bodega de datos que pueda servir para la identificación de patrones históricos por tratarse de series de tiempo. Una primera aproximación se presenta en Figura 21.

Column	Type
◇ Node_id	int(11)
◇ Node_name	varchar(100)
◇ Assignee_id	int(11)
◇ Assignee_full_name	varchar(500)
◇ Assignee_email	varchar(200)
◇ Sensor_group_id	int(11)
◇ Sensor_group_name	varchar(100)
◇ Sensor_id	int(11)
◇ Sensor_name	varchar(100)
◇ Sensor_precision	decimal(3,2)
◇ SENSOR_LOCATION_...	decimal(10,2)
◇ SENSOR_LOCATION_...	decimal(10,2)
◇ SENSOR_LOCATION_...	decimal(10,2)
◇ SENSOR_LOCATION_...	decimal(10,2)
◇ Measure_type	varchar(100)
◇ Measure_units	varchar(45)
◇ Measure_value	decimal(10,2)
◇ Gateway_name	varchar(100)
◇ MEASURE_DATE	datetime

Figura 21 Vista preparada de datos

La cuarta fase constituye un proceso clave en el desarrollo de la metodología, dado que es en esta donde se realiza el modelado de los datos y se encuentran preliminarmente las relaciones estructurales entre las variables del contexto. En este punto se eligen las técnicas de modelamiento, los casos de prueba y la configuración de parámetros del modelo. Para el caso de este proyecto, es en esta etapa donde podrían ser utilizadas herramientas como *Weka* o *Rapid Miner* para la construcción de modelos basados en algoritmos aplicados sobre los datos.

Teniendo como propósito un enfoque práctico en esta etapa, se busca de forma efectiva presentar los posibles algoritmos aplicables a los conjuntos de datos más que profundizar en los conceptos puros que implica este campo. Lo anterior se menciona teniendo en cuenta que el objetivo de esta etapa se centra en proponer un conjunto de técnicas analíticas básicas más allá de llegar a presentar el contexto matemático o la aplicabilidad de todos los métodos existentes.

A continuación se describe un posible camino de análisis para el modelo minable propuesto previamente:

- **Técnicas no supervisadas**

La aplicación de técnicas no supervisadas sobre el conjunto de datos puede tener un resultado interesante si se realiza continuamente, es decir, la segmentación de la información entrante en grupos de análisis que contenga muestras con características similares puede arrojar información relevante para la toma de decisiones si se compara con ciclos previos.

Para lo anterior puede ser sugerido el uso del algoritmo de agrupación de datos *K-Means*, el cual es ampliamente utilizado por su facilidad y para el caso de este proyecto permitiría la segmentación continua de las muestras entrantes a la bodega de datos para la identificación de patrones. Una ventaja relevante de este algoritmo es que permite elegir el número de clusters a formar, número que es determinante en el éxito del análisis y puede ser definido por especialistas que conozcan profundamente el contexto. Este algoritmo es ampliamente utilizado y puede encontrarse en ambas herramientas mencionadas previamente.

Para su aplicación, deben seleccionarse atributos del conjunto de datos que permitan generar una agrupación diciente entre las muestras. Por ejemplo, en este contexto podría realizarse un intento de clasificación de los tipos de medida, sus unidades, y el valor de la muestra, permitiendo identificar los niveles de operación existentes en el conjunto de datos. Este es solo un ejemplo de los posibles clusters que podrían ser generados, pero se deja como anotación que otro tipo de agrupaciones es posible siempre y cuando los datos analizados no sean categóricos.

- **Técnicas supervisadas**

El uso de técnicas supervisadas implica la existencia de un set de datos históricos que pueda servir de entrenamiento en adición a un conjunto de datos de evaluación independiente que haga posible la validación del modelo. A pesar de que en el presente prototipo no se cuenta con un conjunto de datos reales que pueda ser utilizado para este tipo de análisis, si es posible sugerir el uso de ciertos algoritmos con base en el modelo de datos consolidado y el objetivo de la minería. Para este caso, el objetivo se traza para identificar situaciones de riesgo y lanzamiento de alertas tempranas con base en predicciones hechas a partir de un modelo construido con fundamento en eventos históricos.

Con base en lo anterior se presentan una serie de algoritmos aplicables a este escenario que podrían dejar como resultados robustos modelos matemáticos de predicción en el evento en que se cuente con un soporte histórico de datos suficiente para lograr la generalización. Nuevamente, se debe tener en cuenta que la cantidad de algoritmos disponibles para la clasificación es vasta y que en el alcance de este trabajo solamente se pretende esbozar el uso de algunos entre los más comunes.

Redes Neuronales:

Las redes neuronales son algoritmos de clasificación ampliamente utilizados en el campo de inteligencia de negocios. Su correcto funcionamiento es generalmente logrado mediante ejercicios de prueba y error entre sus parámetros de configuración como número de capas ocultas, número de neuronas en sus capas ocultas, tasa de aprendizaje y función de activación. En este sentido, se propone el uso de estos algoritmos una vez se tengan registros históricos suficientes para obtener resultados aceptables de predicción. Se debe tener en cuenta, que para obtener la clasificación adecuada para la detección de

eventos de riesgo, se debe contar con una base de datos histórica de entrenamiento donde se contengan eventos de riesgo, de lo contrario las redes neuronales no servirían como algoritmo de identificación de alarmas.

Perceptrón multicapa

Se recomienda el uso de 3 o más capas para la configuración del perceptrón con el fin de posibilitar la aproximación de funciones de alta complejidad. Su configuración como se dijo previamente es un proceso de prueba y error entre sus parámetros. *Weka* cuenta con este algoritmo y es posible configurar todos sus parámetros a excepción de su función de activación. Su uso permitiría predecir alertas de riesgo con base en patrones existentes en eventos pasados entre las variables analizadas. Este mismo algoritmo podría ser utilizado en tareas de predicción de valores de las variables numéricas, como la medida tomada por un sensor, en intentos de identificación de tendencias en las series de tiempo.

Redes de base radial

Redes neuronales configurables para la clasificación de muestras en un número determinado de clusters. Este algoritmo podría ser de inmensa utilidad por su precisión si eventualmente este proyecto obtiene una definición clara de los grupos de riesgo ya sea aplicando técnicas supervisadas como se presentó previamente o una formación específica por conocimiento experto. Este algoritmo permitiría clasificar los nuevos datos entrantes en tales cluster predefinidos y de esta manera se obtendrían las alertas de riesgo. También podría ser utilizado en tareas de predicción de valores de las variables numéricas.

Máquinas de soporte vectorial:

Es un algoritmo complejo que se utiliza comúnmente en tareas de clasificación por su alta precisión. Este algoritmo tiene un alto coste computacional y es comparable en relación a las aplicaciones dichas previamente para el perceptrón multicapa.

Naive Bayes

Este algoritmo se utiliza para calcular la probabilidad de que una nueva muestra pertenezca a una clase. Es aplicable a datos categóricos y por tal razón es propuesto solo en aquellos atributos de la vista minable que no son numéricos.

Arboles de decisión

Los arboles de decisión son un grupo amplio de algoritmos aplicables para variables categóricas. Su resultado es un conjunto de reglas visuales que facilitan la toma de decisiones ante la entrada de nuevas muestras. Su limitación radica en que su uso no es recomendable en conjuntos de datos con un alto número de campos.

J48

Este es un algoritmo comúnmente utilizado en tareas de clasificación categórica y podría ser implementado en este proyecto para determinar las reglas de lanzamiento de alertas en caso de tener un soporte histórico que así lo permita. Existen muchos otros algoritmos para la construcción de árboles de decisión que podrían ser utilizados en esta etapa.

Lógica difusa

Con base en [47] se puede proponer el uso del algoritmo FURIA (*Fuzzy UnorderedRule Induction Algorithm*) el cual permite obtener un conjunto de reglas difusas en vez de las tradicional reglas estrictas. Se propone dado que este tipo de algoritmos permitiría en el sistema identificar posibles alarmas sin tener todos los ejemplos cubiertos, lo cual es una restricción común en los demás algoritmos supervisados descritos.

Hasta este punto se han presentado algoritmos que podrían solucionar los problemas de segmentación utilizando la técnica no supervisada propuesta, clasificación con las técnicas supervisadas para datos categóricas, predicción con las técnicas supervisadas aplicadas a datos numéricos. En cuanto las tareas de asociación, podrían hacerse uso de los filtros disponibles en la herramienta *Weka* para establecer reglas de asociación entre las variables presentes en eventos de riesgo reales almacenados en la base de datos explotada.

La quinta etapa indica la ejecución de tareas de evaluación sobre los modelos obtenidos, las cuales deben evaluar los resultados frente a muestras reales, aprobar los modelos adecuados y determinar los pasos a seguir dentro del ciclo de explotación. Esta fase representa el proceso más importante del ciclo dado que sobre su ejecución se toman decisiones que agregan o destruyen valor en el sistema real.

Finalmente, dentro de la etapa 6 se ejecuta el proceso de despliegue que incluye los planes de monitoreo y mantenimiento y el reporte final del ciclo. Para el sistema que sea desarrollado sobre este prototipo, se recomienda la recopilación de lecciones aprendidas así como un énfasis especial en los planes de mantenimiento con el fin de garantizar la sostenibilidad del mismo en el tiempo.

3.2. Parametrización de reglas de negocio

La parametrización de reglas para el lanzamiento de alertas es una funcionalidad que se propone con el fin de demostrar alcances futuros que puede tener el sistema que se desarrolle con base en este prototipo. Su propósito es establecer condiciones paramétricas de alerta sobre un conjunto de datos. Su aplicación podría ser establecida en aquellos contextos donde se conocen las reglas que rigen algunos eventos de riesgo y en consecuencia los lanzamientos de alerta podrían ser automatizados bajo el cumplimiento de alguno de los criterios.

Con el fin de lograr lo anterior desde la infraestructura en la nube establecida para este proyecto, es necesario incluir una actividad adicional al proceso de ETL. La Figura 16 incluye dicha actividad y fue llamada "Alarmas.WSN". Lo que esta actividad anexa realiza es la ejecución de un conjunto de consultas sobre la bodega de datos ya actualizada con las últimas muestras, buscando en los nuevos registros algún comportamiento ajustado a una regla parametrizada previamente. En caso de ser encontrado algún conjunto de muestras que cumplan con los criterios de alarma, este es extraído en un reporte en formato CSV y enviado al grupo de correos deseado por los usuarios. Este proceso requiere del uso de un recurso del tipo EC2 en los servicios de Amazon, el cual debe tener configurado el motor de base de datos para realizar su conexión a la bodega y un cliente de correo para el envío del reporte. El proceso de configuración se realizó de la siguiente manera:

1- Configuración de instancia EC2

Haciendo uso de la plataforma web de Amazon se creó una instancia EC2 tipo t1.micro basado en [49]. Luego de establecer la conexión ssh con la instancia, se continúa la configuración de MySQL y el cliente de correo SendEmail.

2- Configuración MySQL

La configuración del servidor MySQL se realiza de la forma descrita en el Anexo 1 a este documento.

3- Configuración SendEmail

La configuración del cliente de correo para CentOS se realiza siguiendo la documentación expuesta en las secciones de su página web de creación ¹. Sin embargo, no se referencia su documentación dado que cualquier cliente de correo puede ser utilizado para esta tarea.

4- Creación de archivo bash

Una vez configuradas ambas aplicaciones, se creó un archivo bash que contiene los comandos de ejecución necesarios para consultar la bodega de datos y extraer los resultados en un archivo csv que sería enviado por el cliente previamente parametrizado. La plantilla de comandos utilizados se presenta en el Anexo 4 a este documento.

5- Creación de Actividad en Data Pipeline

Finalmente, el proceso de automatización se logra creando una imagen de la instancia configurada como se indica en [50] y creando la actividad mencionada en la introducción de esta sección como se muestra en la Figura 22. Los parámetros de la actividad llamados *Script Argument* son los valores que determinan los criterios de riesgo. En el siguiente capítulo se describe su funcionamiento en el escenario de prueba.

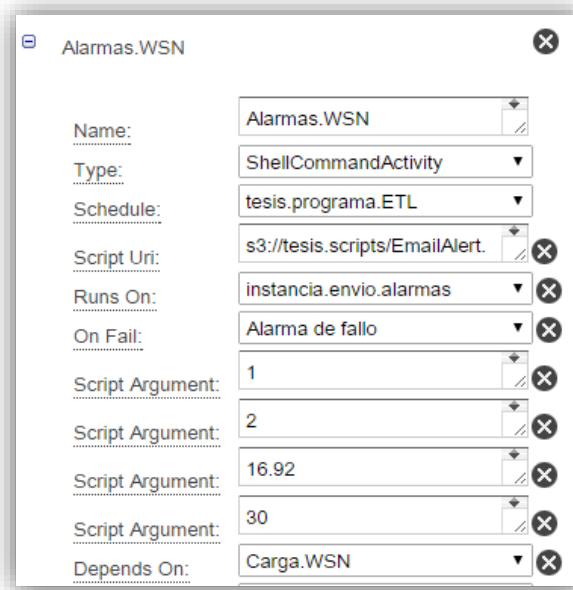


Figura 22 Actividad de lanzamiento de alertas

¹ <http://caspiantdotconf.net/menu/Software/SendEmail/>

El procedimiento descrito representa un valor alto para el proyecto en términos de flexibilidad para la parametrización. Una vez es lograda la configuración de la instancia encargada de ejecutar las consultas y enviar las alertas, la creación de procedimientos almacenados y de consultas parametrizables es ahora el camino abierto hacia la identificación de patrones de riesgo entre los conjuntos de muestras almacenadas en la bodega de datos diseñada. Como se puede observar en la Figura 22, es posible introducir múltiples parámetros para la evaluación de escenarios en los datos, en adición a la posibilidad de crear ilimitadas actividades de lanzamiento de alertas de acuerdo con los requerimientos propios del sistema.

3.3. Visualización de datos

La visualización de datos es un importante campo en la inteligencia de negocios en aquellos procesos de generación de conocimiento a partir de los datos. La interpretación visual permite a los usuarios, en diversos contextos, obtener rápidamente detalles no triviales de lo que puede estar sucediendo como material de entrada en sus tareas de toma de decisiones. Particularmente, las voluminosas series de tiempo son exitosamente resumidas en elocuentes gráficos que pueden servir de referencia para la planeación de acciones en atención al entendimiento de los acontecimientos.

Actualmente, las herramientas de inteligencia de negocios han abierto sus fronteras de análisis visuales en diversos frentes. Los paquetes ofrecen más a menudo la posibilidad de presentar datos visualmente de forma interactiva, permitiendo al consumidor entrelazar numerosos tipos de gráficos en atractivos tableros de control que resultan de alto valor para quienes deben gestionar cambios. En adición, Una importante característica que se encuentra cada vez más abundantemente hoy en día es la posibilidad de crear gráficos geo referenciados, los cuales son de alta importancia y relevancia para este proyecto.

Teniendo en cuenta lo anterior, el uso de gráficos como parte del proceso de análisis de las muestras provenientes de diversas fuentes es una técnica clave que puede ser explotada desde la mayoría de herramientas especializadas en inteligencia de negocios así como también desde el paquete Excel de la suite Office de Microsoft comúnmente utilizado para el procesamiento de datos. Un requerimiento importante en la visualización de series de tiempo, específicamente en las series generadas por muestras de sensores tomadas en los campos implantados, es la posibilidad de evidenciar la evolución en el tiempo de algunas o todas las variables de estudio en zonas determinadas. Adicionalmente, la posibilidad de interactuar simultáneamente relacionando distintas variables, tanto en gráficos tradicionales como se muestra en Figura 19 así como en mapas de calor y de bolas como se evidencia en Figura 23 Figura 24 añade un alto valor en los esfuerzos de toma de decisiones para la gestión de riesgo.

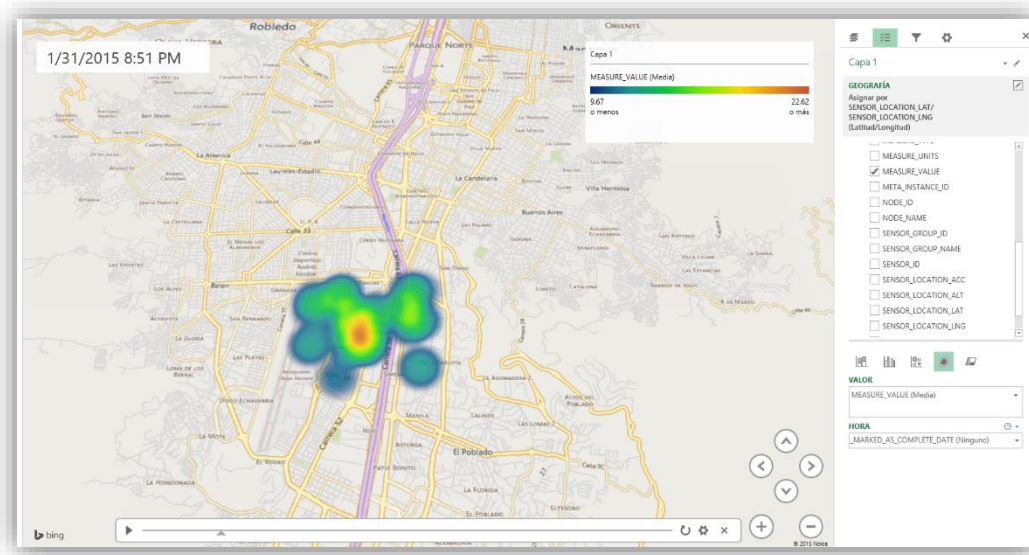


Figura 23 Mapa de calor basado en muestras de sensores

Los gráficos de calor resultan de especial interés en este trabajo dado que mediante su uso puede comprenderse de forma directa el estado de una zona de variable de análisis en las zonas implantadas.

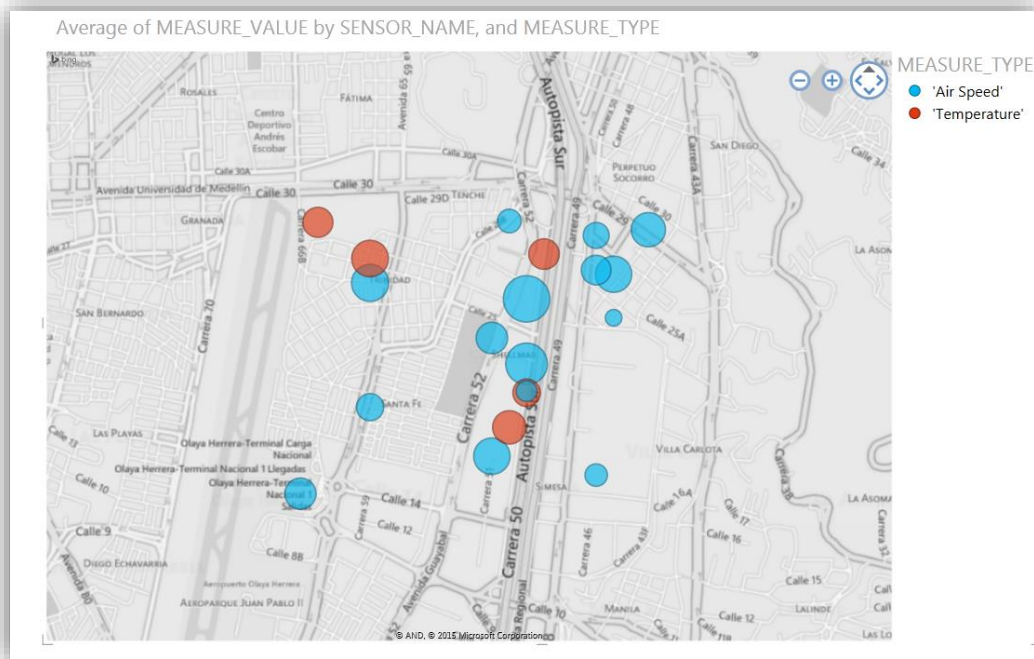


Figura 24 Mapa de bolas usando muestras de sensores

Nuevamente, apelando al requerimiento de bajo costo para el prototipo que se desarrolla en el presente documento, la elección de las herramientas de visualización a utilizar fueron elegidas según su costo. Para el caso de análisis visual se determinó conveniente el uso de la plataforma Pentaho, particularmente utilizando el componente adicional gratuito llamado *Saiku Analytics Community*. Este último componente permite generar gráficos interactivos tradicionales que pueden ser enlazados de forma interactiva en tableros de control para el contexto. En cuanto a los análisis geo referenciados se decide usar la herramienta *Power Map* de Microsoft Excel, la cual es un instrumento de ofimática ampliamente difundido y puede ser encontrado en bajos costos a nivel mundial. Adicionalmente, su uso es intuitivo y puede ser un excelente complemento para la difusión de resultados de análisis que puedan ser logrados sobre los conjuntos de datos.

La configuración y uso de *Saiku Analytics* de este prototipo fue lograda utilizando la información de [51]. Por otro lado, la utilización de la herramienta *Power Map* de Excel requirió la configuración de una conexión DSN a la bodega de datos para realizar las consultas necesarias en la construcción del mapa. Su configuración se pudo completar a partir de la información detallada en [52] y [53]. Las figuras mostradas previamente fueron logradas con datos simulados almacenados en la bodega de datos real, lo cual es una prueba de funcionamiento necesaria para el lanzamiento de un escenario de prueba programado el cual después de completar su fase de ETL podrá ser analizado de una forma similar a la mostrada.

CAPITULO 4

4. DISCUSIÓN DE RESULTADOS

En este capítulo se presentan los resultados obtenidos al ejecutar un caso de prueba propuesto luego de completar la implementación del prototipo planteado como prueba de concepto. Se detalla en las secciones siguientes las condiciones de la prueba hasta llegar a establecer una serie de recomendaciones deducidas a partir de los resultados.

4.1. Descripción del escenario

Con el fin de validar el funcionamiento del prototipo diseñado e implementado en la nube, se propuso un escenario de prueba que permitiera conocer el comportamiento del sistema en cada uno de sus componentes. El escenario consta de una simulación de 1200 muestras provenientes de un cluster de sensores implantados en una zona ficticia de riesgo como se muestra en la Figura 25. En la zona se miden dos variables físicas de interés para la gestión del riesgo; Velocidad del aire en Km/h y Temperatura en °C. Para la primera variable se cuenta con 15 sensores que enviaron 600 mediciones en un lapso de 4 horas de tiempo, en tanto que para la segunda variable se registraron 600 muestras provenientes de 5 sensores diferentes en un periodo de 6 horas.

Dentro de la generación de datos para la simulación se tuvo en cuenta la inclusión de datos que en lapsos de tiempo cumplieran con la regla a continuación: temperatura mayor a velocidad del aire y velocidad del aire mayor a una constante definida. La regla se muestra en la Ecuación 1. Lo anterior con el fin de poder validar el lanzamiento automático de alertas por parametrización de reglas descrito en el capítulo anterior.

$$T > V \wedge V > K$$

Ecuación 1 Regla para el lanzamiento de alertas

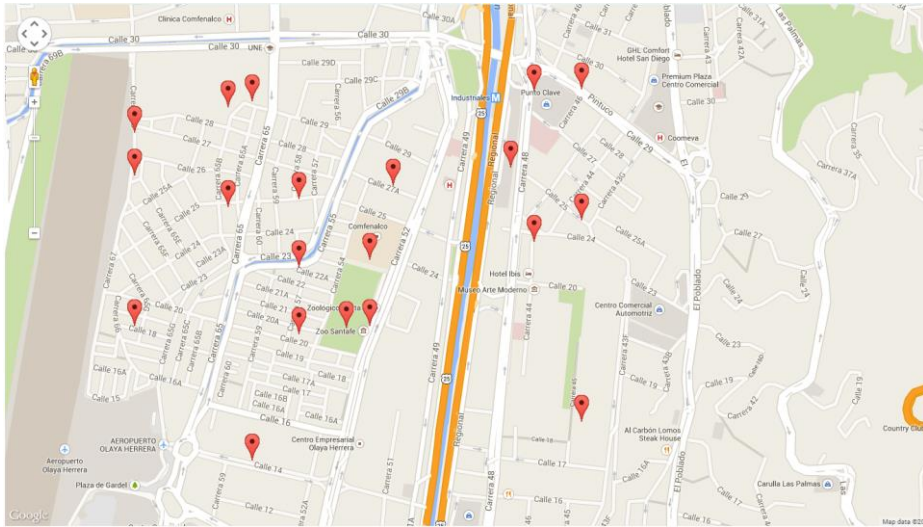


Figura 25 Zona de riesgo implantada

En las Figura 26 y Figura 27 se presenta la composición de las muestras almacenadas en la base de datos designada para el cluster simulado. Ambas resumen como están distribuidos los registros por tipo de variable y por sensor respectivamente.

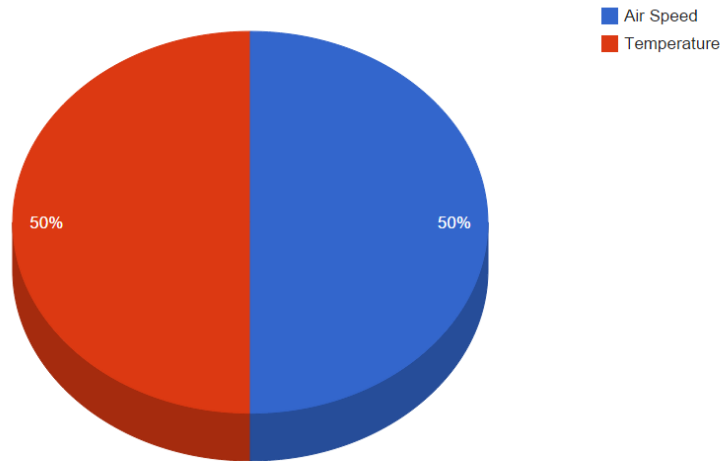


Figura 26 Composición de muestras por tipo de variable

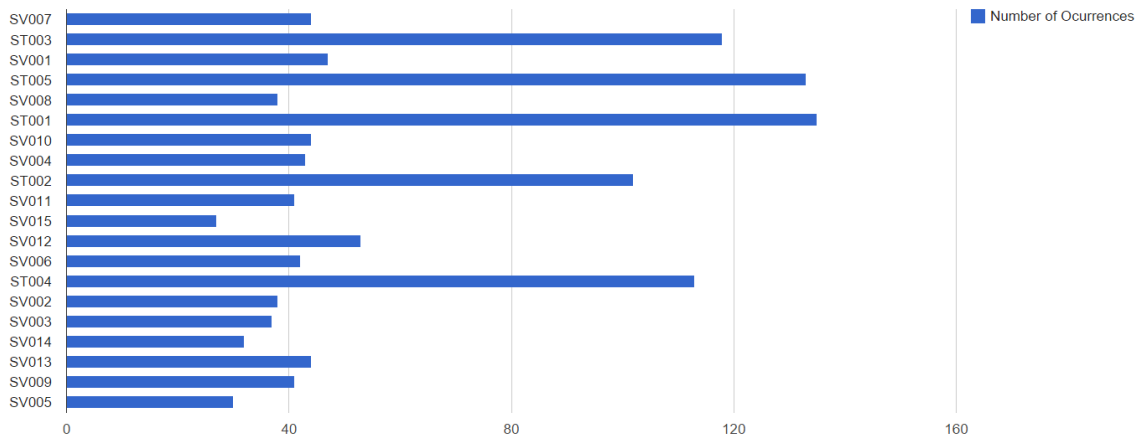


Figura 27 Distribución de muestras por sensor

Finalmente, teniendo en cuenta que una implantación de sensores puede ser vasta en extensión geográfica y cantidad de dispositivos, el modelo transaccional propuesto permite crear agrupaciones arbitrarias entre las unidades pertenecientes a una implantación. De esta manera, en el campo de análisis se podrían crear jerarquías dentro del cluster de sensores, lo cual facilitaría la toma de decisiones dada la segmentación ya disponible. Los grupos formados para esta prueba fueron tres, divididos por el identificador del sensor. La agrupación se hizo con propósitos demostrativos y no ofrece ninguna ventaja de análisis significativa. En implementaciones reales, podría utilizarse la ubicación geoespacial para asignar los grupos por ejemplo.

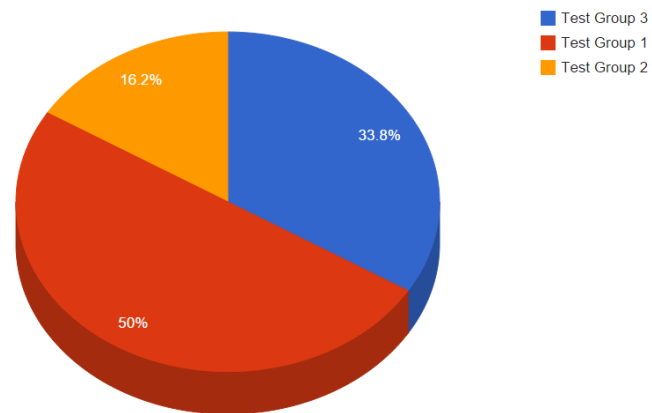


Figura 28 Composición de grupos en la implantación de prueba

4.2. Lanzamiento de prueba

Antes de proceder con el lanzamiento de la prueba, se diseñó el escenario cumpliendo con las reglas expuestas en la sección anterior haciendo uso de Microsoft Excel para generar las muestras ficticias siguiendo la estructura mostrada en

Tabla 12, correspondiente al modelo transaccional de ODK diseñado e implementado hasta este punto.

#_URI
_CREATOR_URI_USER
_CREATION_DATE
_LAST_UPDATE_URI_USER
_LAST_UPDATE_DATE
_MODEL_VERSION
_UI_VERSION
_IS_COMPLETE
_MARKED_AS_COMPLETE_DATE
MEASURE_VALUE
SENSOR_LOCATION_ALT
SENSOR_GROUP_ID
ASSIGNEE_FULL_NAME
META_INSTANCE_ID
SENSOR_GROUP_NAME
MEASURE_UNITS
GATEWAY_NAME
SENSOR_LOCATION_ACC
SENSOR_NAME
SENSOR_LOCATION_LAT
SENSOR_LOCATION_LNG
SENSOR_PRECISION
MEASURE_TYPE
ASSIGNEE_EMAIL
NODE_ID
NODE_NAME
SENSOR_ID
ASSIGNEE_ID

Tabla 12 Estructura de datos para la generación de muestras

Una vez poblada la base de datos con las muestras generadas de la forma más aproximada posible a lo que se tendría en un ambiente implantado real, se procedió a lanzar la ejecución del proceso de ETL en *Data Pipeline* en la consola web de Amazon. El proceso inició de forma inmediata con las tareas de extracción de todas las muestras que fueron registradas 24 horas antes de la hora de ejecución. Tal periodo fue determinado durante la fase de diseño e implementación con el fin de aprovechar la capa gratis de uso de la plataforma Amazon Web Services y así lograr tener un prototipo funcional de bajo costo. Sin embargo, este periodo es configurable y podrá disminuirse hasta ejecuciones repetidas en minutos, teniendo en cuenta la repercusión en los costos de funcionamiento que este tipo de frecuencias produciría.

Una vez consolidadas todas las muestras registradas en el servicio S3, se da inicio al proceso de transformación, el cual se origina con el lanzamiento de un recurso de tipo EMR con 1 maestro y un esclavo tal y como se muestra en el Anexo 3 a este documento. Esta etapa toma un tiempo considerable debido a que el recurso no se encuentra todo el tiempo activo y debe ser lanzado y configurado en cada ciclo de ejecución. Lo anterior se debe a los altos costos que representa el uso de este tipo de clusters cien por ciento del tiempo. En las recomendaciones de este proyecto se comenta acerca de esta alternativa entre costos y tiempo de transformación.

Una vez completado el proceso de transformación de forma satisfactoria, los resultados son almacenados en el servicio S3. Dicho resultado contiene las muestras consolidadas según la estructura y granularidad definida para el modelo analítico diseñado previamente. Este proceso es un logro importante en el presente trabajo dado que su ejecución ocurre sobre un la plataforma *Hadoop* haciendo uso de sentencias en lenguaje HQL. Esto resulta representativo dado que se prueba el correcto funcionamiento del procesamiento masivo y paralelo al cual puede ser escalado cualquier sistema que se implemente usando como base este prototipo.

Finalmente, luego de completar la transformación de muestras y disponer los resultados en el servicio S3, se inició satisfactoriamente la actividad de carga, la cual se encarga de tomar las muestras ya consolidadas en S3 para luego copiarlas en una tabla de recepción dispuesta en la bodega de datos. A partir de tal recepción, un disparador programado fue ejecutado sobre la tabla para distribuir las muestras en las dimensiones correspondientes y su

tabla de hecho. Una vez logrado este proceso, se ejecutó de forma automática la actividad de alertas descrita en la sección 3.2. Con éxito fueron identificados y enviados por correo todas las muestras que cumplieron con la regla de riesgo parametrizada. Esta demostración se consideró un logro alcanzado y fue completada satisfactoriamente siguiendo los procedimientos expuestos a lo largo de este documento. La siguiente figura presenta el correo obtenido y el tipo de reporte adjunto en formato csv luego de ejecutadas las actividades de carga y lanzamiento de alarmas.

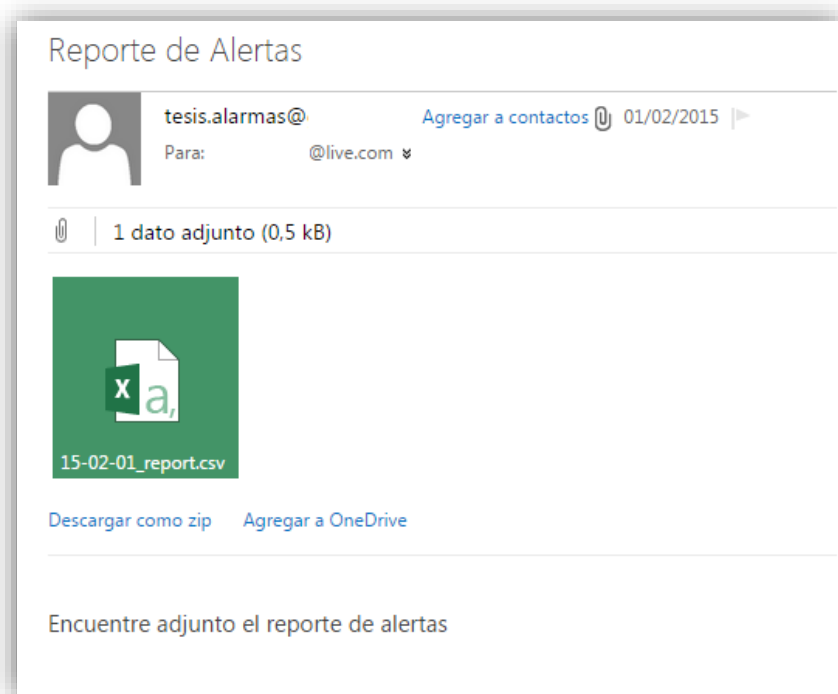


Figura 29 Reporte de alertas vía correo electrónico

4.3. Análisis del escenario

Habiendo completado el proceso de ETL implementado en la nube, todas las muestras recolectadas en las diferentes bases de datos relacionales se encuentran consolidadas en el modelo dimensional propuesto. Esto hace posible la ejecución de análisis a gestionados por parte del usuario teniendo en cuenta las consideraciones de diseño descritas en el capítulo 2 y la propuesta de análisis desarrollada a lo largo del capítulo 3. Haciendo uso de las herramientas presentadas en la sección 2.6 de este documento, se construyeron una serie de reportes que permitieran analizar el contexto físico de interés.

Como ejemplo, se presentan las Figura 30 y Figura 31 con el fin de evidenciar brevemente las posibilidades de análisis que se podrían elaborar. La primera, se construye con el fin de resumir los estadísticos fundamentales en un grupo de sensores que sobre los cuales se cumple la regla de alerta parametrizada en la sección 3.2. En este reporte, se demuestra que existen muestras de velocidad de aire que superan la constante configurada en 16.2 km/h y a su vez, hay muestras provenientes de un sensor de temperatura que superan en magnitud las mediciones de velocidad de la primera variable dicha. Esta condición dio origen de forma satisfactoria al lanzamiento de una alerta vía correo electrónico.

Statistics	Temperature / ST003 / Medida Promedio	Air Speed / SV001 / Medida Promedio	Air Speed / SV003 / Medida Promedio	Air Speed / SV012 / Medida Promedio
Min	10.382	12.540	12.784	12.881
Max	22.456	21.929	21.950	21.912
Sum	76.927	85.220	87.168	86.674
Average	15.385	17.044	17.434	17.335
Std. Deviation	4.555	3.781	3.834	3.626

Figura 30 Estadísticas por elección de sensores

Con el propósito de indagar más profundamente en el evento de riesgo identificado, se incluyó en el reporte la dimensión temporal, la cual deja en evidencia un el día y la hora donde se inició el posible incidente. Se resalta que la mezcla de las dimensiones “Sensor” y “Tiempo” del modelo dimensional se hace bajo jerarquías, permitiendo conocer las variables involucradas, los sensores responsables y un dato temporal compuesto por fecha y hora en conjunto con estadísticas generadas sobre las mediciones tal como se presenta a continuación.

Dia	Hora	Temperature			Air Speed								
		ST003			SV001			SV003			SV012		
		Medida Promedio	Medida Maxima	Cantidad de Muestras	Medida Promedio	Medida Maxima	Cantidad de Muestras	Medida Promedio	Medida Maxima	Cantidad de Muestras	Medida Promedio	Medida Maxima	Cantidad de Muestras
Saturday	20	11.826	13	11	13.409	14.92	7	12.764	14.39	10	13.826	14.02	12
	21	10.362	11.05	17	12.54	14.96	6	13.498	14.91	4	12.881	14.77	13
	22	13.429	15.51	17	16.572	18.8	12	17.549	18.92	12	17.172	18.14	6
	23	18.855	21.49	21	20.772	21.49	12	21.408	21.49	4	20.884	21.48	10
Sunday	0	22.456	22.62	23	21.929	22.27	7	21.95	22.2	5	21.912	22.19	10

Figura 31 Estadísticas por elección de sensores en el tiempo

Un reporte derivado del anterior, más resumido, conteniendo solo las medidas registradas, se presenta en la Figura 32.

Dia	Hora	Temperature	Air Speed			
		ST003	SV001	SV003	SV012	
		Medida Promedio	Medida Promedio	Medida Promedio	Medida Promedio	
Saturday	20	11.826	13.409	12.764	13.826	
	21	10.362	12.54	13.498	12.881	
	22	13.429	16.572	17.549	17.172	
	23	18.855	20.772	21.408	20.884	
Sunday	0	22.456	21.929	21.95	21.912	

Figura 32 Reporte resumido de mediciones en el tiempo

Es importante anotar que dado que el escenario fue simulado y se conocía la existencia de los eventos de riesgo ocultos en los datos, se pudo construir de forma inmediata el reporte que les dejaría en evidencia. Sin embargo, en un contexto real de análisis se deberán llevar a cabo construcciones más elaboradas presenten de forma más general el contexto físico de análisis. Independiente del contexto, lo anterior es una tarea que debe ser ejecutada con el acompañamiento de personal experto en análisis de incidentes.

Finalmente, teniendo en cuenta que la mayoría de análisis se conduce mediante el uso de gráficos más que de reportes como los mostrados previamente, se construyó un gráfico demostrativo que permitiera de forma visual identificar efectivamente el evento de riesgo parametrizado en las reglas del sistema. La fig demuestra el comportamiento de las muestras provenientes de los sensores involucrados en el incidente a lo largo del tiempo. En la parte superior de la imagen, se presenta el lapso durante el cual la regla parametrizada se cumple, es decir, la gráfica de temperatura supera en magnitud los valores de las gráficas de velocidad y a su vez todas superan la constante K estipulada. La configuración del gráfico se presenta en la barra vertical izquierda que indica las dimensiones y las medidas utilizadas.

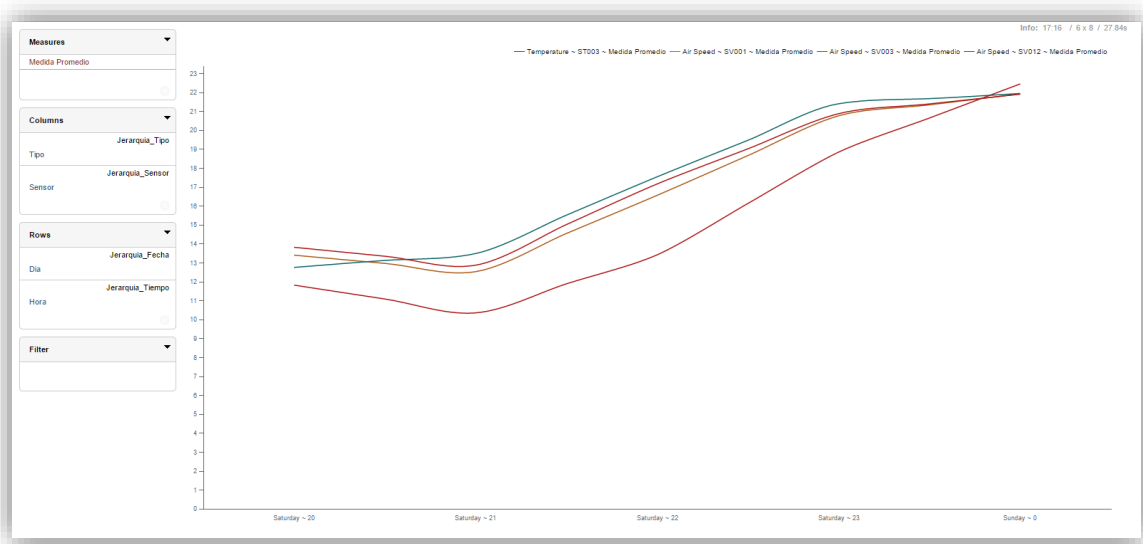


Figura 33 Análisis gráfico del escenario de prueba

Los métodos de análisis usados hasta este punto son presentados con el fin de demostrar el correcto funcionamiento del prototipo diseñado e implementado a lo largo del tiempo de ejecución. Es importante mencionar que de las técnicas de análisis propuestas, se implementaron las opciones de parametrización de reglas y análisis visual, en tanto que los algoritmos de inteligencia de negocio se dejan como opción viable y sugerida en futuras implementaciones que se puedan lograr sobre esta base. Se rescata, que las técnicas aplicadas y presentadas en esta sección van un poco más allá del objetivo de proponer y logran demostrar bajo un escenario de prueba hipotético la calidad del análisis que sobre los modelos planteados podrían llegar a ser creados.

4.4. Recomendaciones y Conclusiones

La realización de este prototipo es un trabajo que ha involucrado diversas tecnologías, desde las tradicionales bases de datos relacionales, hasta los nuevos sistemas de procesamiento paralelo propuestos con el apogeo de Big Data. Por tal motivo, el hecho de intentar abarcar tantos campos y de orquestar su funcionamiento en un solo sistema ha sido un reto que deja numerosas lecciones aprendidas y recomendaciones para trabajos futuros. En este orden de ideas, las principales recomendaciones se detallan a continuación.

- Las tecnologías de código abierto elegidas en el presente proyecto, han sido implementadas de forma satisfactoria permitiendo cumplir con los requerimientos funcionales y no funcionales propuestos. Sin embargo, en el mercado existen opciones comerciales que por un costo ofrecen menor dificultad de implementación, mayor rendimiento en demandas altas y mayor estabilidad en su funcionamiento. Un caso específico, se detalla con respecto al diseño de la bodega de datos propuesta en el prototipo; Es recomendable que cualquier sistema implementado basado en el diseño actual, utilice opciones comerciales de almacenamiento como Amazon Redshift a cambio de un motor MySQL como se planteó en las secciones previas. Esto se recomienda, teniendo en cuenta que todo el sistema ofrece oportunidades de escalamiento vertical y horizontal, en tanto que la bodega propuesta solo permitiría un escalamiento vertical limitado que estaría en contravía a lo que los demás componentes ofrecen. Un cluster dimensionable bajo demanda como Amazon Redshift otorgaría la escalabilidad faltante para un diseño robusto que soporte altos volúmenes en escenarios de internet de las cosas de la talla de una ciudad.
- La implementación actual proporciona sistemas de almacenamiento relacionales para las muestras tomadas por los sensores implantados haciendo uso del diseño ODK descrito a lo largo del documento. No obstante, esta implementación se decidió por restricciones de alcance y tiempos de ejecución. Por tal motivo, es recomendable que los sistemas reales que puedan ser implementados sobre esta base, utilicen diseños propios que cumplan con modelos normalizados de

mayor nivel. Adicionalmente, el desarrollo a la medida de servicios web dispuestos para la recolección de muestras, permite tener un mayor control sobre la seguridad que implica este tipo de sistemas, así como también la posibilidad de utilización de protocolos de transmisión de datos diferentes a http y https como se encuentra restringido en la propuesta actual. No obstante, de forma ilustrativa, a lo largo de la sección 2.2 de este documento se dejó planteada una propuesta del modelo transaccional adecuado para las prestaciones del presente prototipo.

- La implementación de lanzamiento automático de alertas expuesto en este trabajo, es un mecanismo válido a nivel de prototipo teniendo en cuenta que en el contexto de prueba de concepto con bajo presupuesto solo se requería comprobar el correcto funcionamiento. En el evento de consolidar una implementación real de mayores dimensiones, es necesario migrar este tipo de iniciativas a herramientas más robustas con interfaces de parametrización amigables para los usuarios. Es recomendable que los escenarios parametrizables sean lo suficientemente flexibles para evitar costos de mantenimiento que pudieran resultar bajo un caso de uso real.
- La aplicación de las técnicas de minería requieren en su mayoría un conjunto de datos histórico sobre el cual puedan ser construidos modelos predictivos evaluables y aplicables a nuevos registros. Por tal motivo, en la medida de lo posible, es recomendable que durante la implementación del sistema real, se utilicen set de datos reales obtenidos de otras fuentes y ajustados al modelo analítico planteado para poblar inicialmente la base de datos. De esta manera podrían ser implementadas las propuestas de análisis hechas y a su vez obtener el componente de automatización de lanzamiento de alertas deseado en el proyecto padre de este prototipo.
- Los costos del sistema diseñado son dependientes de la demanda de recursos de infraestructura tal como se describió en la fase de diseño e implementación. Sin embargo, el uso de tales recursos puede ser optimizado con el fin de reducir los costos variables de implementación. Por ejemplo, el uso de instancias de tipo EC2 para la recepción de muestras de los sensores puede ser optimizado mediante el uso de balanceadores de carga u otros servicios en la

nube que permitan distribuir la concurrencia de peticiones entre las máquinas disponibles para esto. Adicionalmente, el costo de transformación haciendo uso de un recurso de tipo EMR de Amazon puede ser controlado mediante usos solamente bajo demanda y no teniendo tal recurso un cien por ciento del tiempo en línea. No obstante, lo anterior implica que el tiempo para el proceso de transformación se incrementa notablemente dado que previo a cada ciclo de procesamiento se debe crear y configurar el cluster a utilizar, situación que no se presentaría en un escenario donde se tenga la disponibilidad total de este recurso.

- Como estrategia de reducción de costos en implementaciones futura, en caso de utilizar el proveedor Amazon, se recomienda utilizar instancias de computo EC2 bajo reserva programada al plazo de 3 años. Lo anterior se concluye con base en los hallazgos demostrados en la sección 1.2 de este documento donde se evidencia una reducción sustancial en los precios con respecto a la opción de utilización de recursos exclusivamente bajo demanda.
- La aplicación de técnicas de minería de datos para el sistema diseñado e implementado a nivel de prototipo de prueba de concepto a lo largo de este trabajo, no cobra mayor sentido teniendo en cuenta que los escenarios han sido simulados y se conocen los patrones ocultos en los datos. Sin embargo, se recomienda fuertemente el uso de los algoritmos propuestos sobre implementaciones reales, dado que este tipo de análisis serán las herramientas clave para efectivamente identificar, alertar y gestionar patrones de incidentes de riesgo bajo diversos contextos. Para obtener resultados exitosos de análisis en sistemas desarrollados sobre este prototipo, se concluye que será necesario encontrar y estructurar bases de datos históricas de acontecimientos en el contexto físico que aumenten la precisión de los modelos de predicción creados a partir de la mayoría de técnicas de minería. No obstante, algunas tendencias básicas de comportamiento podrían ser identificadas y modeladas con datos obtenidos en un corto tiempo de operación del sistema.
- De forma concluyente se comenta que el ejercicio de diseño e implementación desarrollado a lo largo de este trabajo fue exitoso según los resultados obtenidos en el escenario de prueba en contraste

con los requerimientos funcionales y no funcionales del proyecto. Se sugiere utilizar el diseño propuesto, incluyendo sus modelos y técnicas de implementación para el desarrollo futuro de sistemas de mayor envergadura.

- El lanzamiento de alertas automáticas por medios electrónicos es una funcionalidad vital en sistemas como los desarrollados a lo largo de este trabajo dada su orientación a la gestión de riesgos. En consecuencia, se debe destacar la capacidad de la presente prueba de concepto por identificar, extraer y distribuir vía correo electrónico reportes de registros que cumplieran con un patrón de riesgo. A pesar de lo recomendado en párrafos anteriores en lo que respecta al uso de herramientas comerciales para el lanzamiento de alertas, se concluye que la distribución electrónica de alarmas es un componente esencial que debe hacer parte de las futuras implementaciones sea cual sea la técnica aplicada. Así mismo, se resalta esta funcionalidad como satisfactoria dentro de la prueba de concepto, teniendo en cuenta que de forma exitosa se pudo comprobar la viabilidad de su aplicación. Finalmente, se recomienda para trabajos futuros invertir esfuerzos en la inclusión de métodos alternativos de distribución tales como mensajes de texto SMS, llamadas telefónicas o incluso el envío directo de material multimedia que pudiera estar presente como imágenes y videos de la zona de riesgo.

REFERENCIAS

- [1] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to Mobile Crowd Sensing" *2014 IEEE Int. Conf. Pervasive Comput. Commun. Work. PERCOM Work. 2014*, pp. 593–598, 2014.
- [2] Apprenda, "IaaS, PaaS, SaaS (Explained and Compared)," 2013. [Online]. Available: <http://apprenda.com/library/paas/iaas-paas-saas-explained-compared/>. [Accessed: 10-Nov-2014].
- [3] M. Dorasamy, M. Raman, and M. Kaliannan, "Knowledge management systems in support of disasters management: A two decade review" *Technol. Forecast. Soc. Change*, vol. 80, no. 9, pp. 1834–1853, Nov. 2013.
- [4] A. S. Vivacqua and M. R. S. Borges, "Taking advantage of collective knowledge in emergency response systems" *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 189–198, Jan. 2012.
- [5] T. Moilanen, "Scalable Cloud Database Solution," University of Oulu, 2013.
- [6] D. Restrepo, O. Ovalle, and A. Montoya, "Manejo e integración de bases de datos en redes de sensores inalámbricas" *Rev. Av. en Sist. e Informática*, vol. 6, no. 1, pp. 145–154, 2009.
- [7] G. Cardone, P. Bellavista, A. Corradi, and L. Foschino, "Effective Collaborative Monitoring in Smart Cities: Converging Manet And WSN for Fast Data Collection" *Kaleidosc. Acad. Conf.*, p. 8, 2011.
- [8] A. Zanella, S. Member, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities" *IEEE Internet things*, vol. 1, no. 1, pp. 22–32, 2014.
- [9] Y. Li and S. Manoharan, "A performance comparison of SQL and NoSQL databases" in *IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing - Proceedings*, 2013, pp. 15–19.
- [10] K. Barmpis and D. Kolovos, "Comparative analysis of data persistence technologies for large-scale models" *Proc. 2012 Extrem. Model. ...*, pp. 33–38, 2012.
- [11] Amazon, "Documentation Amazon Web Services (AWS)" 2013. [Online]. Available: http://aws.amazon.com/documentation/?nc1=f_dr. [Accessed: 10-Oct-2014].

- [12] Amazon, "AWS Documentation" 2014. [Online]. Available: <http://aws.amazon.com/documentation/>. [Accessed: 10-Oct-2014].
- [13] Microsoft, "Azure Documentation Center" 2013. [Online]. Available: <http://azure.microsoft.com/en-us/documentation/>. [Accessed: 10-Oct-2014].
- [14] Google, "Google Cloud Platform Documentation" 2014. [Online]. Available: <https://cloud.google.com/docs/>. [Accessed: 10-Oct-2014].
- [15] Rackspace, "Rackspace Support Network" 2013. [Online]. Available: <http://support.rackspace.com/>. [Accessed: 10-Oct-2014].
- [16] C. Harmony, "State of The Cloud Report by RightScale". July, 2014.
- [17] Apache Software Foundation "Apache Hadoop," 2.6.0, 2014. [Online]. Available: <http://hadoop.apache.org/docs/current/>. [Accessed: 17-Apr-2015].
- [18] Nasuni Corporation, "The State of Cloud Storage" 2013.
- [19] B. Satish, P. Pawar, and A. S. M. S. Ibmr, "Smart City with Internet of Things (Sensor networks) and Big Data" Pune, 9860027825, 2013.
- [20] A. Gates, *Programming Pig (Google eBook)*, 1st ed. Cambridge: O'Reilly, 2011, p. 224.
- [21] K. J. Schmidt and C. Philips, *Programming Elastic Map Reduce*, 1st ed. 2013.
- [22] E. Capriolo, D. Wampler, and J. Rutherglen, *Programming Hive*, 1st ed. O'Reilly, 2012, p. 328.
- [23] O. Reilly and M. Seu, "Hadoop, The Definitive Guide" *Online*, vol. 54, pp. 13, 258, 2012.
- [24] B. Lublinsky, K. T. Smith, and A. Yakubovich, *Professional Hadoop Solutions*, 1st ed. Wrox, 2013, p. 504.
- [25] E. Barbierato, M. Gribaudo, and M. Iacono, "Performance evaluation of NoSQL big-data applications using multi-formalism models" *Future Generation Computer Systems*. DI, Università degli Studi di Torino, corso Svizzera, 185, 10129 Torino, Italy, 2014.
- [26] W. Kernochan, "Business Intelligence 101: A Brief History." [Online]. Available: <http://www.enterpriseappstoday.com/business->

intelligence/business-intelligence-101-a-brief-history.html. [Accessed: 01-Feb-2015].

- [27] E. Turban, R. Sharda, D. Delen, and D. King, "Introduction to business intelligence" *Bus. Intell. a Manag. approach*, pp. 3–18, 2011.
- [28] G. Sansu, "Inmon vs. Kimball: Which approach is suitable for your data warehouse?" 2012. [Online]. Available: <http://searchbusinessintelligence.techtarget.in/tip/Inmon-vs-Kimball-Which-approach-is-suitable-for-your-data-warehouse>. [Accessed: 06-Feb-2015].
- [29] I. Abramson, "Data Warehouse: The Choice of Inmon versus Kimball," *IOUG. Najdeno*, 2004.
- [30] J. Caserta and R. Kimball, *THE DATA WAREHOUSE ETL TOOLKIT*. 2004, p. 491.
- [31] B. H. Rita L. Sallam, Joao Tapadinhas, Josh Parenteau, Daniel Yuen, "Magic Quadrant for Business Intelligence and Analytics Platforms." [Online]. Available: [http://www.gartner.com/technology/reprints.do?id=1-1QLGACN&ct=140210&st=sb&ref=lp&signin=ae246c1bc05fca45ab83a8f2c90e077b&1\[os\]=windows](http://www.gartner.com/technology/reprints.do?id=1-1QLGACN&ct=140210&st=sb&ref=lp&signin=ae246c1bc05fca45ab83a8f2c90e077b&1[os]=windows). [Accessed: 08-Feb-2015].
- [32] Microsoft, "Reference catalog" 2015. [Online]. Available: <https://msdn.microsoft.com/library>. [Accessed: 17-Apr-2015].
- [33] "What is Data Mining, Predictive Analytics, Big Data." [Online]. Available: <http://www.statsoft.com/Textbook/Data-Mining-Techniques#mining>. [Accessed: 10-Mar-2015].
- [34] M. Habib, "Agile software development methodologies and how to apply them" 2013. [Online]. Available: <http://www.codeproject.com/Articles/604417/Agile-software-development-methodologies-and-how-t>. [Accessed: 17-Apr-2015].
- [35] Capgemini, "Agile Business Intelligence How to make it happen?" Rightshore, Amsterdam, p. 16, 2013.
- [36] Scrum Alliance, "Scrum - a description" pp. 1–11, 2012.
- [37] D. Peterson, "What is Kanban?" 2015. [Online]. Available: <http://kanbanblog.com/explained/>.
- [38] ODK, "About ODK," 2010. [Online]. Available: <https://opendatakit.org/about/>.

- [39] W. Brunette, M. Sundt, and N. Dell, "Open data kit 2.0: expanding and refining information services for developing regions" Washington, 2013.
- [40] R. Kimball and M. Ross, *The Data Warehouse Toolkit, The Definitive Guide to Dimensional Modeling*. 2013, p. 600.
- [41] Facebook, "Facebook developers" 2015. [Online]. Available: <https://developers.facebook.com/>. [Accessed: 17-Apr-2015].
- [42] Yahoo, "Yahoo developers" 2015. [Online]. Available: <https://developer.yahoo.com/analytics/>. [Accessed: 17-Apr-2015].
- [43] Twitter, "Twitter developers" 2015. [Online]. Available: <https://dev.twitter.com/>. [Accessed: 17-Apr-2015].
- [44] Pentaho, "Pentaho Mondrian Documentation" 2014. [Online]. Available: <http://mondrian.pentaho.com/documentation/installation.php>. [Accessed: 05-Mar-2015].
- [45] W. D. Back and J. Hyde, *Mondrian in Action*. 2013.
- [46] T. A. Basile, N. Mauro, S. Ferilli, and F. Esposito, "Relational Temporal Data Mining for Wireless Sensor Networks" in *AI*IA 2009: Emergent Perspectives in Artificial Intelligence SE - 42*, vol. 5883, R. Serra and R. Cucchiara, Eds. Springer Berlin Heidelberg, 2009, pp. 416–425.
- [47] E. Sarajevo, "Mining and predicting temperature and smoke sensors data" vol. 13, no. March, pp. 343–348, 2014.
- [48] C. Pete, C. Julian, K. Randy, K. Thomas, R. Thomas, S. Colin, and R. Wirth, "Crisp-Dm 1.0," *Cris. Consort.*, p. 76, 2000.
- [49] Amazon, "Launch an Amazon EC2 Instance - Amazon Elastic Compute Cloud" 2014. [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance_linux.html. [Accessed: 09-Mar-2015].
- [50] Amazon, "Amazon Machine Images (AMI) - Amazon Elastic Compute Cloud" 2013. [Online]. Available: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>. [Accessed: 09-Mar-2015].
- [51] Joy of data, "Getting Started With Pentaho BI Server 5, Mondrian and Saiku" 11-May-2014. [Online]. Available: <http://www.joyofdata.de/blog/getting->

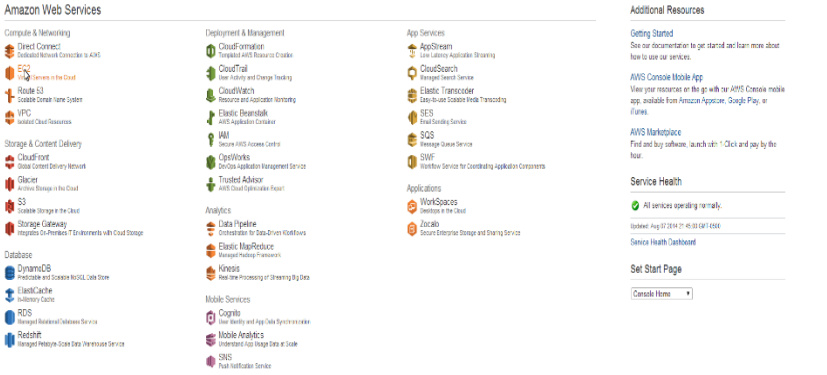
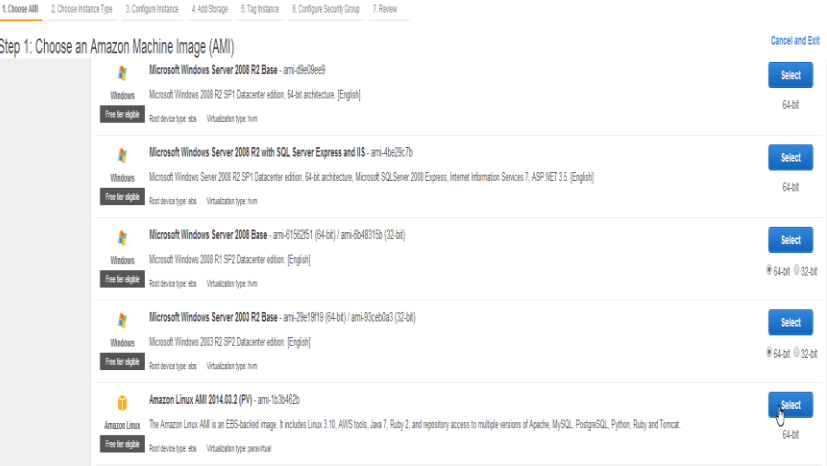
started-with-pentaho-bi-server-5-mondrian-and-saiku/. [Accessed: 09-Mar-2015].

- [52] Oracle, “Configuring a Connector/ODBC 5.x DSN on Windows” 2013. [Online]. Available: <http://dev.mysql.com/doc/connector-odbc/en/connector-odbc-configuration-dsn-windows-5-2.html>. [Accessed: 09-Mar-2015].
- [53] Microsoft, “Connect external data to your workbook” 2012. [Online]. Available: <https://support.office.com/en-in/article/Connect-external-data-to-your-workbook-945f2cbb-d50b-4ee2-bae8-c4c9381000c6>. [Accessed: 09-Mar-2015].

ANEXOS

ANEXO 1

CONFIGURACIÓN Y LANZAMIENTO DE INSTANCIAS EC2 EN LA NUBE PARA RECOLECCIÓN DE MUESTRAS

<p>Ingresar a la consola AWS / EC2</p> <p>Para ingresar al panel de administración de Amazon Web Services se debe tener una cuenta valida creada previamente en Amazon.</p>	
<p>Elegir AMI</p> <p>El tipo de imagen que se elija depende de la aplicación que se pretenda implementar. De lo anterior dependerá también el tamaño de la instancia física y el costo de la misma. En este trabajo se eligió una imagen Linux en una instancia t1.micro sin costo con el fin de realizar la prueba de concepto.</p>	

Elegir tipo de instancia
Dependiendo del tamaño requerido, elegir la instancia más adecuada.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
Micro instances	t1.micro	1	0.615	EBS only	-	Very Low
General purpose	t2.micro	1	1	EBS only	-	Low to Moderate
General purpose	t3.small	1	2	EBS only	-	Low to Moderate

Aceptar comportamiento predeterminado y lanzar instancia
Se presenta el resumen por defecto de configuración antes de hacer el lanzamiento de la instancia y la instalación de la imagen.

Descargar llave privada de encriptación para conexión SSH con la instancia.
Al finalizar la instalación se debe descargar la llave privada de conexión con el fin de realizar una conexión remota segura posteriormente con la instancia.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name
T1Instance

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

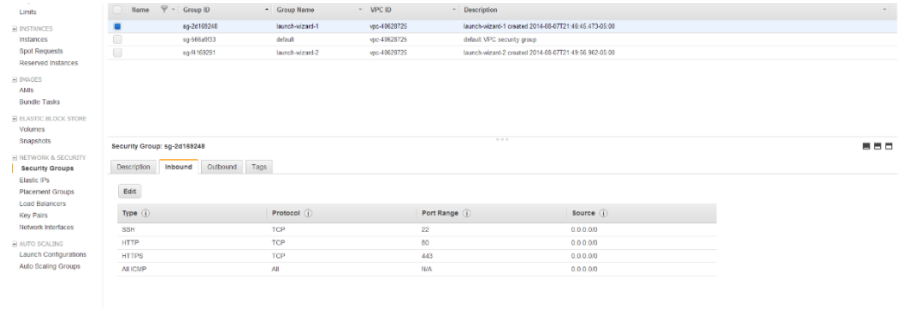
Cancel **Launch Instances**

Verificar el estado de la instancia lanzada.
Ingresar el panel de administración de instancias EC2 y verificar el estado de la maquina lanzada en los pasos anteriores.

Instances	ID	Type	Platform	Status	Checks	Tags	Created	AMI
	i-6610d6c3	t1.micro	us-west-2a	running	2/2 checks passed	None	2014-09-19T10:19:50.000Z	ami-763b46b5

Configurar el grupo de seguridad para acceso a la instancia.

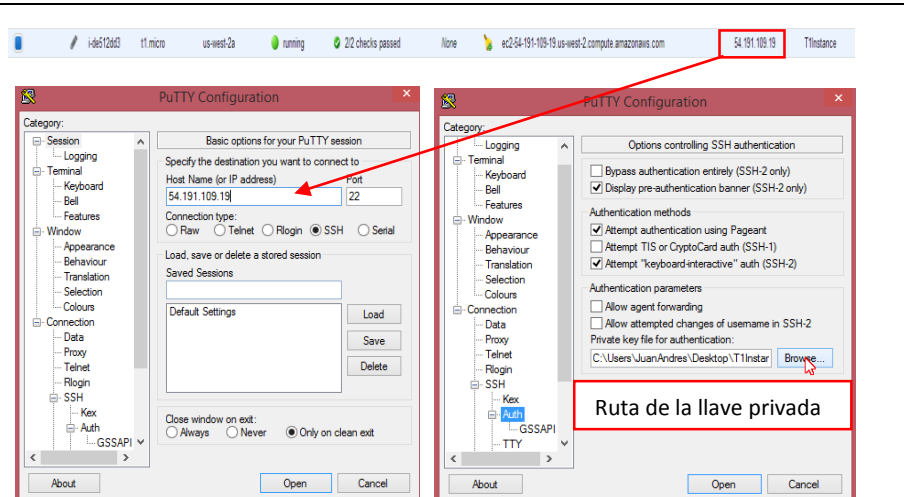
En el panel de administración de grupos de seguridad, se debe modificar las reglas de ingreso a la instancia creada permitiendo conexiones entrantes por los protocolos SSH, HTTP, HTTPS y ICMP.



Configurar conexión remota a la instancia.

Utilizar una herramienta de conexión por el protocolo SSH para configurar el acceso remoto a la máquina virtual creada. Se utilizó Putty para esto.

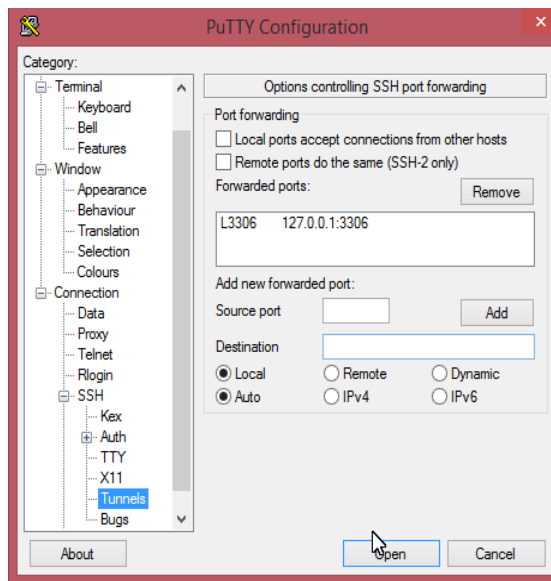
Se debe identificar la IP pública de la instancia en el panel de administración de Amazon Web Services y tener la llave privada con el fin de realizar la conexión exitosamente.



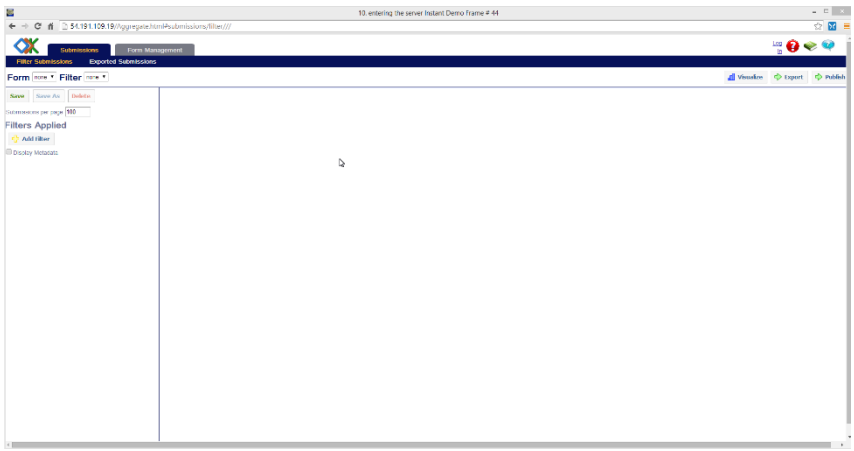
Crear túnel de acceso a base de datos

Luego de configurar la ip de conexión y la ubicación de la llave privada requerida por el protocolo SSH, se debe configurar el túnel por el puerto 3306 hacia la instancia remota como se muestra en la imagen con el fin de poder acceder a la base de datos con un cliente MySQL posteriormente.

Una vez abierta la consola, el nombre de usuario por defecto para ingresar al sistema es "ec2-user".



<p>Generar despliegue local de la aplicación ODK Aggregate.</p> <p>Se debe descargar la última versión del instalador de ODK Aggregate y seguir los pasos del asistente.</p>	<p>Durante la configuración, es importante indicar que será una instalación de MySQL, y también se debe especificar la dirección IP que se utiliza para acceder a su servidor Aggregate.</p> <p>También se debe especificar una dirección de cuenta de Google con la que posteriormente pueda iniciar sesión en la instancia agregada.</p> <p>La instalación genera un archivo llamado <i>"create_db_and_user.sql"</i> que debe ser subido al directorio <i>/home/ec2-user</i>.</p> <p>La instalación también genera un archivo llamado <i>"ODKAggregate.war"</i>. Se debe cambiar el nombre de esta a <i>"ROOT.war"</i> y subirlo a la carpeta <i>/tomcat6/webapps /usr /share</i>.</p>
<p>Configurar MySQL.</p>	<p>En la consola Putty se debe ejecutar el comando <i>"/usr /bin /mysql_secure_installation"</i> para establecer una contraseña de super usuario a MySQL</p> <p>Luego para conectarse al motor de base de datos se debe ejecutar <i>"mysql -u root -p"</i>. Se debe especificar la contraseña creada previamente.</p> <p>Una vez establecida la conexión, se introduce el comando <i>"source ~/create_db_and_user.sql"</i>. Esto creará el usuario ODK y base de datos.</p> <p>Por último, ejecutar <i>"sudo /sbin /chkconfig --levels 235 mysqld on"</i> para iniciar el servicio de MySQL automáticamente.</p>
<p>Configurar el servidor Tomcat.</p>	<p>Descargar el <i>MySQL Connector / J</i> del sitio de descargas de MySQL (http://dev.mysql.com/downloads/connector/j/), descomprimirlo, y transferir el archivo <i>"mysql-connector-java-xxx-bin.jar"</i> hasta <i>"/tomcat6/lib/usr/share"</i> de la instancia.</p> <p>Editar el archivo <i>"/etc/tomcat6/server.xml"</i> con el fin de personalizar la configuración:</p> <ol style="list-style-type: none"> 1. Cambiar <i>"<Connector port="8080" protocol="HTTP/1.1" a "<Connector port="8080" proxyPort="80" protocol="HTTP/1.1" (es decir, añadir el atributo proxyPort).</i> 2. Cambiar <i>"<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true" a "<Connector port="8443" proxyPort="443" protocol="HTTP/1.1" SSLEnabled="true"</i>. <p>Ejecutar los siguientes comandos para terminar la configuración de puertos:</p> <ol style="list-style-type: none"> 1. <i>"sudo /sbin/iptables -t nat -I PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080"</i>

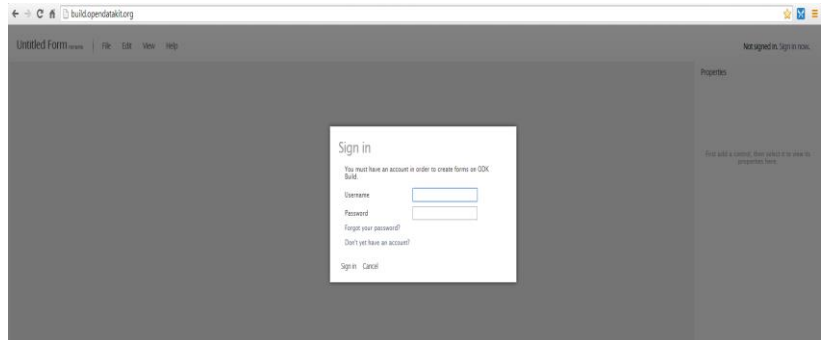
	<ol style="list-style-type: none">2. <code>sudo /sbin/iptables -t nat -I PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 8443</code>3. <code>sudo /sbin/service iptables save</code> <p>Finalmente se debe iniciar el servicio Tomcat para realizar el despliegue y configurarlo para iniciar automáticamente.</p> <ol style="list-style-type: none">1. <code>sudo service tomcat6 start</code>.2. <code>sudo chkconfig --level 345 tomcat6 on</code>
<p>Ingresar al sitio web y verificar que se encuentra disponible.</p>	

ANEXO 2

CONFIGURACIÓN DE ENVÍO DE MUESTRAS VÍA INTERNET DESDE DISPOSITIVOS REMOTOS

Crear una cuenta.

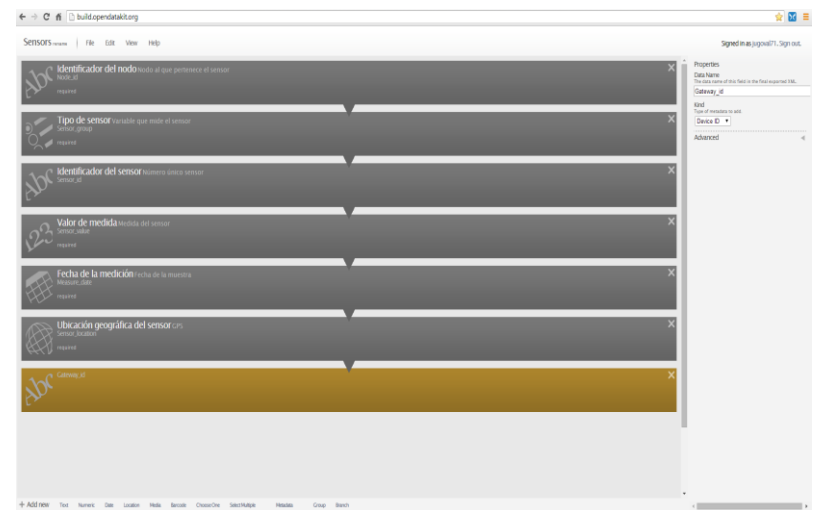
Ingresar al sitio de construcción "<http://build.opendatakit.org/>" y crear una cuenta de usuario.



Crear formularios de muestreo.

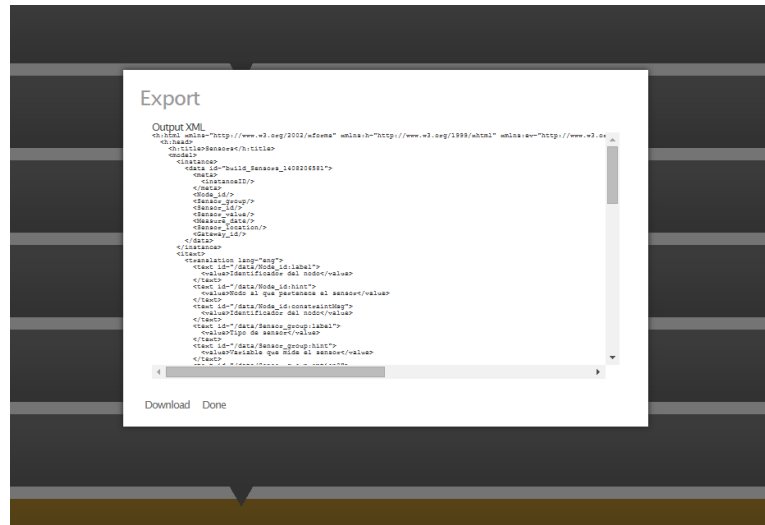
Una vez creada la cuenta, se debe crear un formulario con los campos requeridos y las validaciones necesarias según los datos que se deseen recolectar.

En este caso se realiza una recolección de mediciones de sensores y el formulario se muestra en la imagen.



Exportar a formato XML

En la opción “*File/Export to xml..*” del sitio se permite descargar el formulario creado.



Importar Formulario xml al servidor.

Ingresar a la instancia configurada como servidor “*Aggregate*”.

A continuación, en la sección “*Form Management*” se debe cargar el archivo xml generado en el paso anterior.

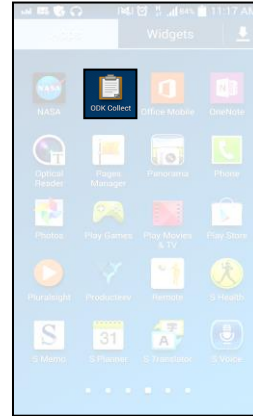
En este punto, el formulario cargado al servidor podrá ser visto y editado por peticiones HTTP de forma remota con alguna herramienta que utilice las librerías Java Rosa para transmisión de datos.



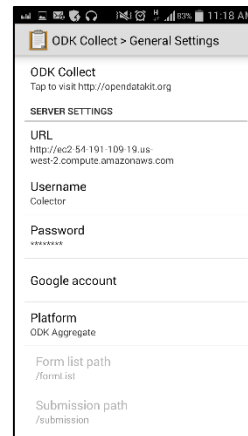
Utilizar ODK Collect para envío remoto de muestras.

En un dispositivo móvil con sistema operativo Android, se debe descargar la aplicación ODK Collect.

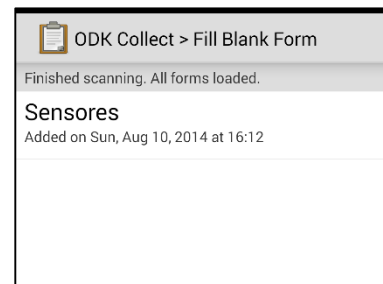
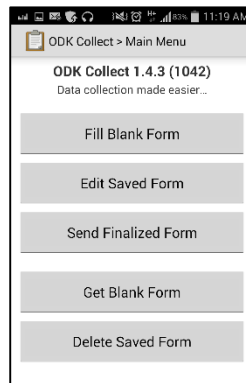
1. La aplicación ODK Collect instalado en Android.



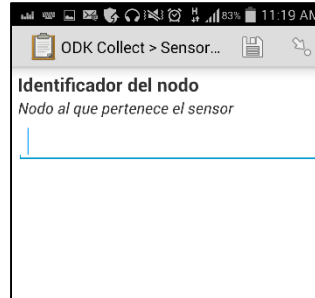
2. Configuración del servidor Aggregate.



3. Descarga del formulario cargado al servidor Aggregate.



4. Registro de mediciones en el formulario y envió al servidor.



Verificación de datos en el servidor.

Ingresar al servidor configurado en Amazon Web Services y verificar la recepción del envío de las muestras.

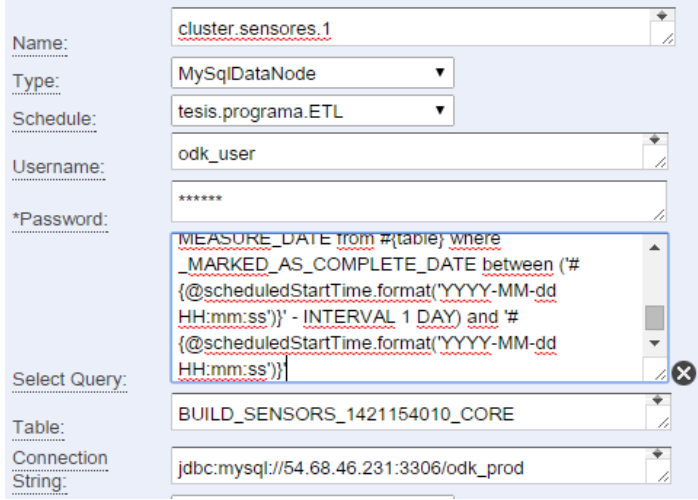
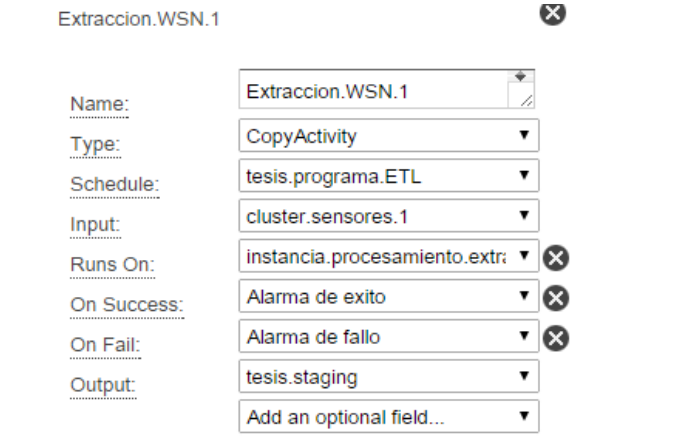
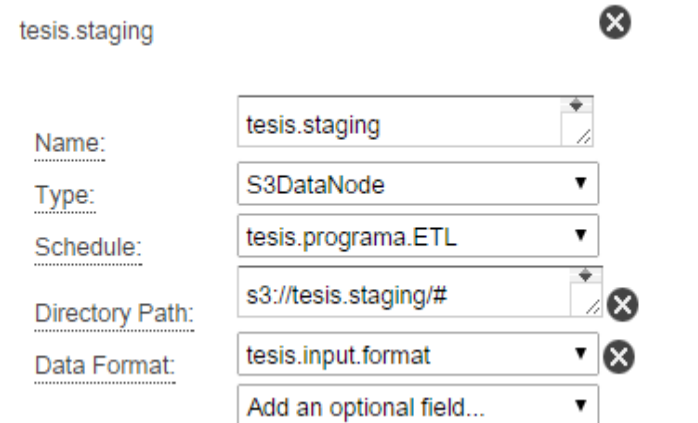


Form	Sensores	Filter	View
Form	Sensores	Filter	View
Form	Sensores	Filter	View

ANEXO 3

CONFIGURACIÓN DATA PIPELINE EN LOS SERVICIOS EN LA NUBE DE AMAZON

<p>Componentes de Data Pipeline</p> <p>Activities:</p> <p>Tareas a ser ejecutadas recibiendo parámetros, después de cumplida cierta condición, en un tiempo programado y utilizando recursos de procesamiento. Ejemplo: Copiar datos de un origen SQL a S3 todos los días a una hora.</p> <p>Data Nodes:</p> <p>Representan lugares de almacenamiento como S3, RDS, Redshift, DynamoDB etc.</p> <p>Schedules:</p> <p>Tiempo de ejecución programado</p> <p>Resources:</p> <p>Servidores de tipo EC2 y Clusters EMR.</p> <p>Preconditions:</p> <p>Condiciones configurables para ejecutar actividades.</p> <p>Other:</p> <p>Alarmas, formatos de entrada y salida para usar en Nodos de datos.</p> <p>Parameters:</p> <p>Datos usables en actividades.</p>	<ul style="list-style-type: none">▼ Activities<hr/>▶ DataNodes<hr/>▶ Schedules<hr/>▶ Resources<hr/>▶ Preconditions<hr/>▶ Others<hr/>▶ Parameters
---	--

<p>Bases de datos de muestras:</p> <p>Son data nodes de tipo MySql que deben contener los datos de conexión e la base de datos donde se encuentran las muestras registradas por un cluster de sensores. Adicional a la conexión, debe ser escrita una sentencia SQL que se ejecuta sobre la base de datos para extraer todas las muestras según un delta estipulado. En el caso de la imagen, todos los datos registrados en un día.</p>	 <p>The screenshot shows a configuration window for a data node named 'cluster.sensores.1'. The 'Type' is 'MySqlDataNode', the 'Schedule' is 'tesis.programa.ETL', and the 'Username' is 'odk_user'. The password is masked with asterisks. A 'Select Query' field contains a complex SQL statement: <code>MEASURE_DATE from #{table} where MARKED_AS_COMPLETE_DATE between ('#{@scheduledStartTime.format("YYYY-MM-dd HH:mm:ss")}' - INTERVAL 1 DAY) and '##{@scheduledStartTime.format("YYYY-MM-dd HH:mm:ss")}'</code>. The 'Table' is 'BUILD_SENSORS_1421154010_CORE' and the 'Connection String' is 'jdbc:mysql://54.68.46.231:3306/odk_prod'.</p>
<p>Extracción de muestras:</p> <p>Actividad de extracción de datos que traslada los registros desde los nodos de la imagen anterior hacia una sistema de almacenamiento S3 llamdo tesis.staging como se aprecia en la imagen. Registra alarmas de éxito y fallo cada ciclo de ejecución. Utiliza un recurso del tipo EC2 para ser ejecutada.</p>	 <p>The screenshot shows a configuration window for a 'CopyActivity' named 'Extraccion.WSN.1'. The 'Schedule' is 'tesis.programa.ETL' and the 'Input' is 'cluster.sensores.1'. The 'Runs On' resource is 'instancia.procesamiento.extr.', and the 'Output' is 'tesis.staging'. There are also fields for 'On Success' (Alarma de exito) and 'On Fail' (Alarma de fallo), both with alarm icons.</p>
<p>Almacenamiento Intermedio:</p> <p>Nodo de datos del tipo S3Data para el almacenamiento de todos los datos extraídos por las actividades descritas en el paso anterior. Se utiliza un formato de almacenamiento para los datos ingresados y una estructura de carpetas basadas en la fecha de ejecución para garantizar la no duplicidad.</p>	 <p>The screenshot shows a configuration window for an 'S3DataNode' named 'tesis.staging'. The 'Schedule' is 'tesis.programa.ETL' and the 'Directory Path' is 's3://tesis.staging/'. The 'Data Format' is 'tesis.input.format'.</p>

Transformación:

Actividad del tipo Hive, ejecutada usando un recurso EMR para el procesamiento paralelo de todas las muestras encontradas en el nodo de almacenamiento intermedio "tesis.staging". Ejecuta un script HQL para integrar todas las muestras de diversas bases de datos y las agrupa en un formato de salida para ser escritas en un nodo de almacenamiento S3 llamado tesis.transform. Al igual que la actividad de extracción, en esta se lanzan alarmas de éxito o fallo en cada ciclo de ejecución. Esta actividad tiene como pre condición la ejecución exitosa de las actividades de extracción como se muestra en la figura.

Transformacion.WSN

Name: Transformacion.WSN

Type: HiveActivity

Input: tesis.staging

Hive Script: INSERT OVERWRITE TABLE \${output1} SELECT Node_id, Node_name, Assignee_id, Assignee_full_name, Assignee_email, Sensor_group_id, Sensor_group_name, Sensor_id, Sensor_name, avg(Sensor_precision) as Sensor_precision, avg(SENSOR_LOCATION_LAT) as

Schedule: tesis.programa.ETL

Runs On: instancia.procesamiento.tran:

On Success: Alarma de exito

On Fail: Alarma de fallo

Depends On: Extraccion.WSN.1

Depends On: Extraccion.WSN.2

Output: tesis.transform

Almacenamiento de datos Transformados

Nodo de datos de tipo S3 para el almacenamiento de los resultados obtenidos en la actividad de transformación. Se utiliza un formato de salida para los archivos introducidos y se construye una estructura de carpetas basada en la fecha de ejecución para evitar duplicidad.

tesis.transform

Name: tesis.transform

Type: S3DataNode

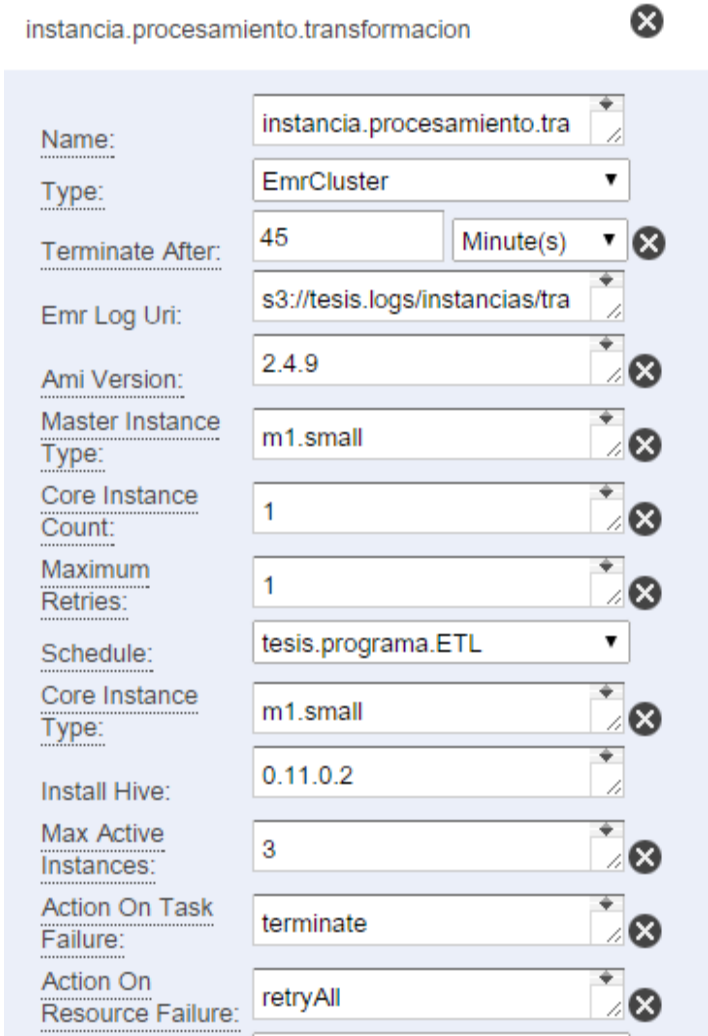
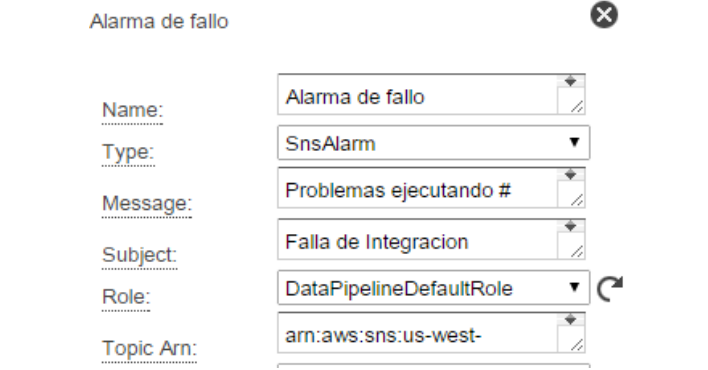
Schedule: tesis.programa.ETL

Directory Path: s3://tesis.transform/#

Data Format: tesis.input.format

<p>Carga</p> <p>Actividad de tipo Copy encargada de trasladar todos los archivos existentes en el nodo de almacenamiento de datos transformados hacia la bodega de datos MySQL configurada en tesis.warehouse. Al igual que las demás actividades, esta hace un envío de alertas en caso de éxito o fallo durante su ejecución. Esta actividad tiene como pre condición la ejecución de transformación.WSN como muestra la figura.</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: right;">Carga.WSN ✕</p> <p><u>Name:</u> Carga.WSN</p> <p><u>Type:</u> CopyActivity</p> <p><u>Schedule:</u> tesis.programa.ETL</p> <p><u>Input:</u> tesis.transform</p> <p><u>Runs On:</u> instancia.procesamiento.carg ✕</p> <p><u>On Success:</u> Alarma de exito ✕</p> <p><u>On Fail:</u> Alarma de fallo ✕</p> <p><u>Output:</u> tesis.warehouse</p> <p><u>Depends On:</u> Transformacion.WSN ✕</p> </div>
<p>Bodega de Datos</p> <p>Nodo de almacenamiento de tipo SqlData. Contiene la cadena de conexión hacia la base de datos en adición a la tabla donde serán insertados los registros transformados. Se debe incluir una sentencia SQL para la inserción de los registros durante la ejecución de la actividad de carga.</p>	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: right;">tesis.warehouse ✕</p> <p><u>Name:</u> tesis.warehouse</p> <p><u>Type:</u> SqlDataNode</p> <p><u>Schedule:</u> tesis.programa.ETL</p> <p><u>Insert Query:</u> insert into #{table} (Node_id, ✕</p> <p><u>Table:</u> TblCarga</p> <p><u>Database:</u> tesis.wh ✕</p> </div>

<p>Programador</p> <p>Este programa indica la frecuencia de ejecución de todas las actividades de la ETL. Estipula una ejecución diaria desde la primera vez que sea activado. Este programa rige la ejecución de todos los procesos en la nube.</p>	<p>tesis.programa.ETL ✕</p> <p><u>Name:</u> tesis.programa.ETL</p> <p><u>Type:</u> Schedule</p> <p><u>Start At:</u> FIRST_ACTIVATION_DATE_TIME ✕</p> <p><u>Period:</u> 1 Day(s)</p>
<p>Instancias de Extracción</p> <p>Recurso del tipo EC2 sobre el cual se ejecutan las actividades de extracción. Su configuración total se aprecia en la imagen.</p>	<p>instancia.procesamiento.extraccion1 ✕</p> <p><u>Name:</u> instancia.procesamiento.ext</p> <p><u>Type:</u> Ec2Resource</p> <p><u>Terminate After:</u> 15 Minute(s) ✕</p> <p><u>Instance Type:</u> t1.micro ✕</p> <p><u>Schedule:</u> tesis.programa.ETL</p> <p><u>Log Uri:</u> s3://tesis.logs/instancias/ext</p> <p><u>Role:</u> DataPipelineDefaultRole ↻</p> <p><u>Resource Role:</u> DataPipelineDefaultResource ↻</p> <p><u>Instance Count:</u> 1 ✕</p>

<p>Instancia de Transformación</p> <p>Recurso de tipo EMR o Hadoop en la nube. Estipula la creación de un cluster con una instancia maestra del tipo EC2 m1.small y una instancia esclavo del mismo tipo. Su configuración total se evidencia en la figura.</p>	 <p>The screenshot shows the configuration for an EMR cluster named 'instancia.procesamiento.transformacion'. The settings are as follows:</p> <ul style="list-style-type: none"> Name: instancia.procesamiento.tra Type: EmrCluster Terminate After: 45 Minute(s) Emr Log Uri: s3://tesis.logs/instancias/tra Ami Version: 2.4.9 Master Instance Type: m1.small Core Instance Count: 1 Maximum Retries: 1 Schedule: tesis.programa.ETL Core Instance Type: m1.small Install Hive: 0.11.0.2 Max Active Instances: 3 Action On Task Failure: terminate Action On Resource Failure: retryAll
<p>Alarmas</p> <p>La imagen evidencia la forma de configurar alarmas de éxito o fallo haciendo uso del servicio SNS de amazon para envió de correos electrónicos después de la ejecución de actividades.</p>	 <p>The screenshot shows the configuration for an SNS alarm named 'Alarma de fallo'. The settings are as follows:</p> <ul style="list-style-type: none"> Name: Alarma de fallo Type: SnsAlarm Message: Problemas ejecutando # Subject: Falla de Integracion Role: DataPipelineDefaultRole Topic Arn: arn:aws:sns:us-west-

