

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación



**HERRAMIENTAS DE SOPORTE
ANALÍTICO DENTRO DE UN ENTORNO
INTEGRADO MONGODB**

TRABAJO FIN DE MÁSTER

Miguel Ángel López Macías

2015

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos**

TRABAJO FIN DE MÁSTER

**HERRAMIENTAS DE SOPORTE
ANALÍTICO DENTRO DE UN ENTORNO
INTEGRADO MONGODB**

Autor

Miguel Ángel López Macías

Director

Juan Carlos Dueñas López

Co-director

Hugo Alexer Parada Gélvez

Departamento de Ingeniería de Sistemas Telemáticos

2015

Resumen

Actualmente las empresas generan y administran grandes cantidades de datos, estos suelen estar estructurados, semi estructurados y no estructurados por lo que requieren de sistemas de almacenamiento acorde a sus necesidades. También resultan de vital importancia el diseño de herramientas analíticas que brinden la oportunidad de tomar decisiones de forma ágil y dinámica para poder ser competitivas en su sector y adaptarse a los cambios del contexto.

El objetivo de este trabajo son: investigar el estado actual y contexto de las bases de datos no relacionales, las tecnologías de soporte para un entorno analítico, dentro de las cuales estarán las características de cada una de ellas y el diseño de un entorno analítico, el análisis y la selección de estas tecnologías que den soporte al entorno creado para dar solución a lo que las empresas u organizaciones requieren.

En este trabajo se analiza un caso de estudio: “la obesidad infantil en el Estado de Aguascalientes, México”, en el cual se implementa el entorno creado para el análisis de datos semi estructurados a través de MongoDB y el uso de una herramienta de soporte para el análisis visual de los datos SlamData, logrando así los objetivos planteados.

El uso e implementación de MongoDB permite el almacenamiento de datos estructurados, semi estructurados y no estructurados y actualmente es utilizado por varias empresas como apoyo para el almacenamiento de este tipo de datos, es rápido y eficaz al momento de realizar consultas en grandes cantidades de datos.

Por otra parte SlamData es una herramienta que tiene la madurez suficiente, y por ahora ha cumplido con los objetivos de análisis visual, pero con la limitación que aún no permite el uso e implementación de gráficas dinámicas para comparación de datos.

Abstract

Currently the companies generate and manage large amounts of data, these are usually structured, semi-structured and unstructured therefore they require storage system to meet your needs. Are also of vital importance to design analytical tools that provide the opportunity to make decisions in a flexible and dynamic to be competitive in its sector and adapt to the changing context.

The aim of this study are: to investigate the current status and context of non-relational data bases, supporting technologies for an analytical environment, within which are the characteristics of each and designing an analytical environment, analysis and selection of these technologies that support the environment created to solve what businesses or organizations require.

In this paper a case study is analyzed, "childhood obesity in the state of Aguascalientes, Mexico", in which the environment created for the analysis of semi-structured data via MongoDB and using a support tool is implemented for visual data analysis SlamData, thus achieving the objectives.

The use and implementation of MongoDB allows storage of structured data, semi-structured and unstructured and is currently used by several companies as support for the storage of such data, it is fast and effective when querying large amounts of data .

Moreover SlamData is a tool that is mature enough, and so far has met the objectives of visual analysis, but with the limitation that still does not allow the use and implementation of dynamic graphs to compare data.

Índice general

Resumen	i
Abstract	iii
Índice general	v
Índice de figuras y tablas	vii
Siglas	ix
1. Introducción	10
1.1 Definición del problema	12
1.2 Contexto del trabajo	12
1.3 Objetivos	14
1.3.1 Objetivo general.....	14
1.3.2 Objetivos específicos	14
2. Tecnologías de soporte para un entorno analítico.....	15
2.1 Bases de datos NoSQL	15
2.1.1 BigTable.....	15
2.1.2 Cassandra.....	16
2.1.3 HBase.....	16
2.1.4 MongoDB.....	17
2.1.4.1 Características	21
2.2 Analítica y explotación visual.....	23
2.2.1 SlamData.....	23
2.2.2 Pentaho.....	25
2.2.3 R Studio.....	26
2.2.4 D3 (Data Driven Documents)	26
2.3 Despliegue del entorno.....	26
2.3.1 Puppet	27
2.3.2 Vagrant.....	27

3. Características y diseño del entorno analítico	28
3.1 Requisitos.....	28
3.2 Diseño del entorno analítico	28
3.3 Análisis y selección de las tecnologías de soporte	29
3.4 Arquitectura del entorno analítico	32
Proceso de Importación.....	34
Consulta MongoDB	35
Consulta SlamData	36
4. Caso de estudio	37
4.1 Configuración y script de despliegue.....	37
4.2 Instituto Nacional de Estadística y Geografía (INEGI)	41
4.3 Contexto del caso de estudio	41
4.4 Análisis de datos y rendimiento.....	43
Uso de variable.....	44
Uso de MongoDB.....	44
Uso de SlamData.....	45
5. Conclusiones y trabajos futuros	50
Bibliografía.....	53

Índice de figuras y tablas

Figura 1. Tipos de datos	11
Figura 2. Versiones de MongoDB	17
Tabla 3. Comparativas de Query MongoDB vs SQL [22]	18
Figura 4. Comparativa de conceptos de SQL vs MongoDB	19
Figura 5. Gráfica de Gartner [12]	20
Figura 6. Versiones de SlamData	24
Tabla 7. Comparativa de versiones de SlamData	24
Tabla 5. Ejemplos de Querys Pentaho [21]	26
Figura 9. Diseño del entorno analítico	28
Tabla 10. Comparativa de características de las bases de datos NoSQL	29
Tabla 11. Comparativa de herramientas de análisis y explotación visual.....	30
Imagen 12. Análisis SlamData.....	30
Imagen 13. Prueba Pentaho	31
Tabla 14. Análisis R Studio	32
Figura 15. Diagrama de la Arquitectura del entorno analítico.....	33
Figura 16. Diagrama del proceso de importación	34
Figura 17. Diagrama Consulta MongoDB	35
Figura 18. Diagrama Consulta de SlamData.....	36
Figura 19. Diagrama estructura de Vagrant y Puppet	37
Figura 20. Obesidad y sobrepeso en población infantil del estado de Aguascalientes [26]	42
Imagen 21. Rendimiento importación de datos MongoDB	43
Imagen 22. Uso de MongoDB.....	44

Imagen 23. Visualización de datos en MongoDB.....	45
Imagen 24. Visualización de datos en SlamData.....	46
Imagen 25. Uso de instalaciones deportivas por Municipio del estado de Aguascalientes.....	47
Imagen 26. No uso de instalaciones deportivas por Municipio del estado de Aguascalientes.....	48
Tabla 26. Rendimiento de SlamData.	49

Siglas

SQL	Structure Query Language
RDBMS	Relational DataBase Management System
NoSQL	No SQL, No Only SQL
IDC	International Data Corporation
PB	Pentabytes
JSON	JavaScript Object Notation
JSON	JavaScript Object Notation
BSON	Binary JSON
CSV	Comma Separated Values
GNU AGPL	Affero General Public Licence
P D I	Pentaho Data Integration
P B A	Pentaho Business Analytics
D3	Data Driven Documents

1. Introducción

La evolución de las tecnologías (tablets, móviles, etc.) y la web han incrementado los datos de manera exponencial [1]. Google procesa datos de cientos de petabytes (PB), mientras que Facebook genera datos de registro de más de 10 PB por mes [2], por lo tanto este crecimiento trae consigo nuevos desafíos para las empresas de diferentes sectores, como lo son el recolectar, almacenar y procesar grandes cantidades de datos que provienen de diferentes fuentes y en distintos formatos.

Para ello existen dos movimientos dentro del universo de las bases de datos: el SQL y el NoSQL, las bases de datos SQL también son llamadas bases de datos relacionales y son las que actualmente se usan en la mayor parte de las organizaciones, este tipo de bases de datos almacenan datos estructurados, sirven de gran ayuda para aplicaciones transaccionales en las que mantener y proteger la integridad es vital, pero el problema es que empiezan a ser insuficientes respecto a la forma de almacenamiento de la información de internet debido a que tiene una estructura muy rígida que impiden el crecimiento constante a la par con las fuentes de datos.

Por otro lado el movimiento NoSQL, no es exactamente un tipo de bases de datos, si no un conjunto de tipos de bases de datos que puede almacenar datos estructurados, semi estructurados y no estructurados, ejemplo de ello son las bases de datos documentales que son las mayormente usadas porque prácticamente se podría hacer todo lo actualmente con una base de datos relacional [27].

NoSQL es la combinación de dos palabras: No y SQL por lo que es una tecnología que contrarresta con SQL (Structured Query Language), NoSQL se utiliza hoy en día como un termino genérico para todas la bases de datos y almacenes de datos que no lo hacen de la forma tradicional o bajo un sistema manejador de bases de datos relacional (RDBMS), y a menudo se refieren a grandes cantidades de datos. Esto significa que NoSQL no es solo un producto o una simple tecnología, si no que representa una clase de productos y diversas colecciones, pero también relacionados con conceptos de almacenamiento y manipulación de datos.

Los desafíos para el procesamiento de grandes cantidades de datos no son específicos de un RDBMS, si no que pertenecen a todas las clases de bases de datos relacionales, por lo que una base de datos no relacional asume una estructura bien definida de los datos, estos se basan en unos requisitos en las que las propiedades de los datos se pueden definir previamente y que sus relaciones están bien establecidas, NoSQL hace frente a estos desafíos permitiendo trabajar con grandes cantidades de datos los cuales

no requieren de una estructura bien definida, permitiendo así almacenar y procesar grandes cantidades de datos.

Inicialmente Doug Laney, analista de META (actualmente Garner), define los retos y oportunidades que se presentan por el aumento de datos a través de un modelo 3V's, es decir, el aumento de *Volumen*, *Velocidad* y *Variabilidad*. [4], el **Volumen** se refiere a la recolección de grandes cantidades de datos que son generados por empresas, usuarios (dispositivos móviles), internet (web) y entre otros como los GPS, la **Velocidad** es la capacidad para hacer uso de la recolección de datos y realizar un análisis de la información que se encuentra almacenada, de tal forma que debe ser rápida y oportuna, por ultimo la **Variabilidad** indica los diferentes tipos de datos que están almacenados, estos datos pueden ser videos, fotos, textos, redes sociales, etc. El problema de tratar con diferentes tipos y formatos de dato como se muestra en la figura 1, también incluye el contexto donde los datos se generan en grandes cantidades y cambian rápidamente, por lo tanto las bases de datos deben de considerar esta situación y adaptarse a las necesidades.



Figura 1. Tipos de datos

Con la creación de nuevas aplicaciones y la generación de datos. Jim Gray propone que para hacer frente a estos retos se tienen que desarrollar nuevas herramientas informáticas para la gestión, visualización y análisis de datos [6].

Las tecnologías de bases de datos han ido evolucionando durante más de 30 años conforme las necesidades de las empresas, también se han desarrollando sistemas de bases de datos para la explotación y análisis de datos a diferentes escalas para poder dar apoyo a diversas aplicaciones. Las bases de datos SQL no pueden satisfacer las necesidades provocadas por los tipos de datos, dando lugar a las bases de datos NoSQL que cada vez son mas populares para el almacenamiento de grandes cantidades de datos.

1.1 Definición del problema

Las tecnologías están en una constante evolución, con ello trae como consecuencia la generación de nuevos sistemas que ocupan y generan grandes cantidades de datos estructurados, semi estructurados o no estructurados, los cuales requieren de un amplio espacio de almacenamiento. Este es uno de los principales problemas a resolver porque se necesita una base de datos capaz de almacenar los diferentes tipos de datos, almacenar grandes cantidades de datos, procesarlos y extraer estos datos de tal forma que nos permita realizar la explotación de esta información y poder realizar un análisis en el cual sea de importancia para una empresa en la toma de decisiones. Hoy en día ya existen bases de datos capaces de almacenar y procesar este tipo de datos como lo son BigTable, Cassandra, HBase, MongoDB y entre otros por mencionar los mas importantes.

1.2 Contexto del trabajo

Los datos no estructurados y semi estructurados no pueden ser almacenados en las bases de datos SQL tradicionales, por lo que se han implantado las bases de datos NoSQL como alternativa, sin embargo para la explotación de estos datos se necesitan herramientas adicionales para consultar, procesar y visualizar los resultados.

En base al problema descrito anteriormente la consulta y explotación de los datos aun no esta totalmente resuelto, a continuación explicare los tipos de datos relacionados con el problema.

Datos estructurados son todos aquellos que tienen definida su longitud y su formato, suelen ser: números, fechas, combinaciones de números y palabras llamadas strings (nombre de un cliente, numero de DNI, dirección postal un mail, etc.), estos datos son los que encontramos en las empresas, como por ejemplo los datos ubicados en DataWarehouses y Datamarts, datos de finanzas, ventas, almacenes, etc.

Este tipo de datos se consultan generalmente a través del lenguaje SQL (Structured Query Language), la mayoría de soluciones de Business Intelligence y Business Analytics trabajan con este tipo de datos.

Estos datos también tienen un papel importante dentro de Big Data siendo la piedra angular de las bases de datos relacionales sobre las que operan casi toda la totalidad de

los sistemas informáticos dentro de las empresas así como también en la administración.

Datos semi estructurados serían aquellos datos que no residen de bases de datos relacionales, pero presentan una organización interna que facilita su tratamiento, tales como documentos XML y datos almacenados en bases de datos NoSQL.

Datos no estructurados son aquellos datos no estructurados que no pueden ser almacenados en una base de datos tradicional (SQL). Este tipo de datos pueden ser generados por maquinas a través de imágenes satelitales, datos científicos (gráficos sísmicos, atmosféricos, etc.), fotografías, videos (cámaras de vigilancia) y datos de radares ó bien por nosotros mismos como lo son los textos incluidos dentro de los sistemas de información internos de las organizaciones (documentos, presentaciones, correo electrónicos, etc.), datos provenientes de redes sociales (Facebook, Twitter, LinkedIn, Instagram, etc.), dispositivos móviles (mensajes y aplicaciones) y contenidos de sitios web (youtube, blogs, etc).

Algunas de las características de los datos son:

- **Volumen y crecimiento:** el volumen y la tasa de crecimiento de los datos no estructurados es muy superior en comparación a los datos estructurados, Por ejemplo, twitter genera 12 Terabytes de información cada día, la tasa anual de crecimiento de datos es del 40% ó 60%, pero para los datos no estructurados en empresas, la tasa de crecimiento puede llegar al 80% (informe 2012).
- **Almacenamiento:** debido a su estructura no se pueden emplear arquitecturas relacionales, de tal forma que es necesario trabajar con bases de datos NoSQL, siendo de importancia en este tipo de arquitecturas los aspectos relacionados con la escalabilidad y paralelismo.
- **Seguridad:** Hay que considerar que algunos datos no estructurados de tipo texto, pueden no ser seguros. Por otra parte el control de accesos a los mismos es complejo debido a cuestiones de confidencialidad y la difícil clasificación de los datos [7].

Los datos semi estructurados y estructurados se consideran que son de tipo documental por lo que su explotación es mas compleja, unos de los retos es el uso de bases de datos NoSQL que permitan almacenar grandes cantidades de datos así como también de herramientas que permitan extraer dicha información almacenada y procesarla, existen sistemas de bases de datos que almacena este tipo de datos pero aun no existe algún entorno para la explotación y análisis de estos datos, por lo tanto en este trabajo propongo construir un entorno de herramientas alrededor de MongoDB

con el objetivo de consultar, extraer, procesar y explotar la información de manera visual. Pretendo definir un conjunto de características o requisitos que me permitan establecer una selección de las herramientas para construir el entorno propuesto y con esta selección propondré un diseño del conjunto integrado que dará soporte a la explotación de los datos.

1.3 Objetivos

1.3.1 Objetivo general

Analizar y seleccionar las herramientas disponibles en el ámbito del código abierto para construir un entorno de almacenamiento, proceso y explotación de datos estructurados, semi estructurados y no estructurados en el ecosistema MongoDB.

1.3.2 Objetivos específicos

- *Investigar el estado actual y contexto de las bases de datos NoSQL.*

Investigar la tendencia actual de las bases de datos NoSQL, como ha sido su evolución y cual es el problema que actualmente se tiene para poder dar una solución.

- *Indagar sobre las diferentes tecnologías de soporte para un entorno analítico.*

Investigar cuales son las tecnologías relacionadas y elaborar un pequeño estado de arte pero enfocándome en el uso de MongoDB.

- *Descubrir las diferentes características y efectuar el diseño del entorno analítico*

Realizar una lista con los requisitos que el entorno MongoDB debe cumplir para dar una solución al problema y así mismo la elaboración y diseño del entorno

- *Análisis y selección de las tecnologías de soporte.*

Investigar, seleccionar y realizar un resumen de las herramientas a implementar en el entorno MongoDB, justificando su utilidad en la solución del problema.

- *Arquitectura el entorno analítico.*

Realizar la arquitectura de las herramientas en base a los requisitos del diseño del entorno analítico para la solución.

- *Realizar el despliegue del entorno analítico*

Las herramientas y el entorno se desplegara de manera automática con el uso de Puppet y Vagrant.

- *Caso de uso.*

Tras la realización y la creación del entorno analítico se implementa una base de datos con grandes cantidades de datos para el análisis visual de los datos y obtención del rendimiento de las herramientas.

- *Elaboración de la memoria.*

Escribir una memoria que se empezara a realizar desde los primeros días del proyecto.

2. Tecnologías de soporte para un entorno analítico

2.1 Bases de datos NoSQL

Las bases de datos NoSQL son un conjunto de bases de datos que no se ajustan al modelo de bases de datos relacionales y sus características, estas no usan esquemas y resuelven el problema de las grandes cantidades de datos. Por ejemplo algunas bases de datos no estructuradas son: BigTable, Cassandra, HBase y MongoDB, de las cuales se pueden explotar con herramientas como SlamData y Pentaho.

2.1.1 BigTable

BigTable es precursora de las bases de datos Hbase y Cassandra, fue creado por Google y se empezó a desarrollar a principios de 2004, es un sistema de almacenamiento de datos que esta distribuida y estructurada, esta diseñada para procesar los datos a gran escala (PB) entre miles de servidores comerciales. [8] La estructura básica de datos en BigTable es multidimensional, distribuida y persistente, se construye en la parte superior de las tecnologías como Google File System y SSTable.

BigTable es usada por el propio buscador, Google Maps, Google Earth, Google Finance, Blogger, etc. De esta manera, la cantidad de información almacenada es enorme y del orden de Petabytes. Aunque tiene algún parecido con los sistemas tradicionales relacionales de bases de datos, rompe alguna de sus principales premisas. Un ejemplo es la organización de las propias tablas. Estas se dividen en conjuntos de columnas y éstos en otras columnas. Es posible añadir columnas en cualquier momento y no es posible borrar las filas.

2.1.2 Cassandra

Es un sistema de almacenamiento columnar distribuido de código abierto altamente escalable para manejar grandes cantidades de datos estructurados distribuidos entre varios servidores comerciales [9]. El sistema fue desarrollado por Facebook y se convirtió en una herramienta de código abierto en 2008. Es un gestor de datos no relacional y no SQL, utiliza un estilo denominado por BigTable y esta diseñada para guardar campos-valores, lo que permite dinamismo en el diseño. Nos encontramos ya con sitios web como Facebook o Twitter que han hecho el cambio de bases de datos relacionales (típicamente MySQL) a Cassandra de forma exitosa. Apache Cassandra está disponible bajo la licencia de Software Apache v2.0 y que es supervisada por un comité de gestión de proyecto (PMC), que orienta sus operaciones diarias, incluyendo versiones de desarrollo y producto de la comunidad, permite la importación de datos desde un archivo origen en formato CSV y esta desarrollado en Java.

2.1.3 HBase

HBase comenzó como un proyecto por la empresa Powerset por la necesidad de procesar grandes cantidades de datos a efectos de búsqueda de lenguaje natural. Ahora es un proyecto de Apache y ha generado un interés considerable, es de código abierto, y esta basado en las bases de datos distribuidas dentro del modelo de Google BigTable y esta escrito en Java, es tolerante a fallos de almacenar grandes cantidades de datos dispersos [10]. No es un reemplazo directo para bases de datos SQL, aunque últimamente su rendimiento ha mejorado y ahora está sirviendo en varios sitios web impulsada por datos, incluyendo la mensajería en la plataforma de Facebook.

2.1.4 MongoDB

MongoDB es una base de datos no relacional (NoSQL) de tipo documental en formato BSON que está diseñado para tener un almacenamiento y velocidad más eficiente [28]. Permite a las empresas ser más ágiles y escalables, son ya muchas las organizaciones que están usando MongoDB para crear nuevos tipos de aplicaciones, mejorar la experiencia del cliente, acelerar el tiempo de comercialización y reducir costes. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, ofrece fiabilidad a nivel empresarial y flexibilidad operativa, también permite almacenar grandes cantidades de datos estructurados, semi estructurados y no estructurados, este tipo de bases de datos documental acepta la importación de grandes cantidades de datos que se encuentran en un archivo fuente con formatos JSON y CSV, tiene una licencia pública que es GNU AGPL 3.0 de modo que se puede descargar gratuitamente desde su sitio web [11], también ofrece escalabilidad y además de ello tiene un gran rendimiento en la escritura y lectura de los datos.

MongoDB es una base de datos actualmente usada el mercado empresarial que integra alta escalabilidad y permite la escritura y lectura de datos con mayor rapidez, a continuación presento una línea del tiempo donde presento las últimas 3 versiones de MongoDB, como se muestra en la Figura 2.

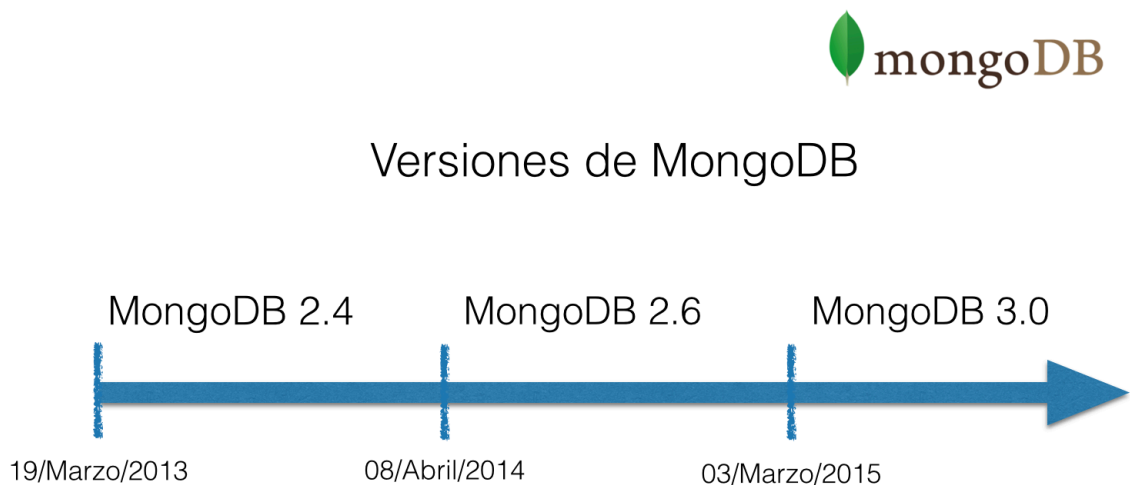


Figura 2. Versiones de MongoDB

Las características más destacadas son su velocidad y su sencillo pero potente sistema de consulta de datos, por lo que es un sistema de bases de datos que tiene rendimiento

y funcionalidad en el que se puede realizar casi todas las consultas que se usan en sistema relacional pero sin sacrificar el rendimiento [11].

Las consultas en MongoDB se realizan a través de la terminal de Mongo la cual realiza una petición y esta genera una respuesta en formato JSON, la sintaxis de las funciones son muy parecidas a las de SQL por lo que a continuación presento algunos ejemplos y comparativas de las consultas entre MongoDB y el lenguaje SQL que se muestran en la Tabla 3.

Comparativa Query MongoDB vs SQL		
Función	MongoDB	SQL
Insert	db.users.insert ({name: "sue", age: 26, status: "A" })	INSERT INTO users (name, age, status) VALUES ("sue", 26, "A")
Update	db.users.update ({ age: { \$gt: 18 } }, { \$set: { status: "A" } }, { multi: true })	UPDATE users SET status = "A" WHERE age > 18
Remove	db.users.remove ({ status: "D" })	DELETE FROM users WHERE status = "D"
Select	db.users.find ({ age: { \$gt: 18 } }, { name: 1, address: 1 }).limit(5)	SELECT _id, name, address FROM users WHERE age > 18 LIMIT 5

Tabla 3. Comparativas de Query MongoDB vs SQL [22]

Las ventajas que ofrece MongoDB son:

- **Gratuito y multiplataforma.** Esta disponible para todas las plataformas.
- **Rápida y funcional.** Normalmente se tiene que sacrificar rendimiento por funcionalidad o viceversa, incluso usar dos sistemas (RDBMS + Cache) redundando los datos, MongoDB alcanza el equilibrio entre rendimiento y funcionalidad.
- **Fácil de probar.** Se puede levantar una instancia en cuestión de minutos, solo se descargan los ejecutables, descomprime, se crean los directorios y se ejecuta la instancia. MongoDB contiene drivers mantenidos para los lenguajes como: C, C#, Java, Java Script, .NET, PHP, Ruby.
- **Fácil de entender.** En comparación con las bases de datos relacionales se utilizan los siguientes conceptos como se muestra en la Fig. 3 en donde de una manera sencilla se observan las diferencias de los conceptos en SQL y en MongoDB.

- **Escalabilidad, Replicación y Alta disponibilidad.**
- **Formación** (MongoDB University)
- **Soporte comercial**

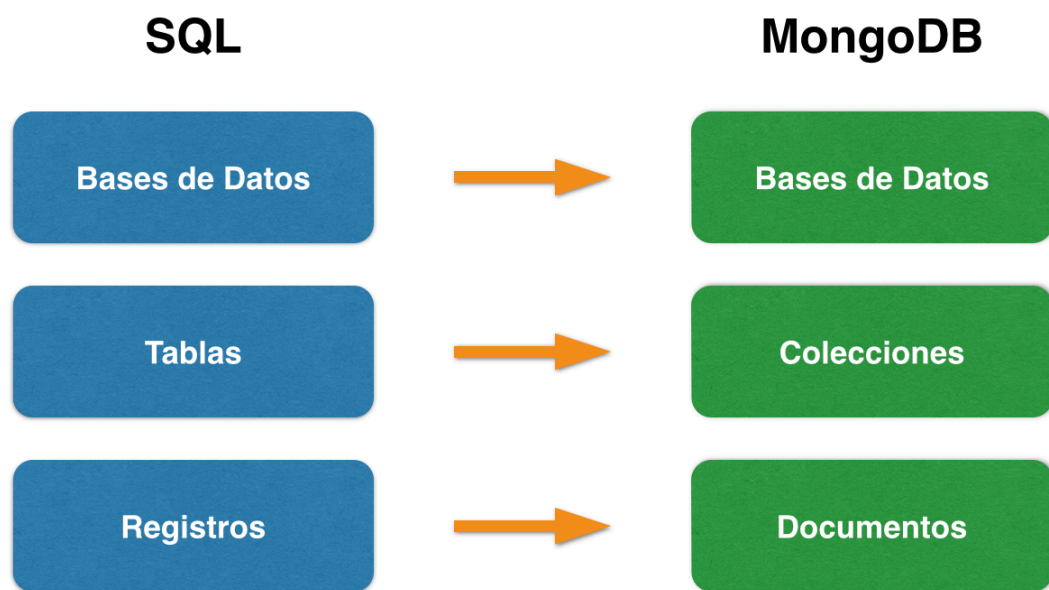


Figura 4. Comparativa de conceptos de SQL vs MongoDB

Como se observa en la figura 4, en MongoDB los documentos se agrupan en colecciones, por lo tanto se puede decir que los documentos son similares a los registros en una base de datos relacional. Las colecciones son parecidas a las tablas relacionales pero con la diferencia que no imponen una estructura fija a los documentos que contienen, ni siquiera al tipo de datos y longitud de cada campo. Entonces se puede decir que en MongoDB la estructura de los datos (bases de datos, colecciones, documentos, campos...), es implícita, flexible y dinámica. Implícita porque lo normal es que los documentos de una colección compartan estructura, y ésta no se declara de manera explícita antes de introducir los documentos. Flexible porque esa estructura no se impone a ningún documento y dinámica porque la estructura puede cambiar.

Por otro lado en el cuadrante de Gartner, MongoDB se posiciona como retador en los sistemas de gestión de bases de datos operacionales, entre ellas se incluyen las bases de datos relacionales y las NoSQL como se muestra en la Figura 5.

En el cuadrante de retadores o aspirantes se encuentran los proveedores bien posicionados y que ofrecen altas posibilidades de éxito a la hora de implementar una solución , que suelen centrarse en un aspecto único de los que demanda el mercado.



Figura 5. Gráfica de Gartner [12]

Este grafico fue publicado por Gartner, Inc. Como parte de un documento de investigación [12].

MongoDB ha logrado posicionarse en el cuadro de Gartner debido a:

- Da respuesta a la necesidad de almacenamiento de todo tipo de datos: estructurados, semi estructurados y no estructurados.
- Tiene un gran rendimiento en cuanto a escalabilidad y procesado de la información.
- Puede procesar la cantidad de información que se genera hoy en día.
- Se adapta a las necesidades actuales (millones de usuarios, miles de peticiones por segundo).
- Permite a las empresas ser mas ágiles y crecer más rápidamente y crear nuevos tipos de aplicaciones.
- Esta orientada a documentos lo que quiere decir que en un único documento es capaz de almacenar toda la información necesaria que define un producto, un cliente, etc., esto sin tener que seguir un esquema predefinido.
- Permite adaptar el esquema de la base de datos a las necesidades de la aplicación rápidamente, disminuyendo el tiempo y el coste de la puesta en producción de la misma.

2.1.4.1 Características

Flexibilidad. MongoDB almacena los datos en documentos JSON y estos ofrecen un modelo de datos que se asigna a la perfección a los tipos de lenguaje de programación nativa y el esquema dinámico que hace que sea más fácil para evolucionar su modelo de datos a comparación con un sistema de esquemas forzados como lo son los manejadores de bases de datos relacionales (RDBMS).

Potencia. MongoDB ofrece muchas de las características de un RDBMS tradicionales como índices secundarios, consultas dinámicas, clasificación, actualizaciones (actualización si el documento existe, inserta si no existe), y fácil agregación.

Velocidad. Al mantener los datos relacionados entre sí en los documentos hace que las consultas puedan ser mucho más rápidas que en una base de datos relacional. MongoDB también hace que sea fácil ampliar las base de datos, también es posible

aumentar la capacidad sin ningún tiempo de inactividad, lo cual es muy importante en la web cuando la carga puede aumentar repentinamente y derribar el sitio web.

Facilidad de uso. MongoDB es muy fácil de instalar, configurar, mantener y utilizar. Con este fin, MongoDB ofrece algunas opciones de configuración, y en su lugar trata de hacer automáticamente "lo correcto" siempre que sea posible. Esto significa que MongoDB funciona solamente instalándolo y se puede entrar directamente en el desarrollo de su aplicación, en lugar de gastar un montón de tiempo de ajuste configuraciones de bases de datos oscura.

Orientada a documentos. En lugar de tomar un tema de negocios y dividirlo en múltiples estructuras relacionales, MongoDB puede almacenar el tema de negocios en el número mínimo de documentos. Por ejemplo, en lugar de almacenar información de título y autor en dos estructuras distintas relacionales, título, autor, y otra información relacionada con el título, todos pueden ser almacenados en un solo documento denominado libro, que es mucho más intuitiva y por lo general más fáciles de trabajar.

Consultas ad hoc. MongoDB soporta la búsqueda por campo, consultas de rango, búsquedas de expresiones regulares y también se incluyen las funciones de JavaScript definidas por el usuario. Las consultas pueden devolver los campos específicos de documentos.

Replicación. MongoDB proporciona alta disponibilidad con conjuntos de réplicas. [14] Un conjunto de réplicas se compone de dos o más copias de los datos. Cada miembro del conjunto de réplicas puede actuar en el papel de la réplica primaria o secundaria en cualquier momento. La réplica primaria realiza todas las escrituras y lee por defecto. Réplicas secundarias mantienen una copia de los datos en el primario utilizando una función de replicación. Cuando una réplica principal falla, el conjunto de réplicas automáticamente lleva a cabo un proceso de elección para determinar qué secundaria debería convertirse en el principal. Los secundarios también pueden realizar operaciones de lectura, pero los datos son consistentes con el tiempo de forma predeterminada.

El equilibrio de carga. MongoDB puede ejecutar en varios servidores, equilibrando la carga y / o la duplicación de datos para mantener el sistema en funcionamiento en caso de fallo de hardware. La configuración automática es fácil de implementar, y las nuevas máquinas se puede agregar a una base de datos en ejecución.

Almacenamiento de archivos. MongoDB puede ser utilizado como un sistema de archivos , aprovechando el equilibrio de carga y las características de duplicación de datos a través de múltiples máquinas para almacenar archivos.

En un sistema de MongoDB multi-máquina, los archivos se pueden distribuir y copiar varias veces entre las máquinas de forma transparente, por lo tanto creando efectivamente un sistema de equilibrio de carga y tolerancia a fallos.

Consultas. MongoDB ofrece una amplia gama de opciones de consultoría para ayudar a los clientes con cualquier escenario de desarrollo, desde la creación de nuevas aplicaciones, hasta la migración a MongoDB, para escalar despliegues ya existentes, y mucho más [13].

2.2 Analítica y explotación visual

Analítica se emplea a los esfuerzos en la explotación de datos de diversas fuentes para ayudar a las empresas a ser más eficaces y a evaluar las acciones pasadas para estimar el potencial de las acciones futuras, con las cuales tomar mejores decisiones y adoptar estrategias más eficaces. La explotación visual tiene como objetivo apoyar el razonamiento analítico a través de interfaces visuales interactivas.

2.2.1 SlamData

SlamData es una herramienta OpenSource que tiene soporte para MongoDB en su versión más reciente 3.0.4. la cual permite realizar consultas en lenguaje SQL sobre la base de datos no relacional para la explotación de los datos, esta herramienta es de gran utilidad dado que ofrece todas las posibilidades para la obtención de la información a través de consultas en lenguaje SQL [14].

Características:

- 100% ejecución en la base de datos para cada consulta, incluidos los que tienen uniones. Los datos no se transmiten siempre de nuevo al cliente, por lo que SlamData puede manejar tantos datos como el clúster de MongoDB pueda manejar.
- Apoyo a todas las cláusulas SQL importantes, así como numerosos operadores y funciones.
- Contiene un planificador de optimización que elige el método de mayor rendimiento cuando realiza la ejecución de una consulta en MongoDB.

- Al aprovechar SQL, SlamData hace posible una amplia gama de usuarios y herramientas para interactuar con MongoDB, y ayuda de forma rápida y fácil a comprender los datos generados por las aplicaciones.

En su ultima versión SlamData integra el análisis visual de los datos generando gráficas a través de consultas o Querys en lenguaje SQL. A continuación presento una línea del tiempo donde muestro la constante actualización de la herramienta.



Versiones de SlamData

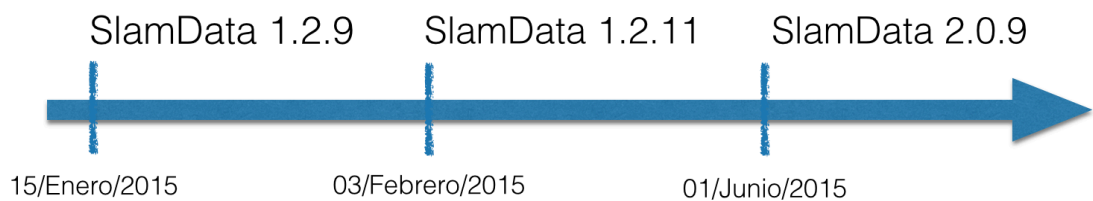


Figura 6. Versiones de SlamData

Algunas de las características que han tenido cada versión se muestran en la siguiente tabla:

Comparativa versiones SlamData					
Características	Interfaz HTLM	SQL	MongoDB	OpenSource	Gráficas
1.2.9	X	✓	✓	✓	X
1.2.11	X	✓	✓	✓	X
2.0.9	✓	✓	✓	✓	✓

Tabla 7. Comparativa de versiones de SlamData.

2.2.2 Pentaho

Pentaho es una herramienta de entorno visual, tiene dos herramientas de utilidad que son Pentaho Data Integration (P D I) y Pentaho Business Analytics (P B A), ambas herramientas permiten el análisis de los datos de forma visual a través de diferentes tipos de gráficas.

P D I permite realizar una conexión directa con MongoDB y aplicar Querys de consultas para la extracción de datos tal como se muestra en la Tabla. 5, está herramienta está mas relacionado con la elaboración de informes que puede ser exportados en formato PDF.

P B A es una herramienta moderna, simple e interactiva que permite al usuario la explotación de datos masivos, siendo así útil en el análisis y la visualización de los datos para la toma de decisiones. Esto se facilita porque incorpora herramientas de análisis visual interactivo y dashboards gráficos para mejorar el desempeño organizacional, además de que cuenta con soluciones integrales para informes de los cuales se generan en múltiples formatos como HTML, Excel, CSV, PDF y RTF, cuenta con una amplia conectividad a cualquier fuente de datos con soporte nativo para Hadoop, NoSQL, y Bases de datos analíticos [15].

Ejemplos de Querys Pentaho	
Query ejemplos	Descripción
<code>{name: "MongoDB"}</code>	Consulta todos los valores en el campo "nombre" que tengan un valor igual a "MongoDB"
<code>{ name : { '\$regex' : "m.*", '\$options' : "i" } }</code>	Utiliza una expresión regular para encontrar campos de nombre comenzando con m, mayúsculas y minúsculas
<code>{ name : { '\$gt' : "M" } }</code>	Busca en todas las cadenas de mayor que M
<code>{ name : { '\$lte' : "T" } }</code>	Busca en todas las cadenas de menor o igual a T
<code>{ name : { '\$in' : ["MongoDB", "MySQL"] } }</code>	Encuentra todos los nombres que son MongoDB o MySQL
<code>{ name : { '\$nin' : ["MongoDB", "MySQL"] } }</code>	Encuentra todos los nombres que no son MongoDB o MySQL, o cuando no se ha establecido el campo
<code>{ created_at : { \$gte : { \$date : "2014-12-31T00:00:00.000Z" } } }</code>	Buscará todos los documentos created_at que son mayores o igual a la fecha especificada

<code>{ \$where : "this.count == 1" }</code>	Uso de JavaScript para evaluar una condición
<code>{ \$query: {}, \$orderby: { age : -1 } }</code>	Devuelve todos los documentos de la colección ordenados por el campo de la edad en orden descendente.

Tabla 8. Ejemplos de Querys Pentaho [21]

2.2.3 R Studio

R es una herramienta OpenSource que puede integrarse con distintas bases de datos, entre ellas se puede realizar una conexión directa con MongoDB a través de las librería RMongo para la extracción de datos a partir de Querys realizadas en R.

Posee una capacidad grafica que permite realizar gráficos con alta calidad además proporciona una amplia variedad de herramientas estadísticas por ejemplo: modelos lineales y no lineales, análisis de series temporales, algoritmos de clasificación y agrupamiento, entre otras, además esta herramienta esta disponible para cualquier sistema operativo [16].

2.2.4 D3 (Data Driven Documents)

Es una herramienta visual de código abierto que integra una biblioteca de JavaScript para la manipulación de documentos basados en datos, funciona a través de los estándares web como: HTML5, SVG y CSS, es rápido y permite comportamientos dinámicos de interacción y animación. Permite la importación de datos en un formato JSON y CSV [18].

2.3 Despliegue del entorno

Despliegue es la acción de distribuir entornos o herramientas en base a instrucciones para su procesamiento, las ventajas de automatizar el despliegue es que solo con un enter o un click se realice el despliegue de lo que hemos realizado en base a las instrucciones que se tienen y en un momento se cree lo que se requiere. Por lo tanto las herramientas a usar para el despliegue del entorno analítico son Puppet y Vagrant a continuación se muestra una breve reseña de las características de cada una de ellas.

2.3.1 Puppet

Es una herramienta desarrollada por Puppet labs para administrar la configuración de sistemas Unix y Windows de forma declarativa, de tal forma que no le decimos a la maquina lo que tiene que ejecutar, si no que cual es el estado que queremos que se encuentre.

Alguna característica de Puppet es que permite realizar la configuración de forma abstracta, especificando el estado en el que queremos que se encuentre la maquina y no las ordenes que tiene que ejecutar, esto permite instalar paquetes independientemente del gestor de paquetes que tenga el sistema.

Algunos componentes y características los explico a continuación:

Puppet esta desarrollado en Ruby y actualmente existen dos versiones:

- Puppet OpenSource.
- Puppet Enterprise.

Puppet Open Source está bajo la licencia de Apache 2.0 y Puppet Enterprise es la solución de pago, aunque se puede probar el producto con un limite de 10 nodos.

Se puede ejecutar Puppet en todas las principales plataformas como Linux y Unix, Mac OS X y Windows y está disponible para su descarga en la pagina oficial de Puppet [19].

2.3.2 Vagrant

Es una herramienta OpenSource que se utiliza para la creación y configuración de entornos virtualizados, es compatible con cualquier sistema operativo, teniendo como principal características que ayuda a gestionar entornos de desarrollo de manera independiente [20].

3. Características y diseño del entorno analítico

3.1 Requisitos

A continuación realizo un listado de los requisitos importantes que se requieren para la creación del entorno integrado de herramientas:

- Debe de ser capaz de importar grandes cantidades de datos a partir de un archivo origen con diferentes formatos, por ejemplo JSON, CSV, entre otros.
- Contendrá un manejador de base de datos no relacional.
- El manejador de bases de datos no relacional debe de ser capaz de almacenar datos estructurados, semi estructurados y no estructurados.
- Se requieren de herramientas capaces de consultar los datos almacenados en la base de datos no relacional.
- Debe de ser capaz de exportar la información consultada.
- Debe permitir generar gráficas para el análisis visual.

3.2 Diseño del entorno analítico

En base a los requisitos y características del entorno analítico, se concluye con el diseño del entorno como se muestra en la Figura 9.

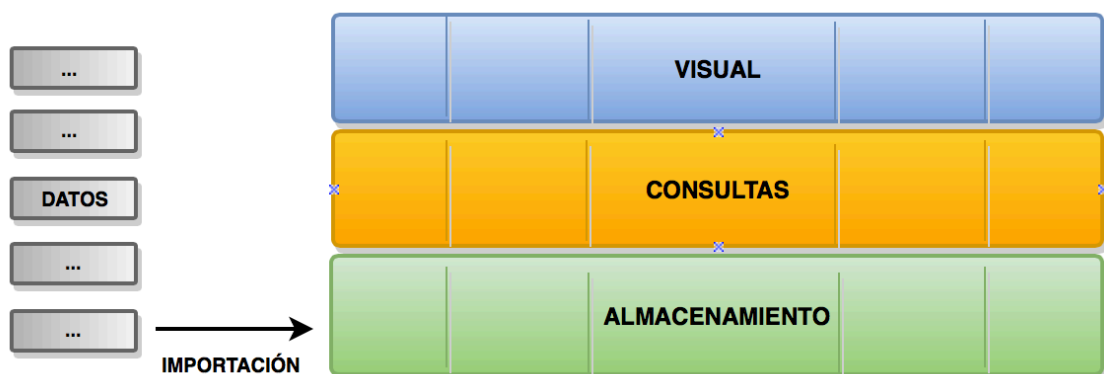


Figura 9. Diseño del entorno analítico

3.3 Análisis y selección de las tecnologías de soporte

Para el análisis y la selección de estas herramientas realice unas tablas comparativas en relación a las investigaciones realizadas de las bases de datos no relacionales (NoSQL) y las herramientas de análisis y explotación visual. La Tabla 10 contiene las comparaciones de las bases de datos NoSQL y Tabla 11 contiene las comparaciones de cada una de las herramientas de explotación visual.

Tabla comparativa de bases de datos NoSQL			
Características	MongoDB	HBase	Cassandra
Tipo	Documental	Columnar	Columnar
Formato	BSON	HTTP	CQL3
Licencia	AGPL	Apache	Apache
Carga masiva	MongoImport (CSV, JSON)	Sqoop (HDFS)	CSV
Escrito en	C++	Java	Java

Tabla 10. Comparativa de características de las bases de datos NoSQL

He seleccionado la base de datos no relacional y documental MongoDB porque es capaz de importar, almacenar y procesar grandes cantidades de datos estructurados, semi estructurados y no estructurados, la importación es a partir de un formato CSV y JSON, MongoDB posee una licencia gratuita y es de formato BSON (representación binaria) por lo que esta diseñado para tener velocidad más eficiente al momento de realizar consultas, MongoDB ahora es la tecnología que esta de moda y es usada por grandes empresas como Bosch, Cisco, MetLife y entre otras [23].

Las demás bases de datos no las he seleccionado porque aunque cumplen los requisitos del entorno no tienen un formato binario BSON, por lo cual serian un poco mas lentas en la consulta sobre grandes cantidades de datos.

Ahora después de la selección de la base de datos NoSQL, realizare un análisis y selección de las herramientas que se puedan implementar en el entorno analítico con MongoDB.

Tabla comparativa de herramientas					
Características	SlamData	P.D.I.	P.B.A.	R Studio	D3
Interprete de SQL	✓	X	X	X	X
OpenSource	✓	X	X	✓	✓
Multiplataforma	✓	✓	✓	✓	✓
MongoDB	✓	✓	✓	✓	X

Tabla 11. Comparativa de herramientas de análisis y explotación visual

He seleccionado SlamData porque como se observa en la Tabla 8 cumple con todas las características y objetivos que se requieren. Permitiendo así el análisis de grandes cantidades de datos y la visualización en gráficas a partir de una consulta en lenguaje SQL, es una herramienta OpenSource por lo que es gratuita para su uso y esta disponible para todas las plataformas.

En la Imagen 12 se puede observar que SlamData es capaz de analizar y mostrar datos semi estructurados o no estructurados como un tabla estructurada, por lo que esta herramienta cumple con los requisitos, de tal forma que la he elegido para formar parte del entorno analítico con MongoDB.

The screenshot shows the SlamData interface with a search bar at the top containing "/CEMABE/CEMABE/CENTROS". Below the search bar, there is a table with the following columns: CENTROSO, P159, P158, P157, P156, P155, P154, P153, P152, P151, P150, P149, P148H, P148G, P148F, P148E, P148D, P148C, P148B, P148A, P4K, P4J, P4I, P4H, P4G, P4F. The table contains several rows of data, including addresses and names, such as "A 500 METROS APROXIMADAMENTE AL NORESTE DE LA CALLE GERONIMO DE LA CUEVA" and "CELIA MARIA MARTINEZ".

Imagen 12. Análisis SlamData

Pentaho es una herramienta muy potente que permite la interacción con gráficas dinámicas, pero estas versiones de Pentaho son de pago por lo tanto es una limitante, para ello he probado la versión Pentaho Community Edition que es la versión gratuita, pero esta herramienta no incluye todas las utilidades que se tienen en la versión de pago, por lo que no cumple con los requisitos del entorno analítico como se muestra en la Imagen 13, la cual no permite agregar más de 100,000 entradas, por lo tanto no la he seleccionado para formar parte de este entorno.

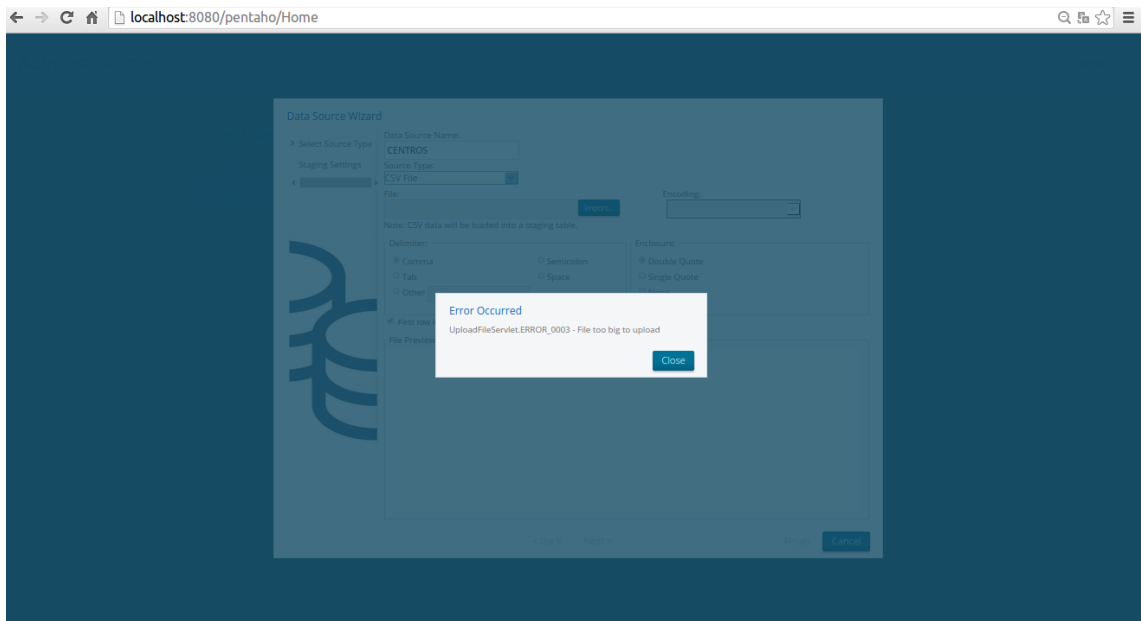


Imagen 13. Prueba Pentaho

R Studio cuenta con dos librerías las cuales realizan la conexión a MongoDB, y son: rmongodb y RMongo, cada una de ellas contiene sus paquetes para funcionar con R Studio.

R Studio es una herramienta muy potente para el análisis de datos, pero no cumple con los objetivos debido a que no supero las pruebas de extracción de grandes cantidades de datos con la conexión MongoDB a través del uso de las librerías rmongodb y RMongo. En base a las pruebas realizadas para el análisis de esta herramienta obtuve los resultados que se muestran en la Tabla 14.

Análisis R Studio		
Numero de registros (filas)	Numero de variables (columnas)	Resultado de carga de datos
1,000	267	satisfactorio
10,000	267	satisfactorio
50,000	267	satisfactorio
69,750	267	satisfactorio
69,772	267	satisfactorio
69,773	267	error

Tabla 14. Análisis R Studio

Las librerías RMongo y rmongodb solo permiten el uso de 69,772 registros por lo que no es viable para grandes cantidades de datos.

D3 (Data Driven Documents) es una herramienta muy útil porque permite interacciones dinámicas con los datos, permite la importación de datos en JSON y CSV, pero debido a que no tiene una conexión directa con MongoDB la he descartado para formar parte del entorno analítico con MongoDB.

3.4 Arquitectura del entorno analítico

En base a la selección de la base de datos NoSQL y las herramientas de análisis y visualización, incorporo la arquitectura del entorno analítico como se muestra en la Figura 15.

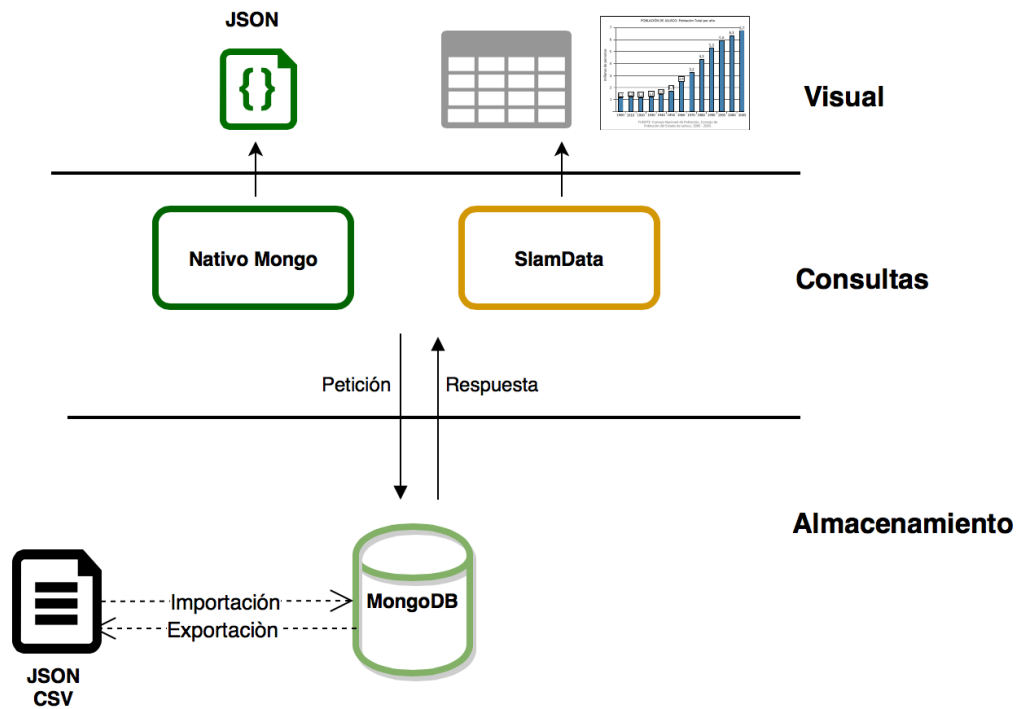


Figura 15. Diagrama de la Arquitectura del entorno analítico

En este diagrama de la arquitectura de herramientas el entorno está compuesto por tres niveles los cuales son:

Almacenamiento. En este nivel se encuentra la base de datos MongoDB que permite la importación y exportación de grandes cantidades de datos a partir de un archivo en formato JSON y CSV que son capaces de almacenarse en MongoDB.

Consultas. Este nivel esta representado por las herramientas para la explotación de los datos.

Visual. Este nivel esta representado por la visualización de los datos en formato JSON como lo es en el caso de la herramienta nativa de MongoDB y también las gráficas generadas por la herramienta SlamData.

Proceso de Importación

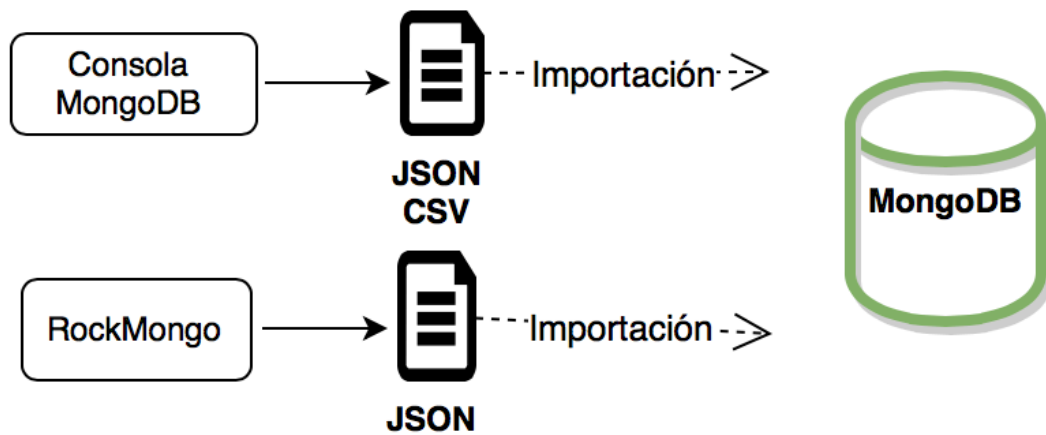


Figura 16. Diagrama del proceso de importación

El proceso de importación se puede realizar de dos formas:

- Herramienta nativa mongo en la cual se hace uso de la siguiente instrucción para la importación de datos a partir de un archivo en formato JSON.

```
mongoimport --db (NombreBD) --collection (NomColl) -type(csv json) -  
headerline --file (.csv, .json)
```

NombreBD. Indica el nombre de la base de datos a seleccionar.

NombreColl. Indica el nombre de la colección dentro de la base de datos seleccionada para la importación de datos.

Archivo json o csv. Indica el nombre del archivo que contiene los datos a importar, en este caso el archivo puede ser en formato json y csv.

- El uso de la herramienta "RockMongo" que es una interfaz web similar a PHPMyAdmin basada en PHP para la administración de MongoDB la cual permite crear bases de datos, colecciones y la importación de datos a través de un archivo JSON sin incluir la importación de CSV, todo esto a través de la interfaz grafica.

Ambas herramientas permiten la importación de datos la única diferencia es que la primera es a través de líneas de comando ejecutándose en la terminal y la segunda es en un entorno grafico.

Consulta MongoDB

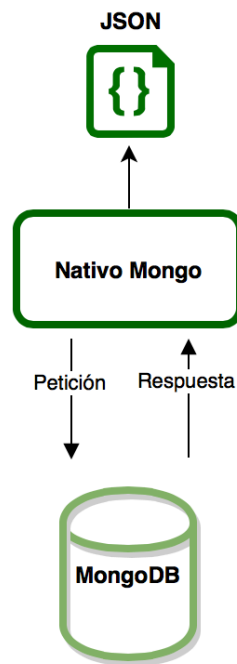


Figura 17. Diagrama Consulta MongoDB

MongoDB también hace uso de su propio lenguaje para las consultas y explotación de los datos. En el apartado 2.1.4 en la Tabla 3 se muestran unos ejemplos de consultas sobre MongoDB.

La sintaxis varía un poco pero en realidad la ventaja de usar MongoDB es la velocidad con la que esta accede a una cantidad muy grande de datos y los procesa.

Consulta SlamData

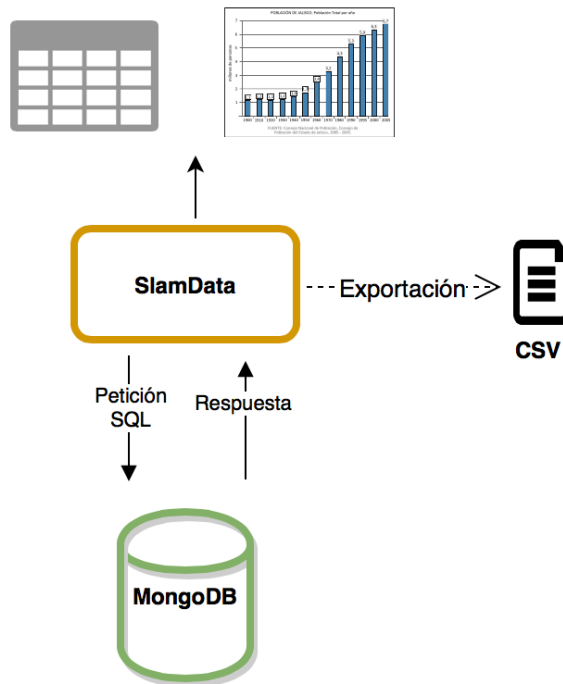


Figura 18. Diagrama Consulta de SlamData

SlamData realiza consultas a través del lenguaje SQL, la cual realiza una petición a la base de datos MongoDB y esta da una respuesta, teniendo como resultado la vista de una tabla estructurada. Posteriormente se pueden realizar consultas específicas para la obtención de gráficas, SlamData también brinda la posibilidad de exportar los datos extraídos de la consulta en un archivo CSV.

4. Caso de estudio

El objetivo de este caso de estudio consiste en desplegar el entorno analítico, implementar y utilizar el entorno creado para el análisis de datos estructurados, semi estructurados y no estructurados a través de MongoDB y el uso de la herramienta de soporte SlamData para el análisis visual de los datos. Debido a que ha sido difícil conseguir una base de datos no estructurada, hago uso de una base de datos estructurada que contiene grandes cantidades de datos que se encuentran en un data set en formato CSV, esto con la finalidad de comprobar que el entorno analítico creado satisface las necesidades de procesar, analizar y explotar los datos de manera visual.

En la primera parte muestro el uso y las configuraciones de Vagrant y Puppet. Posterior a eso se muestra como funciona el ecosistema MongoDB y la herramienta de análisis visual SlamData.

4.1 Configuración y script de despliegue

Vagrant requiere de una estructura específica de ficheros para su funcionamiento, a continuación simbolizo en la Figura 19 la estructura de los directorios para Vagrant y la configuración de Puppet son los siguientes:

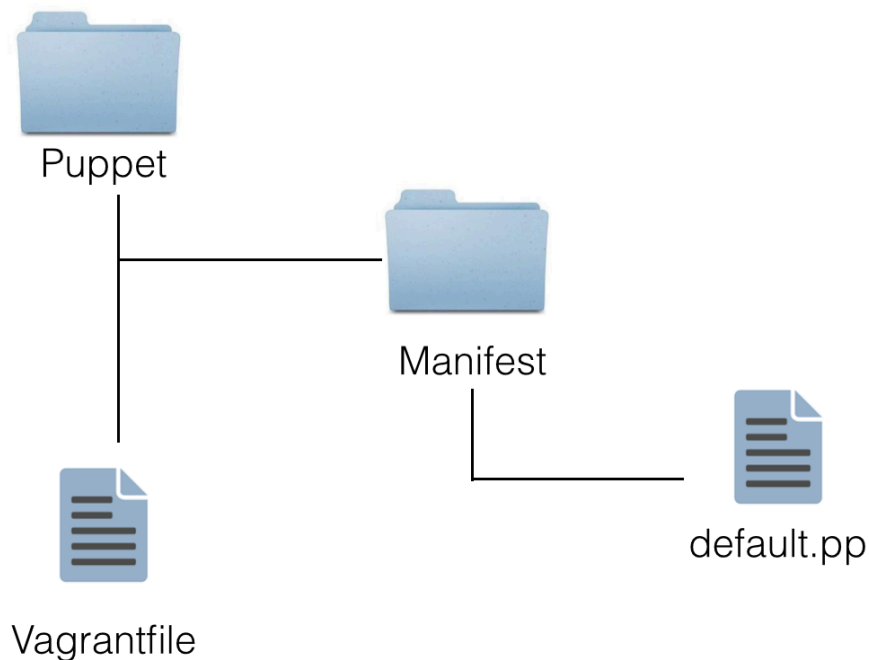


Figura 19. Diagrama estructura de Vagrant y Puppet

El archivo Vagrantfile es un fichero en donde se encuentra el script de configuración de la maquina virtual a crear, en el cual se le puede indicar que sistema operativo tendrá, la dirección IP, así como también se puede definir la localización de los aprovisionamientos que en este caso es Puppet.

El script del fichero Vagrantfile es el siguiente:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure('2') do |config|
  # Sistema operativo
  config.vm.box = 'ubuntu/trusty64'
  # IP privada
  config.vm.network 'private_network', ip: '10.0.0.100'
  # Configuración de VirtualBox
  config.vm.provider :virtualbox do |vb|
    # Nombre de la máquina virtual
    vb.name = 'entornomongo'
    # Memoria
    vb.memory = 2048
    # Número de procesadores
    vb.cpus = 2
  end
  # Habilitar aprovisionamiento por Puppet
  config.vm.provision :puppet do |puppet|
    # Localización de los ficheros de configuración
    # manifiestos
    puppet.manifests_path = 'puppet/manifests'
    # Nombre del manifiesto que se va a ejecutar inicialmente
    puppet.manifest_file = 'default.pp'
    # Opciones de Puppet. Se activa el modo debug y verbose
    puppet.options = [
      '--verbose',
      '--debug',
    ]
  end
  # Compartir un directorio de nuestro host con la VM
  config.vm.synced_folder "~/home/miguelangel/slamdata2.0.9",
  "/vagrant_shared"
end
end
```

El archivo `default.pp` es un fichero en el que se contiene el script donde se establecen los programas o herramientas con las cuales se desplegará la maquina virtual en Vagrant.

El script del fichero `default.pp` es la siguiente:

```
# Actualizar los repositorios de paquetes
exec { "apt-get update":
  command => "/usr/bin/apt-get update"
}

# Instalación de Apache
package { "apache2":
  ensure => present,
  require => Exec["apt-get update"]
}

# Instalacion de MongoDB

$server_ksbdistcodename = downcase($::ksbdistcodename)

apt::source { 'mongodb-org-3.0':
  location      => 'http://repo.mongodb.org/apt/debian',
  release       => "${server_ksbdistcodename}/mongodb-
org/3.0",
  repos        => 'main',
  key           => '7F0CEB10',
  key_server    => 'keyserver.ubuntu.com',
  include_src  => false
}

class { 'mongodb::globals':
  manage_package_repo => false,
  server_package_name => 'mongodb-org',
  service_name        => 'mongod',
  version              => '3.0.3',
}->

class { '::mongodb::server': }
```

```

# Arrancar el servicio de MongoDB
service { "mongod":
    ensure => running,
    require => Package["mongodb-org"]
}

# Arrancar el servicio de Apache
service { "apache2":
    ensure => running,
    require => Package["apache2"]
}

# Lista de paquetes de PHP para instalar
$packages = [
    "php5",
    "php5-cli",
    "php5-mysql",
    "php5-dev",
    "php5-curl",
    "php-apc",
    "libapache2-mod-php5"
]

# Instalación de los paquetes de PHP
package { $packages:
    ensure => present,
    require => Exec["apt-get update"],
    notify => Service["apache2"]
}

```

- **Exec.** Utilizado para la ejecución de comandos.
- **Package.** Utilizado para la instalación de paquetes. Este comando internamente utilizará el gestor de paquetes del sistema operativo que tengamos instalado (apt-get, yum, pacman...).

- **Service.** El recurso utilizado con todo lo relacionado con los servicios

Después de haber terminado la creación de los directorios y scripts de los ficheros se ejecuta el siguiente comando para encender la maquina.

```
$ vagrant up
```

4.2 Instituto Nacional de Estadística y Geografía (INEGI)

El INEGI es una empresa del Gobierno Federal Mexicano la cual genera estadística obtenida de tres tipos de fuentes: censos, encuestas y registros administrativos, así como estadística derivada, mediante la cual produce indicadores demográficos, sociales y económicos, además de contabilidad nacional [24].

La obtención de datos la realice a través del contacto realizado con el INEGI, se me envió una copia de una base de datos en un data set con formato CSV. La base de datos contiene información de los Censos de Escuelas, Maestros y Alumnos de Educación Básica y Especial (CEMABE) de los Estados Unidos Mexicanos. Dicha base de datos tiene su última actualización con fecha de 2 de Marzo del 2015.

La base de datos surge de la exportación de una base de datos relacional como lo es SQL, el data set es heterogéneo pero no todos los registros aportan información, esto debido a que Mongo DB no requiere de tener una estructura definida de tal forma que no es necesario tener creada la estructura como los nombres de los campos, tipo y longitud, por tal motivo MongoDB permite dicha importación.

4.3 Contexto del caso de estudio

Actualmente, México ocupa el primer lugar mundial en obesidad infantil, y el segundo en obesidad en adultos, precedido sólo por los Estados Unidos. El problema está presente no sólo en la infancia y la adolescencia, sino también en la población con edad preescolar. La Secretaría de Salud informó que México, líder mundial en obesidad infantil, registró de enero a noviembre pasado 35 mil 157 nuevos casos de obesidad entre niños de 1 a 14 años y 15 mil 626 nuevos casos entre jóvenes de 15 a 19 años.

Aguascalientes cerrará el año con similares índices de obesidad y sobre peso que el resto del país, A poco menos de una semana de que concluya el 2014, una de las principales problemáticas de salud en la entidad, referente a la obesidad y el sobrepeso tanto en adultos como en menores, cerrará con índices similares a los del resto del país,

ubicando cerca de la media nacional, con un 70 por ciento de población adulta que sufre de dichos padecimientos, así como el 30 por ciento de la población infantil. Lo dio a conocer Alexander Luévano Contreras, director de Atención Primaria a la Salud del Instituto de Servicios de Salud del Estado de Aguascalientes (ISSEA), quien además de dar a conocer el estado en que cierra el padecimiento para 2014, señaló que pese a que en los niños exista una menor preocupación, es necesario que se atienda, principalmente en lo relativo a la prevención, debido a que pueden sumarse a futuro al preocupante porcentaje de adultos con obesidad o sobrepeso [25].

En relación a la última Encuesta Nacional de Salud y Nutrición (ENSANUT) 2012 de la entidad federativa Aguascalientes, tiene resultados los siguientes resultados como se muestra en la Figura 20 donde se presenta prevalencias de sobrepeso y obesidad, y la suma de estas dos condiciones para el ámbito estatal, por tipo de localidad (urbana y rural) y por sexo. En 2012 las prevalencias de sobrepeso y obesidad fueron 23.2 y 11.6%, respectivamente (suma de sobrepeso y obesidad, 34.7%) a nivel estatal [26].

Categoría	Condición	ENSANUT 2012			
		Muestra n	Expansión		
			N (miles)	%	IC95%
Estatal	Sobrepeso	114	43.2	23.2	19.1-27.9
	Obesidad	65	21.5	11.6	8.9-14.9
	SP+O	179	64.7	34.7	30.3-39.4
Sexo	Masculino				
	Sobrepeso	63	23.5	25.1	20.8-30.0
	Obesidad	34	10.7	11.5	7.7-16.8
	SP+O	97	34.3	36.6	30.9-42.7
	Femenino				
	Sobrepeso	51	19.6	21.2	15.6-28.2
	Obesidad	31	10.8	11.7	7.5-17.7
Localidad	Urbana				
	Sobrepeso	85	35.1	25.6	20.3-31.7
	Obesidad	39	16.2	11.8	8.5-16.1
	SP+O	124	51.3	37.4	31.7-43.4
	Rural				
	Sobrepeso	29	8.0	16.4	10.7-24.4
	Obesidad	26	5.3	10.8	7.1-16.2
SP+O	55	13.4	27.3	21.5-34.0	

SP+O: sumatoria de la prevalencia de sobrepeso más obesidad
IC= Intervalo de confianza

Figura 20. Obesidad y sobrepeso en población infantil del estado de Aguascalientes [26]

4.4 Análisis de datos y rendimiento

En relación con el contexto del caso de estudio descrito anteriormente realizaré un caso de estudio haciendo uso de los datos que se encuentran almacenados en la base de datos CEMABE ya mencionada anteriormente, la base de datos contiene 177 mil 829 registros (filas) que representan el total de centros de educación básica en el territorio Mexicano, cada registro contiene 267 variables (columnas), esta base de datos esta en un data set con formato CSV donde se encuentra el Id de la escuela, nombre de la escuela, dirección, estado, municipio, turno, modalidad, nivel y entre otras variables, de las cuales me enfocaré en la variable "P192" que representa el uso de las instalaciones deportivas por escuela, "NOM_ENT" que representa el estado y "NOM_MUN" que representa el municipio al que pertenece.

La importación de la base de datos en formato CSV con un tamaño de fichero de 112.4 MB hacia MongoDB se ha realizado satisfactoriamente a través de la siguiente forma:

```
mongoimport --db CEMABEDOS --collection CENTROSODOS --type csv --headerline --file /TR_CENTROS.CSV
```

En este proceso solo se requiere previamente tener creada la base de datos y la colección, por lo que MongoDB permite la importación de datos semi estructurados como es el caso, sin la necesidad de tener creada una estructura.

El rendimiento de este proceso ha tardado 1 min con 9 segundos, analizando la Imagen 21 se puede observar que MongoDB realiza la importación en bloques de 3 segundos hasta llegar al 100 % de la importación de la base de datos.

```
miguelangel@miguelangel-Satellite-L635:~$ mongoimport --db CEMABEDOS --collection CENTROSODOS --type csv --headerline --file /home/miguelangel/Escre
torio/CEMABE/TR_CENTROS.csv
2015-07-10T12:00:03.781+0200      connected to: localhost
2015-07-10T12:00:06.755+0200      [#####] CEMABEDOS.CENTROSODOS 6.3 MB/107.2 MB (5.9%)
2015-07-10T12:00:09.755+0200      [#####] CEMABEDOS.CENTROSODOS 12.2 MB/107.2 MB (11.3%)
2015-07-10T12:00:12.755+0200      [#####] CEMABEDOS.CENTROSODOS 18.6 MB/107.2 MB (17.3%)
2015-07-10T12:00:15.756+0200      [#####] CEMABEDOS.CENTROSODOS 22.0 MB/107.2 MB (20.5%)
2015-07-10T12:00:18.756+0200      [#####] CEMABEDOS.CENTROSODOS 24.7 MB/107.2 MB (23.0%)
2015-07-10T12:00:21.755+0200      [#####] CEMABEDOS.CENTROSODOS 30.1 MB/107.2 MB (28.1%)
2015-07-10T12:00:24.755+0200      [#####] CEMABEDOS.CENTROSODOS 30.4 MB/107.2 MB (28.4%)
2015-07-10T12:00:27.755+0200      [#####] CEMABEDOS.CENTROSODOS 36.3 MB/107.2 MB (33.9%)
2015-07-10T12:00:30.755+0200      [#####] CEMABEDOS.CENTROSODOS 42.3 MB/107.2 MB (39.5%)
2015-07-10T12:00:33.755+0200      [#####] CEMABEDOS.CENTROSODOS 44.1 MB/107.2 MB (41.1%)
2015-07-10T12:00:36.755+0200      [#####] CEMABEDOS.CENTROSODOS 48.6 MB/107.2 MB (45.4%)
2015-07-10T12:00:39.755+0200      [#####] CEMABEDOS.CENTROSODOS 54.9 MB/107.2 MB (51.2%)
2015-07-10T12:00:42.755+0200      [#####] CEMABEDOS.CENTROSODOS 60.9 MB/107.2 MB (56.8%)
2015-07-10T12:00:45.755+0200      [#####] CEMABEDOS.CENTROSODOS 66.8 MB/107.2 MB (62.3%)
2015-07-10T12:00:48.755+0200      [#####] CEMABEDOS.CENTROSODOS 70.8 MB/107.2 MB (66.0%)
2015-07-10T12:00:51.758+0200      [#####] CEMABEDOS.CENTROSODOS 75.1 MB/107.2 MB (70.0%)
2015-07-10T12:00:54.755+0200      [#####] CEMABEDOS.CENTROSODOS 78.8 MB/107.2 MB (73.5%)
2015-07-10T12:00:57.755+0200      [#####] CEMABEDOS.CENTROSODOS 84.7 MB/107.2 MB (79.0%)
2015-07-10T12:01:00.755+0200      [#####] CEMABEDOS.CENTROSODOS 90.7 MB/107.2 MB (84.6%)
2015-07-10T12:01:03.755+0200      [#####] CEMABEDOS.CENTROSODOS 92.9 MB/107.2 MB (86.6%)
2015-07-10T12:01:06.757+0200      [#####] CEMABEDOS.CENTROSODOS 96.9 MB/107.2 MB (90.4%)
2015-07-10T12:01:09.755+0200      [#####] CEMABEDOS.CENTROSODOS 102.8 MB/107.2 MB (95.9%)
2015-07-10T12:01:12.755+0200      [#####] CEMABEDOS.CENTROSODOS 107.2 MB/107.2 MB (100.0%)
2015-07-10T12:01:12.771+0200      imported 177829 documents
miguelangel@miguelangel-Satellite-L635:~$
```

Imagen 21. Rendimiento importación de datos MongoDB

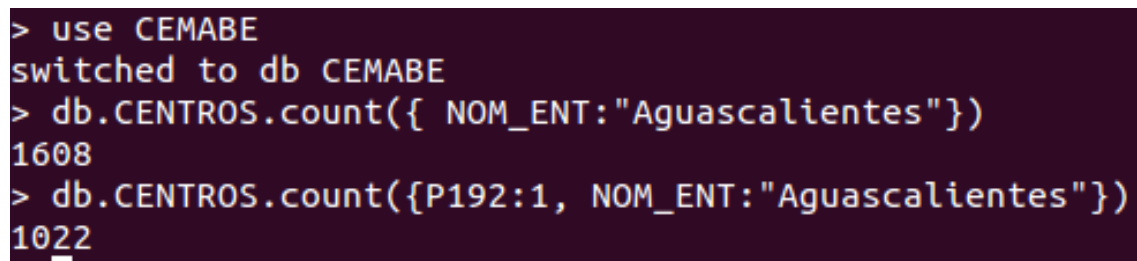
Uso de variable

La variable a emplear en base al diccionario de la base de datos será "P192" que es la variable que contiene la información sobre el uso de instalaciones deportivas.

```
Deportes <-- "P192"  
#Hace uso de las instalaciones deportivas  
"P192": 1  
  
#No hace uso de instalaciones deportivas  
"P192": 2  
"P192": 9  
"P192": (null) <-- campo vacío o inexistente
```

Uso de MongoDB

A través de la consulta desde la consola de MongoDB se puede obtener la cantidad de escuelas en Aguascalientes que hace uso de las instalaciones deportivas.



```
> use CEMABE  
switched to db CEMABE  
> db.CENTROS.count({ NOM_ENT:"Aguascalientes"})  
1608  
> db.CENTROS.count({P192:1, NOM_ENT:"Aguascalientes"})  
1022
```

Imagen 22. Uso de MongoDB

Donde:

```
#Cuenta el total de las escuelas del estado de Aguascalientes  
db.CENTROS.count({NOM_ENT: "Aguascalientes"})  
  
#Cuenta las escuelas que hacen uso de instalaciones deportivas  
db.CENTROS.count({P192:1, NOM_ENT: "Aguascalientes"})
```

Como resultados de la imagen 22 se concluye lo siguiente:

- El total de escuelas en el estado de Aguascalientes es de 1,608.
- El total de las escuelas que hacen uso de las instalaciones deportivas son 1,022. Como se muestra en la Imagen 25.
- El resto de las escuelas que son 586 no hacen uso de las instalaciones deportivas. Como se muestra en la Imagen 26.

Uso de SlamData

El uso de SlamData permite realizar consultas a través del lenguaje SQL, de tal forma que muestra los resultados de la consulta en una tabla estructurada haciendo mas fácil su entendimiento, como se muestra en la imagen 23 donde los datos consultados en MongoDB no tienen una estructura resultado complicado su entendimiento y en comparación con el uso de SlamData donde los datos semi estructurados se muestran en una tabla relacional como se muestra en la Imagen 24.

```
{ "_id" : ObjectId("55849a2ecba0b67c90585f43"), "ID_INM" : 513, "CLAVE_CT" : "01DPR0647K1", "ENT" : 1, "NOM_ENT" : "Aguascalientes", "MUN" : 1, "NOM_MUN" : "Aguascalientes", "LOC" : 1, "NOM_LOC" : "Aguascalientes", "AGEB" : 2691, "MZA" : 15, "NOMBRECT" : "MARIANO JIMENEZ", "NIVEL" : 3, "MODALIDAD" : 1, "TURNO" : 1, "CONTROL" : 1, "ENT_ADMON" : 1, "P4A" : "CALLE", "P4B" : "MEXICO LIBRE", "P4C" : "SN", "P4D" : "", "P4E" : "FRACCIONAMIENTO", "P4F" : "MORELOS I", "P4G" : 20298, "P4H" : "SIGLO XXI", "P4I" : "CONGRESO DE CHILPANCINGO", "P4J" : "HERMANOS GALEANA", "P4K" : "", "P4L" : "P148A" : 8, "P148B" : 0, "P148C" : 13, "P148D" : 0, "P148E" : 99, "P148F" : 99, "P148G" : 99, "P148H" : 99, "P149" : 1, "P150" : "", "P151" : "", "P152" : "", "P153" : "", "P154" : "", "P155" : "", "P156" : "", "P157" : 2, "P158" : "", "P159" : "", "P160" : "", "P161" : "", "P162" : 1, "P163" : "", "P164" : "", "P165" : "", "P166" : 385, "P167" : 420, "P168" : 2, "P169" : "", "P170" : "", "P171" : 1, "P172" : "", "P173" : "", "P174" : "", "P175" : 1, "P176" : "", "P177" : "", "P178" : 2, "P179" : 2, "P180" : 2, "P181" : 4, "P182" : "", "P183" : "", "P184" : "", "P185" : "", "P186" : "", "P187" : "", "P188" : "", "P189" : "", "P190" : "", "P191" : "", "P192" : 1, "P193" : "", "P194" : "", "P195" : 1, "P196" : 1, "P197" : "", "P198" : "", "P199" : "", "P200" : "", "P201" : "", "P202" : 1, "P203" : "", "P204" : "", "P205" : "", "P206" : "", "P207" : 1, "P208" : 2, "P209" : 2, "P210" : "", "P211" : "", "P212" : "", "P213" : "", "P214" : "", "P215" : "", "P216" : 1, "P217" : "", "P218" : 5, "P219" : 1, "P220" : "", "P221" : 0, "P222" : 1, "P223" : "", "P224" : 0, "P225" : "", "P226" : "", "P227" : "", "P228" : "", "P229" : "", "P230" : "", "P231" : 1, "P232" : "", "P233" : 4, "P234" : 1, "P235" : "", "P236" : 4, "P237" : 4, "P238" : 2, "P239" : "", "P240" : 2, "P241" : "", "P242" : 0, "P243" : "", "P244" : "", "P245" : 2, "P246" : 2, "P247" : "", "P248" : 2, "P249" : 2, "P250" : 2, "P251" : 2, "P252" : 2, "P253" : 2, "P254" : 1, "P255" : 2, "P256" : 2, "P257" : 1, "P258" : 2, "P259" : 2, "P260" : 1, "P261" : 2, "P262" : 2, "P263" : 2, "P264A" : 1, "P264B" : "PROGRAMA BECAS DE OPORTUNIDADES", "P264C" : "", "P264D" : "", "P265" : 2, "P266" : 2, "P267A" : "", "P267B" : "", "P267C" : "", "P267D" : "", "P267E" : "", "P267F" : "", "P267G" : "", "P267H" : "", "P268" : 1, "P269" : 1, "P270" : 1, "P271" : 1, "P272" : 2, "P273" : 1, "P274" : 2, "P275" : "", "P276" : 20, "P277" : 20, "P278" : 0, "P279" : 20, "P280" : 15, "P281" : 19, "P282" : 1, "P283" : 19, "P284" : 9999, "P285" : 4, "P286" : 9999, "P287" : 9999, "P288" : 15, "P289A" : 1, "P289B" : "MAESTRO PARTICULAR", "P290" : 1, "P291" : 1, "P292" : 10, "P293" : 1, "P294" : 10, "P295" : 999, "P296" : 1, "P297" : 1, "P298A" : "45EB", "P298B" : "", "P298C" : "", "P299A" : "fundadores.tv@hotmail.es", "P299B" : "", "P300A" : "", "P300B" : 1, "P301" : 21, "P302" : 7, "P303" : 14, "P304" : "", "P305" : 1, "P306" : 1, "P307" : "", "P308" : "", "P309" : "", "P310" : "", "P311" : "", "P312" : "", "P313" : 18, "P314" : 6, "P315" : 12, "P316" : "", "P317" : 2, "P318" : "", "P319" : 2, "P320" : "", "P321" : "", "P322" : "", "P323" : "", "P324" : "", "P325" : 377, "P326" : 205, "P327" : 172, "P328" : 68, "P329" : 35, "P330" : 33, "P331" : 51, "P332" : 19, "P333" : 32, "P334" : 68, "P335" : 46, "P336" : 22, "P337" : 57, "P338" : 32, "P339" : 25, "P340" : 65, "P341" : 41, "P342" : 24, "P343" : 68, "P344" : 32, "P345" : 36, "P346" : "", "P347" : "", "P348" : "", "P349" : "", "P350" : "", "P351" : "", "P352" : "", "P353" : "", "P354" : "", "P355" : "", "P356" : "", "P357" : "", "P358" : "", "P359" : "", "P360" : "", "P361" : "", "P362" : "", "P363" : "", "P364" : 119 }
```

Imagen 23. Visualización de datos en MongoDB

Explore

/CEMABE/CEMABE/CENTROS

Finished: took 9608ms.

CENTROS	P4D	P4C	P4B	P4A	ENT_ADMON	CONTROL	TURNO	MODALIDAD	NIVEL	NOMBRECT	MZA	AGEB	NOM_LOC	LOC	NOM_MUN	MUN	NOM_ENT	ENT	CLAVE_CT	ID_INM
ACCIONAMIENTO	S/N	SEBASTIAN DE LARA	CALLE	1	1	2	1	3	2010 CENTENARIO DE LA REVOLUCION MEXICANA	3	3539	Aguascalientes	1	Aguascalientes	1	Aguascalientes	1	01DPR0698R2	664	
ACCIONAMIENTO	350	MAHATMA GANDI	AVENIDA	1	2	1	1	4	COLEGIO LINCOLN, A.C.	31	869	Aguascalientes	1	Aguascalientes	1	Aguascalientes	1	01PES0046P1	644	
ACCIONAMIENTO	SN	MEXICO LIBRE	CALLE	1	1	1	1	3	MARIANO JIMENEZ	15	2691	Aguascalientes	1	Aguascalientes	1	Aguascalientes	1	01DPR0647K1	513	
ACCIONAMIENTO	SN	JARDIN E ZARAGOZA	CALLE	1	1	2	1	3	JARDINES DE AGUASCALIENTES	12	2102	Aguascalientes	1	Aguascalientes	1	Aguascalientes	1	01DPR0073O2	477	
ACCIONAMIENTO	128	DELLAGO	AVENIDA	1	2	1	1	2	INSTITUTO ERIK ERIKSON	43	159A	Aguascalientes	1	Aguascalientes	1	Aguascalientes	1	01P.JN0124T1	650	
ACCIONAMIENTO	600	REPUBLICA COSTA RICA	CALLE	1	1	2	1	3	BENEMERITO DE LAS AMERICAS	19	084A	Aguascalientes	1	Aguascalientes	1	Aguascalientes	1	01DPR0127B2	636	
ONIA	S/N	FRANCISCO MARQUEZ	CALLE	1	1	1	1	8	UNIDAD DE SERVICIOS DE APOYO A LA EDUCACION REGULAR NUM. 41	5	731	Aguascalientes	1	Aguascalientes	1	Aguascalientes	1	01FUA0041F1	472	
ACCIONAMIENTO	S/N	JOSE CALVILLO	CALLE	1	1	1	1	4	22 DE OCTUBRE	3	1439	Aguascalientes	1	Aguascalientes	1	Aguascalientes	1	01DES0007I1	598	

Imagen 24. Visualización de datos en SlamData

A continuación presento varias gráficas del análisis de los datos a través de la herramienta SlamData.

Análisis 1:

1.- Query que dividen por municipio las escuelas del estado de Aguascalientes que hacen uso de las instalaciones deportivas.

```
SELECT NOM_MUN AS MUNICIPIO, NOM_ENT AS ESTADO FROM  
"/CEMABE/CEMABE/CENTROS" WHERE P192=1 AND  
NOM_ENT='Aguascalientes'
```

2.- Query que cuenta las escuelas agrupándolas por municipio

```
SELECT COUNT(MUNICIPIO) AS TOTAL_ESCUELAS, MUNICIPIO, ESTADO  
FROM "out1" GROUP BY MUNICIPIO
```

3.- Query que elimina las filas duplicadas

```
SELECT DISTINCT MUNICIPIO, TOTAL_ESCUELAS, ESTADO FROM "out2"
```

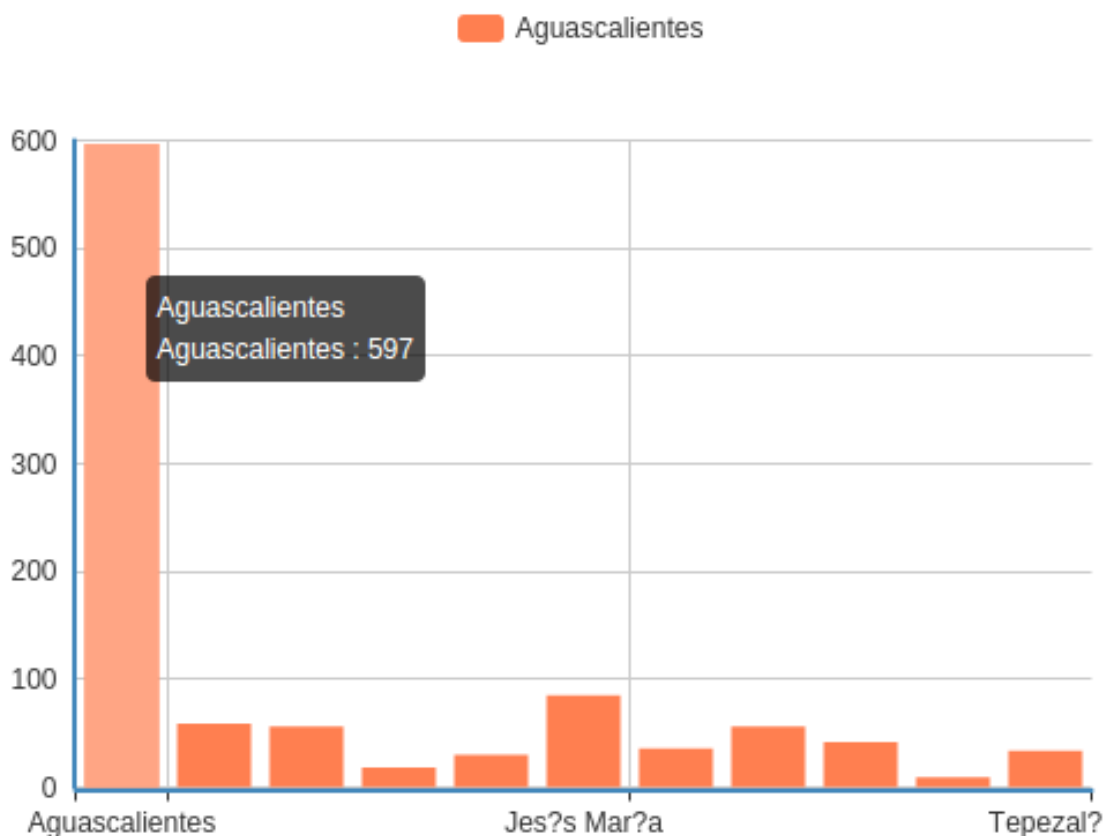


Imagen 25. Uso de instalaciones deportivas por Municipio del estado de Aguascalientes.

Análisis 2:

1.- Query que dividen por municipio las escuelas del estado de Aguascalientes que no hacen uso de las instalaciones deportivas.

```
SELECT NOM_MUN AS MUNICIPIO, NOM_ENT AS ESTADO FROM  
"/CEMABE/CEMABE/CENTROS" WHERE P192 in (2,9,') AND  
NOM_ENT='Aguascalientes'
```

2.- Query que cuenta las escuelas agrupándolas por municipio

```
SELECT COUNT(MUNICIPIO) AS TOTAL_ESCUELAS, MUNICIPIO, ESTADO  
FROM "out1" GROUP BY MUNICIPIO
```

3.- Query que elimina las filas duplicadas

```
SELECT DISTINCT MUNICIPIO, TOTAL_ESCUELAS, ESTADO FROM "out2"
```

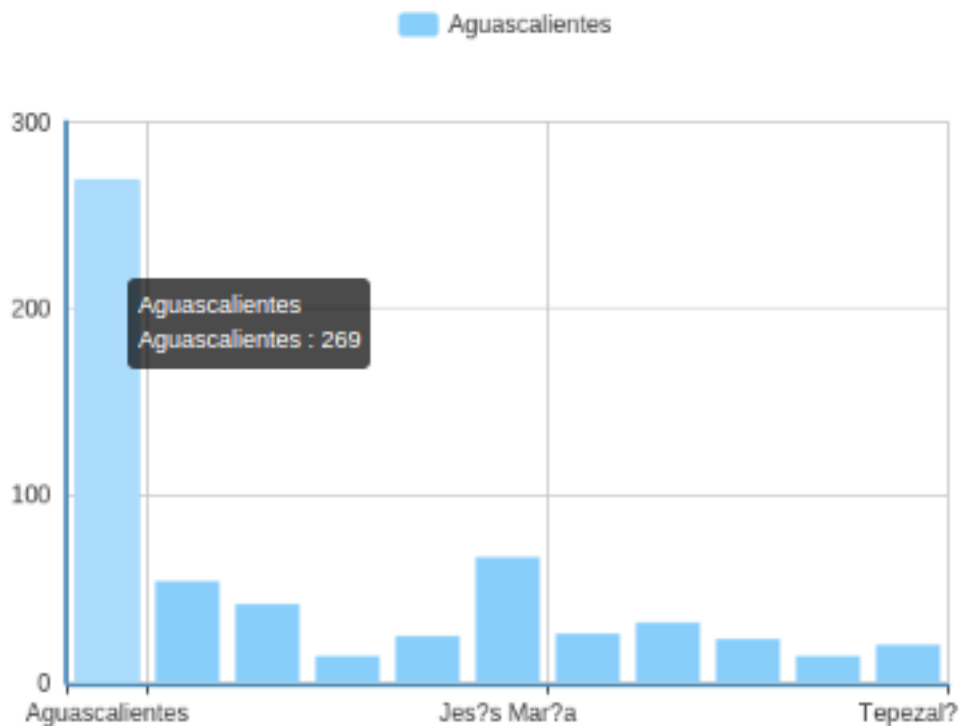


Imagen 26. No uso de instalaciones deportivas por Municipio del estado de Aguascalientes.

Rendimiento de SlamData:

Rendimiento promedio	
Procesos	Tiempo (Seg)
Carga de datos	22
Query 1	15
Query 2	7
Query 3	5

Tabla 27. Rendimiento de SlamData.

El tiempo de la carga de datos y las consultas puede variar dependiendo de la capacidad del ordenador que se use, en este caso el ordenador usado es el siguiente:

- **Toshiba Satellite-L635.**
 - Memoria RAM 4 GB
 - Procesador Intel Core i5 CPU M 460 2.53GHz x 4
 - S.O. Ubuntu 14.04 LTS de 64 Bits
 - Disco Duro 700 GB

Conclusiones sobre el caso de uso de obesidad:

En relación a las gráficas y los datos analizados se tiene lo siguiente:

- El estado de Aguascalientes esta conformado por 1,608 escuelas en total, de las cuales solo 1,022 hacen uso de instalaciones deportivas representado el 63.55%.
- Las escuelas que no hacen uso de las instalaciones son 586 representado el 36.45% del total de las escuelas en el estado de Aguascalientes.
- Aguascalientes esta integrado por 10 Municipios que son considerados como zona rural y la capital Aguascalientes como zona urbana, teniendo como total 11 municipios.
- El total de las escuelas de la capital de Aguascalientes es de 866.
- Un total de 597 escuelas que hacen uso de las instalaciones deportivas en la capital, que equivale a un 68.93%

- El no uso de las instalaciones deportivas esta representada por el 31.07% que corresponden a 269 escuelas en la capital.
- Las escuelas en zona rural son en total 156 escuelas que representan un 15.27% del estado de Aguascalientes.
- En relación a las cifras del 2012 de la ENSANUT donde la capital “zona urbana” tenia un 37.4% de obesidad y sobrepeso infantil, y que también actualmente esta cifra se mantiene en el 2014, puede tener relación con el 31.07% de las escuelas que no hacen uso de las instalaciones deportivas en la capital Esto puede ser debido a que no se inculca a la población infantil la practica de actividades deportivas que son de ayuda para combatir la obesidad y el sobrepeso. La cifra de obesidad y sobrepeso a nivel estado es de 34.7% y la cifra de las escuelas que no hacen uso de instalaciones deportivas en el estado es de 36.45%.

5. Conclusiones y trabajos futuros

A través de la realización del entorno analítico en base a la investigación del estado actual de las bases de datos no relacionales, el análisis y selección de las herramientas que existen para la explotación de datos concluyo lo siguiente:

- He revisado, analizado, probado y seleccionado las herramientas para construir el entorno integrado.
- El entorno de análisis creado cumple con los objetivos planteados porque permite la importación de datos estructurados, semi estructurados y no estructurados en formato CSV y JSON.
- MongoDB cumple con los objetivos porque permite el almacenamiento de los diferentes tipos de datos.
- SlamData cumple con los objetivos de procesar y extraer los diferentes tipos de datos almacenados, de tal forma que permite realizar análisis visual construyendo gráficas básicas, así mismo puede generar CSV para explotarlos con otras herramientas más potentes.

- Se ha desarrollado un script en Puppet utilizando Vagrant para el despliegue y configuración del entorno analítico de tal manera que se realice de forma automática y pueda ser utilizado para futuros trabajos.

Las limitantes que tuve en la realización de este proyecto son:

R Studio es una herramienta muy buena para el análisis científico de datos pero en relación con MongoDB aún queda mucho por hacer siendo trabajo a futuro el desarrollo de librerías que den solución a este problema relacionado a las funciones y librerías para extraer grandes cantidades de los datos.

Pentaho en su última actualización que es la versión 5.4 soporta la conexión directa con MongoDB 3.0 enfocándose al 100% al estudio de BigData, pero es una herramienta de pago de la cual hacen uso las grandes empresas que requieren analizar grandes cantidades de datos.

En lo que respecta a futuros trabajos el entorno integrado se puede ampliar con la implementación de nuevas herramientas de análisis de tipo OpenSource, que permitan la realización de graficas más dinámicas para un análisis más profundo de los datos. También se debe considerar para mejorar el entorno analítico la implementación de las actualizaciones de la herramienta SlamData que puedan surgir en un futuro, esto debido a su constante actualización e innovación.

Bibliografía

- [1] Gantz J, Reinsel D (2011) Extracting value from chaos. IDC iView, pp 1-12.
- [2] Manyika J, McKinsey Global Institute, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C, Byers AH (2011) Big data: the next frontier for innovation, competition, and productivity. McKinsey Global Institute.
- [3] Shashank Tiwari, (2011) Professional NoSQL ISBN 978-0-470-94224-6
- [4] Laney D (2001) 3-d data management: controlling data volume, velocity and variety. META Group Research Note, 6 February
- [5] Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107-113.
- [6] Hey AJG, Tansley S, Tolle KM et al (2009) The fourth paradigm: data-intensive scientific discovery.
- [7] Gestión de datos no estructurados <http://www.dataprix.com/blog-it/big-data/big-data-gestion-datos-no-estructurados>
- [8] Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, Burrows M, Chandra T, Fikes A, Gruber RE (2008) Bigtable: a distributed storage system for structured data. ACM Trans Comput Syst (TOCS) 26(2):4
- [9] Lakshman A, Malik P (2009) Cassandra: structured storage system on a p2p network. In: Proceedings of the 28th ACM symposium on principles of distributed computing. ACM, pp 5-5
- [10] George L (2011) HBase: the definitive guide. O'Reilly Media Inc.
- [11] MongoDB, <https://www.mongodb.org/>
- [12] Magic Quadrant for Operational Database Management Systems, <http://www.gartner.com/technology/reprints.do?id=1-23B94M3&ct=141017&st=sb>
- [13] Características de MongoDB <http://www.mongodbspain.com/es/2014/08/17/mongodb-characteristics-future/>
- [14] SlamData, <http://slamdata.com>
- [15] Pentaho, <http://www.pentaho.com>

- [16] R Studio, <http://www.rstudio.com>
- [17] Python Software Foundation, <https://www.python.org>
- [18] Data Driven Documents, <http://d3js.org>
- [19] Puppet Labs, <https://puppetlabs.com>
- [20] Vagrant Documentation, <https://docs.vagrantup.com/v2/>
- [21] Querys Pentaho, <http://wiki.pentaho.com/display/EAI/MongoDB+Input>
- [22] SQL to MongoDB Mapping chart
<http://docs.mongodb.org/manual/reference/sql-comparison/>
- [23] Empresas MongoDB, <https://www.mongodb.com/who-uses-mongodb>
- [24] Instituto Nacional de Estadística y Geografía (INEGI), <http://www.inegi.org.mx>
- [25] La Jornada Aguascalientes 2015, <http://www.lja.mx/2014/12/aguascalientes-cerrara-el-ano-con-similares-indices-de-obesidad-y-sobrepeso-que-el-resto-del-pais/>
- [26] Encuesta Nacional de Salud y Nutrición,
<http://ensanut.insp.mx/informes/Aguascalientes-OCT.pdf>
- [27] NoSQL como el futuro de las bases de datos,
<http://www.maestrosdelweb.com/nosql-como-el-futuro-de-las-bases-de-datos/>
- [28] BSON, <https://es.wikipedia.org/wiki/BSON>