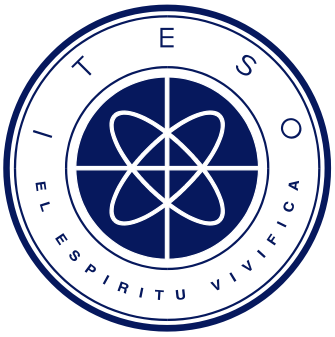
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

MAESTRÍA EN SISTEMAS COMPUTACIONALES



BUSINESS INTELLIGENCE Y BIG DATA APLICADA PARA ANALIZAR INFORMACIÓN DE SALUD PÚBLICA DEL ÁREA METROPOLITANA DE GUADALAJARA

Trabajo recepcional que para obtener el grado de

MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: Ing. Genaro García Fermín
CVU: 711443

Tutor: Mtro. Víctor Hugo Ortega Guzmán

Tlaquepaque, Jalisco, octubre 2017.

AGRADECIMIENTOS

El autor desea dar las gracias al sistema educativo mexicano que ofrece vías para el desarrollo profesional como lo son escuelas públicas y privadas cuyo nivel académico es de gran calidad. Especialmente, agradezco al Instituto Tecnológico y de Estudios Superiores de Occidente en el que tuve el honor de desfilar como estudiante de licenciatura y posgrado. Agradezco enormemente los apoyos económicos que recibí para cursar mis estudios en esta casa de estudios.

Extiendo mis agradecimientos al cuerpo académico del Departamento de Electrónica, Sistemas e Informática que, con su incondicional apoyo, me formé profesionalmente en el área de Ingeniería y Maestría en Sistemas Computacionales. Especialmente a los profesores que me impartieron clases, así como los coordinadores y asesores, especialmente a el Maestro Víctor Hugo Ortega Guzmán, la Doctora Mildreth Isadora Alcaraz Mejía, y el Doctor Luis Fernando Gutiérrez Preciado.

Además, quiero agradecer al Consejo Nacional de Ciencia y Tecnología, CONACYT, que me otorgó una beca con número (CVU/Becario): 711443/591587, a partir de la fecha 01 de agosto de 2015 hasta el 31 de julio de 2017, para realizar mis estudios de maestría en el programa Maestría en Sistemas Computacionales, mediante su convocatoria 290981, Convocatoria Para Posgrados Con La Industria 2015.

DEDICATORIA

El autor dedica esta tesis a los médicos, enfermeros y colaboradores que laboran en el sector público y privado, y los investigadores en el área de la salud que, en conjunto, sus esfuerzos ayudan a mejorar la calidad de vida de pacientes y enfermos. Así mismo, extiendo la dedicatoria a todos aquellos que investigan y desarrollan herramientas de *Big Data* y *Business Intelligence* enfocadas en mejorar y crear herramientas de software para el sector salud.

RESUMEN

El presente trabajo tiene por objetivo analizar la información pública de las dependencias del sector salud y presentarla al público en general de una manera accesible. El análisis se acota al área metropolitana de Guadalajara. Para lograr este objetivo se utilizan las técnicas de *Business Intelligence* y Big Data.

El trabajo consta de una investigación del estado del arte de los métodos y procedimientos utilizados para manejar grandes cantidades de Datos, y del proceso de creación de la herramienta final con su respectiva documentación.

El problema principal que se aborda es el desabasto de medicinas en los hospitales públicos. Los hospitales públicos en México sufren de constante desabasto de medicinas. Los pacientes, por tanto, no pueden llevar al pie de la letra los tratamientos indicados.

Los pacientes de enfermedades crónicas, estacionales, degenerativas o de cualquier tipo, al ser recetados por sus médicos están a la expectativa de la disponibilidad de los medicamentos. Ya sea por falta de recursos, o falta de estudios de inventarios, los Hospitales públicos constantemente envían a sus pacientes a adquirir sus medicamentos en el sector privado, o los hacen esperar hasta que haya en existencia, deteriorando significativamente la calidad de vida de los pacientes.

Al analizar la información combinada de los pacientes y sus requerimientos de medicina, las instituciones de salud pública podrían predecir el inventario requerido y anticiparse a su adquisición. Esto podría evitar desabasto, caducidad, costos de inventario y logística, y dar mejor servicio a todos los pacientes.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ACRÓNIMOS Y ABREVIATURAS

API	*Application programming interface*
AWS	*Amazon Web Services*
AWS Console	*Amazon Web Services Console used to manage cloud services*
BI	*Business Intelligence*
CCD	Ciudad Creativa Digital
CUCEA	Centro Universitario de Ciencias Exactas y Administrativas de Guadalajara
DSS	Sistemas de Soporte de Decisiones
E/R	Modelo Entidad Relación. Se refiere a los modelos de Bases de datos Transaccionales.
EC2	*Elastic Cloud Computing. A service in Amazon Web Services*
EMR	*Elastic Map Reduce. A service to used Map Reduce as a service*
ETL	*Extraction Transformation and Loading*
GFS	*The Google File System*
HOLAP	*Hybrid OLAP*
IIEG	Instituto de Información Estadística y Geográfica
IMEPLAN	Instituto Metropolitano de Planeación del Área Metropolitana de Guadalajara
IMTJ	Instituto de Movilidad y Transporte del Estado de Jalisco
ITEI	Instituto de Transparencia, Información Pública y Protección de Datos Personales del Estado de Jalisco
MOLAP	*Multidimensional OLAP*
OLAP	*On Line Analytical Processing*: Agiliza las consultas de grandes cantidades de datos mediante le modelado dimensional de datos (Cubos), partiendo de bases de datos Entidad Relación Transaccionales (OLTP - On Line Transaction Processing)
RDD	*Resilient Distributed Data Frame used by Spark and Hadoop*
ROLAP	*Relational OLAP*
SaS	*Software as a Service*
SSJ	Servicios de Salud Jalisco
Token	Palabra clave que sirve a los programas o robots entender una secuencia de información en un texto plano
ZMG	Zona Metropolitana de Guadalajara

1. INTRODUCCIÓN

Resumen: *En este capítulo se presenta brevemente los antecedentes del trabajo de Big Data y Business Intelligence aplicados en general y al sector salud. Además, se presenta la justificación del presente trabajo, finalizando con la definición del problema al que se debe este estudio, que es el análisis de datos abiertos sin restricciones de derechos y lograr predicciones en base a esto. Finalizamos con los objetivos principal y secundarios que se propone a desarrollar, y con la aportación que este trabajo desarrolla.*

1.1. Antecedentes

La información es poder. Sin embargo, la información no es poderosa por sí misma. Tiene que ser relevante, en tiempo y fácil de procesar, con el fin de tomar decisiones correctas. Las dependencias de gobierno de la ZMG han trabajado arduamente para proveer información a los ciudadanos mediante sitios públicos de información, tal es el caso del IIEG el IMEPLAN, el ITEI, la SSJ, el IMTJ, entre otros. Cada sitio de información tiene distintas fuentes de información basados en los servicios que ofrece. Además, cada sitio tiene cuentas oficiales de redes sociales que originan datos en tiempo real.

Las personas, grupos de personas u organizaciones, necesitan información combinada de todos estos sitios para obtener datos relevantes que les ayuden a las decisiones de su vida cotidiana. Combinar toda esta información puede ser un trabajo maratónico.

1.2. Justificación

Dada la cantidad de información pública que se puede obtener de las dependencias gubernamentales de la ZMG, es posible construir un producto de software que recolecte, almacene, procese y analice dicha información. Para esto, existen herramientas de software de licencia de propietario, y de licencia de código abierto. En el presente trabajo se analizan y utilizan las herramientas basadas en código abierto.

Las dependencias gubernamentales de la ZMG están realizando la labor de hacer pública la información mediante Sitios de internet, Aplicaciones móviles y Datos en archivos de texto. Día a día se puede observar cómo los procesos de dichas dependencias están basando sus procesos en herramientas electrónicas interconectadas. Desde obtener una cita para realizar un trámite, hasta consultar datos en línea de los resultados del trabajo del gobierno.

1.3. Problema

Cada dependencia gubernamental de la ZMG genera sus propios datos y los ofrece al público en diferentes escalas y proporciones, enfocadas a su labor específica. Esto implica una rápida generación de datos estructurados y no estructurados en crecimiento constante. A su vez la necesidad de acceder a estos datos también va en crecimiento. Las soluciones de *Business Intelligence* y Big Data, pueden aplicarse a esta problemática.

Este trabajo es acerca de la creación una solución de software basado en herramientas de código abierto enfocada en el análisis de la información pública disponible del sistema de salud de la ZMG para ofrecerla al público en general.

1.4. Objetivos

1.4.1. Objetivo General:

El objetivo principal es crear una solución de software que recolecte, almacene, procese y analice la información pública disponible de las dependencias públicas de salud de la Zona Metropolitana de Guadalajara, y entregarla al público en general en un formato fácil de utilizar.

1.4.2. Objetivos Específicos:

Los objetivos particulares de este trabajo los enumeramos a continuación:

14.2.1 Predecir el requerimiento de medicina para hospitales públicos de la ZMG.

14.2.2 Analizar la información pública de salud de la ZMG combinándolas con otras pertinentes.

14.2.3 Crear una solución fácil de usar y accesible

14.2.4 Analizar y combinar 10 o más fuentes de información de distintas dependencias

14.2.5 Trabajar con Datos Estructurados y No estructurados

14.2.6 Basarse en herramientas de código abierto

14.2.7 Realizar un análisis del estado del arte, para aplicar las técnicas más actuales en el área

1.5. Novedad Científica, Tecnológica o Aportación

La aportación de este trabajo es la combinación de distintas técnicas, herramientas, *frameworks* y tecnologías de reciente aparición, que convergen para producir un resultado gráfico de información de salud pública fácil de analizar. Esto implica la recolección de distintas fuentes de datos, realizar las conversiones apropiadas y mezclarlas para generar una aplicación de inteligencia de negocios aplicadas al sector salud.

Utilizamos técnicas de minería de datos, de extracción y transformación de *datasets* públicos. Los *frameworks* utilizados van desde librerías matemáticas como Spark, entornos de almacenamiento y procesamiento como Hadoop, combinado con servicios de nube bajo demanda. Aprovechamos las nuevas tecnologías como Nodejs y Angularjs para crear servicios que son consumidos por *dashboards* completamente adecuados a la necesidad de este trabajo.

En general, comprobamos que la unión de los recursos informáticos de Hardware, Software y servicios son útiles para propósito general, y específico que nos han ayudado a generar esta solución.

2. ESTADO DEL ARTE O DE LA TÉCNICA

Resumen: *En este capítulo se presenta un resumen de los trabajos relacionados con Big Data y Business Intelligence y su aplicación al sector salud, enfocándose en los proyectos locales, que han sido desarrollados gracias a la información que entidades públicas y de gobierno han puesto a disposición del público en general. Analizamos los esfuerzos de ciudadanos e instituciones para obtener información valiosa, con datos de distintas fuentes y los logros obtenidos.*

2.1. Proyectos relacionados

En todas las grandes ciudades existen muchos proyectos Big Data que se enfocan en solucionar problemas de la ciudad tales como falta de energía, aumento del crimen, saturación del sistema de salud, crecimiento del tráfico vehicular, baja calidad del aire, etc. Por su parte, en la Zona Metropolitana de Guadalajara existen proyectos con este enfoque, que están basados en métodos de Big Data.

2.1.1. Proyectos Big Data relacionados

2.1.1.1. Entidades de Gobierno

El Gobierno tiene un gran reto, entregar información a la ciudadanía [4]. Para esto se están gestando distintos proyectos Big Data. Consideremos además que el tiempo y el presupuesto son limitados. En respuesta a esto, el Gobierno de la ZMG lanzó en 2013 un plan para preparar a la ciudad y convertirse una *Smart City*. El plan incluye implementar métricas estructuradas de los indicadores de la ciudad, monitorear la actividad de la ciudad y analizar la información de salida. El resultado deberá ser entregado a la ciudadanía a través de Quioscos de información con interfaces interactivas [4].

Además, el IEEG está recolectando información con métodos de Big Data. Su mayor interés, dada la naturaleza del instituto es adquirir información y maximizar la información que entregan.

2.1.1.2. Ciudad Creativa Digital (CCD)

Ciudad Creativa Digital es un capítulo de la IEEE de Ciudades Inteligentes y pretende transformar a la ZMG en la primera ciudad inteligente de México en un plazo de 10 años. Su principal objetivo es traer el gobierno a sus ciudadanos, logrando esto con tecnología y mecanismos para crear una mejor calidad de vida. La ZMG es la primera ciudad, pero la IEEE incluye otras ciudades más en el país, estas son: Ciudad de México, Monterrey, Tijuana y Querétaro, [4]. El internet de las cosas, cómputo en la nube, centros de datos y una gran cantidad de sensores por toda la ciudad, son los principales temas propuestos por el proyecto. El proyecto CCD se desarrolla en el CUCEA dentro del laboratorio “Living LAB”. Actualmente el ITESO también forma parte de esta iniciativa.

2.1.1.3. Soluciones de Gran Escala

En una escala mayor, las principales compañías de software como Oracle, IBM, Microsoft, Google, Amazon y Facebook han comenzado sus proyectos de Big Data [2]. Han invertido principalmente en adquirir herramientas relacionadas con Big Data. A la escala de billones de *Kilobytes*, estas compañías

almacenan información de sus usuarios y están analizando cómo explotar esa información. Para ellas, la información es el nuevo petróleo.

En general las implementaciones de Big Data son el siguiente paso en lo que respecta a los sistemas de información. Las aplicaciones comerciales combinan información estructurada de sus bases de datos relacionales con la no estructurada proporcionada por sus usuarios, por ejemplo, videos, imágenes, archivos de audio, etc., y la examinan para obtener análisis predictivos.

La información almacenada en sus sitios provee información fresca para actualizar sus productos.

Las redes sociales están recolectando a través de la minería de textos, emails, tweets, imágenes y documentos, para conocer mejor el comportamiento de sus usuarios, y de la sociedad en general. La información que recolectan puede proveer un entendimiento certero de los grupos sociales, grupos políticos, crisis de salud, conflictos bélicos, y problemas económicos.

Como ejemplo en 2009 con la crisis de la influenza, las preguntas de este tópico en el buscador Google, o los comentarios en Facebook, dieron un acertado comportamiento de cómo se expandía el virus, mucho más amplio del que fue establecido por el sistema de Salud.

2.1.1.4. Soluciones en el área científica

Otra aplicación que se están gestando actualmente de Big Data es en el área científica. Los investigadores necesitan grandes conjuntos de datos para combinarlos y obtener resultados más exactos. El reto en esta área es la generación y recolección de datos, porque están basados en sensores colocados en regiones físicas específicas de investigación (ejemplo en los océanos) [5]. La calidad de los sensores afecta significativamente los resultados.

En el área de la investigación de salud, la cantidad de datos digitalizados va en aumento constante. Esta gran cantidad de datos genera un reto para ser almacenados y después, genera la posibilidad de analizarlos con la intención de crear conocimiento que ayude a la investigación médica y promueva la generación de mejores técnicas para tratar enfermedades. Lo relevante de esta área es que, dentro de las grandes ciudades, lo que se aplique para tratar las enfermedades depende de otras áreas relacionadas a la genética de las personas, el tiempo, las estaciones, el clima y la región. Un medicamento creado en base a datos de una región puede no tener los mismos efectos, si es aplicado a personas que vivan en una región diferente. Big Data ayudará a regionalizar los medicamentos [34].

2.1.1.5. DATATON 2014 en Zapopan

En 2014 se organizó un concurso abierto a la población, en donde se convocaba a realizar un proyecto de *Business intelligence* enfocado a mejorar la calidad de vida de los ciudadanos, al crear ya sea rutas de tráfico, herramientas para prevenir el delito, o mejorar el servicio policiaco y de tránsito. Para esto el

gobierno puso a disposición de los participantes, cubos de información procedentes de las siguientes dependencias públicas SHCP, el INEGI, CONAGUA, la SEP, Telefonía celular de Movistar, además de incluir redes sociales como *Twitter* y *Foursquare*. La convocatoria culminó con varios proyectos finalistas y uno ganador [36]. A continuación, se mencionan las características más importantes de estos proyectos y su relevancia. Todos los proyectos se acotaron al municipio de Zapopan, Jalisco.

El proyecto ganador se avocó a predecir los delitos mediante el análisis de mensajes en Twitter y búsquedas en Google. El principal logro de este proyecto es que realizó una caracterización de la delincuencia en zonas específicas o secciones de colonias, dependiente de los establecimientos comerciales, escuelas, tianguis y tipo de vivienda. Sin embargo, el mayor aporte fue de descubrir y prevenir el delito, al hacer análisis de lo que se publica en tiempo real en la red social Twitter, o por búsquedas en Google.

Por otra parte, los demás proyectos destacan en varios aspectos. Por ejemplo, encontrar la relación entre la cantidad de accidentes viales y el número de acceso a las redes sociales en los alrededores o dentro de los automóviles. Dado que, en una relación directa se pretende mitigar mediante la vigilia de los agentes viales y evitar que se utilicen dispositivos móviles al ir manejando.

Otro punto importante tocado por los finalistas fue la planificación de las rutas turísticas en el municipio, según lo que se encuentra en redes como Foursquare, para dar a cada turista un recorrido económico y a gusto personal.

Similar al ganador, un finalista analizó los índices delictivos por zona y por región del municipio, para encontrar los mejores puntos estratégicos, y ubicar en dicho lugar una cámara de seguridad. Lo más interesante de este proyecto fue su análisis, pues se basa en métodos heurísticos, de acotamientos a mínimos de dispersión.

2.2. Proyectos de *Business Intelligence* (BI)

El área de *Business Intelligence* (BI) es un concepto no tan nuevo. Desde la década de 1970 las grandes computadoras ofrecían información relevante a las compañías para innovar sus procesos y productos. Sin embargo, en estos días esta área de estudio abarca muchos más términos y tiene más aplicaciones, pues se ha combinado con Big Data. Una vez que los procesos de Big Data han acumulado una cantidad onerosa de datos, *Business Intelligences* entra en acción para ofrecer una mayor cantidad de datos de salida. Esto significa que el total de datos analizados es mayor y por consecuencia puede ser más acertada la predicción [1]. La información que se analiza es más compleja y proviene de muchas fuentes de datos distintas. En general *Business Intelligence* puede ser visto desde sus diferentes áreas de estudio como lo son los Sistemas de Decisión, los Sistemas de información, los Sistemas de Competición y los Sistemas de Administración del Conocimiento.

2.2.1. Sistemas de Soporte de Decisiones (DSS)

Decision Support Systems (DSS), sirven a las personas o grupos de personas que deben tomar decisiones ejecutivas. Los sistemas integrados WEB, las aplicaciones móviles y los servicios en línea son parte de los DSS. Al final del completo proceso de Big Data y análisis, el resultado debe ser compacto y mostrar en una forma útil los resultados unificados de varias fuentes.

Los sistemas de información son un estándar de la industria. Sin importar el producto, existe un sistema de información donde se almacena la información que se adapta a la empresa. Los sistemas de información están basados en Bases de Datos Relacionales, y por así decirlo, es relativamente simple recolectar información de ellos. Existen herramientas que colectan datos de estas bases y los transforman en bases dimensionales listas para ser consumidas por un sistema DSS. Los sistemas DSS encuentran su complejidad cuando es necesario recolectar información desde distintas fuentes, aún estructuradas, pero con diferentes formas de datos. Este es un reto mayor si se hablan de bases de datos no estructuradas [1].

BI tiene una relación muy fuerte con los sistemas de administración del conocimiento y los sistemas de Competitividad [1]. La recolección de datos de fuentes externas o no propias, también afecta directamente la toma decisiones. Es indispensable obtener información de fuentes externas, donde el reto va más allá de las herramientas pues siempre debe estar a pegado a derechos de terceros, y mantener una conducta ética.

2.3. Caso de Estudio en México Programa RE-AIM

La herramienta RE-AIM examina los factores individuales y organizacionales públicos de los programas de salud del Gobierno, para determinar el grado de alcance los programas de mejora del estado físico de los participantes de dichos programas [38]. Este proyecto fue motivado por las estadísticas nacionales encontradas en las Encuesta Nacional de Salud y Nutrición ENSANUT. El reporte arrojó que el 19.4% de los adultos de México son físicamente inactivos, y a su vez esto indica que del 2006 al 2012 hubo un aumento de la inactividad en los adultos del 44.0%. Otro dato preocupante es que el 70.0% de los adultos mexicanos y el 26.0% de los niños presentan sobrepeso, conduciendo al incremento paulatino de casos de diabetes tipo 2 y problemas cardiovasculares [38].

Por estas razones, los programas de salud y las iniciativas son reconocidos por su impacto en la vida de los niños y adultos. Un ejemplo de este tipo de programas son las Ciclo vías en las capitales del país. Además, existe un constante reforzamiento de los espacios públicos donde las personas pueden realizar un sinnúmero de actividades físicas. Estos programas han sido bien recibidos por la población, sin embargo, han existido pocas investigaciones sistemáticas que realicen un análisis del impacto real que estos programas tienen.

El programa RE-AIM es una forma de evaluación que incluye los factores individuales y organizacionales que pueden proveer evidencia del resultado de los programas públicos y su impacto en la sociedad y organizaciones, para poblaciones que estén interesadas en promover este tipo de

programas. RE-AIM desde el punto de vista de la investigación pretende proveer información balanceada entre factores internos y externos. Desde la perspectiva práctica el objetivo de este Programa es proveer la información adecuada a los educadores y organizaciones para decidir cómo implementar los programas y adoptarlos a un costo razonable.

Para desarrollar RE-AIM siguieron la siguiente metodología. Identificaron los programas mediante la publicidad que se le daba a cada uno entre los años 2008 y 2013 en México. Todos los programas encontrados tenían además de la publicidad, páginas de internet con información, consejos y en algunos casos, los motivadores. Para proceder se analizó cuál de los programas contenía resultados y análisis de los mismos, así como retroalimentación o evaluación. Se dio prioridad a estos.

De todos los programas seleccionados, se realizó un filtro dependiendo de la información que se podía encontrar en internet respecto al programa. Para esto se filtraron con base en varios factores, principalmente:

- Alcance del programa, Nacional, Estatal, Municipal,
- Alcance étnico, de edades y de género,
- La eficacia o Efectividad del programa, si se encuentra información de las evaluaciones y sus resultados, además de la facilidad de obtención de la información,
- Cómo fue implantado el programa y si tenía políticas y reglamentos claros,
- Información sobre su implementación,
- Por último, información sobre su mantenimiento, cuanto tiempo duró, y si fue replicable [38].

Después de pasar todos los filtros, en este caso de estudio, se terminaron con 12 programas a nivel nacional que contenían suficiente información para realizar un análisis completo.

Este caso de estudio es importante como ejemplo de selección de información pública. Considerando por ejemplo responder todos los cuestionamientos que los autores propusieron para determinar si una fuente de datos era utilizable o no. La selección adecuada de los sitios confiables cuando se analiza información pública es de vital importancia y puede ser la diferencia entre el fracaso y el éxito del proyecto de Big Data.

Otro tema importante a relucir para un proyecto de continuidad, es el mantenimiento de las fuentes de información, ¿Con que frecuencia la información se actualiza? ¿Es información fidedigna? ¿La fuente de información es oficial? ¿Existe un programa a largo plazo que mantenga el sitio de información activo? ¿Los formatos de información cambian constantemente?

2.4. Retos y oportunidades del Big Data en el sector Salud

Al igual que en cualquier sistema de Big Data, cuando se aplica en el sector salud, existen 2 retos importantes tanto para modelar como interpretar la información. Sin un sistema de representación, pre-análisis e inferencia, mucha información recolectada puede malinterpretarse o esconderse en la gran cantidad de datos. A pesar de los avances en todas las técnicas de minería de datos, se requiere más innovación y desarrollo, dada la complejidad de las entradas en este contexto, como pueden ser imágenes, recetas, comentarios de doctores y pacientes. [22]

Al momento de escribir este documento, no encontramos en las comunidades de código abierto, o con los desarrolladores de software propietario, una solución canónica capaz de representar, analizar e inferir resultados, con entradas incongruentes de procedencia diversa, y de múltiple formato. Esto se debe a que la cantidad de información crece a mayor velocidad, que las herramientas para manejarla [22].

Enfocando en el análisis de datos generados en el sector salud, se definen 4 fases propias de este contexto. Primero se debe conocer la complejidad de los datos a analizar. Después es necesario representar la información de forma simple y fácil de manejar. La siguiente fase consiste en el modelado y por último la interpretación o inferencia [39].

Las entradas en el contexto de Big Data de la salud, son principalmente del tipo no estructurado, lo que significa que mucho de su valor radica en la calidad y no la cantidad, la congruencia y la relación con el contexto. Ejemplos claros de este tipo de información son las imágenes biomédicas, los ultra sonidos, videos, ondas de resonancia, secuencias de genomas, observaciones médicas, imágenes en 3D [22].

Las herramientas que se utilizan para analizar este tipo de datos son las siguientes. Para las notas de los médicos o las observaciones, se utiliza el aprendizaje máquina, el análisis lingüístico, arroja información sumariada que puede ser interpretada. Se hace un conteo de palabras repetidas para determinar la importancia, y el análisis lingüístico, sirve para contextualizar la información.

Otra técnica importante que puede aplicarse para analizar los datos de Big Data es mediante Redes de grafos. En general como se ha visto, las redes sociales, la información biomédica, los sensores, e incluso la información proporcionada por el Gobierno contiene mucho valor. Sin embargo, ese valor puede estar oculto, o no ser visible en un texto plano. Mediante las redes de grafos se pueden categorizar las características de los grupos, y unirlos con sus vínculos en común. Esta técnica pone a una sola vista, decenas o centenas de relaciones entre los datos como se agrupan entre ellos [22].

Otro método utilizado en Big Data en Salud es el de Clasificación. Existe una gran cantidad de algoritmos disponibles para clasificar la información. Entre los más utilizados se aplica el aprendizaje de máquinas y el reconocimiento de patrones. Una técnica utilizada es la *Gaussian Mixture Model* (GMM), que consiste en un modelo paramétrico de la probabilidad de la distribución de las mediciones continuas de un sistema biométrico. Otra técnica muy utilizada es la clasificación “K” del vecino más cercano, basado en el método *K-means*.

Una Fuente de métodos y procedimientos que debe utilizarse para encontrar lo más nuevo es la herramienta scholar de Google, tal como lo hacen en [28] y [30], donde documentan una serie de estudios de Big data enfocada a los servicios de salud. Logran analizan los retos de almacenamiento y procesamiento de análisis clínicos para determinar las causas y síntomas de pacientes mediante la predicción de enfermedades por su recurrencia e incidencia.

3. MARCO TEÓRICO/CONCEPTUAL

Resumen: *En este capítulo se presentan las bases teóricas y conceptuales sobre Big Data y Business Intelligence y su aplicación al sector salud, que servirán de soporte para el desarrollo de este trabajo. Comenzamos con la teoría elemental que nos ayuda a entender los fundamentos de Big Data, sus retos y aplicaciones y su relación con las demás disciplinas en el área de tecnologías de la información. Continuamos con un análisis de los elementos que conforman el tema de Inteligencia de negoscios. El análisis se enfoca principalmente a las herramientas de código abierto y su disponibilidad. Todos los conceptos convergen para mostrar un abanico de opciones disponibles para encontrar una solución para el tema propuesto.*

3.1. Bases de Datos Relacionales y Dimensionales

Las bases de datos son conjuntos de datos información relacionada entre sí, almacenados de manera ordenada para que sea simple agregar más información leerla, etc. Existen diferentes modelos que dependen su uso. Algunos buscarán acrecentar su capacidad de lectura, mientras que otros su capacidad de lectura. En relación con este trabajo, las bases de datos que analizaremos serán las bases de datos relacionales basadas en el modelo Entidad / Relación, cuya principal aplicación son los sistemas ERP y Administrativos; mientras que las bases de datos Dimensionales tienen una aplicación más concreta para el análisis de Información.

Estos modelos tienen relación de proveedor – consumidor en un sistema de Inteligencia de negocios. El Sistema ERP produce información primaria y la almacena en una base de datos relacional. Por su parte un proceso extractor se encarga de guardar esa misma información en una base de datos dimensional, que permite explotarla desde el ángulo de la inteligencia de negocios. A continuación, se analizará el funcionamiento de este proceso y sus implicaciones.

Lo primero a revisar es la conversión de modelo Entidad / Relación (E/R), a Dimensional. El proceso para convertir un modelo de datos Entidad / Relación, E/R a modelo dimensional es el siguiente:

Paso 1: Identificar los diferentes procesos del modelo E/R. un modelo ER puede tener muchos procesos, y con ello las tablas involucradas

Paso 2: Identificar las relaciones muchos a muchos en el modelo ER, pues cada relación N a M genera una tabla de hechos - *Fact Tables*. Ejemplo, Facturas de Venta y Detalle de Facturas de Venta. En general Las tablas con relaciones muchos a muchos son las tablas transaccionales

Paso 3: De las tablas sobrantes involucradas en el proceso, que no fueron seleccionadas en el paso 2, se crean tablas de dimensiones. Esto es, generar una tabla que incluya todos los elementos de varias tablas relacionadas, por ejemplo, se combina las tablas Cliente con los catálogos Ciudad, Estado, País. Este proceso se llama des normalización, pues implica repetición de datos.

Paso 4: Identificar las tablas de transacciones y las que no son de transacciones, ejemplo Facturas y detalles de factura son tablas de transacción. Las tablas que no son de transacción generalmente almacenan catálogos como productos, clasificaciones, empleados, categorías, etc.

Paso 5: Identificar los campos de fecha y hora en las tablas de transacciones. Estos campos sirven como una dimensión más, Dimensión Tiempo, Dimensión Fecha. En los modelos Dimensionales es importante identificar la Granularidad de la información, es decir, cuan desglosada debe estar la información transaccional, o a que nivel de detalle. En cada proceso, puede haber diferentes tipos de granularidad; sin embargo, la recomendación es hacer diferentes modelos dimensionales para diferentes grados de granularidad [10].

Nivel de Granularidad. El nivel de granularidad deberá ser determinado por la complejidad del negocio que se está modelando, o la cantidad de información en la tabla de hechos. Mientras mayor sea el grano, existirá menos detalle para el análisis posterior. Es importante analizar los requerimientos actuales del

modelo según el negocio, y además tratar de prever los futuros. La granularidad también afecta el espacio físico que abarca la base de datos [10].

Existe la posibilidad de crear diferentes tablas de hechos con diferente nivel de detalle, aun proviniendo de la misma información de las tablas transaccionales. Las tablas de Tiempo y Fecha pueden convertirse en tablas de hechos si se relacionan con sumatorias, con corte a una Hora y Día especifico, ejemplo, todos los días a las 12 horas. Cuando los movimientos transaccionales abarcan diferentes zonas horarias. Se especifica un meridiano de referencia, por ejemplo, el meridiano de *Greenwich*.

Cuando existen diferentes relaciones muchos a uno en cascada, cada nivel se convierte en un atributo de una dimensión. Por ejemplo: Región, Ciudad, Estado. Ciudad y Estado son 2 atributos de la dimensión.

Tablas de Hechos. Las tablas de hechos se basan en distintos eventos que cambian o se suscitan en el tiempo. Existen lo Hechos No aditivos, cuyo valor no puede aplicársele una sumatoria SUM; estos pueden ser Texto, Razones de un valor contra otro, porcentajes o radios, promedios, o valores específicos. Existen los hechos Semi aditivos, cuyos valores pueden ser sumados, o vistos individualmente a un evento dado.

La llave primaria de la tabla de hechos se compone de las llaves foráneas de las tablas de dimensión. Esto es una llave que concatena muchas llaves foráneas. Para garantizar una llave única para cada registro de la tabla de hechos, es necesario utilizar la combinación de todas las llaves foráneas [10].

Las tablas de hechos, también pueden almacenar eventos. Esto las convierte en una tabla de hechos basada en Eventos. Almacena registros según suceda un evento, ejemplo, alta de un contrato.

Administración de cambios. El modelo dimensional debe ser diseñado para manejar distintos tipos de cambios en el tiempo. Estos cambios pueden ser Cambios en los datos, Cambios en la estructura y Cambios en los requerimientos del modelo de negocio. Los de datos tiene que ver con que tan rápido o con cuanta frecuencia deben actualizarse las tablas de hechos. Los de Estructura, tienen que ver con la adicción de nuevas dimensiones, o sub -dimensiones; en este apartado debe entenderse que, de requerirse más dimensión, se requiere más granularidad. Mientras más granulado sea el diseño del modelo dimensional inicial, es más sencillo realizar cambios a la estructura del modelo a nivel de dimensiones o de atributos de las dimensiones.

3.2. Big Data

Big Data se relaciona con grandes cantidades de información, centenares o millares de servidores trabajando en conjunto, diferentes tipos de formatos de archivos y almacenamiento, grandes redes de comunicación de datos que trabajan en conjunto para obtener valor de la información. El término Big Data está ligado intrínsecamente a cantidades descomunales de información, y sobre a los procesos que son necesarios para extraer valor de la misma. Dichos procesos los podemos clasificar en 4, que trabajan en conjunto como una línea de producción. A continuación, hacemos un bosquejo de estos pasos o etapas que componen Big Data.

Big Data se compone de 4 etapas que trabajan en un ciclo infinito; los pasos son:

- Generación de datos,
- Adquisición de datos,
- Almacenado de datos, y
- Análisis de datos.

Cada paso se repite a de forma continua a una velocidad determinada por el propio diseño del modelo [17]. Además, cada uno debe ser entendido de forma separada.

3.2.1. Generación de Datos

La generación de datos incluye todos los dispositivos que convierten las actividades o eventos reales a datos digitales que los representan. Esta etapa entonces, requiere de sensores, instrumentos y dispositivos que recolectan datos del ambiente en donde se encuentran. Los principales generadores de datos son las páginas de internet, dispositivos móviles, sensores, sistemas de información, y redes [13].

Dada la gran cantidad de generadores, la información que se obtiene puede ser relevante o irrelevante para el sistema diseñado. A pesar de que los datos generados son relevantes o irrelevantes, en esta etapa todo es recolectado. Para esto, la etapa de generación de datos se diseña con filtros de información que permite eliminar desde esta instancia todos los datos repetidos o innecesarios [21].

3.2.2. Adquisición de Datos

El segundo paso es la adquisición de datos. Esto implica colectar los datos y trasmitirlos. Depende del sistema de Big Data, en este paso además se incluye un pre procesamiento que hace más fácil analizar los datos. Al igual que una cadena de producción, los datos aquí, son la materia prima de la cual se extraerá la información relevante.

Algo muy importante en esta fase es la redundancia. La materia prima que se envíe a la siguiente etapa debe ser adecuada, precisa y exacta. En algunos sistemas los datos deben ser comprimidos para reducir y filtrar la información correcta [2]. Colectar de sensores implica que existirán muchos datos irrelevantes o ruido; por ejemplo, los datos de una videocámara que visualiza el movimiento en pocos eventos al día; sólo se requiere colectar los datos cuando existe movimiento. Para este caso, además de comprimir los datos, muchos de ellos deben ser eliminados.

El proceso de selección de datos relevantes está incluido en el proceso ETL, en el paso de la transformación. Como hemos mencionado, la compresión, selección y conversión como parte de su transformación se realiza basados en la naturaleza propia de los datos. Como ejemplo imaginemos sistema de fidelización de clientes que obtiene un conjunto de datos de redes sociales. Los datos básicos son las transacciones de texto entre diferentes cuentas; la extracción nos permitirá filtrar información mediante las palabras clave; la transformación convertirá el texto en un conteo de adjetivos positivos y negativos, que podrá cargarse en un gráfico con estadísticas de percepción positiva contra la negativa.

A su vez, todos los procesos de ETL están interrelacionados y su función depende del tipo de información masiva que se genera y el objetivo de su análisis. La misma información del ejemplo anterior puede tratarse para transformarla en un gráfico de intención de compra, donde las palabras clave de filtrado serán distintas y la transformación de información será para un fin diferente.

Las fuentes comunes de adquisición de datos en esta fase son los archivos de log, sensores, videos, archivos de texto plano, o texto posteado por usuarios en las redes sociales. Para cada tipo de fuente de datos existen estándares generalmente aceptados, creados por las principales compañías como IBM, Microsoft, Google, Facebook, etc. Así es que en esta etapa se debe confiar en las fuentes de datos [8].

La fase de adquisición requiere de las mejores prácticas para liberar la mejor salida de datos posible. La mejor salida de datos significa el menor volumen de datos, la mayor velocidad de recolección, la máxima variedad de datos que, en conjunto, puedan generar la información de salida más valiosa [31].

Mientras menor sea el volumen de datos generados, menor será la cantidad de infraestructura requerida para almacenarla. Mientras más rápido se generen los datos, más fresca será la información colectada. Mientras exista más variedad en los datos, se obtendrá la mejor combinación de datos. Y finalmente todo en conjunto, puede maximizar el valor de la información de datos de salida [29].

3.2.3. Almacenamiento de Datos

La siguiente etapa es el almacenado de datos. Cada *bit* almacenado en un sistema de archivos o en bases de datos requiere de energía, espacio físico, y software especial para manejarlo. En esta etapa cada dato es almacenado en una forma manejable y replicable. Sin importar su relevancia, en esta fase, se almacena toda la información [9].

Big data establece requerimientos estrictos de almacenado. Debido a que la información generada cada día va en aumento, así también aumenta el costo de almacenamiento. Además, el almacenamiento debe ser confiable y disponible; esto implica retos de infraestructura como los planes de recuperación en desastres, respaldos de información, y poderosas herramientas [2].

El almacenamiento de sistemas de Big data, comúnmente se diseñan como un sistema distribuido. En este caso se requieren múltiples servidores; los datos deben ser consistentes en toda la infraestructura. Los datos son separados, pero deben ser consistentes sin importar el ambiente de cada infraestructura. El reto aquí es crear un plan que haga los datos disponibles sin importar el espacio y la distancia. La conexión entre los distintos servidores es a través de las redes de datos, y dado que los sistemas de redes pueden fallar, es necesario crear planes de redundancia [7].

3.2.4. Análisis de los Datos

El último paso es el análisis de los datos. Esta etapa requiere métodos, arquitecturas y software para extraer la información almacenada y proveerla en distintas formas y diferentes niveles de vista. Este es el paso más importante en la cadena de Big Data, porque debe proveer información útil, crear sugerencias, y en algunos casos, decisiones [2]. El valor de la información de salida depende de la interpretación que le dé el usuario final. Así, es importante que la salida de este paso se presente de una forma impersonal de forma que evita la malinterpretación.

El reto es expandir las formas de salida de una manera variada. Para esto existen métodos para reducir la malinterpretación de datos. El método de *cluster* analiza grupos de objetos de acuerdo a algunas características que los distinguen y clasifican. El análisis de factores analiza las diferencias entre los objetos respecto a un factor común. El método de correlación determina la correlación matemática entre los objetos o los fenómenos. El método estadístico se basa en modelos matemáticos y en modelos estadísticos. El método de minería de datos extrae información oculta en datos masivos de ruido y datos aleatorios [2].

3.3. Relaciones al tema de Big Data

El concepto de Big Data se relaciona con múltiples conceptos de sistemas informáticos. En general se relaciona con temas como el Internet de las cosas (IoT), cómputo en la nube, grandes centros de datos, *Smart Grids* y *Business Intelligence*.

3.3.1. El internet de las cosas (IoT)

El internet de las cosas significa todos los dispositivos conectados en redes. La gran cantidad de producción de dispositivos móviles hace rentable el crecimiento de las redes. Al final, cada dispositivo conectado es de alguna forma alcanzable desde otro dispositivo en otro punto de la red. Si algo está conectado a la red, existe una forma de entregar y extraer información a él, desde los sistemas de servidores distribuidos.

Todos los dispositivos conectados a la red, utilizan estándares de comunicación para enviar y recibir información de los servidores. La gran masa de datos que estos dispositivos conectados adquieren puede distribuirse fácilmente. Los datos generados vienen en distintos formatos como simple texto plano, video, números, coordenadas, etc. Dada la variedad de datos que provienen de distintas fuentes, los estándares proponen métodos heterogéneos para procesar la comunicación entre los dispositivos.

Además, el internet de las cosas ofrece datos según su relación espacio tiempo, por ejemplo, el lugar y fecha donde se efectúa una videoconferencia. Cada dispositivo se encuentra en diferentes lugares, y de

aquí surge otro reto, el de sincronización. El concepto de ruido es un tema muy relevante en el internet de las cosas, porque entre la información almacenada por cada dispositivo y la información relevante, pueden estar los datos importantes. Por ejemplo, en un video, el ruido creado por el movimiento es el dato importante, y no el resto de imágenes si movimiento.

El internet de las cosas puede describirse por sus 3 partes esenciales: Sensores, Redes y Aplicaciones. Los sensores determinan la adquisición de datos y combinado con las redes, la transmisión. Las aplicaciones son las interfaces con los usuarios, mientras que los sensores son parte de la infraestructura y provee datos, aun cuando los usuarios no participan activamente. Todos en conjunto forman parte del internet de las cosas.

3.3.2. Redes Sociales

Las redes sociales son estructuras creadas por los propios usuarios. Los usuarios dan información a la red sobre los puntos de interconexión y la alimentan con información de todo tipo, relevante o irrelevante. Individualmente, ninguna información es irrelevante, pero para un grupo alguna información individual es irrelevante; para la red total, sólo las sumatorias de datos son relevantes. Es por eso que las redes sociales proveen una cantidad importante de entrada para los sistemas Big Data. Si una página del Gobierno tiene una cuenta de red social, se facilita la entrada de datos a su sistema de Big Data.

La información almacenada en una red social puede proveer información estructurada de le comportamiento humano. Mediante el uso de teorías matemáticas, de una red social es posible determinar la interacción social de los grupos, su estructura, y cómo la información se esparce por toda la red. El público en general determina lo que es aceptado o rechazado en una red social. El reto en las redes sociales, desde el punto de vista de Big Data, es clasificar las entradas de los usuarios para sumariarla y convertirla en información estructurada para ser analizada en un sistema de BI.

3.3.3. Aplicaciones para el sistema de Salud

Big data se relaciona con algunas otras aplicaciones como las de investigación en salud. La información para estos sistemas se recolecta de las investigaciones médicas. Una compañía de seguros con seleccionó pacientes con síndrome metabólico, y les realizó pruebas que dieron salida de más de 500 mil resultados en aproximadamente 3 años. Los resultados fueron combinados con la información de la personalidad de cada paciente. El experimento dio como resultado el principal factor de riesgo que debe evitar cada paciente según su personalidad y el tipo de tratamiento que debería seguir [4]. Con Big data es posible determinar tratamientos personalizados, o basados en el ambiente en que se aplicarán. Otro ejemplo del uso de Big data en medicina, son las investigaciones sobre el ADN y sus variantes.

3.3.4. Computo en la nube

El cómputo en la nube ofrece una gran capacidad de operatividad y gran almacenamiento, para los escenarios de Big data. La nube concentra una gran cantidad de recursos manejados por un solo administrador. Esta forma de trabajo da a los sistemas de BI grandes conjuntos de datos sin importar dónde y cómo están almacenados. El tema de Big data está empujando el desarrollo de los grandes centros de datos que alojan el computo en la nube; existe gran interés en el tema, debido a que sin estos escenarios, las soluciones de Big data no serían posible. Gracias a los esquemas distribuidos, los sistemas Big Data pueden manejar eficientemente el cómputo paralelo y la distribución de información para almacenar y adquirir datos.

Cada solución en la nube tiene su propia personalidad, y tiene variantes según el resultado de comportamiento deseado. Las variantes entre cada sistema pueden influir en cómo se diseñe el ambiente de nube. Esto promueve un reto más que es el de hacerlo conforme a estándares, y aun así que cumpla con los objetivos para los que se crea.

La solución en Big data en la nube deben ser vistas como organismos vivos. Cada nube tiene características propias, pero al final, funcionan de la misma manera, y existen similitudes entre ellas. Todas las funciones principales existen en todas las nubes. Por esto las nubes desarrollan servicios virtuales para procesar las etapas de Big data. De manera estándar se utiliza mucho software como servicio.

3.3.5. Centros de Datos

Los centros de datos con plataformas tecnológicas donde los datos son concentrados y almacenados. Organizan la información, proveen servicios de lectura y escritura de datos y manejan los formatos más actualizados de datos. Los centros de datos alojan datos de forma masiva y desarrollan caminos para darle valor real. Big data promueve el crecimiento de los centros de datos, tanto en infraestructura como en software de administración, y de servicio.

Físicamente también crecen día con día, y esto crea un reto de energía que debe estar presente 24 horas al día, 7 días a la semana, al igual que las redes que transmiten la información de ellos. Las compañías están trabajando en encontrar una forma de proveer energía continua de una manera segura a los centros de datos y a las redes que los interconectan. Cuando la energía de los Centros de datos es estable, continúa y redundante, el servicio que proveen es de calidad y confiable, para proveer flujo constante de datos.

A la par de sistemas de Big data, también existen grandes centros de datos. De otra forma Big data no puede dar un servicio real. Big data se soporta sobre muchos servicios de los centros de datos para compartir información valiosa de forma distribuida. Sin los servicios de los centros de datos, se compromete el resultado de los sistemas de Big data [2].

3.3.6. Sistemas Business Intelligence (BI)

Los sistemas BI también están ampliamente relacionados con Big data. Desde principio de la década de 1980, los datos analizados por los sistemas de BI provenían de los sistemas de información de las compañías. Ahora, además de los sistemas de datos de las compañías, la entrada para un sistema de información proviene de múltiples fuentes distribuidas que generan resultados más exactos. Esto porque mientras más variada sea la entrada de los sistemas BI más exacta y relevante será la salida de estos. Big Data provee a los sistemas BI una gran cantidad de información de entrada para ser analizada [19].

Los eventos aleatorios son resultado de conocer todas las variables que los provocan o generan. Mientras más factores se conozcan alrededor de un evento, más sencillo será predecir el siguiente evento. La exactitud de los sistemas BI, entonces depende de la cantidad y variedad de información que pueda ofrecer Big Data de salida.

3.4. Herramientas de Código abierto

Existen herramientas de software de código abierto que ayuda a los desarrolladores a crear soluciones de Big data. Entre estas tenemos a *Pentaho*, *MongoDB*, *Cassandra*, *R-Studio*, *Kettle*, *Hadoop*, *MySQL*, Sistema Operativo *Ubuntu*, como se muestra en la Figura 1.

Figura 1. Algunas herramientas de Código abierto utilizadas en soluciones Big Data.

Apache Hadoop es un ejemplo. Es un framework ampliamente utilizado para almacenar y procesar grandes cantidades de datos generados por Big Data, mediante consultas sencillas. Hadoop utiliza el método de Mapeo y reducción para reducir la información de grandes conjuntos de datos y los almacenan en nodos distribuidos. Este framework tiene su propio sistema de archivos distribuido que se conoce como HDFS, que garantiza alto rendimiento para entregar datos sumariados.

Cassandra es otro ejemplo de una base de datos de código abierto que maneja una gran cantidad de información estructurada. Trabaja con servidores distribuidos para garantizar alta disponibilidad, así como lo asegura Hadoop. Los datos almacenados en bases de datos de Cassandra pueden ser accedidos mediante consultas estructuradas similares a las que se utilizan en SQL. Sus principales características son alto rendimiento, fácil de implementar, fácil de utilizar. Cassandra nació bajo el esquema del cómputo en la nube, por lo que nativamente maneja servicios distribuidos [14].

Pentaho es una suite de herramientas open source que integran además extracción y análisis de datos. Integra herramientas como Kettle, que es utilizado en el proceso de extracción y transformación de datos. Pentaho provee librerías para análisis visual y predictivo. Su principal enfoque es en la etapa final de Big data que es el análisis.

Como parte de los entornos de código abierto más utilizados tenemos distribuciones de Linux como Ubuntu, que además de su interfaz gráfica, posees todas las características necesarias para desarrollar con framworks como Nodejs, Netbeans o Apache.

Nodejs es un framework recién incorporado al conjunto de herramientas basados en Javascript. Su uso se extiende rápidamente por que ofrece desarrollos rápidos, ambientes ligeros y basados en web que permiten obtener soluciones en tiempo récord y compatible con la mayoría de los navegadores de los dispositivos móviles y computadoras de escritorio. Nodejs permite establecer rutinas a nivel de cliente y de servidor, ofreciendo funcionalidades síncronas y asíncronas que elevan el tiempo de respuesta de las aplicaciones.

3.4.1. Apache Hadoop

El Framework Apache Hadoop esta creado en Java, y su objetivo en administrar datos distribuidos que por su naturaleza pueden o no estar disponibles en un tiempo dado. Hadoop implementa un procesamiento con base a algoritmos de Mapeo y Reducción que ofrecen información pre procesada para los sistemas Business Intelligence. Hadoop utiliza los estándares del software desarrollado como código abierto [24].

El Framework Apache Hadoop es ampliamente utilizado en aplicaciones Big Data [3]. Se utiliza como filtrado de contenido, análisis de redes, análisis de clics, comportamiento social, e investigación de mercado. Además, tiene un nicho de clientes muy importantes en las áreas académicas y de investigación. Grandes empresas como Yahoo, Google, Facebook, Amazon e IBM, corren Hadoop en miles de nodos para maximizar el análisis de conjuntos de datos que se miden en decenas de cientos de Penta Bytes [11].

Las bases de datos relacionales tradicionales pueden también manejar una gran cantidad de información. Sin embargo, para cantidades de datos que operan sobre los Penta Bytes, estos sistemas pueden colapsar. Aquí la importancia de un Framework como Hadoop, con sus técnicas de Mapeo y Reducción, puede reducir la información de los Penta Bytes a niveles de Giga Bytes, o incluso Mega Bytes, dependiendo de la variedad de datos almacenados. Además, las bases de datos relacionales no pueden, ni deben reducir la información almacenada [24].

3.4.2. Framework Hadoop

Hadoop se enfoca en 2 principales problemas: las posibles fallas del hardware, y el almacenamiento en diferentes discos duros distribuidos en una red [11]. Hadoop crea un ecosistema donde los datos pueden fluir rápidamente a la misma velocidad sin importar la distancia física, haciendo la información disponible debido a su redundancia.

Hadoop tiene su propio sistema de archivos para almacenar archivos de forma distribuida en diferentes servidores, para cumplir con el requerimiento de la confiabilidad y redundancia, y es no depende de un almacenado tradicional tipo RAID. Hadoop crea archivos nodos en lugar de archivos comunes, como un sistema de archivos tradicional.

El sistema de archivos Hadoop no es a prueba de cualquier fallo. Pero pretende replicar los datos en diferentes nodos. Cada nodo puede contener información que otros nodos requerían. Los algoritmos de Mapeo y Reducción combinan la información almacenada en un nodo, y la salida en otro, que será el resultado de una consulta [6]. Esto previene duplicación de tráfico por una misma consulta.

3.4.3. Mapeo y Reducción

El Mapeo y Reducción es un modelo de procesamiento distribuido que se ejecuta en un ambiente que contiene una gran cantidad de servidores de *Cluster* [11]. Utiliza programación paralela que corre sobre varios servidores a gran escala. El mapeo y reducción consiste en 2 pasos o funciones, Mapeo y Reducción. El mapeo genera una llave única o índice que es la dirección de un conjunto de datos; esto para que sea accesible por una consulta. La función de reducción hace sumatorias de los datos que ofrecen una versión reducida o compacta de los datos. Esta es la visión básica del mapeo y reducción. Puede sonar simple, pero cada función requiere de gran procesamiento en paralelo, y muchas sub funciones, que trabajan en conjunto sobre el ambiente creado por el sistema de archivos Hadoop.

La programación paralela que utilizan las funciones de Mapeo y Reducción, proveen al usuario final una interface de fácil uso para adquirir datos con consultas simples. Las fórmulas de reducción además eliminan tiempo de procesamiento al sumariar datos cada vez que se ejecuta una consulta.

El Mapeo y Reducción puede ser implementado en el *framework* de Hadoop que nació bajo este concepto. Dado que Hadoop es una herramienta concebida como distribuida, puede distribuir la carga computacional entre decenas, centenas o miles de Clusters y nodos generando un tiempo de respuesta rápida.

La solución de Mapeo y Reducción puede implementarse en redes distribuidas conformadas por un conjunto de *Clusters* conetados por una red y administrados por el sistema de archivos nativo del Hadoop. Esta red de computadoras puede tener costos elevados de implantación y mantenimiento.

Existen proveedores como AWS que permite explotar las bondades del Mapeo y Reducción sin la necesidad de instalar una gran red de computadoras unidas por el sistema Hadoop. Estos servicios son configurados bajo de demanda, y permiten un pago ajustado a la necesidad de computo.

3.4.4. Hive SQL

Apache Hive es un almacenador de datos distribuidos del tipo de Columna. Hive administra datos de una forma similar a las sentencias de SQL para consultar información [11]. El software de Apache Hive provee herramientas para consultar y administrar información con sentencias del tipo SQL, en un lenguaje que se llama HiveSQL [15].

3.4.5. Pentaho

Pentaho *Business Intelligence* es una herramienta popular de código abierto. Tiene una interfaz de tipo WEB para generar gráficos, integrar datos, y hacer minería de datos. Además, provee funciones de proceamiento, selección clasificación, regresión, asociación y visualización de datos.

Pentaho ayuda a las organizaciones a descubrir a través de los procesos de las empresas a encontrar oportunidades y formas de desarrollar soluciones. El objetivo final de Pentaho es habilitar a las empresas y organizaciones a minimizar el riesgo y maximizar la eficiencia.

Pentaho puede tomar datos de una amplia variedad de fuentes de información, y tienen conectores con servidores, redes, dispositivos móviles, computadoras y la nube. Se caracteriza por ser una herramienta fácil de usar dentro de las aplicaciones de *Business Intelligence*.

3.4.6. Spark

Spark es un framework que puede utilizarse para tratamiento de datos de forma distribuida. Nació como un proyecto de Apache vinculado con Hadoop y preparado para utilizar su sistema de archivos distribuido. Mientras el usuario trabaja con un conjunto de datos único y consolidado, Spark distribuye los datos y los proporciona como un *Dataframe* único.

Entre las librerías más utilizadas de Spark se encuentra la matemática MLlib [27]. Esta librería permite aplicar fórmulas matemáticas para transformar la información, realizar predicciones y aplicar funciones complejas sobre una gran cantidad de datos.

3.5. Minería de datos en Big Data

Big data trabaja con grandes volúmenes de datos, complejos, de múltiples formatos y provenientes de fuentes autónomas. Y por ello se está expandiendo a prácticamente todas las áreas de estudio de ingenierías, incluyendo la física, la biológica y la biomédica. Muchas de están ramas manejan información en demanda cuya primera extracción de datos se hace mediante la minería de datos. Existe un Teorema llamado HACE (por sus siglas en inglés ***H****eterogeneous and* ***A****utonomous sources,* ***C****omplex and* ***E****volving data*), que analiza el proceso de extracción de información mediante la minería de datos [20].

Se consideran que gran parte de la información disponible de en un sistema Big Data tiene las siguientes principales características: Es de gran tamaño, Es Heterogénea, proviene de sistemas Autónomos probablemente descentralizados y distribuidos, y tiene por objetivo explorar en datos Complejos y Evolutivos para encontrar las mayores relaciones posibles [20].

En conjunto todos presentan un reto especial. Derivado de que cada sistema utiliza sus propios parámetros el resultado es diferentes dimensiones, medidas y esquemas. El hecho de que las fuentes de datos estén descentralizadas, distribuidas y sean autónomas provoca información de salida incompleta, pues cada fuente o sistema decidirá lo que comparte y lo que no con el sistema Big Data. A esto debemos sumarle el hecho de que cada vez existe mayor información, y por tanto mayor relación entre los datos [20].

La minería de datos debe entenderse como un proceso iterativo de 3 etapas: Etapa 1 se enfoca en el acceso a bajo nivel de los datos y sumatorias. En la etapa 2 se lidia con la semántica y el significado de los datos. Finalmente, en la Etapa 3 es la más compleja y evolutiva en la que se aplican los algoritmos de minería propios. Cada etapa o nivel tiene sus retos [20].

Los retos de la etapa 1 consisten en la capacidad de cómputo, pues es importante que conforme la cantidad de información aumente, a su vez también, la capacidad de procesamiento; junto con ello, esta primera etapa también enfrenta el reto de almacenamiento creciente.

Los retos de la etapa 2 se producen en el nivel del entendimiento. Es necesario utilizar una semántica adecuada para lograr mezclar la información de distintas las fuentes, bajo un esquema o lenguaje común.

En la etapa 3 los retos implican un adecuado algoritmo de minería que permita rescatar información valiosa, de los datos que pueden estar dispersos, incompletos y complejos. Esta tercera etapa a su vez se divide en 3 partes que manejan cada característica. En la Figura 2 puede observarse el ciclo de las 3 etapas trabajando en conjunto, además de los retos implícitos.

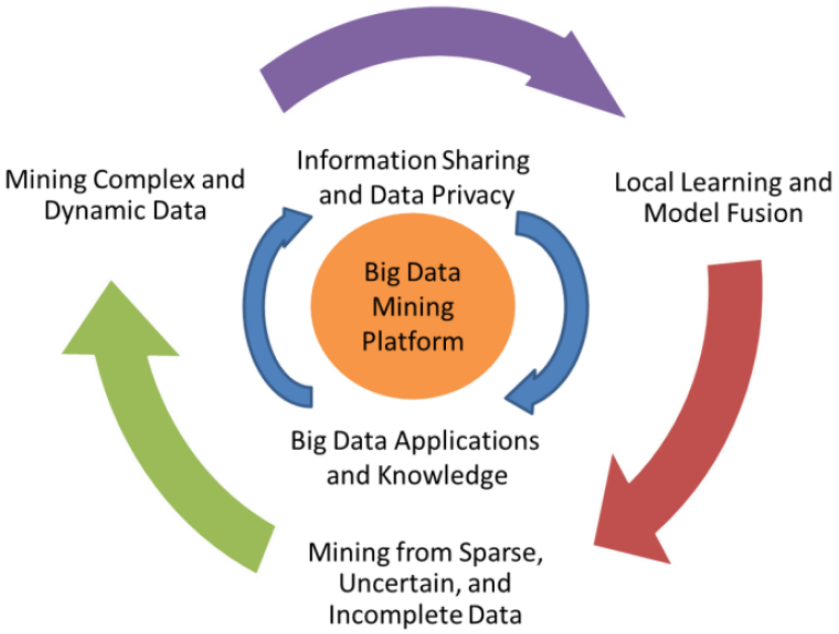


Figura 2. El esquema de las 3 etapas de Big Data en Salud y sus retos. Tomado de [20].

Cada reto visto se enfrenta con diferentes herramientas. La etapa 1 que requiere una alta capacidad de cómputo, se está solucionando con el procesamiento en paralelo, y con técnicas de Mapeo y Reducción de datos. Esto implica un pre procesamiento en paralelo que pueda simplificar la cantidad de datos que trabaja cada procesador en forma separada para después unirla en formas más pequeñas para ser almacenadas. Estos algoritmos son ampliamente utilizados para sobrellevar las complicaciones de trabajar con datos que llegan a los PENTA, o incluso EXA Bytes [20].

Los retos de la segunda etapa se enfrentan con modelos multicapa, que pueden sumariar la información desde distintos puntos de vistas, para llegar un mismo punto de referencia. Cuando se trabajan con datos privados, es posible construir un modelo que permita entregar información bajo diferentes grados de granularidad. Mientras menos granular, menor será el peligro de comprometer la privacidad de la información [20].

La etapa 3 se enfrenta con algoritmos precisos y adaptativos. Se sabe que el conocimiento, mientras va creciendo además evoluciona, por lo que los algoritmos utilizados deberán evolucionar a la par [20].

En conjunto, las técnicas utilizadas para enfrentar los retos, permiten hasta ahora, solucionar la creciente demanda de sistemas que exploten la información de Big data. Mientras que poco a poco, estas tecnologías están siendo probadas, aplicadas y mejoradas, se visualiza un gran avance que puede darnos información muy relevante que a simple vista no es posible observar de nosotros mismos o de nuestro entorno [18].

Es importante entender que existe una diferencia entre los datos que podemos encontrar en sistemas de negocios tradicionales comparado con los del sector salud. A continuación, veremos las principales diferencias entre estos tipos de datos [37].

3.6. Tipos de datos en el sector Salud

En el sector salud, relacionado a Big Data, encontramos las variables volumen, variedad, velocidad y valor, todos en gran cantidad. La Figura 3 muestra la interrelación entre las variables, sus retos y lo que es esperado inherentemente, como lo es, confidencialidad, seguridad y aplicaciones sostenibles. En la siguiente sección se analizan las técnicas de minería de datos, para este tipo de información.

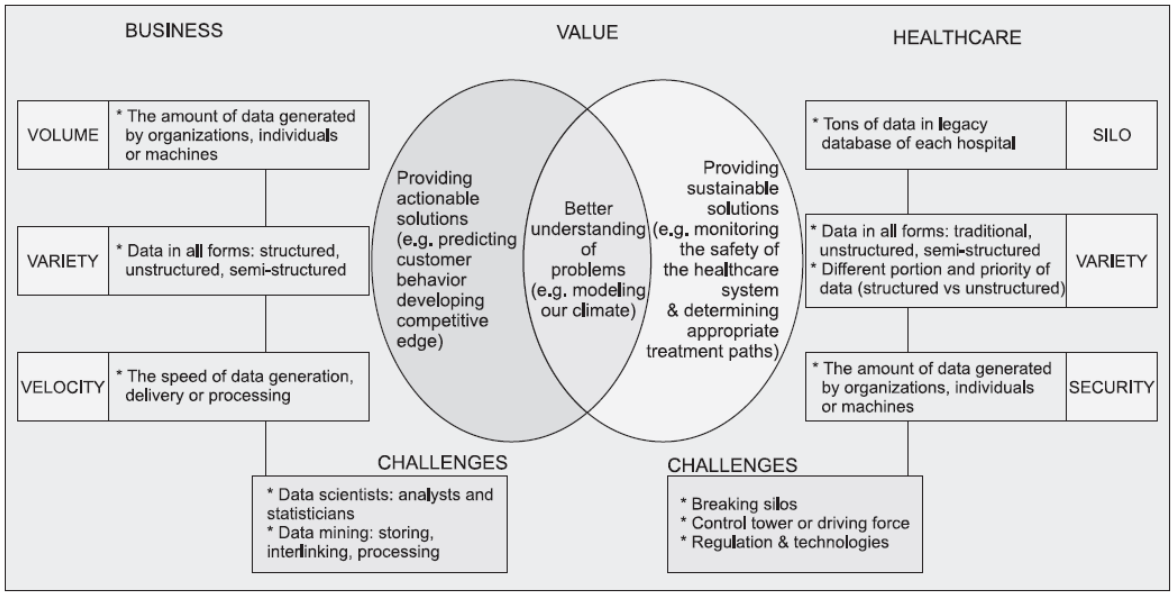


Figura 3. Comparación de Datasets entre el sector salud y los negocios tradicionales Tomado de [37]

3.7. Técnicas de Minería de datos en Big Data aplicados al sector salud

Big data utiliza distintos métodos y técnicas para analizar la información. Algunas deberán ser más adecuadas que otras, si enfocamos los esfuerzos de Big data a temas específicos como lo es a la información generada en el sector salud [23]. La minería de datos se basa en técnicas como el reconocimiento de patrones, el aprendizaje de computadoras, modelos estadísticos y sistemas expertos. Lo importante es seleccionar el método adecuado para el conjunto de datos adecuado.

En el sector salud, la información generada proviene de múltiples fuentes como Bases de datos, Recetas, Información de expedientes de pacientes, apreciación de los doctores, experiencias escritas de los pacientes, etc. Es por eso que para información de bases de datos comunes puede utilizarse métodos

estadísticos, mientras que, para las observaciones del médico o los comentarios de los pacientes, pueden utilizarse técnicas de clasificación de información por reconocimiento de patrones [23].

Las principales estrategias utilizadas en aplicaciones Big data para el sector salud, pueden ser: Algoritmos de predicción, Algoritmos de Clasificación, Estrategias de Exploración de datos adimensionales, y Análisis de afinidad para el reconocimiento de Factores y secuencias [23]. A su vez estas estrategias pueden implementarse para que funcionen de manera supervisada o sin supervisión. Dependerá de la capacidad de autocontrol del sistema. Para los métodos que implican el aprendizaje de máquinas, dependiendo el modelo será útil que sea sin supervisión a fin de que dé resultados en el que no intervenga el factor humano.

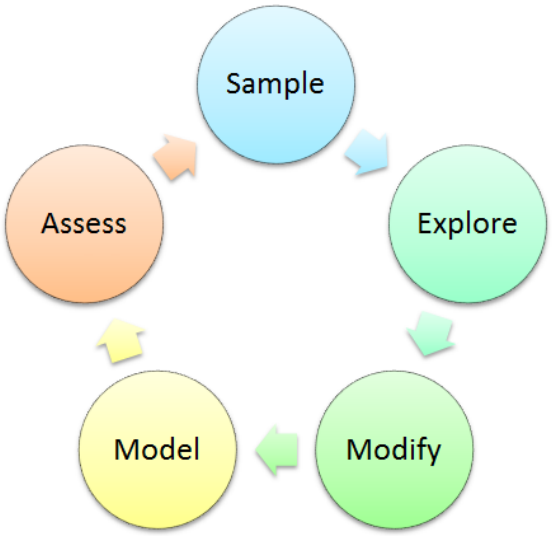


Figura 4. Proceso continuo y cíclico de minería de datos, SEMM. En español, Muestreo, Exploración, Modificación, Modelación y Evaluación.

Antes del uso de computadoras y sistemas de Big data, el proceso de creación de nuevas medicinas podría llevarse muchos años, e implicaba correr muchos más riesgos y resultados sesgados a la lugar y tiempo de experimentación. Ahora con las nuevas técnicas de minería de información se puede reducir el tiempo, el costo, el riesgo y maximizar la eficiencia del resultado de la medicina [23]. El proceso de minería de datos puede verse en distintas etapas: Muestreo, Exploración, Manipulación, Modelación, y Evaluación, trabajando en forma cíclica. Este ciclo completo y su secuencia se muestra en la Figura 4.

Muestreo. Dada la cantidad de información, ahora una muestra puede dar información muy amplia del universo de datos. Más aún si la muestra consiste de datos pre-procesados, de un universo aún mayor. Utilizando la reducción en múltiples iteraciones, puede tomarse un muestreo total de los datos de una forma sumariada.

Exploración. Para explorar los datos resultado del muestreo, contamos también con herramientas gráficas que ayudan acelerar el proceso de análisis. Aún si los datos no son factibles de verse desde una forma gráfica, existen hojas de cálculo donde se pueden hacer otro tipo de validaciones.

Modificación. Al provenir de distintas fuentes, la calidad de datos o los datos faltantes pueden ser demasiados. Por tanto, puede optarse por transformar datos de tal manera, para hacer irrelevantes los faltantes.

Modelación. Una vez que se han localizado los patrones en el desarrollo lo siguiente es realizar el modelo de información. Aquí es importante discriminar los métodos y utilizar el más aplicable según lo que se esté analizando.

Evaluación. Con el modelo completo, puede aplicarse a los datos ya obtenidos y validar que la salida sea válida. De ser así, el modelo puede aplicarse para lo demás. De otra forma, si la salida no corresponde con la misma información de entrada, el modelado esta incorrecto y será necesario reevaluar todo el proceso.

3.8. Sistemas Distribuidos

Los sistemas de almacenamiento para *Big Data* tienen 2 características básicas: son distribuidos, y tienen capacidades de almacenamiento en el orden de *petabytes* o superior. Un sistema distribuido es un conjunto de computadoras que trabajan de manera independiente y colaboran entre sí, para dar al usuario la sensación de que está trabajando con único sistema.

Los sistemas distribuidos deben cumplir con 4 objetivos clave: Hacer accesible los recursos, Ser transparentes, Ser abiertos con estándares claros, y Ser escalables de forma simple a nivel de aplicación y a nivel geográfico. A continuación, se presentan algunas arquitecturas de sistemas distribuidos.

3.8.1. Bigtable de Google

Entre los sistemas pioneros de arquitecturas de almacenamiento distribuido, tenemos el caso de Bigtable de Google inc., mismo que fue publicado en 2006. Bigtable es un sistema de almacenamiento distribuido de cientos o miles de servidores estándares, que puede manejar datos en la escala de *petabytes* [33]. Para el año de su publicación, Bigtable ya era utilizado en productos de la empresa Google inc., como el Indexado de las páginas web de su buscador, Google Earth y Google Finance.

Bigtable está diseñado para almacenar datos estructurados, pero no ofrece un modelo estricto de entidad relación, como sí lo hacen las bases de datos estructuradas. En vez de ello su enfoque se centra en proveer un modelo dinámico y evolutivo, que permita hacer cambios a la estructura y propiedades de los datos. Los datos se almacenan como cadenas de texto, donde los usuarios pueden introducir esquemas de datos estructurados o semiestructurados, que incluyan información sobre ellos mismos [32].

Bigtable es un mapa ordenado de datos que están dispersos y distribuidos, son persistentes y multidimensionales. Dicho mapa se compone de cadenas de caracteres que están indexadas por las coordenadas de ubicación de la columna, fila y unidad de tiempo (*timestamp*) del momento de su

incorporación en el sistema que también sirve para tener un control de la versión de los datos. Esta combinación es univoca [16]. La combinación Columna, Fila y *timestamp* produce una matriz de 3 dimensiones que puede visualizarse como se muestra en la Figura 5.

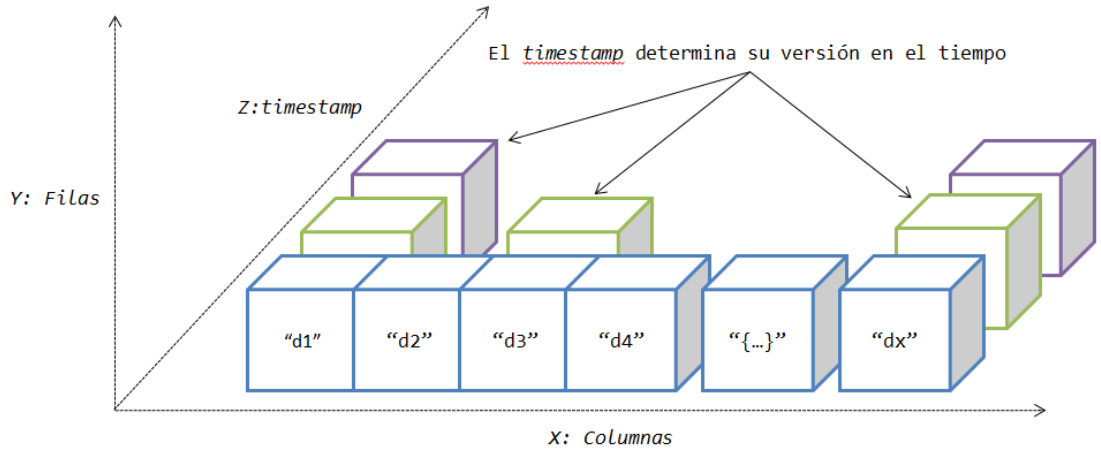


Figura 5. Modelo del mapeo de datos en Bigtable

La Filas en el sistema de Bigtable son llaves de cadena de caracteres que permite hasta 64 kilobytes de longitud. Cada identificador de fila es único y depende del diseñador mantener tener control sobre la secuencia. La longitud permitida ofrece al diseñador múltiples maneras de identificación, que puede incluir hasta oraciones de lenguaje humano. Por su parte Bigtable ordena de forma léxico-gráfica cada fila insertada. Una llave común para la indexación de páginas web es la propia URL [33].

Las columnas son agrupadas en conjuntos y se les conoce como familias de columnas. Las familias de columnas se agrupan por similitud semántica o de tipo de dato. En el caso de la indexación de páginas web, las familias de columnas pueden agrupar las características de cada página web, sus vínculos a otras páginas o su contenido. La agrupación permite a los usuarios de Bigtable controlar el acceso a distintos niveles y manejar la privacidad de información. Se pueden configurar acceso a algunos grupos de columnas para un perfil, y otros grupos de familia para otro perfil [32].

El control de la versión de datos se controla mediante la tercera coordenada: el dato *timestamp*. Cada dato Fila-Columna puede contener diferentes versiones que son indexadas por el tiempo de inserción en Bigtable. La coordenada *timestamp*, es un valor de tipo Entero de 64 bits. Si Bigtable las asigna, *timestamp* contiene el valor en microsegundos de la fecha y hora exacta de inserción. También el control de esta coordenada puede estar bajo el control del usuario, asignándola de forma explícita cuando se inserte un nuevo dato [33].

Bigtable permite controlar la creación de tabla, insertado y actualización de tablas mediante una consola de administración, y también cuenta con una API (*Application programming interface*). Con la API se puede crear y borrar tablas, crear y borrar familias de columnas, administrar el control de acceso a los datos, y finalmente se puede administrar la arquitectura de los *clusters* [33].

Bigtable utiliza el Sistema de Archivos de Google (*Google File System*, *GFS*), para almacenar los datos y el log de actividad de Bigtable. Este sistema le permite distribuir la información en diferentes *clusters*,

mismos que son controlados desde una consola que utiliza los recursos disponibles para distribuir las cargas de trabajo y el lugar de almacenamiento final [16].

El almacenamiento de datos masivo y distribuido es uno de varios componentes que deben ser distribuidos. Otro componente indispensable de los sistemas distribuidos es la plataforma de mensajes entre los diferentes servidores. En el siguiente tema analizaremos el sistema de mensajes distribuidos Kafka, un proyecto *open source* bajo la licencia de Apache.

3.8.2. Plataforma Distribuida Streaming Kafka

El sistema distribuido Kafka fue creado como un sistema de mensajes distribuido para recolectar los mensajes log de páginas de internet a una gran velocidad. Fue creado en LinkedIn para recolectar la información de uso de la red social. La información que se genera en las páginas de internet corresponde a la actividad de los usuarios, clics, “Me gusta”, compartir, búsquedas, etc. Recolectar dicha información ayuda a las compañías a encontrar búsquedas relevantes, crear recomendaciones para los usuarios, encontrar tendencias y para crear aplicaciones de seguridad. Además, ayuda a la mejora continua de los sitios [35].

Kafka Adopta las características de replicación de los sistemas distribuidos y su resistencia a fallos. Utiliza diferentes servidores y resuelves los problemas de hardware mediante software. Como los diferentes sistemas de mensajes, en el sistema existen productores de mensajes y consumidores, que se ayudan de administradores o *brokers*. Kafka agrupa los mensajes en tópicos y cada productor puede crear nuevos tópicos, mientras que los consumidores requieren inscribirse a los tópicos a fin de leer los mensajes ahí almacenados. Como plataforma distribuida, Kafka replica los tópicos a diferentes *brokers*, habilitando a los consumidores a leer los mensajes al mismo tiempo que se están creando. La arquitectura de Kafka se puede ver en la siguiente figura.

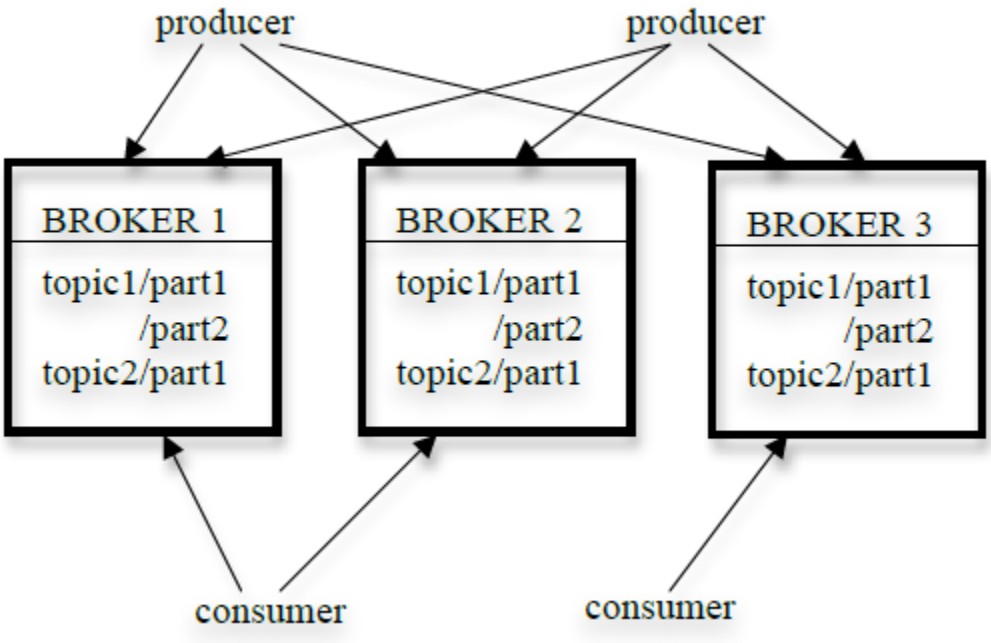


Figura 6. Arquitectura del sistema Kafka. Tomado de [35]

Kafka Puede ser utilizado para diferentes propósitos. Se utiliza como sistema de mensajes que provee mayor tasa de transferencia comparado con sistemas de mensaje tradicionales. Su principal aplicación, que fue su propósito original, es la de llevar un registro de la actividad de un sitio de internet. La arquitectura de Kafka, como se muestra en la Figura 6, permite trabajar con el concepto de Streaming, para consumir información en tiempo real o de baja latencia [35].

El uso de Kafka cada vez está más difundido. Por ejemplo, LinkedIn utiliza Kafka para enviar información de métricas de la página de internet. Twitter Utiliza Kafka para procesar los nuevos mensajes. Por su parte NetFlix utiliza Kafka para hacer un monitoreo en tiempo real del consumo wn su web site. Mozilla utiliza Kafka para recolectar los datos de performance de uso del navegador. Día a día se suman más compañias a utilizar esta herramienta distribuida de Streaming. Las compañias que utilizan Kafka son publicadas en la página de internet kafka.apache.org.

4. DESARROLLO METODOLÓGICO

Resumen: *En este capítulo se presenta en detalle el desarrollo metodológico que incluye la obtención de información pública su procesamiento con herramientas de código abierto y un resumen de los logros obtenidos, los cambios implementados para mejora de procesos y costos. Se incluyen las fuentes de datos iniciales, y las arquitecturas propuestas para logar los objetivos.*

4.1. Levantamiento de requerimientos

Los hospitales públicos en México sufren de constante desabasto de medicinas. Los pacientes, por tanto, no pueden llevar al pie de la letra los tratamientos indicados. Los pacientes de enfermedades crónicas, estacionales, degenerativas o de cualquier tipo, al ser recetados por sus médicos están a la expectativa de la disponibilidad de los medicamentos. Ya sea por falta de recursos, o falta de estudios de inventarios, los Hospitales públicos constantemente envían a sus pacientes a adquirir sus medicamentos en el sector privado, o los hacen esperar hasta que haya en existencia, deteriorando significativamente la calidad de vida de los pacientes. Al analizar la información combinada de los pacientes y sus requerimientos de medicina, las instituciones de salud pública podrían predecir el inventario exacto requerido por día y anticiparse a su adquisición. Esto podría evitar desabasto, caducidad, costos de inventario y logística, y dar mejor servicio a todos los pacientes.

El Objetivo Principal es Predecir el requerimiento de Medicina controlada en Hospitales públicos de la Zona Metropolitana de Guadalajara. A continuación, el plan de Trabajo.

4.1.1. Fuentes de datos

1. Listado oficial de medicamentos controlados Edición 2015

Fuente [25]: Consejo de Salubridad General, http://www.csg.gob.mx/

Plan de acción:

- Obtener el Listado de medicamentos actualizado; convertir el listado en texto plano
- Procesar el Texto para ser introducido a una Base NoSQL
- Almacenar el texto en una base de datos de MySQL
- Medicinas que se utilizan con las enfermedades
- Con un sistema de inteligencia artificial, determinar las medicinas que puede utilizar un enfermo

2. Estadísticas de Enfermedades

Fuente: IMMS http://datos.imss.gob.mx/

Plan de acción:

- Obtener Estadísticas de enfermedades del 2000 - 2015
- Obtener cuantos enfermos se tratan en los hospitales públicos y que enfermedades atienden de 2000 - 2015
- Almacenar las estadísticas en una base de datos de relacional MySQL
- Proyectar las enfermedades por correlación del 2016 - 2017

3. Listado de Unidades Médicas Familiares UMF del IMSS y sus derechohabientes

Fuente: IMSS http://datos.imss.gob.mx/

Plan de acción:

- Obtener estadísticas de derechohabientes por Entidad
- Proyectar las estadísticas por correlación del 2016 – 2017
- Almacenar las estadísticas en una base de datos de MySQL

La Tabla 1 muestra el resumen de la tabla de fuentes de información.

Fuente	Nombre	Rango de Datos	Fuente Pública	Formato	URL
1	Listado oficial de medicamentos controlados	[Edición 2016]	IMSS: Consejo de Salubridad General	PDF	csg.gob.mx
2	Estadísticas de Enfermedades IMSS	[2000:2015]	IMSS: Datos Abiertos	CSV	datos.imss.gob.mx
3	Unidades Médicas Familiares UMF del IMSS	[2000:2015]	IMSS: Datos Abiertos	CSV	datos.imss.gob.mx

***Tabla 1*. Resumen de la tabla de fuentes de información**

4.1.2. Cruzar información

Obtener proyecciones por correlación del 2016 – 2017 para predecir el requerimiento de Medicina controlada en Hospitales públicos de la Zona Metropolitana de Guadalajara. Encontrar las medicinas necesarias para las enfermedades que el IMSS ofrece en sus datos abiertos. Estas enfermedades son en las que la institución ha puesto mucho interés por considerarlas las que producen mayor mortandad [26]:

- Diabetes
- Sífilis
- Cancér de Mama
- Tuberculosis
- VIH
- Hepatitis
- Hipertensión
- Paludismo

4.1.3. Arquitectura de la solución on premise

Como primer acercamiento a la solución, obtuvimos una arquitectura *on premise* donde se incluye un servidor Hadoop, y librerías de Spark [27]. Además, utilizamos el *framework* de Pentaho. La solución fue muy compleja a nivel de infraestructura. Esta solución esta descrita en la Figura 7. Encontramos complejidad infraestructural por la administración de los servicios Hadoop, Spark, Pentaho. En el caso de Hadoop requeiere de 4 servidores, 1 administrador y 3 nodos. El framework Spark, trabaja sobre el servidor de Hadoop, por lo que se reutilizarían los mismos servidores. Spark por su parte, toma los datos depositados en forma de CSV y los guarda en RDD's. Una vez que los datos son procesados, los deposita en archivos de texto que son consumidos por Impala.

Por su parte, la solución incluyendo a MySQL y Pentaho, se diseñan para que convivan en un mismo servidor, pero separado de los demás. De aquí se concibe el uso de una nueva instancia que proveerá el servicio hacia la página web. Impala por su parte, trabajará dentro de los nodos de Hadoop.

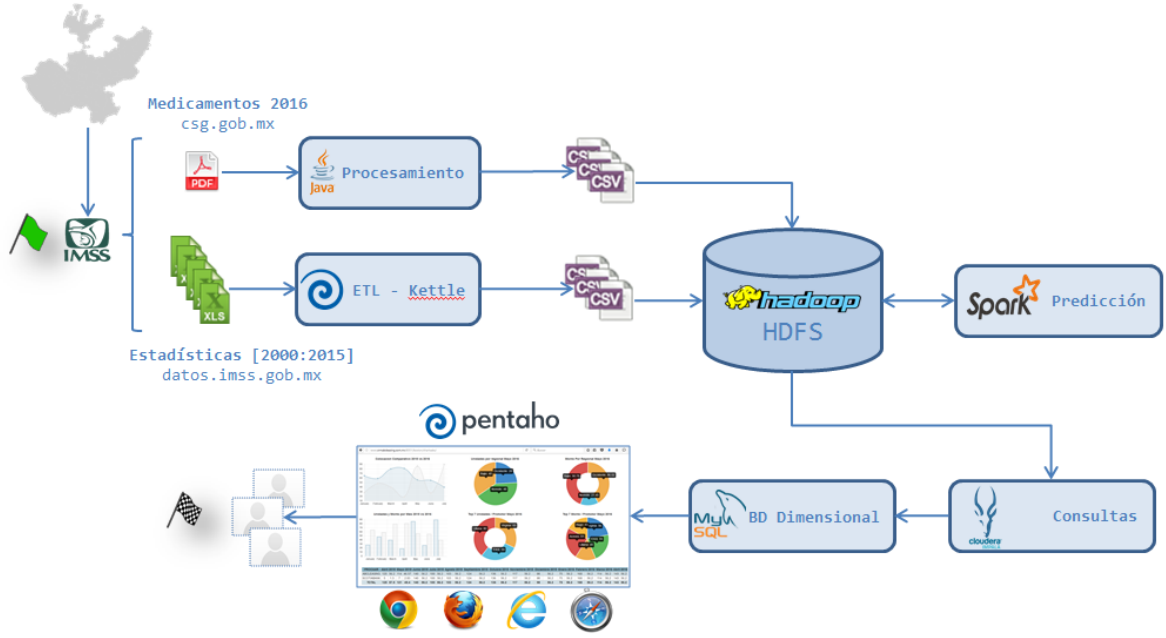


Figura 7. Primera arquitectura de la solución

La cantidad de servidores requeridos y la complejidad infraestructural (a nivel administrativo y de costos) nos condujo idear una segunda versión en la nube, que nos ayuda a utilizar servicios sólo cuando sean necesarios, y evitar costos de mantenimiento que, gracias a las tecnologías de nube pueden evitarse.

Entre los cambios más radicales fue utilizar un servicio de ejecución única para toda la infraestructura de Hadoop, evitando el costo de los servicios y el tiempo requerido para la administración y puesta punto. Otra herramienta que se logró sustituir fue Pentaho. Lo sustituimos por desarrollos simples en Node.js con librerías de Angular.js. Pentaho nos ofrecía una forma de trabajo robusta y con modelos muy sólidos. Por otra parte, también nos exigía mucho tiempo de implementación. Al utilizar librerías

JavaScript logramos el mismo tablero de datos objetivo con mucho ahorro en tiempo de administración y desarrollo.

4.1.4. Transformación hacia Arquitectura en la nube

Consideremos ahora el costo de implementar la infraestructura presentada en el tema anterior. Para el servicio de Hadoop son necesarios 4 servidores, 1 nodo maestros y 3 nodos esclavos. En el nodo maestro se implementarían los servicios de Hadoop con su administrador de archivos, el framework de Spark y el de Impala. Los 3 nodos restantes. Son los mínimos necesarios para efectuar el procesamiento paralelo. Esta es la recomendación en la documentación [11].

Dado el tipo de información de entrada, que cambia de forma anual, implementar la infraestructura completa del ecosistema Hadoop, implicaría una relación costo beneficio muy alta. Por lo tanto, la mejor forma de solucionar esto es utilizar un servicio bajo demanda. Tal es el caso del servicio EMR que ofrece Amazon. Esto es, crear la programación de las consultas a los datos planos de entrada, realizar las proyecciones y depositar los archivos accesibles en la nube. EMR se utiliza bajo demanda, y provee todas las características del ecosistema de Hadoop instalado como recomiendan los autores [6]. A continuación, analizaremos cómo utilizando los servicios de Amazon AWS, llegaremos al mismo resultado sin la inversión en los servidores *on-premise*.

Los servicios de AWS para este proyecto utilizados son **Analytics EMR, Spark Impala SQL y S3**, todo mediante un programa que consume todos estos servicios en una sola ocasión por año. Analytics EMR se utilizará para analizar los datos a través de un programa java con las bibliotecas de **Spark**, mostrado en Figura 8. Esto reemplazará todo un entorno de Hadoop y se ejecutará una sola vez.

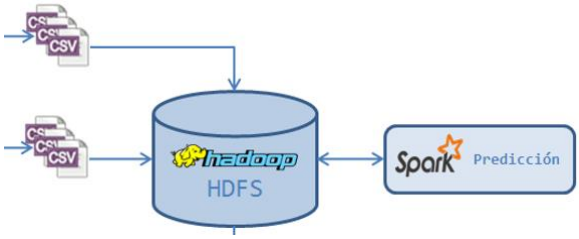


Figura 8. AWS EMR con Hadoop / Spark

Las consultas hacia el texto de los archivos CSV se realiza mediante **Impala SQL**, y son convertidos a archivos de salida que son almacenados también en la nube de Amazon. **Amazon S3** se utilizará para almacenar los resultados del análisis Spark en CSV. La figura 8 muestra la interacción esperada entre los servicios de Amazon.

Ahora analizaremos los servidores MySQL y Pentaho. Como se estableció en el tema anterior, la solución *on premise*, podría combinar los servicios MySQL y Pentaho en un solo servidor. La solución en la nube permitirá flexibilidad de uso y separación de funciones. Esto significa que en un servidor será

necesario instalar ambos servicios y mantenerlos para dar el servicio. Por su parte en nube, es posible separar el servicio de MySQL y el servicio de BI en otro. A su vez, para hacer una separación más amplia podría separase el sistema BI en 2, uno que realice las consultas hacia la base de datos y otro que ofrezca los datos para ser consumidos en la página web. En las pruebas descubrimos que esta separación no era posible con el servicio Pentaho, y su administración fue costosa en tiempo. Para desarrollar un gráfico fue necesario de 5 a 10 horas de desarrollo más el consumo continuo de recursos que Pentaho utiliza para estar disponible.

A lo largo de este trabajo, tuvimos la oportunidad de trabajar con herramientas más ligeras como lo es NodeJs. Hicimos un giro importante apegado al requerimiento inicial y descartamos funcionalidades de Pentaho que no son necesarias para llegar al objetivo. Estas funcionalidades son la administración de ingreso, la interacción con el usuario para que modifique los gráficos y la complejidad de su uso.

Una vez descartadas estas funciones del sistema de BI, minimizamos las funcionalidades a alta disponibilidad, diseño de gráficos, separación de funciones y ligereza de consulta. Al girar hacia frameworks como Nodejs encontramos facilidad de implementación, completa compatibilidad con MySQL y funciones nativas de servicios WEB como REST Ful, que permita tener el acceso a la base de datos separado del servicio que ofrece los gráficos hacia el usuario. Para los gráficos utilizamos la librería Angular JS que trabaja nativamente con NodeJS. Destacamos además que la implementación de cada gráfico fue reducida de 5 a 10 horas en Pentaho a 1 a 2 horas con Angular JS.

Finalmente, para la nube separamos las funciones de la siguiente manera. Para MySQL utilizamos **Amazon RDS MySQL** y evitamos tiempos de instalación y administración. Esta funcionalidad encontrada también en Amazon se utilizará para almacenar una base de datos dimensional para dar formato a los resultados, La sección separada se muestra en la Figura 9.

Figura 9. Base de datos dimensional de MySQL de AWS RDS

Ahora, la separación de funciones de acceso a la base de datos y el servicio de gráficos en 2 será mediante 2 servidores básicos de Linux, con el servicio ligero de NodeJS. La funcionalidad de nube **Amazon EC2** se utilizará para desplegar un servicio Web de Nodejs que se lee desde el RDS MySQL y entrega los resultados en formato JSON para ser consumido por una aplicación de Android o una página Web con plantillas en Angular JS.

Utilizamos 2 instancias EC2 lo que lo dividirá en dos fases: una para el acceso Mysql y la otra para desplegar el servicio web en nodejs. Esta instancia de EC2 podría configurarse como escalamiento automático. Para esto también incluimos un balanceador de carga, preparando la solución para alta demanda. Amazon permite el auto escalamiento, por lo que el costo es incremental y dependiente del uso. Otra de las tantas ventajas del uso de nube.

El resultado final esperado se muestra en la Figura 10. El usuario verá una página Web resultado, y detrás de esta una instancia de base de datos, una instancia de consulta y otra instancia auto balanceada que ofrece la parte menos ligera de la solución que son los gráficos. En la siguiente sección analizaremos el resultado final, y la infraestructura montada en nube con esta novedosa forma de implementar la solución

Figura 10. AWS RDS, EC2 de consulta y EC2 de servidor a los clientes WEB

5. RESULTADOS Y DISCUSIÓN

Resumen: *En este capítulo se presentan los resultados obtenidos del desarrollo de este trabajo y una discusión sobre Big Data y Business Intelligence y su aplicación al sector salud. Se incluye también la descripción de los métodos utilizados como lo son la minería de datos, la predicción y la puesta en marcha de servicios WEB, así como el desarrollo de los pasos propuestos en el marco metodológico. Se presenta una infraestructura final que comprende la misma funcionalidad esperada con los beneficios del cómputo en la nube. Se muestran finalmente los resultados para una mayor comprensión del logro obtenido. Finalizamos con una discusión de los principales aprendizajes y logros en el desarrollo de este proyecto.*

5.1. Resultados

Esta solución se separa en dos partes que se fusionan al final para producir el resultado esperado, que será la entrada para el siguiente proceso. La primera parte es el procesamiento de un archivo PDF en java, y el segundo es el proceso de conjuntos de datos de estadísticas de enfermedades para la institución pública IMSS. El PDF contiene la información de la medicina controlada, y también la proporciona esta institución, y está disponible al público a través de su página web.

5.1.1. Conversión de un archivo PDF a un archivo de texto

El documento PDF recuperado de csg.gob.mx, contiene una descripción de la medicina controlada que se necesita para las enfermedades que el IMSS trata en sus pacientes. La información dentro del archivo tiene una vista humana. El desafío es convertir esa vista humana en una vista de tabla, por lo que podría ser manejable para buscar información. Un ejemplo de la vista humana del archivo se muestra en la Figura 13. El subproceso para la conversión describe el resultado esperado en la Figura 11.

Figura 11. Subproceso que convierte el archivo PDF a CSV.

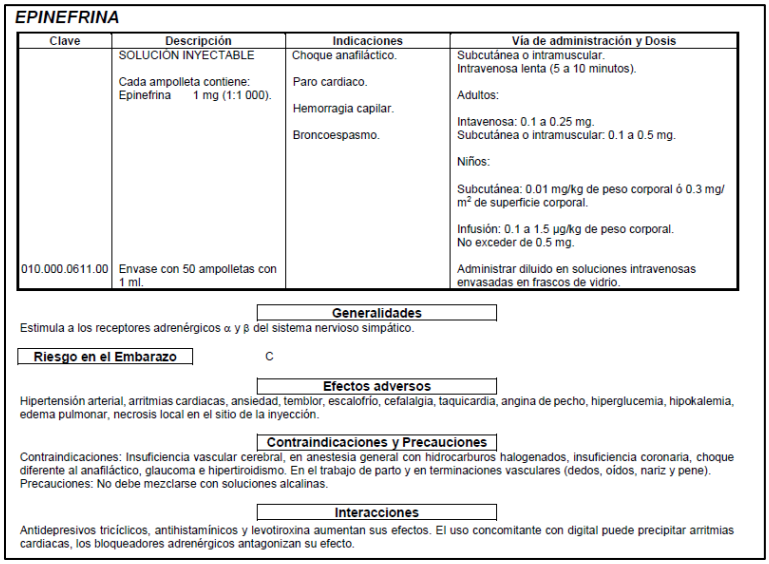
La minería de texto depende de la entrada en este caso el procesamiento del documento PDF, se basa en la estructura de los datos. Un ejemplo de un método adecuado para buscar información dentro de un documento PDF se presenta en [5], donde los autores determinan su propio método de procesamiento de acuerdo con las estructuras de los archivos de entrada. El enfoque para este trabajo se presenta en la Figura 12.

Figura 12. El enfoque del proceso de conversión para esta solución

La sección central, búsqueda de *tokens*, permite localizar información relevante y relacionada. Cada *token* es una bandera dentro del archivo de texto plano que permite seccionar la información y estructurar datos que originalmente no están estructurado.

El código para el proceso mostrado en la Figura 10 fue codificado en Java. La clave principal de la solución es encontrar los tokens que separan la información. Para este archivo PDF los tokens encontrados son:

- `<<Grupo Nº ##>>`
- `<<ÍNDICE GENERAL>>`
- `<<Catálogo>>`
- `<<Cuadro Básico>>`
- `<<Clave Descripción Indicaciones Vía de administración y Dosis>>`
- `<<Generalidades>>`
- `<<Contraindicaciones>>`
- `<<Riesgo en el embarazo>>`
- `<<Interacciones>>`
- `Line Separator <<\r\n>>`



EPINEFRINA

Clave	Descripción	Indicaciones	Vía de administración y Dosis
	SOLUCIÓN INYECTABLE Cada ampolleta contiene: Epinefrina 1 mg (1:1 000).	Choque anafiláctico. Paro cardiaco. Hemorragia capilar. Broncoespasmo.	Subcutánea o intramuscular. Intravenosa lenta (5 a 10 minutos). Adultos: Intavenosa: 0.1 a 0.25 mg. Subcutánea o intramuscular: 0.1 a 0.5 mg. Niños: Subcutánea: 0.01 mg/kg de peso corporal ó 0.3 mg/ m^2 de superficie corporal. Infusión: 0.1 a 1.5 µg/kg de peso corporal. No exceder de 0.5 mg.
010.000.0611.00	Envase con 50 ampolletas con 1 ml.		Administrar diluido en soluciones intravenosas envasadas en frascos de vidrio.

Generalidades

Estimula a los receptores adrenérgicos α y β del sistema nervioso simpático.

Riesgo en el Embarazo C

Efectos adversos

Hipertensión arterial, arritmias cardiacas, ansiedad, temblor, escalofrío, cefalalgia, taquicardia, angina de pecho, hiperglucemia, hipokalemia, edema pulmonar, necrosis local en el sitio de la inyección.

Contraindicaciones y Precauciones

Contraindicaciones: Insuficiencia vascular cerebral, en anestesia general con hidrocarburos halogenados, insuficiencia coronaria, choque diferente al anafiláctico, glaucoma e hipertiroidismo. En el trabajo de parto y en terminaciones vasculares (dedos, oídos, nariz y pene).
Precauciones: No debe mezclarse con soluciones alcalinas.

Interacciones

Antidepresivos tricíclicos, antihistamínicos y levotiroxina aumentan sus efectos. El uso concomitante con digital puede precipitar arritmias cardiacas, los bloqueadores adrenérgicos antagonizan su efecto.

Figura 13. Ejemplo de la estructura del archive PDF. La vista humana.

El resultado final del proceso son dos archivos de texto que están en representaciones de tablas. El primer archivo contiene la lista de todos los medicamentos y su clasificación. El segundo contiene información de cada medicina de sus generalidades, indicaciones y contraindicaciones. Extracto del primer archivo ***catalog.txt***

```
GRUPO|MEDICAMENTO
Grupo Nº 1: Analgesia |BUPRENORFINA
Grupo Nº 1: Analgesia |CAPSAICINA
…
…
Grupo Nº 21: Reumatología y Traumatología |PROBENECID
Grupo Nº 21: Reumatología y Traumatología |TOCILIZUMAB
```

Extracto del segundo archivo ***medicamentos.txt***

```
MEDICAMENTO|OBSERVACIONES|GENERALIDADES|CONTRAINDICACIONES
ÁCIDO ACETILSALICÍLICO|010.000.0101.00 TABLETA Cada tableta contiene: Ácido acetilsalicí...
IBUPROFENO|010.000.5940.00 010.000.5940.01  010.000.5940.02 010.000.5940.03 TABLETA O LA...
PARACETAMOL|010.000.0104.00 TABLETA Cada tableta contiene: Paracetamol 500 mg. Envase co...
BUPRENORFINA|040.000.2100.00 040.000.2100.01 TABLETA SUBLINGUAL Cada tableta sublingual....
CLONIXINATO DE LISINA|010.000.4028.00 SOLUCIÓN INYECTABLE Cada ampolleta contiene: Cloni...
```

Ambos archivos serán útiles para trabajos futuros. En esta solución nos centramos en el segundo archivo. Este archivo se cargará en el sistema de archivos Hadoop y luego se almacenará en caché en un RDD en Spark para buscar el texto en su interior. La parte del código que carga la información en RDD Spark Shell es:

```
[cloudera@quickstart proyecto]$ hadoop fs -copyFromLocal datos/medicamentos.txt genaro/datos

scala> sqlContext.sql("DROP TABLE IF EXISTS default.medicamentos")
sqlContext.sql("CREATE  TABLE  default.medicamentos(code  string,description  string,  indications  string,
contraindications string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' STORED AS TextFile")
sqlContext.tables.show()
sqlContext.sql("LOAD DATA INPATH 'genaro/datos/medicamentos.txt' OVERWRITE INTO TABLE medicamentos")
val df = sqlContext.sql("SELECT * FROM medicamentos")
df.count()
val searchFor = "Hipertensión"
val nuevodf = df.filter($"description".contains(searchFor) || $"indications".contains(searchFor))
val medsFor = nuevodf.map{row => row.getString(0)}
```

La variable **`medsFor`** contiene un RDD con la medicina requerida para enfermedad HIPERTENSIÓN.

5.1.2. Predicción de enfermedades utilizando Spark MLlib

La segunda parte de este trabajo es utilizar el conjunto de datos recuperado de datos.imss.gob.mx, para predecir los nuevos casos de enfermedades para 2017 dada la información anual de 2000 a 2015. El objetivo es predecir la próxima cantidad de casos De una enfermedad dada, el subproceso para conseguir esta información esta descrito en la Figura 14.

Figura 14 La sección del Proyecto para predicción

La fuente datos.imss.gob.mx publicó estos conjuntos de datos de algunas enfermedades:

- datos/datosDiabetes.txt
- datos/dataSifilis.txt
- datos/dataCancerMama.txt
- datos/dataTuberculosis.txt
- datos/dataVIH.txt
- datos/dataHepatitis.txt
- datos/dataHipertension.txt
- datos/dataPaludismo.txt

Cada conjunto de datos tiene el siguiente formato, clasificado por los estados de México:

```
Year,Total,Aguascalientes,BajaCalifornia,BajaCali...
2000,2201975,26009,62837,13967,17871,53247,24920,...
2001,2290956,29255,67355,14736,18326,61285,25590,...
2002,2454077,28309,64090,14053,18987,78064,19364,...
2003,3355434,32601,90481,15426,29797,110003,18299...
2004,2351736,27458,56642,25272,24363,107070,20187...
2005,2382524,30973,59385,23093,25001,113961,22108...
2006,3716332,42093,126909,33082,35237,192351,2905...
2007,2855539,32602,77600,25826,23728,107601,14985...
2008,4522141,98184,82561,50348,29947,182500,21498...
2009,4982626,88425,97274,48486,36664,198804,28315...
2010,5862895,125712,122762,82865,58523,222388,369...
2011,5966064,108924,164273,81929,60446,233023,323...
2012,6680613,105285,205494,93480,59448,242627,414...
2013,6431606,93726,202891,95907,53828,239684,3653...
2014,6108442,124124,213716,68300,56491,230789,297...
2015,6194922,142513,227494,70706,60814,277211,365...
```

Para predecir los próximos años usamos la regresión lineal. La biblioteca de SPARK MLlib tiene métodos para crear modelos de predicción. En este trabajo probamos el API Linear Methods - RDD-based descrito en [12]. En la Figura 15 se muestra un ejemplo gráfico de regresión lineal.

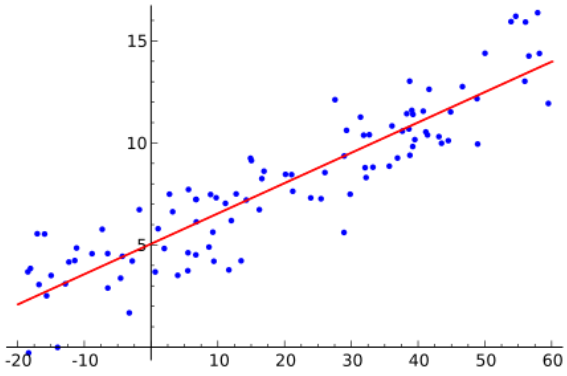


Figura 15. Ejemplo de regresión lineal. Determine f (x) = x para un conjunto de puntos dispersos

El código en Spark se ejecuta en el Scala Shell. Es necesario hacer algunas importaciones para usar las bibliotecas

```
// imports
import org.apache.spark.mllib.feature.{StandardScaler, StandardScalerModel}
import org.apache.spark.mllib.linalg.Vectors
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.mllib.regression.LinearRegressionModel
import org.apache.spark.mllib.regression.LinearRegressionWithSGD
```

Después de leer el conjunto de datos, se prepara como un RDD con LabeledPoints; El RDD siguiente es escalado porque después de alguna prueba, se hizo necesario escalar los datos antes de entrenar el modelo. Con el modelo, ejecutamos una prueba con el mismo conjunto de datos utilizado para entrenar.

El error se almacena en la variable MSE. Finalmente, la predicción contiene el valor de la predicción. Un ejemplo del resultado final, que agrupa ambas partes de este Proyecto, se muestra en la Figura 16. Los valores de entrada fueron Enfermedad: "Cáncer de mama", Estado de análisis: "Guerrero", Año de predicción: 2017

```
Resultados
Cáncer de mama -> Nuevos casos : 79904
16/11/21 09:33:34 WARN lazy.LazyStruct: Extra bytes detected at the end of the
13 Medicamentos necesarios para Cáncer de mama en Guerrero para 2017
16/11/21 09:33:35 WARN lazy.LazyStruct: Extra bytes detected at the end of the
TESTOSTERONA
ANASTROZOL
BEVACIZUMAB
DOCETAXEL
DOXORUBICINA
EPIRUBICINA
GOSERELINA
HIDROXICARBAMIDA
LETROZOL
MECLORETAMINA
MITOXANTRONA
TRASTUZUMAB
TRASTUZUMAB
```

Figura 16. Un ejemplo del resultado en la línea de comandos

5.1.3. Implementación de Infraestructura en la Nube AWS

Como se analizó en el tema 4, la solución cambió de la versión con servidores contratados a la versión en la nube. La mayor parte del proceso se migró a la nube. La versión de la nube utiliza el EMR para procesar el programa de predicción Spark escrito en Scala. Los resultados se almacenan en S3. La base de datos dimensional de Mysql está en la instancia de la nube de RDS. Y finalmente, dos servidores implementan los servicios web y gráficos en Angularjs para presentar los resultados. La Figura 17 muestra la versión definitiva:

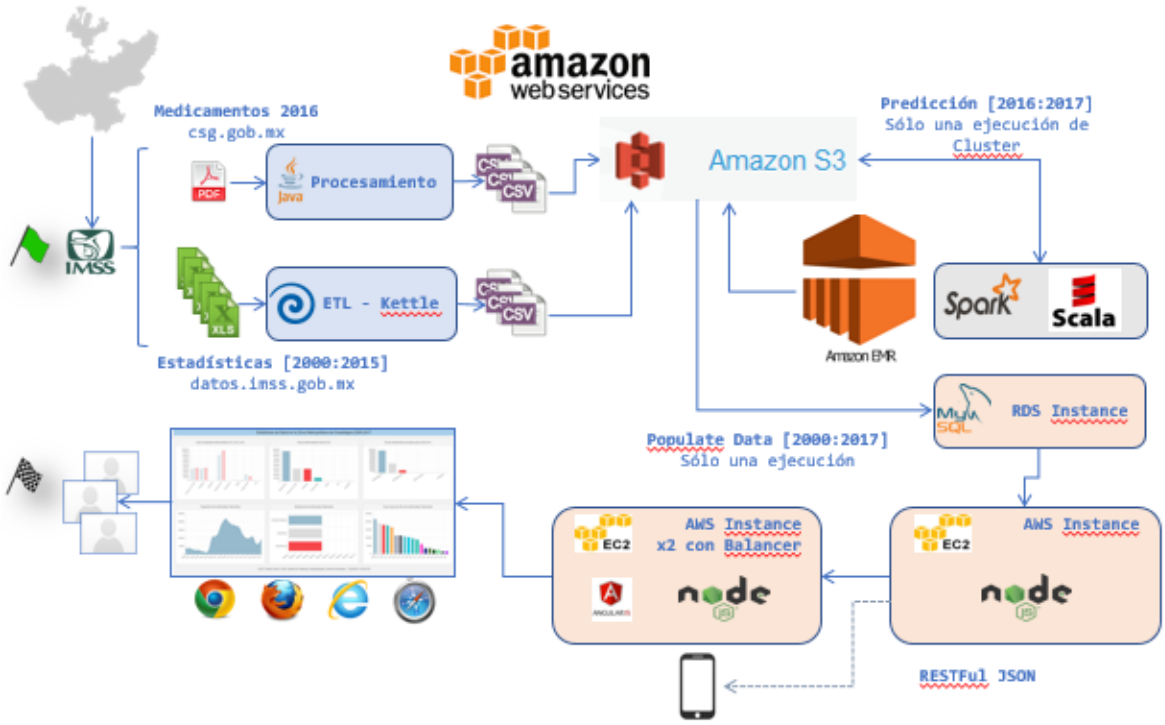


Figura 17. La infraestructura definitiva en AWS Cloud

Los elementos RDS y EC2 instancias siempre se están ejecutando, mientras que el Amazon EMR se utilizaron para una toma. Para calcular los costos de las instancias, los intereses de EC2, el equilibrador y la instancia de RDS se calcularán para un costo de la madre, mientras que el EMR se calculará como un servicio único.

A continuación, analizaremos cada servicio por separado y su función en la solución final. Amazon EMR mostrado en gris en la Figura 17, fue utilizado en una única ocasión. El resultado se almacenó en el servicio de archivos S3. Las funciones en Scala fueron cargadas en un Cluster que utilizó 24 instancias para completar su función. Esto superó la capacidad de respuesta que nos podría ofrecer la versión inicial con 4 instancias. En la Figura 18 se muestra el resultado obtenido en 11 minutos, con una capacidad aumentada de 24 procesadores. Esta sección se explica como de uso único que deberá ejecutarse 1 vez por año, que es la periodicidad con que las autoridades actualizan la información en línea, con nuevos datos de las enfermedades.

El archivo de salida almacenado en S3, sirve de entrada directa a la base de datos dimensional en MySQL desplegada en el servicio Amazon RDS. La entrada de datos fue preparada por las instrucciones en Scala y con la ayuda de los RDDs que opera la librería matemática en Spark. Para lograr esto, la instancia EMR fue creada con los servicios de MLlib y de consulta Impala. La combinación de los servicios permite manejar los archivos de entrada distribuidos con consultas de SQL. La ejemplificación de esta solución se muestra en la Figura 16.

El Resultado en Amazon EMR se observa en la Figura 18. El proceso se completó en 11 minutos, pero las horas normalizadas calculadas por el servicio de Amazon EMR son 24.

Figura 18. 24 horas normalizadas para el programa Spark

Una vez finalizada la función de Spark sobre el ecosistema EMR, el archivo depositado en S3, es consumido por la instancia RDS, para llenar la información en las tablas dimensionales. La facilidad de uso y aprendizaje sobre como consumir un servicio RDS permitió desplegar la base de datos en muy poco tiempo. Además, la instancia quedó configurada exclusivamente con la funcionalidad de MySQL, aportando eficiencia a la solución a la par de la buscada separación de funciones. Un análisis de la instancia se observa en la Figura 19.

La instancia de Amazon RDS MySQL ejecutándose en la nube se muestra en la Figura 19. La cantidad de procesamiento y almacenamiento se configuró para decenas de accesos. Para más capacidad, es posible aumentar la memoria.

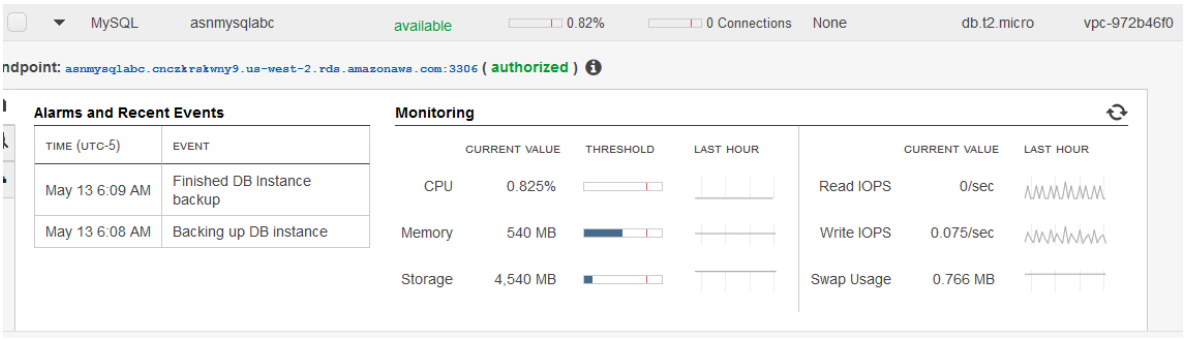


Figura 19. La instancia de RDS MySQL db.t2.micro

Las bases de datos dimensionales trabajan con tablas de dimensiones y tablas de hechos. Esto permite explotar los datos mediante consultas SQL sencillas y de ejecución ligera. La estructura dimensional incluyo 4 dimensiones básicas, Año, Estado de la estadística, Enfermedad, Medicamento y con datos de hechos como la cantidad cruzada de enfermedades contra cantidad de medicamentos necesarios.

Un ejemplo de una consulta simple es la sumatoria de las enfermedades por Año, por Enfermedad o por Estado. La combinación de esta información vista desde las diferentes aristas fue uno de los objetivos principales del desarrollo de esta solución.

En la Figura 20 se observa una consulta a la tabla de hechos. La tabla de hechos es el resultado de la acumulación de datos históricos más los proyectados para los próximos años.

```
mysql> select * from hechos  limit 1 ;
+----+------+---------+--------------+-----------------+--------+
| id | year | estado  | medicina     | enfermedad      | qty    |
+----+------+---------+--------------+-----------------+--------+
|  1 | 2000 | Jalisco | TESTOSTERONA | Cáncer de mama  | 151239 |
+----+------+---------+--------------+-----------------+--------+
1 row in set (0.19 sec)
```

Figura 20. Estructura de tabla hechos en MySQL

Ahora, una vez almacenada la información estructurada en una base de datos dimensional, se procede a la explotación de dicha información. Para esto utilizamos la instancia EC2 con NodeJs desplegado que consulta la información y la ofrece en Formato JSON a través de un servicio web. En la Figura 21 se observa el *webservice* ofreciendo los datos que fueron requeridos desde la página web. Esta función de consulta y conversión de datos en formato JSON puede ser consumida por una página web o por una aplicación en un dispositivo móvil.

```
The solution is:  {"data":[151239,149913,163406,179517,114617,166196,271393,209892,342943,303022,3
58646,363045,408694,391995,383521,364619,355879,348667],"labels":["2000","2001","2002","2003","200
4","2005","2006","2007","2008","2009","2010","2011","2012","2013","2014","2015","2016","2017"]}
The solution is:  {"data":[408694,391995,383521,364619,363045,358646,355879,348667,342943,303022,2
71393,209892,179517,166196,163406,151239,149913,114617],"labels":[2012,2013,2014,2015,2011,2010,20
16,2017,2008,2009,2006,2007,2003,2005,2002,2000,2001,2004]}
Connection to 54.186.252.37 closed by remote host.
```

Figura 21. Ejemplo de la solicitud de resultado REST Ful JSON para Instance con Node.js

Una vez que está desplegado el servicio REST, se requiere un servicio para mostrar los resultados en un tablero con toda la información. Este servicio se desplegó en otra instancia EC2 de Amazon utilizando NodeJs como base y Angular para la creación de gráficos. Cada gráfico tiene un temporizador separados que actualiza la información por separado, dando al usuario la sensación de dinamismo.

Finalmente, el resultado de la página web de la EC2 con Node.js y Angular se muestra en la Figura 22. Son 6 diferentes gráficos dentro del tablero, que muestra la información desde diferentes ángulos y dimensiones. Cada gráfico es independiente y procesa su propia información de forma asíncrona respecto a los demás, para ofrecer dinamismo a la página. En la Figura 22 se la vista completa del tablero de datos. A continuación, analizaremos cada gráfico para observar la información que despliega cada uno.

Los gráficos nos ayudan a conocer la progresión de enfermedades a través del tiempo, y la progresión de requerimientos de medicina que será necesaria para cubrir dichas enfermedades. Los datos del año 2016 y 2017 están proyectados. Los datos de 2016 son actualizados al final del año 2017 y a su vez los del 2017 son publicados al final del 2018.

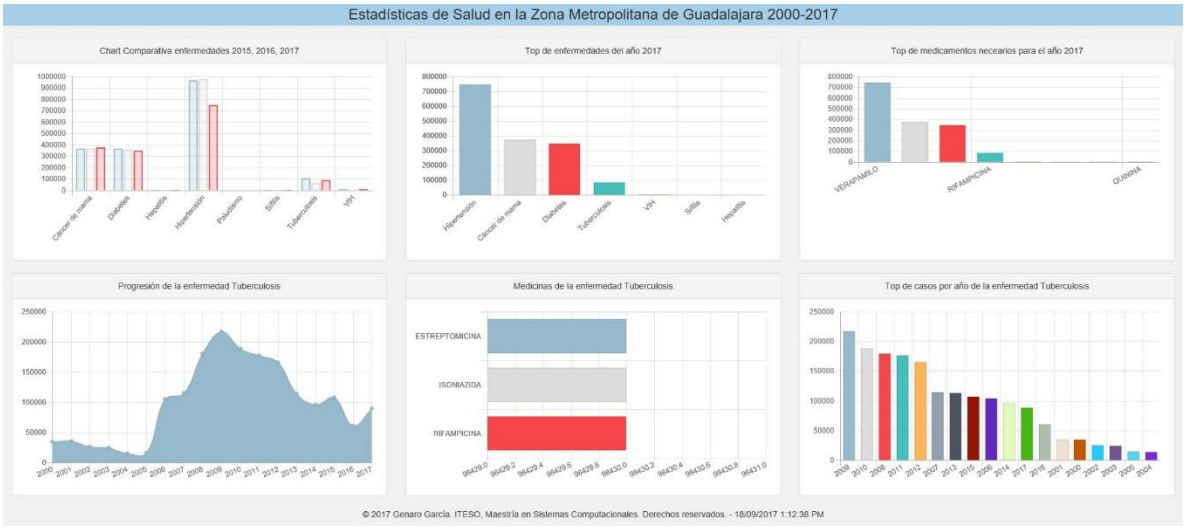


Figura 22. Resultado final de la aplicación

Analizamos por separado la información que nos ofrece por separado. Las instituciones públicas nos ofrecen la información de 8 diferentes enfermedades Cáncer de Mama, Diabetes, Hepatitis, Hipertensión, Paludismo, Sífilis, Tuberculosis y VIH. En la figura 23 se observa la progresión de la enfermedad de Cáncer de Mama.

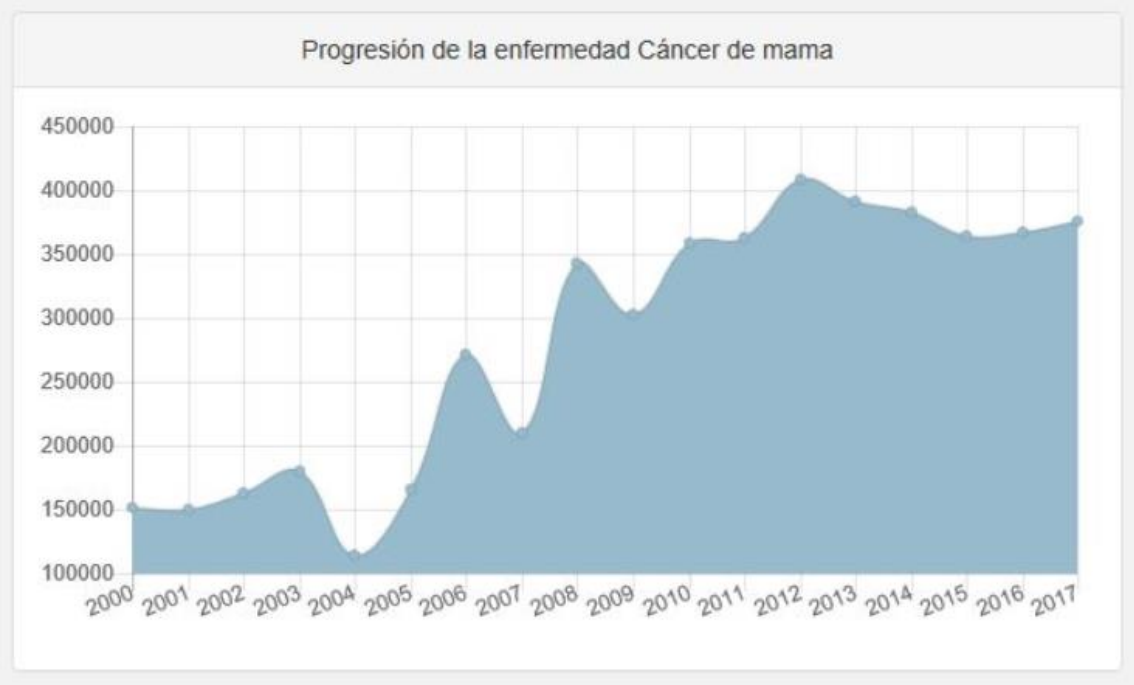


Figura 23. Progresión de la enfermedad Cáncer de Mama

En la gráfica de la Figura 23 se observa la subida de casos desde el año 2000 hasta el presente y su proyección. Junto a esa gráfica tenemos el requerimiento de medicina para el año.

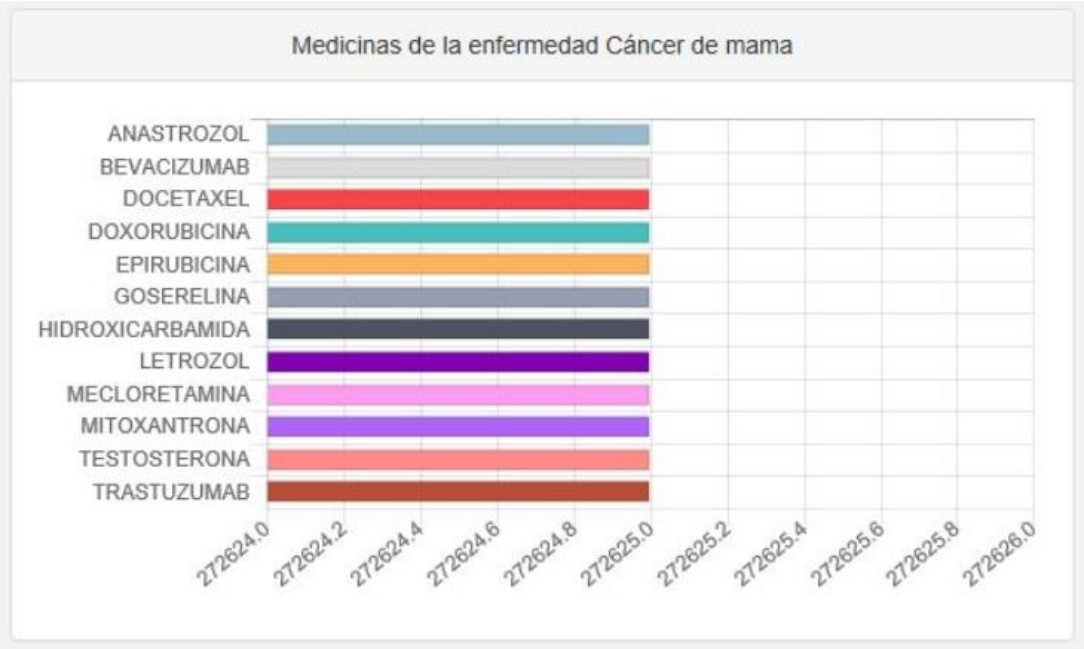


Figura 24. Medicina requerida para el Cáncer de Mama en la Zona Metropolitana de Guadalajara

Al hacer el cruce de información de la estadística de enfermedades contra el informe de medicinas controladas para las enfermedades, se obtiene la lista de medicinas para el Cáncer de Mama proyectado para el año 2017. Este listado se presenta en la Figura 24. En contra parte en la Figura 25 se muestra el top de enfermos por año incluyendo la información predicha para los años 2016 y 2017.

Este mismo tablero se actualiza cada 20 segundos para presentar la información de todas las enfermedades que el IMSS ofrece en su portal de datos abiertos.

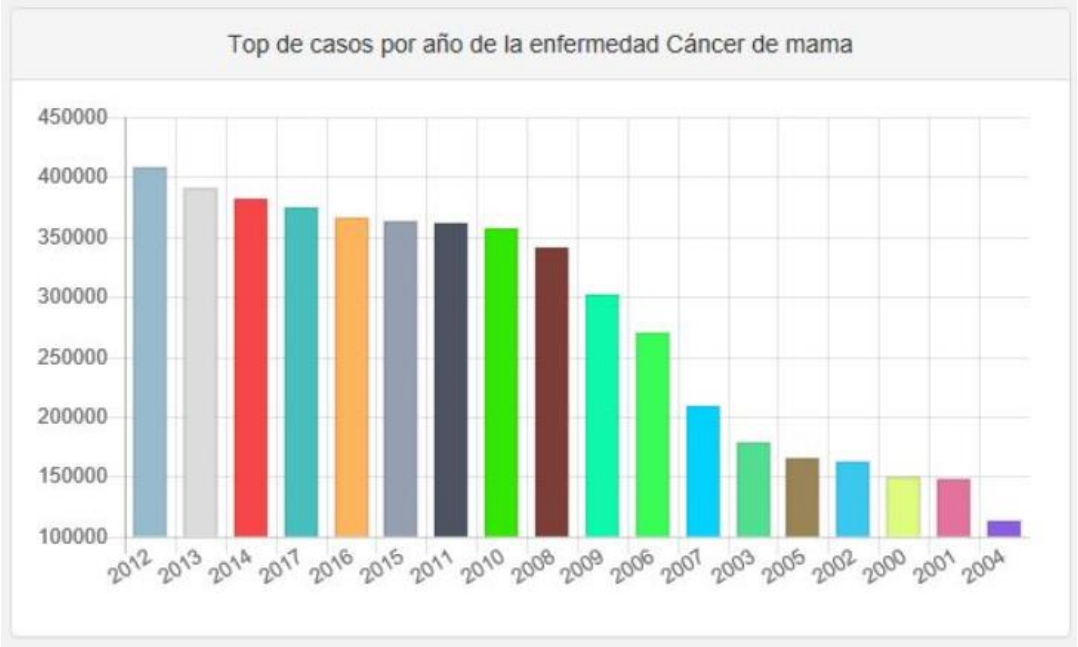


Figura 25. Registros del Cáncer de Mama en la ZMG

Las 3 gráficas superiores nos muestran una comparativa de 3 años de la progresión de la enfermedad en la ZMG. En la Figura 26 se observan las comparativas de casos y su relevancia. Mientras la enfermedad como el Cáncer de Mama va en aumento, los casos de Diabetes e Hipertensión están disparados y probablemente bajo control. Algunas enfermedades se ven de menor impacto contra otras y por tanto requerirán mayor cantidad medicamentos.

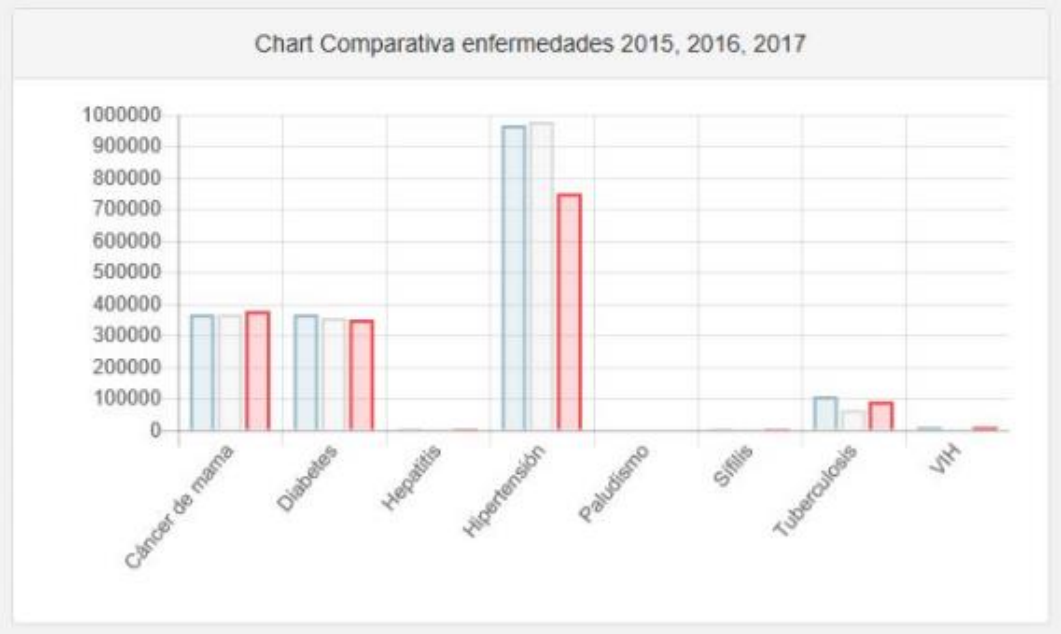


Figura 26. Comparativa de enfermedades de su avance en los últimos 3 años

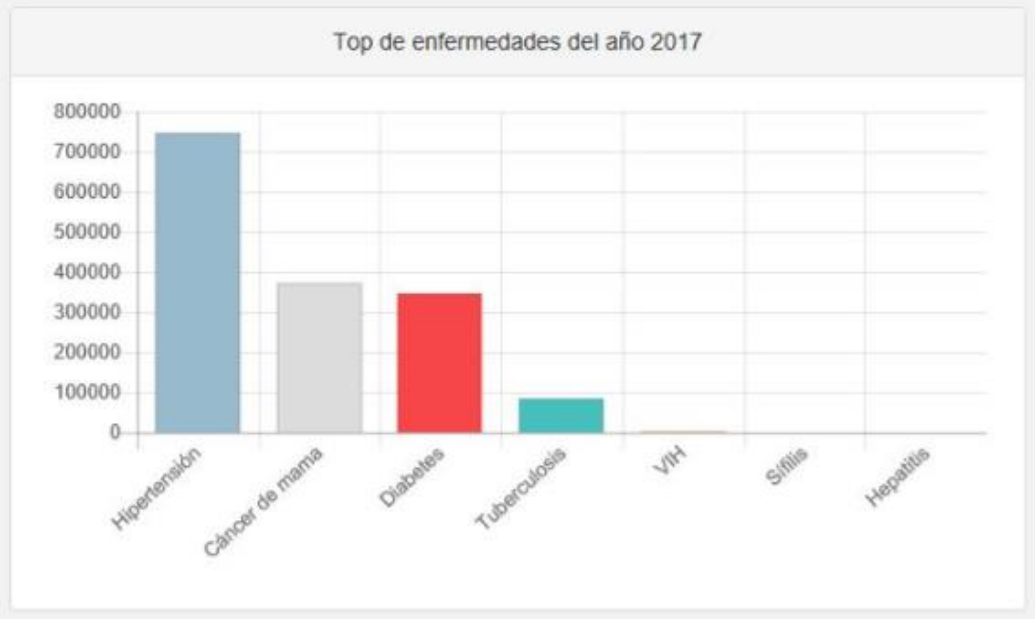


Figura 27. Comparativa de enfermedades en la ZMG proyectadas para el año 2017.

Como se muestra en la Figura 27 para el año 2017 la enfermedad de mayor impacto será la hipertensión el Cáncer de mama y la Diabetes. Esto concuerda con los esfuerzos que las instituciones de salud pública como el IMSS están realizando a nivel nacional para disminuir la cantidad de nuevos casos que se registrarán para el año 2017.

Junto con el top de enfermedades para el año 2017, se tiene el top de medicamentos requeridos para el año. Los más utilizados son los relacionados a la hipertensión y la Diabetes. Esta información se muestra en la Figura 28.

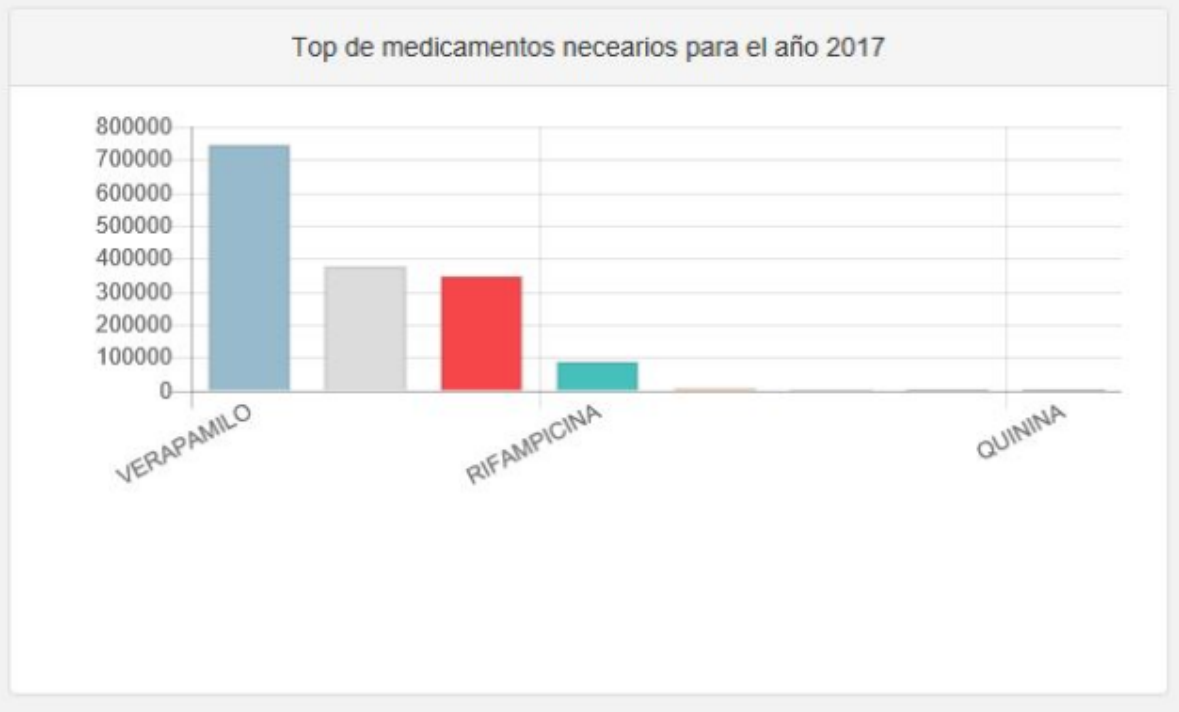


Figura 28. Top de Medicinas requeridas para las enfermedades del 2017 en la ZMG

Para finalizar con los temas de la implementación, es conveniente hacer un análisis del costo de mantenimiento necesario mensual para mantener el servicio del tablero. El costo total es de alrededor de 27 USD al mes incluido el balanceador de cargas automático. En general se considera 27 USD como costo máximo con miras a que puede ser menor si el balanceador de cargas automático no se activa. El resumen se muestra en la Tabla 2.

Costos en la nube AWS

Costo por mes 18.84 USD

**Costo por mes con balanceador 27.48 USD*

No	Servicio	Consumo Máximo	Observaciones	USD por Unidad	Total Mensual USD
1	EMR – Ejecución de Tareas en Spark	24 horas normalizadas	Al terminar la ejecución se termina las instancias.	0.064	1.540
2	S3 Almacenamiento	1 GB	Consumo máximo	0.0125	0.0125
3	Instancia db.t2.micro para mysql	24 * 30 = 720 hrs	Ejecución constante de un db.t2.micro	0.017	12.240
4	Instancia t2.micro para Nodejs y Angular. Ubuntu	24 * 30 = 720 hrs	t2.micro (x2 con Balancer)*	0.012	8.64
5	Instancia t2.micro para Nodejs. Ubuntu	24 * 30 = 720 hrs	t2.micro	0.012	8.64

Costos Adicionales no desglosados pueden incluir compra de dominio, certificado SSL y mantenimiento

Tabla 2. Tabla de costos con el proyecto implementado en AWS Services

5.2. Discusión

Los resultados fueron satisfactorios, dado que muchos archivos de entrada de diferentes fuentes y formatos convergieron para obtener un resultado final. Utilizamos algunos conocimientos de minería de textos, procesamiento de conjuntos de datos y uso de herramientas de código abierto. El objetivo principal fue utilizar diferentes herramientas para lograr predicciones. También usamos sólo datos abiertos sin restricciones de derechos.

El enfoque de minería de texto se abordó para el tipo de archivo PDF de entrada que teníamos. Para otras entradas, tal vez el enfoque necesitará algunos cambios como el archivo de formato de la salida, o tal vez la herramienta que elegimos, una clase en Java. Esta conversión se ejecutó una sola vez. El método de regresión lineal seleccionado fue suficiente para crear un modelo para predecir la cantidad de medicamento para el próximo año. Se resolvió a través de Spark MLlib, que es una poderosa herramienta que en pocas líneas de código en Scala resolvió el problema.

Como parte del proyecto completo, esta sección crea una línea básica para iniciaramos la siguiente que es almacenar los resultados en una base de datos dimensional, que inicialmente sería consumida por Pentaho, una herramienta de código abierto de Business Intelligence; al final optamos por Nodejs por su flexibilidad de uso.

Fuimos capaces de implementar la solución completa en AWS Cloud, aquí están las principales ventajas que descubrimos mediante el uso de AWS Services.

Costos: El costo final del proyecto fue de 30 USD al mes. Esto es más barato que la solución original con la arquitectura local. Anteriormente necesitábamos varios servidores para implementar los servicios.

Implementar rápidamente: Con el *framework* Node.js que es una herramienta ligera, pudimos implementar una nueva aplicación RESTFul en cuestión de horas en lugar de días o incluso semanas.

Funciones separadas: La instancia RDS nos ayudó a separar las funciones y trabajar con MySQL desde una línea de comandos desde la consola de AWS o el Node.js dentro de la instancia de EC2. A veces, cerramos las instancias de EC2 y el RDS sigue estando disponible para realizar consultas. En aplicaciones escalables, se necesita la separación de funciones como bases de datos, *back-end* y *front-end.*

Solución modular: Implementar las instancias fue Fácil y Rápido. Con algunos conocimientos de Node.js, Angular.js y MySQL implementamos la solución completa, y reemplazamos algunos *frameworks* más antiguos que son más difíciles de usar y configurar como el inicialmente propuesto Pentaho.

Ahorrar en costos en la etapa de desarrollo: Ahorramos mucho dinero desactivando las instancias cuando estábamos programando y arreglando errores en una máquina local. Al estar seguros de que funcionaban correctamente, reactivábamos de nuevos las instancias en la nube.

Replicación: La replicación de las instancias es fácil. En lugar de tener una sólida experiencia en las características de la infraestructura, con algunos conocimientos básicos de Redes y Ubuntu, implementamos la solución sin contratar los servicios de personas calificadas.

6. CONCLUSIONES

Resumen: *En esta sección se abordan las conclusiones y trabajo futuro propuesto en base a los resultados obtenidos en el desarrollo de este trabajo. Se realiza una comparación entre objetivos y resultados y las mejoras que pueden implementarse, basándose en lo que ya está implementado.*

6.1. Conclusiones

El objetivo principal fue utilizar diferentes fuentes de datos abiertos combinarlos, analizarlos y ofrecerlos de manera fácil de entender en un formato de página de internet. Decidimos enfocarnos en datos generados por dependencias de la Zona Metropolitana de Guadalajara.

Una vez que fueron localizados los archivos de entrada de diferentes fuentes abiertas de datos y formatos diferentes, convergieron para obtener un resultado final. Utilizamos algunos conocimientos de minería de textos, procesamiento y combinación de conjuntos de datos y uso de herramientas de código abierto.

Logramos predecir información mediante datos estructurados y no estructurados provenientes de distintas fuentes y formatos, todo mediante herramientas de código abierto. Utilizamos técnicas que estudiamos en el estado del arte, como lo son la predicción, minería de datos, bases de datos dimensionales, *Bigtables*, y aplicaciones de Inteligencia de negocios en proyectos de salud.

El enfoque de minería de texto se abordó para el tipo de archivo PDF de entrada que teníamos. Como lo analizamos en el marco metodológico, para otras entradas, tal vez el enfoque necesitará algunos cambios como el formato de la salida. Esta conversión se ejecutó una sola vez, para los datos encontrados en 2015. Para actualizar los datos a una versión del archivo en PDF más reciente, basta con cambiar el archivo de entrada.

El método de regresión lineal seleccionado fue suficiente para crear un modelo para predecir la cantidad de medicamento para el próximo año. Como se analizó en los resultados, resolvimos la predicción a través de Spark MLlib, que es una poderosa herramienta que en pocas líneas de código en Scala generamos la salida esperada.

Los resultados fueron satisfactorios, y dada la infraestructura clásica, la convertimos en una solución totalmente en nube. Esto nos ayudó en el tema de costos, y de uso eficiente de recursos. El uso de la nube nos ayudó a desplegar más rápidamente que de una manera tradicional, con servidores locales. La solución fue preparada para ser desplegada de manera ágil, lo cual fue una buena decisión que también mejoró el resultado final.

Conforme avanzamos en el desarrollo del prototipo fue muy importante el cambio en infraestructura y *frameworks* utilizados, considerando que constantemente aparecen nuevas herramientas más simples de utilizar y con potencial superior.

En este proyecto creamos una base de datos dimensional a partir de datos de entrada estructurados y no estructurados, estadísticos y predichos, que se consumen por Node.js, complementada con Angular como un *Business Dashboard* de Inteligencia, que sustituimos por la infraestructura inicial, donde proponíamos utilizar la herramienta de Pentaho para presentar los resultados.

6.2. Trabajo Futuro

En el presente trabajo analizamos los datos de la ZMG. Sin embargo, los datos abiertos ofrecidos por las instituciones de salud de México dan la posibilidad de analizar datos de todas las entidades del país. Una vez que completamos el modelo, lo siguiente será llenar la base de datos dimensional con la información de todos los estados de la república.

Esto significa que será necesario utilizar más servicios de nube para realizar las predicciones. Esto es, el costo de cómputo en la nube incrementa. La instancia de base de datos incrementará, y el modelo de consulta deberá ser ampliado.

Otra mejora que se puede aplicar a este proyecto es rediseñar la página de muestra de datos para incluir un selector de tipos de gráficos, en vez de mostrar todos los resultados a la vez con un temporizador que cambia los datos en un carrusel de gráficos.

Otro trabajo futuro que se propone comprenderá hacer el comparativo de los datos predichos contra la realidad estadística al final del año 2017. Para esto, tendrá que esperar la publicación de datos.

Respecto a la minería de datos que se efectuó en el archivo, esta puede ampliarse para encontrar más correlaciones entre los datos. Esto implicará introducir nuevos tokens de búsqueda para relacionar las contraindicaciones entre los diferentes medicamentos. Por ejemplo, incluir variables como contraindicaciones en el embarazo o para otras enfermedades.

BIBLIOGRAFÍA

[1] Ana Azevedo y Manuel Filipe Santos. Business Intelligence-State of the Art, Trends, and Open Issues. En KMIS. 2009. p. 296-300.

[2] Min Chen, ShiwenMao y Yunhao Liu, Big data: A survey. Mobile Networks and Applications, 2014, vol. 19, no 2, p. 171-209.

[3] Juan Andrés Alonso González y Andrea Rossi, New trends for smart cities, Open innovation mechanisms in smart cities, 2012.

[4] IEEE on Smart Cities, Guadalajara Ciudad Creativa Digita (CCD), Journal of Smart Cities, 2015, Available: http://smartcities.ieee.org

[5] Min Chen, et al. Big data applications, En Big Data, Springer International Publishing, 2014. p. 59-79

[6] Sara Landset, et al, A survey of open source tools for machine learning with big data in the Hadoop ecosystem. Journal of Big Data, 2015, vol. 2, no 1, p. 1-36.

[7] William H. Inmon: Building the data warehouse. John wiley & sons, 2005.

[8] Ralph Kimbally y Joe Caserta, The data warehouse ETL toolkit. John Wiley & Sons, 2004.

[9] William H. Inmon y Dan Linstedt, Data Architecture: A Primer for the Data Scientist: Big Data, Data Warehouse and Data Vault. Morgan Kaufmann, 2014.

[10] Ralph Kimball y Margy Ross, The data warehouse toolkit: the complete guide to dimensional modeling, Ed. John Wiley & Sons, 2011.

[11] Tom White, Hadoop: The definitive guide. O'Reilly Media, Inc., 2012

[12] Gang-Hoon Kim, Silvana Trimi y Ji-Hyong Chung, Big-data applications in the government sector. Communications of the ACM, 2014, vol. 57, no 3, p. 78-85

[13] Marcelo Beckmann, et al, A User Interface for Big Data with RapidMiner. En RapidMiner World. 2014.

[14] Sugam Sharma, An Extended Classification and Comparison of NoSQL Big Data Models. arXiv preprint arXiv:1509.08035, 2015.

[15] Apache on Hive. The Apache Hive ™ data warehouse software, 2015. Available http:// hive.apache.org

[16] Google, Inc., Big Table: A Distributed Storage System for Structured Data, 2006. Available http://static.usenix.org/events/osdi06/tech/chang/chang_html/?em_x=22

[17] Hsinchun Chen, Roger HL Chiang y Veda C Storey, Business Intelligence and Analytics: From Big Data to Big Impact. MIS quarterly, 2012, vol. 36, no 4, p. 1165-1188.

[18] Tim Menzies, James D. Kiper y Ying Hu, Machine learning for requirements engineering. Submitted to ASE-2001, 2001.

[19] Murugan Anandarajan, Asokan Anandarajan y Cadambi A. Srinivasan, Business intelligence techniques: a perspective from accounting and finance. Springer Science & Business Media, 2012.

[20] Xindong Wu, et al, Data mining with big data. IEEE transactions on knowledge and data engineering, 2014, vol. 26, no 1, p. 97-107.

[21] Jimeng Sun y Chandan K. Reddy, Big data analytics for healthcare. En Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013.

[22] Ivo D. Dinov, Methodological challenges and analytic opportunities for modeling and interpreting Big Healthcare Data. Gigascience, 2016, vol. 5, no 1.

[23] Mary K. Obenshain, Application of data mining techniques to healthcare data. Infection Control & Hospital Epidemiology, 2004, vol. 25, no 08, p. 690-695.

[24] Emad A. Mohammed, Behrouz H. Far y Christopher Naugler, Applications of the MapReduce programming framework to clinical big data analysis: current landscape and future trends. Bio data mining, 2014, vol. 7, no 1, p. 1.

[25] Consejo de Salubridad General, México, Cuadro Básico y Catálogo de Medicamentos, 2015, Available http://csg.gob.mx

[26] Instituto Mexicano del Seguro Social, Datos abierto IMMS, 2016, Datasets available http://datos.imss.gob.mx.

[27] Apache SPARK, MLlib and RDDs documentation, 2016, Available http://spark.apache.org

[28] Weiqi Wang y Eswar Krishnan, Big data and clinicians: a review on the state of the science. JMIR medical informatics, 2014, vol. 2, no 1, p. e1.

[29] Javier Andreu-Perez, et al, Big data for health. IEEE journal of biomedical and health informatics, 2015, vol. 19, no 4, p. 1193-1208.

[30] Katina Michael y Keith W. Miller, Big data: New opportunities and new challenges [guest editors' introduction]. Computer, 2013, vol. 46, no 6, p. 22-24.

[31] Bonnie Feldman, Ellen M. Martin y Tobi Skotnes, Big Data in Healthcare Hype and Hope. October 2012. Dr. Bonnie, 2012, vol. 360.

[32] Fay Chnag, et al, Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 2008, vol. 26, no 2, p. 4.

[33] Thomas H. Davenport y Jill Dyché, Big data in big companies. International Institute for Analytics, 2013.

[34] Peter Groves, et al, The 'big data'revolution in healthcare. McKinsey Quarterly, 2013, vol. 2.

[35] Jay Kreps, et al. Kafka: A distributed messaging system for log processing. En Proceedings of the NetDB. 2011. p. 1-7.

[36] Coordinación de Estrategia Digital Nacional en alianza con INFOTEC, el Gobierno municipal de Zapopan y Telefónica Movistar México, DATATON 2014: Convocatoria para analizar distintas bases de datos, y que emprendedores generen escenarios y soluciones a problemas sociales y de política pública, 2014, Available http://dataton.datos.gob.mx/.

[37] Timothy Schultz, Turning healthcare challenges into big data opportunities: A use-case review across the pharmaceutical development lifecycle. Bulletin of the American Society for Information Science and Technology, 2013, vol. 39, no 5, p. 34-40.

[38] Edtna Jauregui, et al, Using the RE-AIM framework to evaluate physical activity public health programs in México. BMC public health, 2015, vol. 15, no 1, p. 1.

[39] Hugh J. Watson, Tutorial: Big data analytics: Concepts, technologies, and applications. Communications of the Association for Information Systems, 2014, vol. 34, no 1, p. 1247-1268