IMPLEMENTACIÓN DE UNA HERRAMIENTA DE AUTOGESTIÓN Y AUTOCONFIGURACIÓN PARA LA IMPLEMENTACIÓN DE SERVICIOS EN PROYECTOS DE BIG DATA

UNIVERSIDAD CATÓLICA DE COLOMBIA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

BOGOTÁ D.C

Año

2017

IMPLEMENTACIÓN DE UNA HERRAMIENTA DE AUTOGESTIÓN Y AUTOCONFIGURACIÓN PARA LA IMPLEMENTACIÓN DE SERVICIOS EN PROYECTOS DE BIG DATA

EDWIN ANDRES TORRES ROBLES

WILLIAM ALONSO RINCÓN SAAVEDRA

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE

INGENIERO DE SISTEMAS

DIRECTOR

DIEGO ALBERTO RINCÓN YÁÑEZ

INGENIERO DE SISTEMAS

UNIVERSIDAD CATÓLICA DE COLOMBIA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

BOGOTÁ D.C

Año

2017

NOTA DE ACEPTACIÓN

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Facultad de Ingeniería y la Universidad Católica de Colombia para optar al título de Ingenieros de Sistemas.

_____

Jurado

_____

Diego Alberto Rincón
Director

_____

Revisor Metodológico.

24 De Septiembre de 2018

AGRADECIMIENTOS

En primera medida agradecemos a Dios que hace posible todas las cosas, a nuestros padres que con perseverancia y sacrificio han hecho lo posible por brindarnos la oportunidad de ser profesionales, agradecemos a todos los profesores que hicieron parte de nuestro proceso de aprendizaje. También agradecemos a todos aquellos que nos han impulsado a finalizar este proceso, podemos decir hoy que ***"lo logramos".***

TABLA DE CONTENIDO

LISTA DE IMÁGENES

LISTA DE TABLAS

TABLA DE ANEXOS

ANEXO A

- Documento (Manual) Implementación de Herramienta de Autogestión (Chef Server) para el levantamiento de servicios en proyectos de Big Data

ANEXO B

- Manuales de Instalación y configuración de las herramientas usadas para la implementación de los servicios.

GLOSARIO

SQL: (Structured Query Language). El lenguaje de consulta estructurado es un lenguaje de base de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos[1].

ERP: (Enterprise resource planning software) Software que unifica todas las necesidades de todos y cada uno de los departamentos en un único sistema, centralizando la información de la empresa y soportando todas las necesidades particulares de cada departamento[2].

OPEN SOURCE: (Código Abierto) Lo primero que hay que tener en cuenta es que cuando 'compramos un programa' lo que realmente estamos haciendo es pagar por una licencia de uso del programa. El programa sigue perteneciendo al desarrollador[3].

BI: Business Intelligence es la habilidad para transformar los datos en información, y la información en conocimiento, de forma que se pueda optimizar el proceso de toma de decisiones en los negocios[4].

IOT: Internet of Things (IoT), también llamado Internet of Objects, lo cambiará todo, incluidos a nosotros mismos. Esto puede parecer una afirmación atrevida, pero piense en el impacto que Internet ha tenido ya en la educación, la comunicación, los negocios, la ciencia, el gobierno y la humanidad. Es evidente que Internet es una de las creaciones más importantes y potentes de la historia de la humanidad[5].

SAAS: (Software as a Service) Es un modelo de distribución del software que permite a los usuarios el acceso al mismo a través de Internet[6].

TIC: Se definen colectivamente como innovaciones en microelectrónica, computación (hardware y software), telecomunicaciones y optoelectrónica - microprocesadores, semiconductores, fibra óptica - que permiten el procesamiento y acumulación de enormes cantidades de información, además de una rápida distribución de la información a través de redes de comunicación[7].

NAT'S: es un dispositivo que permite el acceso a la Internet a redes privadas, dando a un grupo de máquinas una única IP pública[8].

BIG DATA: Es una definición utilizada en tecnología para referirse a la información o grupo de datos que por su elevado volumen, diversidad y complejidad no pueden ser almacenados ni visualizados con herramientas tradicionales[9].

HDFS: (Hadoop Distributed File System) es un framework que permite el proceso distribuido de grandes volúmenes de datos entre clusters de computación[10].

NoSQL: intenta describir el surgimiento de un número creciente de bases de datos no relacionales y distribuidos que no suelen proveer garantías ACID. El término ACID hace referencia a un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción[11].

JSON: Es un formato de datos muy ligero basado en un subconjunto de la sintaxis de JavaScript: literales de matrices y objetos. Como usa la sintaxis JavaScript, las definiciones JSON pueden incluirse dentro de archivos JavaScript y acceder a ellas sin ningún análisis adicional como los necesarios con lenguajes basados en XML[12].

FRAMEWORK: El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas de software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírsenos[13].

HTML: Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia[14].

BSD: Son las siglas de “Berkeley Software Distribution”. Así se llamó a las distribuciones de código fuente que se hicieron en la Universidad de Berkeley en California y que en origen eran extensiones del sistema operativo UNIX de AT&T Research[15].

ABSTRACT

This project seeks to analyze, investigate and facilitate information to users about the latest free tools, which allow self-management, infrastructure self-configuration and data management in Big Data-oriented projects.

To carry out this project a methodology is established in four phases; which begin with the collection of information, from the concept of Big Data, covering its history, architectures and tools that facilitate its management. Then an evaluation process is carried out based on criteria of impact, resources and time that generate on the organizations when they are implemented.

Afterwards, the mechanism to be implemented is designed, based on the Zeta, Kappa and Lambda architectures; making use of eight selected tools that are managed through Chef Server, through routines (scripts) designed for their administration and management. Finally, comparative tests are performed to measure the degree of optimization generated by the implementation of the tool vs. the normal installation processes, as well as the impact that this generates on IT management in organizations (implementing chef as a service delivery aid in zeta, kappa, and Lambda architectures).

Key words: Autoconfiguration, Big Data, Chef Server, Self-management, IT.

RESUMEN

Este proyecto busca analizar, investigar y facilitar información a los usuarios sobre las actuales herramientas gratuitas, que permiten la autogestión, autoconfiguración de la infraestructura y el manejo de datos en proyectos orientados a Big Data.

Para llevar a cabo este proyecto se establece una metodología en cuatro fases; las cuales inician con el levantamiento de información, desde el concepto de Big Data, abarcando su historia, arquitecturas y herramientas que facilitan su gestión. Luego se realiza un proceso de evaluación basado en criterios de impacto, recursos y tiempo, que generan sobre las organizaciones al ser implementadas.

Posteriormente se diseña el mecanismo a implementar, basándose en las arquitecturas Zeta, Kappa y Lambda; haciendo uso de ocho herramientas seleccionadas que se administran a través de Chef Server, por medio de rutinas (scripts) diseñadas para su administración y gestión. Finalmente se ejecutan pruebas comparativas para medir el grado de optimización generado al implantar la herramienta vs los procesos normales de instalación, al igual que el impacto que esta genera en la administración de TI en las organizaciones. (Implementar chef como ayuda para la prestación de servicio en arquitecturas zeta, kappa, lamba).

Palabras claves: Autoconfiguración, Autogestión, Big Data, Chef Server, TI.

INTRODUCCIÓN

Este proyecto hace parte de una iniciativa para profundizar en temas relacionados a la infraestructura en ambientes orientados a Big Data y de los rasgos de autogestión y autoconfiguración enfocada a las nuevas tendencias de almacenamiento y procesamiento de datos; dada la importancia que estos representan dentro de una organización en pro de su plan estratégico. De igual forma se analiza cómo las herramientas tecnológicas juegan un papel fundamental en la actualidad, dentro del desarrollo de diversas actividades que se realizan al momento de implementar un proyecto.

Teniendo en cuenta que la información constituye el capital más valioso de las organizaciones, surgen necesidades en administrar y configurar plataformas de infraestructura orientadas al análisis de gran cantidad de datos, las cuales buscan ejecutar las tareas de manera más fácil y rápida, permitiendo que las personas que las administran aprovechen su tiempo en procesos relacionados a la investigación y desarrollo, incluso mejorando la competitividad dentro de las compañías y estableciendo al área de IT dentro de un marco estratégico y no dentro del marco actual de la inmediatez y reactividad. Es por eso que las organizaciones implementan ecosistemas de Big Data para enfrentar los desafíos en el momento de soportar sus procesos, generando la necesidad de desarrollar herramientas de autogestión y autoconfiguración para el levantamiento de servicios, aplicaciones e infraestructura, buscando facilitar y optimizar tiempo y recursos en proyectos orientados al manejo masivo de datos. El almacenamiento de la información es clave para un buen resultado, por tanto, tener este tipo de herramientas asegura la competitividad de las empresas en sus mercados.

Para llevar a cabo el proyecto, se hace uso de una arquitectura basada en cuatro (4) componentes principales que debe contener un ecosistema de Big data: La indexación de datos, el almacenamiento de datos, el procesamiento de datos y la visualización de datos; todo con el propósito de facilitar la escalabilidad, confiabilidad, integridad y seguridad orientada a este tipo de proyectos. Para esto se implementa Chef Server, herramienta que mejora y facilita la administración de los componentes anteriormente mencionados, ésta se evalúa mediante pruebas instalando ocho (8) servicios, basándose en parámetros de medición relacionados a la optimización de recursos y mostrando como resultado el impacto que tiene en las organizaciones implementar este tipo de soluciones.

1. GENERALIDADES

1.1 ANTECEDENTES

Para realizar este trabajo de grado fue necesario investigar y dar claridad para que sirve y en qué consiste el mundo del big data. Es por esto que al realizar la investigación se obtuvo conocimiento de la historia del mismo, en el mundo a nivel organizacional se pudo encontrar que es un concepto que significa muchas cosas para muchas personas, dicho concepto ha dejado de estar limitado al mundo de la tecnología. Hoy en día se trata de una prioridad empresarial dada su capacidad para influir profundamente en el comercio de una economía integrada a escala global[16]. Además de proporcionar soluciones a antiguos retos empresariales. Big data inspira nuevas formas de transformar procesos, empresas, sectores enteros e incluso la propia sociedad[17]. Aun así, la amplia cobertura mediática que está recibiendo no nos permite distinguir claramente el mito de la realidad. El cliente aprovecha los datos internos para crear un mejor ecosistema de información.

Se da inicio a esta investigación basándose en estudios, proyectos y las experiencias obtenidas realizadas por académicos de la Universidad de Oxford, expertos en la materia y directivos empresariales. IBM es la principal fuente de las recomendaciones del estudio**,** dando como resultado que El 63% (aproximadamente dos tercios) de los encuestados afirma que el uso de la información (incluido big data) y la analítica está dando lugar a una ventaja competitiva para sus empresas. Si se le compara con el 37% de los encuestados en el "IBM 2010 New Intelligent Enterprise Global Executive Study and Research Collaboration", se habla de un aumento del 70% en tan solo dos años[18].

El estudio de Big Data Inicia en el momento que surge la necesidad de analizar grandes cantidades de Datos, es decir tanto en el pasado como en la actualidad siempre ha existido esta necesidad de almacenar y analizar información, se puede afirmar que el inicio de Big Data parte desde el origen del almacenamiento relacional. En 1970 aparece por primera vez el concepto de Base de Datos Relacional[19] el cual está basado en un concepto Matemático, este descubrimiento que para ese entonces era un concepto generaría la revolución en la manera de almacenar datos, básicamente el almacenamiento relacional se basaba en la forma en la que podía accederse a la información sin saber cómo esta estaba estructurada. Este concepto solo se aplicaría años más tarde. Hoy en día es un concepto que se aplica a diario al hacer cualquier consulta en Internet. Con el pasar de los años el flujo de información seguía en auge y se debía encontrar la manera de gestionar dicha información, tanto así que en censos realizados en países de gran superficie y de gran población como lo son EE. UU y Japón, se daban indicios de que manera y a qué proporción crecía esta información, la cual representa en esta época cifras significativas de datos que

no se podía tratar, de manera tradicional y requería muchos esfuerzos y tomaba demasiado tiempo.

El año de 1975, es considerado un punto crítico del crecimiento de información, en esta época la industria aumentaba a pasos agigantados y las grandes compañías tenían la necesidad de almacenar y organizar su información en pro de la mejora de sus procesos productivos, lo que marcó un cambio en las estrategias de negocio. Es así como en este año se adopta y nace el concepto que aún hoy en día se conoce como SQL (Structure Query Language)[20]. Paralelamente el crecimiento de información también fue posible por avances tecnológicos en el área de la electrónica, estos avances permitieron almacenar grandes cantidades de datos en dispositivos que no ocuparan espacio, por tanto, las empresas y organizaciones tenían una mejor forma de almacenar y organizar su información, tanto así que en este tiempo histórico se dio origen a los estudios predictivos del crecimiento de la información como lo cita Ithiel de Sola Pool en su artículo Tracking The Flow Of Information[21] enfocado en el sector de las telecomunicaciones.

En el año de 1985 se inicia el fenómeno que se llamó la MRP II (Material Requirement Planning), que consiste en la planificación de los recursos de Producción y más adelante seria conocido bajo el concepto de ERP que es el sistema de información que hoy en día las grandes compañías usan para la administración de los recursos, es decir que las empresas y organizaciones ubican a la gestión de información dentro de sus prioridades corporativas adoptando dicha Tecnología[22]. El auge de los sistemas ERP era frenético entre la década de los 80 y 90, permitía generar la sinergia entre las diferentes áreas que componen a una empresa u organización, de aquí que ya no era solo importante el almacenamiento de los datos si no la relación que se generaba entre los procesos, lo cual hacia que los datos y la forma de almacenar los mismos siguieran el camino de la especialización para lograr mejores resultados. En esta misma década hacia el año de 1989 se introduce un Conocimiento que hoy en día es la principal causa de que se hable de big data, nace el concepto de la inteligencia empresarial o en ingles el bussiness inteligent (BI),este mismo año Howard Dresner define la inteligencia empresarial como los conceptos y métodos que mejoran la toma de decisiones de negocios mediante el uso de sistemas de apoyo basados en datos reales[23]. A partir de aquí se desarrollarían herramientas que permitieran visualizar estos datos en tiempo real para tomar decisiones en una organización, nacen empresas como crystal reports y microstrategy las cuales ofrecían el desarrollo de informes y el análisis de datos a las empresas.

A inicios de la década de los 90 en el año 1992 Crystal Reports, genera el primer informe tomado desde una base de datos basado en Windows, esta aplicación permitía a las empresas generar informes de manera sencilla con escasa programación de código[24], permitiendo a las empresas implementar inteligencia empresarial de forma fácil y asequible. Hacia el año 1996 el mundo fue testigo del asombroso crecimiento de la informática y de Internet, en este momento histórico las necesidades del mercado apuntaban a generar herramientas de orden colaborativo, dejando a un lado la Intranet y abriendo

caminos en la globalización de las compañías a través de Internet, este cambio generaría el gran crecimiento exponencial del uso de datos y el consumo de información a niveles que hace 20 años hubiesen sido inimaginables, de aquí la importancia que tiene internet en el desarrollo de Lo que se conocería a futuro como Big Data. Luego del crecimiento de 1996 de la internet hacia el año de 1997, ya se veían las consecuencias en especial en lo que tenía que ver con la infraestructura y la capacidad que se debía tener para almacenar gran cantidad de datos, tal así que en 1997 el almacenamiento Digital empezó a ser más Rentable que el almacenamiento Físico y surgen las primeras plataformas orientadas a BI (Business Inteligent) [25].

Este mismo año es adoptado el termino de Big Data, el término fue usado por primera Vez por la NASA en el artículo realizado por los investigadores Michael Cox y David Ellsworth donde afirman que el ritmo del crecimiento de los datos empezaba a ser un problema para los sistemas informáticos, desde este momento a esto se le denomino el problema del "Big Data" [26]. En el año de 1999 surge el concepto de IOT o por sus siglas en ingles Internet de las cosas, un concepto que se basa en que cualquier dispositivo que genere una radiofrecuencia puede ser conectado a internet, este concepto fue descrito por Kevin Ashton quien dice que a través de las IOT "Si tuviéramos equipos que supieran todo lo que hay que saber acerca de las cosas, a partir de datos que recopilados sin nuestra ayuda, seríamos capaces de monitorizar y controlar todo, y reducir así considerablemente los costes, los desperdicios y las pérdidas" [27]. Dentro de la evolución de los datos siempre ha sido de gran ayuda saber cuántos datos se generan anualmente en internet, por tanto esta estadística ha cambiado considerablemente, es así que hacia 1999 se tenía un Exabyte de Información.

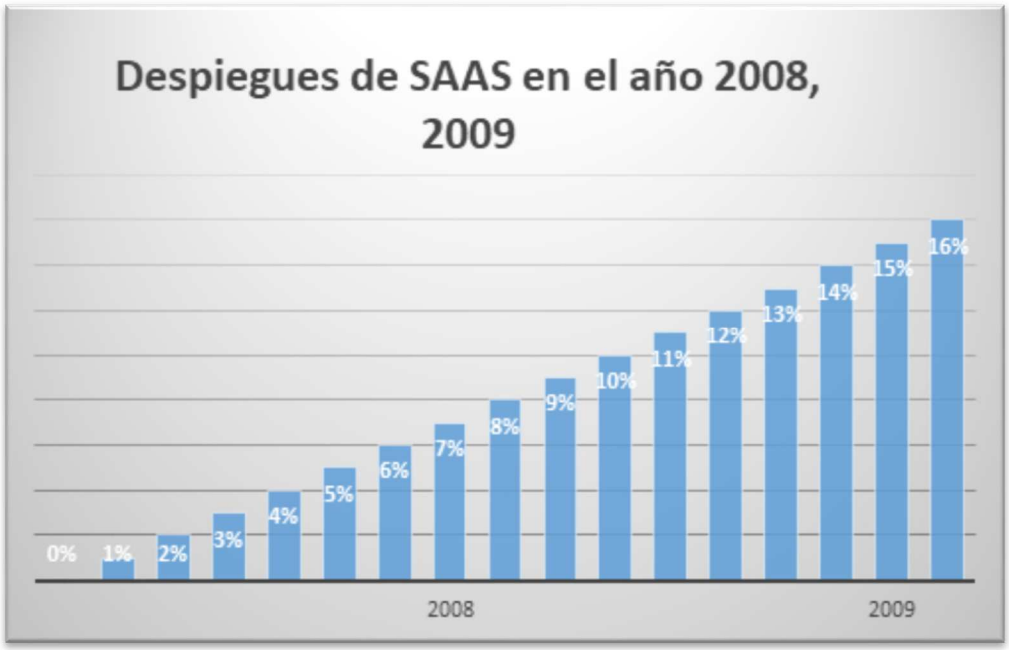
Esta cifra de acuerdo al cálculo realizado que se expresa en el artículo How Much Information[28] Donde se trata el tema de la cantidad de información almacenada en medios informáticos al final del milenio, predecía que el nivel de información iba a aumentar considerablemente y que cada vez se iba a tornar más tedioso manejar esta gran cantidad de datos. En el año 2001 se presenta el primer modelo Orientado hacia Big Data, el cual trata de las Tres V, que se conocen como las dimensiones de Big Data y en la actualidad aún se sigue aplicando este concepto. Las tres V que significan por sus siglas en Ingles (Volumen, Velocity Y Variety) o en español (Volumen, Velocidad y Variedad), son aquellas dimensiones en las cuales se basa el estudio de Big Data, de acuerdo a esto Doug Laney expresa en su artículo "Big Data en los entornos de Defensa y Seguridad," que estas son las 3 principales dimensiones que manejan un negocio, y que se deben tener en cuenta para tener un análisis exitoso de los datos almacenados[29]. Debido a la gran cantidad de Datos que existían hacia el año 2006 se ve la necesidad de implementar soluciones para el análisis de volúmenes significativos de información, tanto que en el año 2006 aparece Hadoop, que es una herramienta libre y de código abierto que permite el procesamiento en paralelo y distribuido de datos en servidores estándar que almacenan y procesan los datos[30], y que pueden escalarse sin límite. Luego del nacimiento de Hadoop y la gran escala de crecimiento que tenían la información en este momento de la historia, se adopta un interés particular por saber qué cantidad de información se deberá procesar y de qué manera esta

información será almacenada para que se tenga siempre disponible. En el año 2007 se genera un nuevo estudio acerca del crecimiento de la información[31], en donde se observa fácilmente que la información respecto al año anterior 2006 ha crecido a razón del Doble con respecto al año anterior y además se encontraban pronósticos de próximos años, donde se decía que hacia el 2010 se tendrían 988 Exabytes es decir aumentaría 16 veces el tamaño inicial en el 2016 y que esta aumentaría paulatinamente cada 18 meses, pero lo que no se imaginaban es que las cifras reales de 2010 superaron las predicciones realizadas, de tal manera que para el año 2010 ya se contaba con un volumen de información de 1227 exabytes, lo que daría pauta para tomar al manejo de los datos de una manera seria e importante. Según estudio de crecimiento de la información las predicciones Mostraban que hacia el año 2010 internet iba a ser 44 veces más grande de lo que era en el año 2009 y que alcanzaría el Orden de los Zeta Byte[32].

En el año 2008 se analiza el potencial que puede tener el estudio de Big Data, se dice que solo se ha visto la funcionalidad y el alcance que tiene Big Data en cuanto al almacenamiento y el análisis de los datos, pero que esta es considerada la mayor innovación en informática de esta última década, tanto así que se debe invertir más en estos estudios porque Big Data es una ciencia que hasta ahora está iniciando[33]. Se le da el respaldo a Big Data para que finalmente obtuviera un nivel de credibilidad intelectual que necesitaba. Al siguiente año en el 2009 según estudio realizado y estadísticas presentadas, se observa que BI (Business Inteligent) es ahora una de las prioridades para los directores en tecnología de información[34], a este punto es importante considerar herramientas que faciliten la autogestión y autoconfiguración para el manejo de la información, al dar este salto de importancia el estudio de BI paso a ser una gran responsabilidad para los administradores de los sistemas de información. En el año 2010 se era consciente que el crecimiento de la información era acelerado, tal como lo describe la revista The Economist en un artículo de este año en el cual resalta lo siguiente: "…el mundo contiene una cantidad de información digital de una magnitud inimaginable, cuyo ritmo de crecimiento es frenético… El efecto es patente en todos los ámbitos de nuestra vida, desde los negocios hasta la ciencia, los gobiernos o el arte"[35] en pocas palabras existe información en lo que hacemos y en lo que vemos desde lo más cotidiano hasta lo más avanzado.

Cerca del año 2011 ya los conceptos de Big Data y de Inteligencia Empresarial eran tendencia a nivel mundial a través de los servicios Cloud (En la Nube) o las SaaS (Software as a Service) las cuales en el 2010 habían tenido un incremento significativo como se observa en la Figura 1.

Figura 1. Incremento de implementaciones SaaS entre el 2008 y el 2009



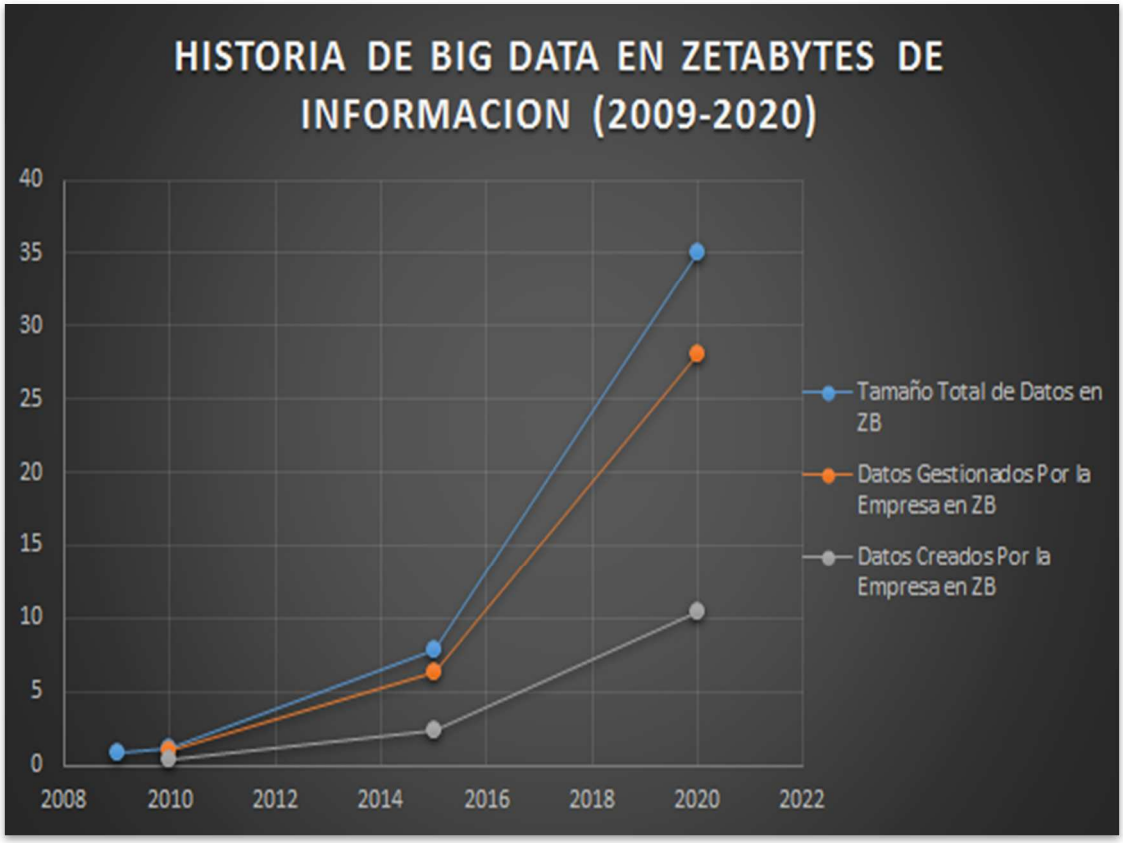
Fuente Autores.

De acuerdo a esto en el 2011 las tendencias se generaron en aspectos como la visualización de datos, el análisis Predictivo y el Big Data[36].

En el año 2012 surge el protocolo IPV6, debido a que el direccionamiento de IPV4 fue agotado; este mismo año también se ve la posibilidad con IPV6 de ampliar el direccionamiento de internet, para que cada vez más cosas se conectaran a esta, lo que supone que generará un aumento en la cantidad de datos que se transmitan por este medio. En el año 2014 luego del crecimiento a IPV6 se da inicio al año que se ha definido como el del internet de las cosas (IOT) este año según estudio el internet de las cosas, conectara a más de 4900 millones de Dispositivos para el año 2015[37] , como se observa el crecimiento del flujo de datos es cada vez mayor y supera todos los niveles predichos en años anteriores, por tanto en este año el IOT se ha convertido en una fuerza poderosa para la transformación de negocios y su enorme impacto afectará en los próximos años a todos los sectores y todas las áreas de la sociedad. Internet de las cosas representa una visión, en la que internet se extiende al mundo real abrazando objetos cotidianos y los elementos físicos ya no están desconectados del mundo virtual [38]. El uso de IOT permitirá el concepto de ciudades Inteligentes donde cada dispositivo se relaciona con su entorno para mejorar el desempeño la calidad y el rendimiento de los servicios urbanos, de acuerdo a la filosofía con el cual fue creado esto permitirá tener mayor eficacia en el uso de los recursos, así como evitar pérdidas significativas por el mal uso de los mismos. Finalmente el futuro de Big Data sigue en Auge y se ha dado a conocer un nuevo pronóstico de como crecerán los datos hasta el año 2020, en este estudio se

realiza un análisis del crecimiento donde se estima que los datos entre el 2015 y el 2020 tendrán un incremento asombroso como se puede ver en la siguiente imagen Figura 2, donde se establece que la información entre el 2015 y el 2020 aumentara considerablemente su tamaño.

Figura 2. Proyección de crecimiento consumo de datos global del año 2015 al año 2020



Fuente Autores.

Como se ve en la historia, el camino de big data es un campo que hasta ahora se está explorando, pero que aprovechado y combinado con otras herramientas de análisis constituye uno de los descubrimientos en el campo de la ingeniería más revolucionarios en los últimos años, que busca mejorar los aspectos de la sociedad en la manera de cómo se administran el tiempo y sus recursos.

1.2 PLANTEAMIENTO DEL PROBLEMA

Se observa que es de vital importancia para las empresas facilitar la administración de la infraestructura y sus recursos tecnológicos, debido al rápido crecimiento que estas experimentan; con el objetivo de mejorar sus capacidades al incorporar e implementar herramientas que permitan realizar estas tareas de forma más fácil y a menor costo, optimizando sus recursos y la administración de las áreas de TI, haciéndolas más económicas, eficientes y eficaces. Esto permite obtener mayores capacidades al momento de enfrentarse al mercado laboral; lo anterior con el propósito de lograr mejores resultados, brindando un servicio rápido y preciso a los usuarios y en especial a aquellos que necesitan la información de una manera oportuna para la toma de decisiones.

Cabe indicar que estas aplicaciones pueden ser ejecutadas en cualquier escenario posible, en el cual se vea la necesidad de buscar un mejor rendimiento en la administración de la infraestructura, tiempos de procesamiento y acceso a los datos, permitiendo así ser más competitivos en los procesos productivos y aprovechar la información para generar su crecimiento, estabilidad y aumentar su cadena de valor.

1.3 PREGUNTA GENERADORA

¿Qué grado de optimización se genera al implementar Chef Server en las organizaciones y cómo esta impacta a los departamentos de T.I.?

1.4 DESCRIPCIÓN DEL PROBLEMA

Hoy en día en las organizaciones a nivel global, consideran de vital importancia el tiempo empleado en las múltiples tareas desarrolladas en los departamentos de T.I. Es por esto que surge la necesidad de optimizar los procesos en busca del beneficio y el aumento de la competitividad, que constituye factor clave en las organizaciones, es así que la problemática se centra al momento de realizar implementaciones tanto de hardware o software por el tiempo que estas demandan, en especial en ambientes orientados al manejo de grandes volúmenes de información.

Tomando como referencia a varios países no solo de Latinoamérica sino también de varios continentes del mundo, se encuentra que existen herramientas que permiten optimizar los tiempos empleados en estas tareas. En la actualidad es fundamental la optimización del tiempo, enfocándose en las áreas de TI, este tiempo podría ser usado en tareas más enfocadas al negocio, aumentando el valor que las áreas de TI aportan a las organizaciones, de manera tal que TI constituya una parte importante de cara a la estrategia de una compañía. Al hablar de herramientas de autogestión y autoconfiguración, se habla de una práctica poco empleada en Colombia, que representa una gran ventana hacia el mejoramiento de la competitividad a nivel empresarial, enmarcando a las áreas de TI como pilares en la búsqueda del éxito dentro de las organizaciones.

1.5 DELIMITACIÓN

1.5.1 ALCANCE

- Definir la herramienta la cual administra la autogestión y autoconfiguración de los servicios a implementar.
- Seleccionar los 8 servicios que se van a implementar, basándonos en las características de los ambientes de Big Data.
- Implementar en ambiente de pruebas los servicios elegidos Orientados al manejo de Big Data.
- Dentro de la implementación de las herramientas solo se tendrán 8 servicios implementados y Disponibles para ser usados.
- Diseñar manuales de infraestructura, implementación, configuración y de usuario para facilitar la continuidad del proyecto a posteriores aportes.

1.5.2 LIMITACIONES

- El tiempo de entrega de la implementación de los servicios estará delimitado por el cronograma de proyectos de grado que tiene estipulado la Universidad católica de Colombia para el segundo semestre del año 2017.
- Los equipos disponibles necesarios para realizar la implementación de los servicios que se plantean como objetivo, ya que se necesitara la implementación de un clúster y equipos para llevar a cabo el desarrollo de la infraestructura.
- Solo se trabajarán Herramientas Open Source.
- Poco acceso a bases de datos indexadas.
- En el momento se tiene poco manejo de las herramientas que permiten la autogestión y autoconfiguración de servicios.
- La disponibilidad de Equipos en los Laboratorios de la Universidad Católica, con los que se espera contar y aprovechar para el Desarrollo del proyecto.
- Más que una limitación lo componen los desafíos que se encuentren en el desarrollo del proyecto, porque pueden causar atrasos y complicaciones para el buen término del mismo.

2. OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

- Implementar una herramienta de tipo open source para levantamiento de 8 servicios orientados a proyectos de Big Data.

2.2 OBJETIVOS ESPECÍFICOS

- Evaluar herramientas Open Source que permitan la autogestión y la autoconfiguración de servicios.
- Realizar la selección de 8 aplicaciones, para construir el ambiente a administrar por la herramienta.
- Implementar la herramienta de autogestión y autoconfiguración para el levantamiento automático de los servicios expuestos.
- Realizar pruebas y mediciones a la implementación de la herramienta de autogestión seleccionada.

3. MARCO DE REFERENCIA

3.1 ESTADO DEL ARTE

3.1.1 Contexto De Big Data Global y Local

En la actualidad se habla de un concepto que para muchos es desconocido y para otros representa un gran horizonte de incertidumbres, este concepto del cual se habla es comúnmente llamado Big Data. Según varios autores afirman que el mundo se encuentra en la era del Big Data la cual definen " Big Data aplica para toda aquella información que no puede ser procesada o analizada utilizando procesos y herramientas tradicionales" [39], desde el punto de vista organizacional se define como "un conjunto de herramientas para la gestión (management revolution)" [40], convirtiéndose en un factor importante a la hora de generar competitividad en las organizaciones a través de los 3 pilares: volumen, velocidad y variedad; siendo así Big Data una fuente que representa un gran potencial con un amplio espectro por explotar. Hoy en día basta con pensar que existen más dispositivos que personas en el mundo y que cada uno de estos dispositivos hacen uso de algún tipo de información, es aquí donde se centra el concepto y la tendencia de esta nueva tecnología, que se basa en obtener datos de todo lo que se encuentre conectado y realizar un análisis exhaustivo para facilitar la toma de decisiones; de esta manera, se encuentra que Big Data constituye una gran herramienta para el estudio de marketing así como para el estudio de otras ciencias.

Al dar un vistazo a esta tecnología se puede observar que abarca grandes campos de potencial para ser usada. En este momento se observa que la mayoría de aplicaciones se centra en temas relacionados con marketing, ciencia y salud; cabe resaltar que el concepto de Big Data puede ser extrapolado a cualquier mecanismo que haga uso de información, de manera tal la tendencia global apunta hacia el concepto de IOT (internet de las cosas),definido como la extensión de internet hacia artículos de uso frecuente, como celulares, tablets, equipos de sonido, televisores, neveras, sensores y todo dispositivo que se pueda comunicar a través de una IP. Con base a esto el estudio de Big Data, permite facilitar herramientas para toma de decisiones a gran escala y de gran precisión, que permiten capturar y procesar toda aquella información brindada por estos.

A escala global se observa que el mundo cambia dinámicamente y que estar preparado para este cambio es la diferencia que se necesita marcar, es por esto que las organizaciones en general adaptan el concepto de Big data en busca de enriquecer sus decisiones y no enfocarse solamente en intuiciones, permitiendo así tomar la decisión correcta en el momento correcto, lo que en las compañías se resume a éxito operativo. Big data no solo centra su atención en el marketing, también es considerada una gran herramienta de análisis en temas de investigación, donde ha logrado sacar provecho en campos de investigación

como física, bioinformática, astronomía y genética) [41], además de otros dominios como se observa en la Tabla1:

Big Data Sector	Big Data Uso
1. Ciencia	A. Descubrimientos Científicos
2. Telecomunicaciones	B. Nuevas tecnologías
3. Industria	C. Manufactura, procesos, control y transporte
4. Negocios	D. Servicios personales campañas
5. Calidad de vida y medio ambiente en las ciudades	E. Apoyo a la calidad de vida
6. Redes Sociales	f. Asistencia sanitaria
7. Salud	

Tabla 1: Dominios de Big Data fuente autores

A nivel local en Colombia hoy en día se ve el incremento del uso de las TIC'S (Tecnologías de la Información y la Comunicación), esto hace que sectores como el sector salud, infraestructura, industria y economía, aumenten la cantidad de información que manejan, constituyendo un marco referencial el cual apunta a tendencias de manejo masivo de información, o dicho de otra manera la implementación de Big Data en las organizaciones. De acuerdo a lo anterior en Colombia ya se tienen iniciativas para la implementación de estas tecnologías, se puede observar que el MINTIC (Ministerio de Telecomunicaciones y TIC en Colombia) dio su primer paso en el año 2016, al dar inicio a la construcción de los CEA (Centros de Excelencia y Apropiación) los cuales "Con esta iniciativa, el sector privado, la academia y el Estado impulsarán la investigación aplicada y el desarrollo de capacidades en Internet de las Cosas para resolver problemáticas reales y crear oportunidades hacia el futuro [42]

Tanto es la importancia que se le está dando a las iniciativas de Big Data en Colombia, que ya en el año 2015 se celebró el primer encuentro mundial relacionado a la tendencias nacionales e internacionales en Big Data, el cual tuvo como principales temáticas "Casos de éxitos a nivel mundial, Big data e Inteligencia de Negocios y Prospectiva del Big data en Colombia[43]

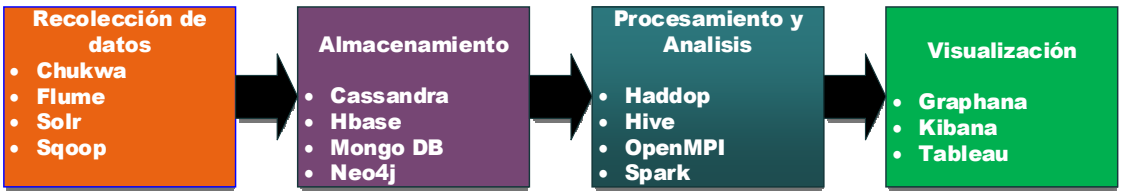
El Big Data se ha convertido en uno de los proyectos principales para el MINTIC en Colombia, con el objetivo que las empresas y gobierno optimicen sus negocios y la administración pública, a partir de un mejor aprovechamiento de la información digital. Por eso, Colombia le está apostando fuerte a esa generación de conocimiento y ha puesto en marcha en asocio con la empresa privada y la academia, un proyecto denominado 'Caoba'. Caoba es el Centro de Excelencia y apropiación en Big Data y Data Analytics – Alianza CAOBA -,que busca impulsar la competitividad y promover una política pública de Big Data y plantea como estrategia el uso de las tecnologías de Big Data y Data Analytics, a través de diferentes frentes que incluyen la formación del talento humano, la investigación aplicada y el desarrollo de productos cuya propuesta de valor está fundamentada en la generación de soluciones alrededor de las tecnologías del

BD&DA[44], es así como en la actualidad Colombia centra sus esfuerzos en constituir herramientas, que permitan el análisis de los datos para generar soluciones a problemáticas establecidas en el país.

3.2 Taxonomía

La taxonomía del Big data, la conforman aquellos componentes que son necesarios a la hora de construir ambientes orientados al manejo masivo de información, los cuales están fundamentados en lo que algunos autores llaman las “5V del Big Data Volumen, Velocidad, Variedad, Valor y Veracidad que constituyen un ecosistema de Big Data”[45] . Por eso para cumplir con estas características se puede dividir su taxonomía en 4 grandes reinos ver figura3, los cuales abarcan (Almacenamiento, procesamiento, distribución y visualización). A continuación se profundizará en cada uno de ellos teniendo en cuenta cada uno de sus componentes.

Figura 3 Taxonomía de Big Data



Fuente Autores.

3.2.1 Herramientas de Recolección de Datos

Chukwa

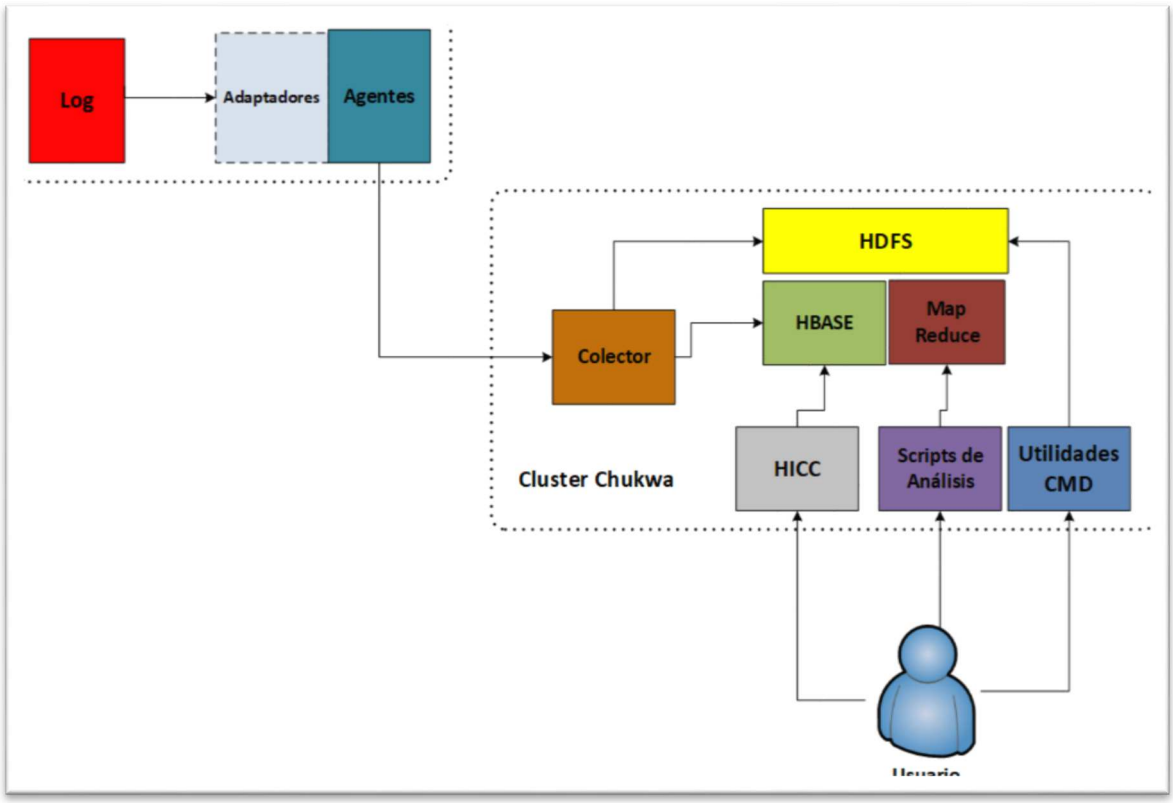
Diseñado para la colección y análisis a gran escala de "logs" de grandes sistemas distribuidos. Incluye un toolkit para desplegar los resultados del análisis y monitoreo[46]. Esta guiada sobre (HDFS), que es el sistema de archivos distribuidos de Hadoop, heredando la escalabilidad y la robustez; apache Chukwa también incluye herramientas para monitorear y analizar resultados, permitiendo el aprovechamiento al máximo de los mismos. Internamente apache Chukwa, está compuesto por:

Agentes: estos se ejecutan en cada máquina y realizan la recolección de datos.
Colectores: Son los que reciben los datos que extraen todos los agentes y se encargan de almacenarlos.
Procesos ETL: Se en carga de Realizar la Extracción, Transformación y Carga de la información contenida en los colectores

HICC: Es el panel central, que permite la visualización y monitoreo de réplicas de la información recolectada.

Para dar claridad a la forma que está compuesta la estructura de chukwa se pude observar en la figura 4 cada uno de sus componentes y como estos interactúan.

Figura 4 Componentes de Apache Chukwa



Fuente Autores.

Flume

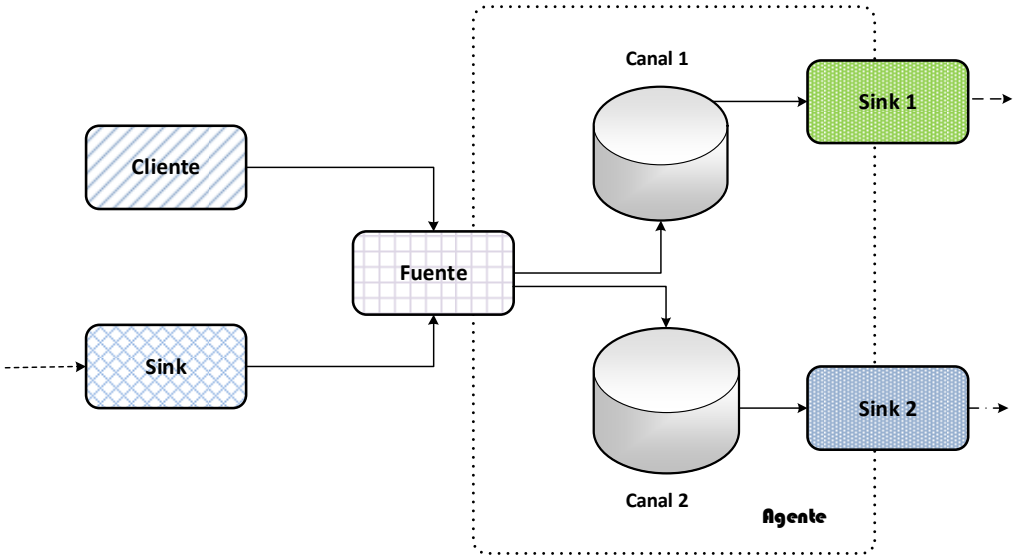
Flume es una herramienta distribuida para la recolección, agregación y transmisión de grandes volúmenes de datos desde diferentes fuentes a un data source centralizado[47]. Ofrece una arquitectura basada en la transmisión de datos por streaming altamente flexible y configurable pero a la vez simple.

Al tener un origen de datos configurable, Flume se adapta prácticamente a cualquier tipo de situación: monitorización de logs, descarga de información de redes sociales o mensajes de correo electrónico, entre muchas otras. Los destinos de los datos también son altamente configurables[48], es decir no solo está limitado a ser usados por Hadoop.

La arquitectura de Flume está compuesta por componentes, que son una serie de conceptos que a continuación se detallan:

Evento: Representa la unidad de datos que Flume pude transportar.
Flujo: Es el movimiento de datos desde un Origen a un Destino.
Cliente: es aquella que se ejecuta en su punto de origen y entrega los eventos del origen a un agente de Flume.
Agente: Es el que se encarga de reenviar eventos
Source: Es aquel que se encarga de consumir los eventos entregados, para luego decidir a qué canal se lo va a enviar.
Channel: Es un almacenamiento temporal para los eventos, desempeñan un papel fundamental en cuanto a garantizar la durabilidad de los flujos.
Sink: Son aquellos que se en cargan de eliminar eventos de un canal, y transmitirlos al siguiente agente o a la finalización del evento.
En la figura 5 Se puede observar como es el flujo de proceso en Flume

Figura 5 Flujo de proceso en Flume



Fuente Autores.

3.2.2 Herramientas de almacenamiento de Datos

Cassandra

Cassandra es una base de datos no relacional (NoSQL) distribuida y basada en un modelo de almacenamiento, dicho servicio fue iniciado pensando en un proyecto de Facebook, elaborado por medio de apache, sus características principales son las de ofrecer una alta disponibilidad en los datos y tolerancia a fallos, es compatible con hardware de bajos presupuestos que pueden ser manipulados y guardados directamente en la nube. Dicho servicio está

desarrollado en Java, muchas compañías en especial aquellas que manejan grandes volúmenes de información como Twitter o Facebook utilizan Cassandra dentro de su plataforma. Cassandra es una base de datos de código abierto cuya principal característica es que funciona de una forma dinámica con implementaciones de código cerrado. Su arquitectura está basada en peer to peer, esto quiere decir que todos sus nodos tienen la misma importancia jerárquica, ninguno obtiene un rol distinto y sus nodos se distribuyen de la misma manera evitando así que hayan fallos únicos. Gracias a peer to peer se garantiza una ubicación y un estado de la información, que está dispuesta a ser usada en el momento que se necesite y esto garantiza su alta nivel de sincronización.

Debido a la verticalidad de soluciones de datos relacionales y a la necesidad de ajustar el coste de la implementación, cada vez que hay un cambio en la escritura de cualquier fichero se activa su log asegurando así su coherencia. Una de las ventajas al implementar Cassandra es que ella cuenta con su propio lenguaje CQL (Cassandra Query Language), logrando que sea lo más fácil en su manejo con esto se convierte en un servicio fácil, ágil y versátil para la interacción con el usuario. A continuación se encuentran en la Tabla 2 las principales ventajas de hacer uso de Cassandra:

Ventaja	Características
1	Open Source
2	Arquitectura Peer to Peer
3	Elástica y Escalable
4	Alta Disponibilidad y tolerancia a fallos
5	Alto Rendimiento
6	Orientada a columnas
7	Esquema Libre

Tabla 2 Principales características de Cassandra fuente autores.

Apache HBase

Es una base de datos no SQL de tipo columna (column-oriented data base) que se ejecuta en HDFS. HBase no soporta SQL, de hecho, HBase no es una base de datos relacional. Se define HBase como un sistema de gestión de bases de datos de correlación ordenada multidimensional persistente, distribuido y vacío que se ejecuta en la parte superior de un sistema de archivos distribuidos (HDFS o GPFS-FPO) [49].

Principalmente HBase está compuesto por dos servicios primarios:

Servidor Maestro: Es el encargado de Administrar las regiones y el encargado del equilibrio de cargas.

Servidor de Región: Estos servidores son los encargados de realizar el trabajo, Y de establecer la comunicación con el cliente quien accede a los datos de Hbase. A su vez estos servidores tienen a cargo un conjunto de regiones que es como están compuestas las tablas en Hbase.

Hfiles: Se consideran la representación física de los datos dentro de Hbase, algo importante para resaltar es que en Hbase, no existen las caracterizaciones de datos, es decir no se habla de tipo de datos, todo el almacenamiento de datos en HBase es almacenado como Bytes, esta ausencia de esquema se debe a que en cada fila de HBase se pueden tener conjuntos de columnas diferentes.

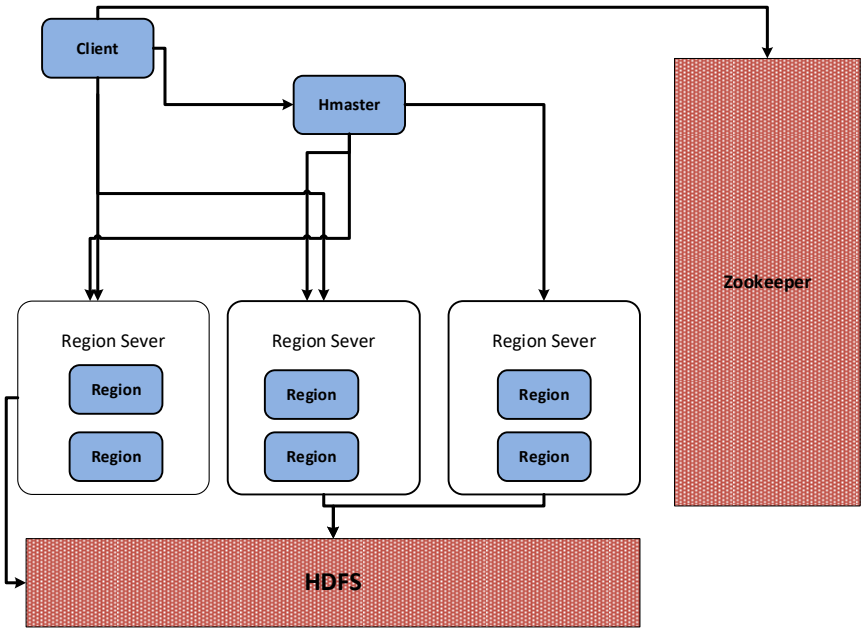
De tal manera que al visualizar una tabla, la representación lógica de la misma es representada como se observa en la tabla 3:

Clave de fila	Valor
11111	cfd: {'cqnm': 'name1', 'cqv': 1111} cfi: {'cqdesc': 'desc11111'}
22222	cfd: {'cqnm': 'name2', 'cqv': 2013 @ ts = 2013, 'cq_val': 2012 @ ts = 2012 }

Tabla3 Representación lógica de una tabla en Hbase fuente autores.

Cada tabla contiene filas y columnas como una base de datos relacional. HBase permite que muchos atributos sean agrupados llamándolos familias de columnas, de tal manera que los elementos de una familia de columnas son almacenados en un solo conjunto[50]. Por tanto para entender como está compuesta, en la figura 6 se encuentra el esquema de la arquitectura de HBase; como los Servidores Maestros y Servidores de regiones se integran sobre una estructura HDFS:

Figura 6 Arquitectura de Hbase



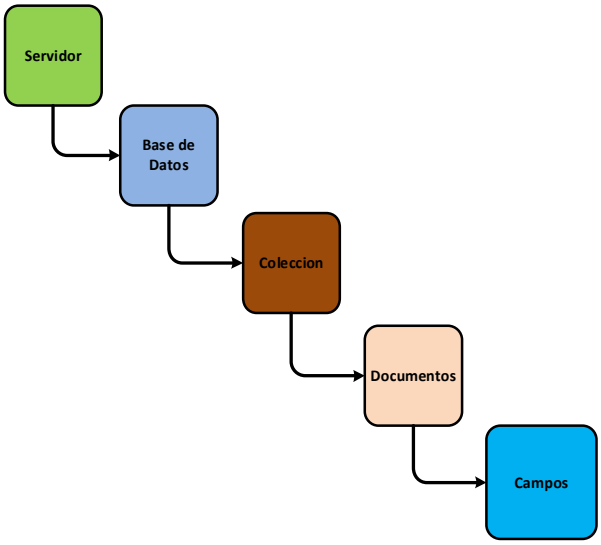
Fuente Autores.

MongoDB

Ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidatos. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria (in-memory) [51].

La arquitectura de MongoDB está basada en los siguientes componentes:

Figura7 Componentes de la arquitectura de MongoBD



Fuente Autores.

Mongo está recomendado Para el uso en empresas con volúmenes de datos altos, que por lo general usan una estrategia de búsqueda de información llamada minería de datos, estrategia que está siendo usada actualmente en grandes entidades y por tal motivo está siendo acogida en las medianas y pequeñas empresas. También se recomienda el uso de mongo a aquellas empresas las cuales utilicen grandes volúmenes de usuarios que realizan accesos concurrentes, las empresas que están acogiendo el uso de big data en sus portafolios también usan este servicio por su velocidad, eficacia y su manejo en grandes flujos de información.

Por tanto mongo con su eficacia en el direccionamiento de inmensos volúmenes de información, presenta una gran diferencia frente a las bases de datos relacionales, puesto que optimiza y agiliza el tiempo en la recolección de datos, logrando una mejor administración y rendimiento, disminuyendo el tiempo en la ejecución de los procesos. Debido a la versatilidad y flexibilidad MongoDB está a un mayor nivel que las bases de datos relacionales.

La replicación nativa de MongoDB y la tolerancia a fallos automática ofrece fiabilidad a nivel empresarial y flexibilidad operativa. Algunas versiones como MongoDB Enterprise ofrece seguridad avanzada, monitorización on-premises, soporte SNMP (Simple Network Management Protocol), certificaciones de SO.

Neo4j

Es una base de datos Orientada a Grafos (BDOG).Especialmente son usadas cuando se tiene información no estructurada y se quiere hallar valor en la misma. Este tipo de base de datos es muy usada en ambientes de Big Data, tanto así que ya son trabajadas en detección de fraudes en el sector de la banca, los seguros o el comercio.

Neo4j hace uso de grafos para representar datos y relaciones entre ellos, para esto hace uso de los siguientes tipos de grafos:

Grafos No Dirigidos: Los nodos y las relaciones son intercambiables

Grafos Dirigidos: Los nodos y relaciones no son bidireccionales

Grafos Con Peso: las relaciones entre nodos tienen un valor para luego realizar operaciones sobre estos.

Grafos Con Etiquetas: Son grafos etiquetados, es decir pueden definir los vértices con base en las etiquetas asignadas.

Grafos Con Propiedad: Es un grafo que tiene peso y etiquetas, con las cuales se pueden asignar propiedades.

Adicionalmente existen razones para usarlo, a continuación se encuentran las 5 razones más importantes según el fabricante[52].

Cinco Razones Por Cuales Usarlo.

- "Base de datos primera y mejor Gráfico del mundo, Neo4j es utilizada por miles de organizaciones, incluyendo 50 + del Global 2000, en aplicaciones de producción de misión crítica."

- "Mayor y más activa comunidad Gráfico en el Planeta, Neo4j tiene la mayor y más vibrante comunidad de entusiastas de la base de datos gráfica que contribuyen a un ecosistema robusto y activo."

- "Gran rendimiento de lectura y escritura escalabilidad, sin compromiso, Neo4j ofrece el rendimiento de lectura veloz y escribir lo que necesita, sin dejar de proteger su integridad de los datos."

- Totalmente nativo Gráfico de almacenamiento y de procesamiento de alto rendimiento adyacencia-índice libre acorta el tiempo de lectura y entrega de ultra-alto rendimiento paralelizado incluso cuando sus datos crece.

- "Más fácil que nunca el aprendizaje, Una interfaz de usuario madura, una función de aprendizaje y una gran cantidad de recursos de educación significan Neo4j es fácil de aprender y fácil de dominar.

En resumen se muestra en la tabla 4 las principales ventajas de Neo4J

Ventaja	Característica
1	Open Source
2	Interfaz Amigable
3	Fácil Modelamiento de Datos
4	Consultas legibles
5	Alto Rendimiento

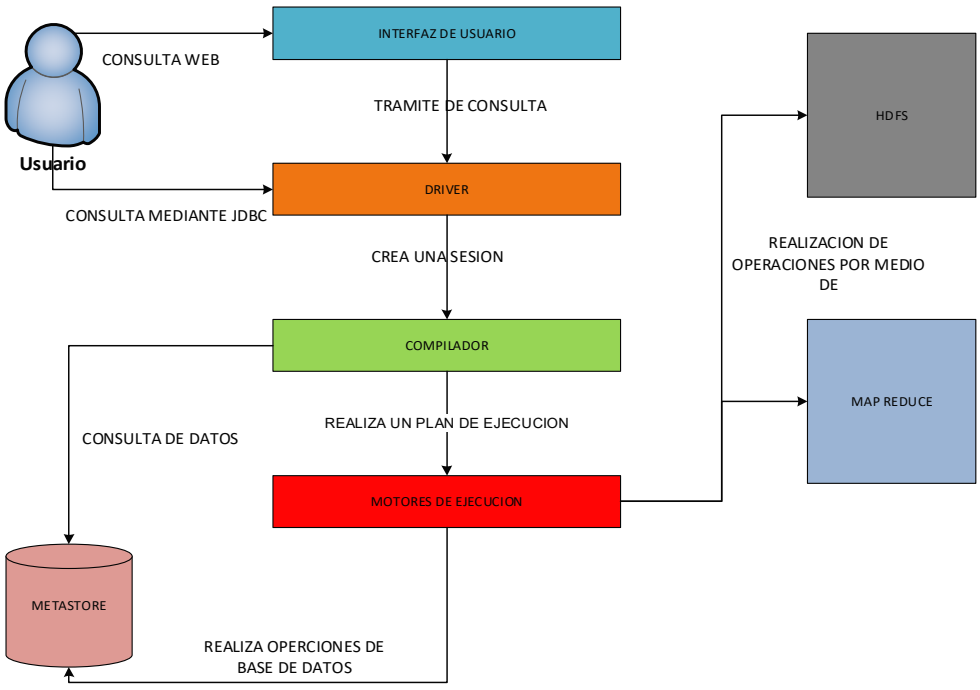
Tabla 4 Características principales Neo4j fuente autores

Hive

Es una infraestructura que inicialmente fue creada para hacer de Hadoop más fácil para operar y administrar grandes conjuntos de datos que se encuentran almacenados en un ambiente distribuido. Hive tiene definido un lenguaje similar a SQL llamado Hive Query Language (HQL), estas sentencias HQL son separadas por un servicio de Hive y son enviadas a procesos MapReduce ejecutados en el clúster de Hadoop.

Funciona en la capa superior a Hadoop y esta optimizado para realizar lecturas en la base de datos, pero no para realizar una gran cantidad de operaciones de escrituras. Tampoco es recomendable cuando se requiere una latencia baja [53]. Para una mejor comprensión a continuación se muestra en la figura 8 como es su arquitectura y el manejo que realiza.

Figura8 arquitectura Hive



Fuente Autores.

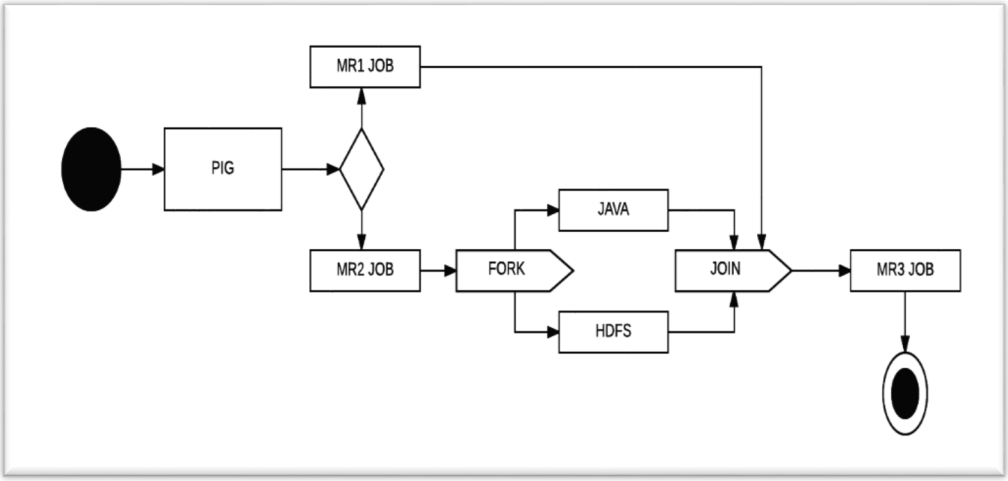
Jaql

Fue donado por IBM a la comunidad de software libre. Query Language for Javascript Object Notation (JSON) es un lenguaje funcional y declarativo que permite la explotación de datos en formato JSON diseñado para procesar grandes volúmenes de información. Para explotar el paralelismo, Jaql reescribe los queries de alto nivel (cuando es necesario) en queries de "bajo nivel" para distribuirlos como procesos MapReduce. Internamente el motor de Jaql transforma el Query en procesos Map y reduce para reducir el tiempo de desarrollo asociado en analizar los datos en Hadoop. Jaql posee de una infraestructura flexible para administrar y analizar datos semiestructurados como XML, archivos CSV, archivos planos, datos relacionales, etc.

Oozie

Están conformados por un conjunto de procesos que son ejecutados en distintos momentos los cuales necesitan ser orquestados, para satisfacer las necesidades de tan complejo análisis de información. Oozie es un proyecto de código abierto que simplifica los flujos de trabajo y la coordinación entre cada uno de los procesos. Permite que el usuario pueda definir acciones y las dependencias entre dichas acciones. Un flujo de trabajo en Oozie es definido mediante un grafo acíclico llamado Directed Acyclical Graph (DAG), y es acíclico puesto que no permite ciclos en el grafo; es decir, solo hay un punto de entrada y de salida y todas las tareas y dependencias parten del punto inicial al punto final sin puntos de retorno. Un ejemplo de un flujo de trabajo en Oozie se representa de la siguiente manera ver figura 9 [54]:

Figura 9. Flujo de trabajo en Oozie



Fuente Autores.

Redis

Redis es un código abierto (licencia BSD), en memoria almacén de estructura de datos, que se utiliza como base de datos, cache e intermediario de mensajes. Es compatible con las estructuras de datos tales como hilos, hash, listas, conjuntos ordenados, conjuntos con consultas de rango, mapas de bits, hyperloglogs y los índices geoespaciales con las consultas de radio. Redis ha incorporado en la replicación, secuencias de comandos Lúa, desalojo LRU, transacciones y diferentes niveles de persistencia en el disco, y proporciona una alta disponibilidad a través de Redis Sentinel y el particionamiento automático con Redis Cluster[55].

3.2.3 Herramientas de procesamiento y distribución

Pig

Inicialmente desarrollado por Yahoo! para permitir a los usuarios de Hadoop enfocarse más en analizar todos los conjuntos de datos y dedicar menos tiempo en construir los programas MapReduce. Tal como su nombre lo indica al igual que cualquier cerdo que come cualquier cosa, el lenguaje PigLatin fue diseñado para manejar cualquier tipo de dato y Pig es el ambiente de ejecución donde estos programas son ejecutados, de manera muy similar a la relación entre la máquina virtual de Java (JVM) y una aplicación Java.

Spark

Spark es el servicio que nos permite hacer trabajos paralelos todos en memoria, gracias a esto Spark logra optimizar tiempos en el procesamiento dándonos grandes cantidades de información en un tiempo racionable, todo este proceso lo realiza utilizando algoritmos iterativos, en especial si estos procesos son iterativos como los que se usan en servicios de machine learning. Este servicio ha pensado en ser un servicio de uso muy fácil y también de ser adecuado para que muchas aplicaciones sean compatibles, tales como, Java, Scala, Python, R entre otros. Spark fue desarrollada en Berkley que lo que busca realizar es un cálculo en la memoria por medio de dichos algoritmos iterativos que puedan ejecutarse en aplicaciones tales como las que anteriormente se mencionaron. Ya que Spark es un servicio relativamente nuevo en el mundo del Big Data ha generado que tenga una gran acogida con respecto a Hadoop[56].

Zookeeper

Zookeeper es otro proyecto de código abierto de Apache que provee de una infraestructura centralizada y de servicios que pueden ser utilizados por aplicaciones para asegurarse de que los procesos a través de un clúster sean serializados o sincronizados. Internamente en Zookeeper una aplicación puede crear un archivo que se persiste en memoria en los servidores Zookeeper llamado znode. Este archivo znode puede ser actualizado por cualquier nodo en el cluster, y cualquier nodo puede registrar que sea informado de los cambios ocurridos en ese znode; es decir, un servidor puede ser configurado para "vigilar" un znode en particular. De este modo, las aplicaciones pueden sincronizar sus procesos a través de un cluster distribuido actualizando su estatus en cada znode, el cual informará al resto del cluster sobre el estatus correspondiente de algún nodo en específico. Como podrá observar, más allá de Hadoop, una plataforma de Big Data consiste de todo un ecosistema tal y como se muestra en la figura 10, de proyectos que en conjunto permiten simplificar, administrar, coordinar y analizar grandes volúmenes de información.

Avro

Es un proyecto de Apache que provee servicios de serialización. Cuando se guardan datos en un archivo, el esquema que define ese archivo es guardado dentro del mismo; de este modo es más sencillo para cualquier aplicación leerlo, posteriormente puesto que el esquema está definido dentro del archivo.

Hadoop

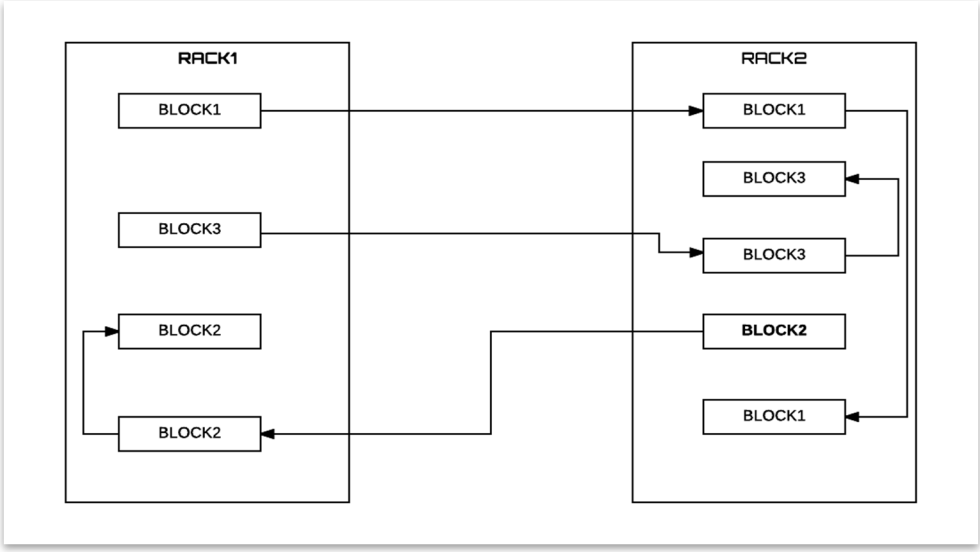
Hadoop está inspirado en el proyecto de Google File System (GFS) y en el paradigma de programación MapReduce, el cual consiste en dividir en dos tareas (mapper – reducer) para manipular los datos distribuidos a nodos de un clúster logrando un alto paralelismo en el procesamiento. Hadoop está compuesto de tres piezas:

- Hadoop Distributed File System (HDFS)
- Hadoop MapReduce
- Hadoop Common.
- Hadoop Distributed File System (HDFS)

Los datos en el clúster de Hadoop son divididos en pequeñas piezas llamadas bloques y distribuidas a través del clúster; de esta manera, las funciones map y reduce pueden ser ejecutadas en pequeños subconjuntos y esto provee de la escalabilidad necesaria para el procesamiento de grandes volúmenes.

La siguiente Figura 10 ejemplifica como los bloques de datos son escritos hacia HDFS. Observe que cada bloque es almacenado tres veces y al menos un bloque se almacena en un diferente rack para lograr redundancia.

Figura 10. Ejemplo los bloques de datos son escritos hacia HDFS



Fuente Autores.

- Hadoop MapReduce es el núcleo de Hadoop. El término MapReduce en realidad se refiere a dos procesos separados que Hadoop ejecuta. El primer proceso map, el cual toma un conjunto de datos y lo convierte en otro conjunto, donde los elementos individuales son separados en tuplas (pares de llave/valor). El proceso reduce obtiene la salida de map como datos de entrada y combina las tuplas en un conjunto más pequeño de las mismas. Una fase intermedia es la denominada Shuffle la cual obtiene las tuplas del proceso map y determina que nodo procesará estos datos dirigiendo la salida a una tarea reduce en específico.

  MapReduce consiste en un framework que proporciona un sistema de procesamiento de datos de una forma paralela y distribuida, fue creado pensando en solucionar problemas utilizando grandes conjuntos de datos de una forma paralela, utilizando un sistema de archivos distribuido como por ejemplo HDFS. Dicho framework esta hecho por medio de una arquitectura que cuenta con un servidor maestro y muchos servidores esclavos, utiliza un servidor esclavo por cada nodo del clúster[57].

  Este nuevo servicio o framework surge como la gran mayoría de servicios, partiendo de una necesidad para el mejoramiento en el procesamiento de los datos, entre las diferentes maquinas que existan en una red ya sea local o exteriorizada y su objetivo principal era optimizar el tiempo del proceso que se tomaba en realizar una tarea necesaria para estas redes.

La idea del nombre para este servicio parte de las funciones que tenían los programadores en ese momento, se habla de un paradigma que surgió en los años 2004 que trataba básicamente de dos obligaciones fundamentales las cuales se refieren a continuación.

Map: transforma un conjunto de datos de partida en pares (clave, valor) a otro conjunto de datos intermedios también en pares (clave, valor). Un formato, que hará más eficiente su procesamiento y sobre todo, más fácil su "reconstrucción" futura.

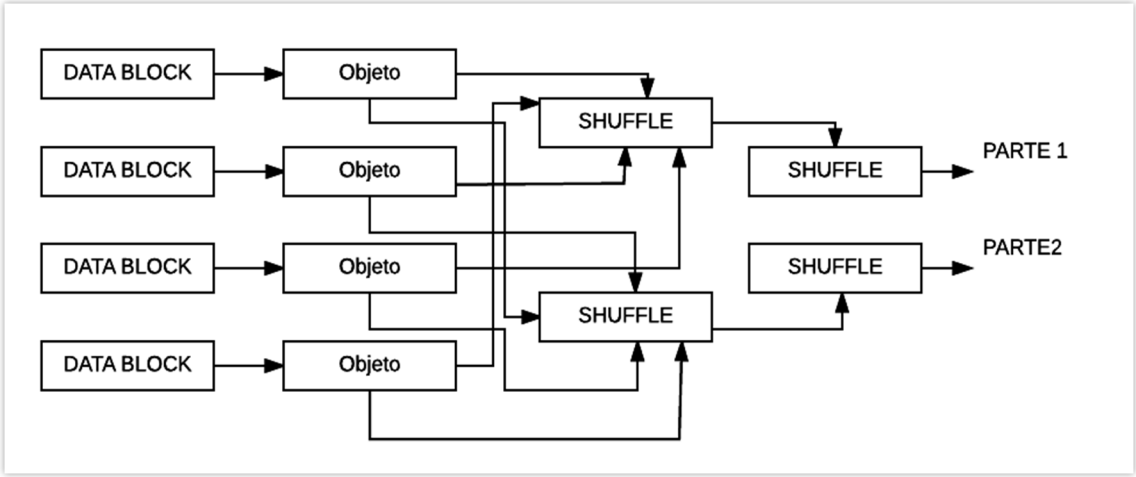
Reduce: recibe los valores intermedios procesados en formato de pares (clave, valor) para agruparlos y producir el resultado final[58].

Gracias a este paradigma es que se produce a la creación de este framework que como se dijo anteriormente se crea con objetivos específicos y cuyos resultados se dan satisfactoriamente abriéndose así paso para ser uno de los framework más utilizados en el mundo del Big Data.

Con Map Reduce se logró obtener mejores resultados que los que se obtenían con las bases de datos paralelas, gracias a que se da una independencia en el sistema de almacenamiento, y se le da un mejor manejo a los errores para los trabajos de gran capacidad. Logrando una herramienta con procesamiento de datos flexibles y de un buen manejo de los errores[59].

- La siguiente figura 11 se ejemplifica un flujo de datos en un proceso sencillo de MapReduce.

Figura 11. Ejemplo de flujo de datos



Fuente Autores.

- Hadoop Common Components son un conjunto de librerías que soportan varios subproyectos de Hadoop. Además de estos tres componentes

principales de Hadoop, existen otros proyectos relacionados los cuales son definidos a continuación.

Open Source High performance Computing (Open M.P.I)

Esta herramienta no es más que una API de código abierto desarrollada con el fin de facilitar la programación paralela y distribuida.

¿Por qué se recomienda usar esta herramienta? Ya que esta herramienta se utiliza por medio de la programación paralela y maneja el paradigma de paso de mensajes, esta desarrolla dicha implementación dejando un espacio para la librería de paso de mensajes y lo mejor es que servicio está a la mano de cualquier usuario que esté interesado en utilizar dicho servicio.

Lucene

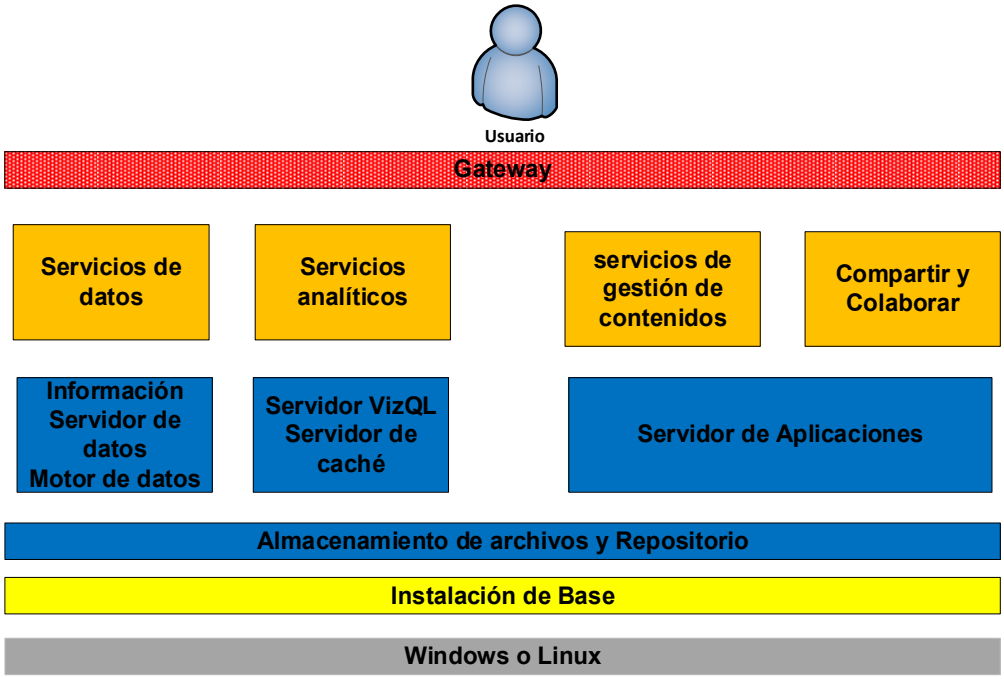
Es un proyecto de Apache bastante popular para realizar búsquedas sobre textos. Lucene provee de librerías para indexación y búsqueda de texto. Ha sido principalmente utilizado en la implementación de motores de búsqueda (aunque hay que considerar que no tiene funciones de "crawling" ni análisis de documentos HTML ya incorporadas). El concepto a nivel de arquitectura de Lucene es simple, básicamente los documentos son divididos en campos de texto (fields) y se genera un índice sobre estos campos de texto. La indexación es el componente clave de Lucene, lo que le permite realizar búsquedas rápidamente independientemente del formato del archivo, ya sean PDFs, documentos HTML, etc.

3.2.4 Herramientas De Visualización

Tableau

Es una herramienta que se encuentra de manera gratuita, Tableau es un sistema gráfico para realizar una exploración ad-hoc y análisis de conjuntos de datos de clientes. Es una continuación comercial del proyecto de investigación Polaris [60],el cual brindaba la oportunidad de generar consultas sobre base de datos, de una manera fácil e intuitiva, por lo tanto resulta ser una buena opción en el momento que se quiera escoger una herramienta para la visualización de datos. Su mayor beneficio es la facilidad en el manejo de la interfaz de usuario con la cual cuenta la herramienta. En la figura 12, se observa la arquitectura que emplea Tableau para su funcionamiento.

Figura 12 Arquitectura de Tableau



Fuente Autores.

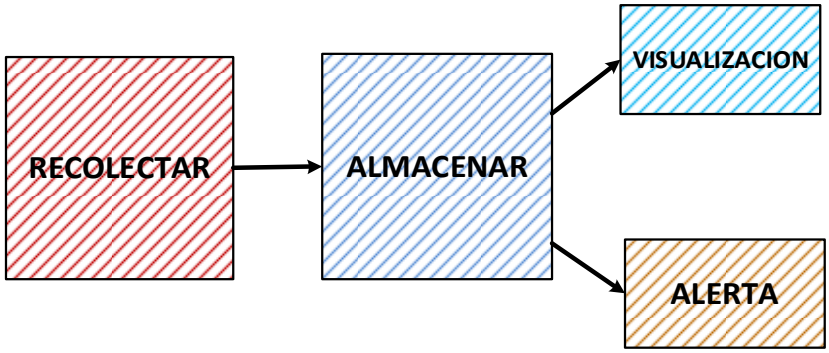
La visualización de los gráficos, son tan buenos en el momento de mostrar los datos tanto en computadores como también en medios móviles, su calidad y buena velocidad en el momento de dar a conocer la información son los atributos perfectos para tener la certeza de obtener informes y dashboards (tableros de control) de una forma sencilla y objetiva.

Grafana

Grafana una herramienta de visualización para indicadores de métricas y editores de gráficos, es muy concurrida por los que buscan una opción de herramientas gratuitas y obtener buenos resultados, es especialmente usada para monitorear infraestructura, aunque también se puede implementar en sectores productivos, una de las ventajas con la que cuenta Graphana es la facilidad y su cuadro de mandos, el cual permite mostrar varios tipos de métricas de graphite a través del navegador web [61]. Graphite es una herramienta de monitoreo lista para las empresas que funciona igual de bien en hardware barato o infraestructura en la nube [62]. También este servicio resulta atractivo por su

Compatibilidad con muchos mecanismos de autenticación para sus usuarios, dando así mayor posibilidad de uso, cumpliendo con niveles de seguridad y fácil mantenimiento. Graphana es una herramienta que está más orientada a monitorización de recursos tales como: Sistema, Tendencias, Programas en Ejecución, Aplicaciones y seguimientos de Log, por tanto constituye una herramienta importante al momento de control de infraestructura y permite anticipar los errores o problemas que lleguen a surgir. Es muy importante disponer de este tipo de recursos cuando se hace uso de ambientes críticos o entornos de Big Data. En la Figura 13, se encuentra los componentes de una herramienta de monitorización como Graphana.

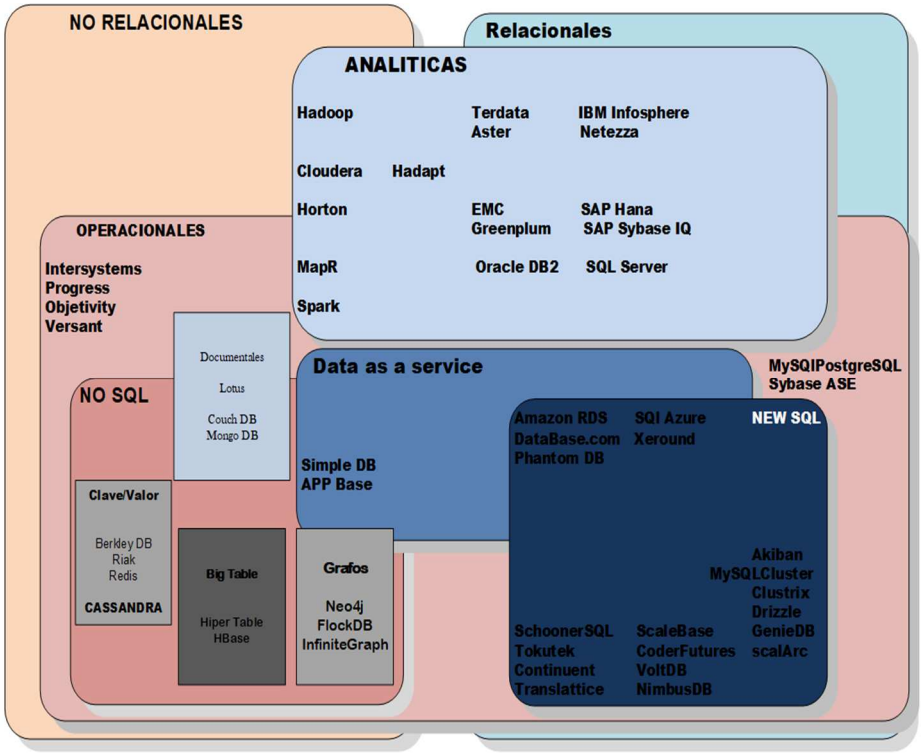
Figura13 Componentes de una Herramienta de Monitorización



Fuente Autores.

Finalmente se puede observar que un ecosistema de big data está compuesto como se observa en la figura 14, donde se integran las herramientas anteriormente descritas.

Figura 14. Componentes de un ecosistema para Big Data



Fuente Autores.

3.3.1 Modelos descriptivos

Para la comprensión de que son los modelos descriptivos, son aquellos que permiten un mejor entendimiento de los resultados que arroja de una manera global la investigación de un determinado proceso. Es decir que existen múltiples maneras de realizar investigaciones las cuales por ejemplo se podrían realizar por medio de la minería de datos es decir todos aquellos resultados que se arrojen en el momento de investigar dicho tema, al encontrar resultados se buscan de una manera descriptiva y critica de que me sirve y que no para poder resolver los interrogantes de los temas que se están investigando.

Con dicho modelo es posible obtener una solución, sin embargo, en este modelo solo se intenta describir la situación y no escoger una alternativa[63]. Ya que dicho modelo lo único que hace es recoger características que pueden hacer parte de los temas a los cuales se están investigando, pero esto no significa que va a darnos un resultado definitivo para la solución de las investigaciones o los temas que se estén tratando.

Los modelos descriptivos se diferencian de los modelos predictivos ya que los predictivos se centran en predecir el comportamiento de un cliente en particular[64], en cambio los descriptivos son utilizados para realizar unas

clasificaciones de los temas que puedan tener relación con lo que se está investigando. Es claro que los modelos descriptivos no brindan una solución total; “a lo que se quiere llegar” pero si nos puede dar una seria de características o temas que pueden dar pequeñas soluciones para así tener una solución globalizada por medio de pequeñas metas.

Para dar un poco más de claridad los modelos predictivos se pueden tener en cuenta en el momento en el cual se quiera segmentar una base de datos de sus clientes. Por medio de la descripción de los mismos

Es posible clasificarlos ya sea por su edad, genero, estrato etc. Solo basta con saber que se quiere y para que se quiere, por medio de la segmentación que se le realice a estos clientes con seguridad arrojara los resultados suficientes para garantizar la eficiencia y optimización en el recurso tiempo, todo depende de que se oferte o para que se necesita esta segmentación de clientes, con lo anterior se garantiza la precisión que se puede obtener con los clientes más propensos a lo que se busca gracias a que se extrajo la información de algo global a algo más preciso que con características esenciales se logra finalmente llegar a algo más específico para así lograr lo que se busca.

3.3.2 Modelos Predictivos

Cuando se hace uso de la palabra modelos predictivos en un entorno de análisis, se refiere a una representación de la realidad basada en un intento descriptivo de relacionar un conjunto de variables con otro[65]. Para cualquier tipo de análisis es necesario reunir una serie de características o similitudes que gracias a esto son los que se piensan analizar, es por esto que en los modelos predictivos se busca reunir una serie de características las cuales quieren llegar a realizar una conclusión a aquellos recursos que se piensan examinar, todo lo anterior basado en la clasificación de recursos o en este caso en particular en grandes volúmenes de datos o de información.

Para poder llegar a ser parte de un grupo de científicos especializados en modelos predictivos, hace falta estar muy bien preparado en la mayoría de ramas que tiene la carrera de ingeniería de sistemas; y además tener cierto manejo en otras carreras afines, ya que es vital tener conocimientos en áreas como, contabilidad, finanzas, marketing, tecnologías de la información, estructura de datos, algoritmos entre otros.

Todo esto y gracias a las tecnologías que hoy en día se manejan para realizar dichos estudios, y más puntualmente gracias a la creación del Big Data ha sido posible el volverse una rama mucho más acertada, con un aumento en la capacidad de almacenamiento, como también en la optimización de los tiempos en las búsquedas de los datos sobre los cuales se estén investigando o recolectando. Todo lo anterior gracias a una clave identificada, la cual es saber

de las demás dimensiones que hagan parte del entorno sobre el cual se está consultando, para dar un pequeño ejemplo se puede tomar una persona viajera, para identificar que modelos predictivos que se pueden obtener basta solo con saber los entornos determinados, como si es tierra fría, cálida, si les gusta hacer determinado deporte entre otras cosas. Con estos componentes es posible identificar que modelos predictivos tienen los viajeros. Haciendo uso de herramientas como el internet y los métodos especializados que se usan para dicho fin.

El anterior ejemplo es para lograr ser un poco más explícito en lo que consiste los modelos predictivos, el ejemplo, puede tomarse para un estudio que se haría en los lugres turísticos donde gracias a estas características se pueden concluir que tipo de turistas son, los que prefieren diferentes climas, saber qué tiempo del año es especial para cada tipo de clima, el número de turistas que se visitan cada mes del año y hasta que número de personas serían las que se deben de emplear para satisfacer las demandas de los turistas.

Para lograr llevar acabo cualquier clase de estudio que se tenga, por medio de los modelos predictivos, se ha llegado a conclusiones específicas, gracias a científicos experimentados, los cuales recomiendan llevar una serie de pasos o tareas que garantizan la obtención de respuestas rápidas, eficaces, y certeras para la toma de decisiones. Estas recomendaciones o tareas son:

- Proporciona un conocimiento profundo de sus datos.
- Busca el pequeño subconjunto de **variables** que son importantes para la predicción.
- El resultado es un producto extremadamente útil para los negocios, dentro el proceso de **toma de decisiones**[66].

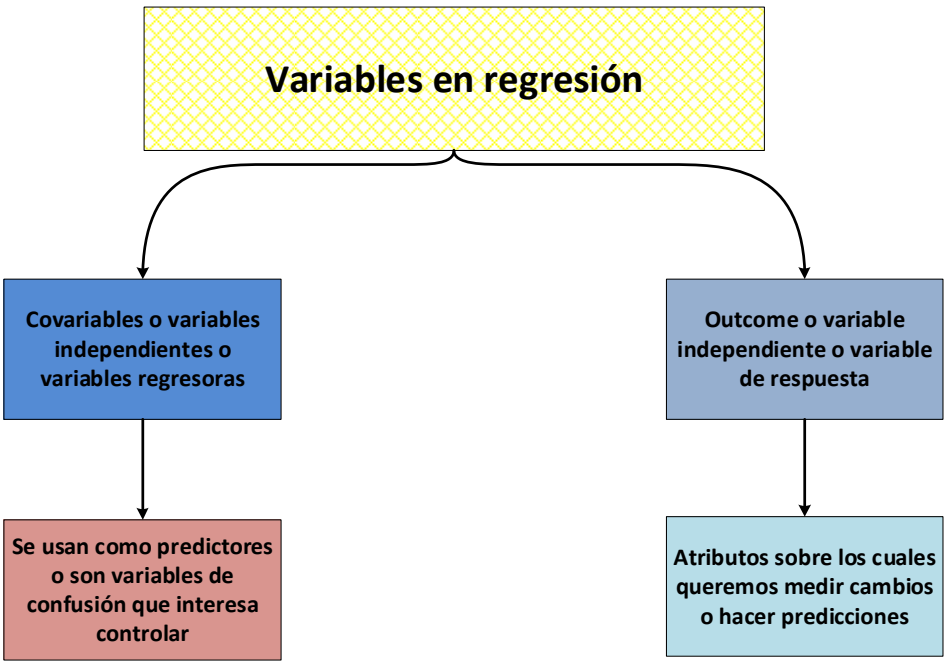
Gracias a la consulta realizada y teniendo en cuenta las anteriores recomendaciones se obtiene que de 4 estudios mínimo tres resultan ser muy efectivos. Con resultados excelentes. También este método es efectivo ya que logra alimentar las variables por medio de recolección de datos o información.

3.3.3 Técnicas de Regresión Modelos de decisión

Los modelos de regresión son los modelos más utilizados para realizar estimaciones, se emplea cuando existe relación entre dos variables [67] .Al tener el conjunto de variables, estas se modelan a través de una ecuación matemática, la cual busca dependiendo de la situación conectar los datos y las acciones, los cuales se pueden simular con una gran variedad de modelos, permitiendo extraer conclusiones acerca de la circunstancias actuales y eventos futuros[68] o dicho

de otra manera, la realización del análisis predictivo. Para realizar el análisis predictivo existen varias herramientas y técnicas estadísticas, que realizan diferentes tareas con resultados efectivos, dichas herramientas pueden ser gratuitas o hay otras que cobran por su uso. Una herramienta o técnica que se suele usar es la de la regresión lineal en la actualidad es una técnica efectiva y se podría decir que es una de las más usadas en estas clases de modelos de decisión, pero ¿en qué consiste o como se realiza esta técnica?; pues básicamente lo que hace la técnica es analizar una relación existente entre dos o más variables como se observa en la figura 15 donde una de estas variables depende de las demás pero hay otras variables que son totalmente independientes, gracias a la clasificación de dichas variables están permiten por medio de una función lineal de los parámetros permite generar una respuesta eficaz y efectiva.

Figura 15 Tipo de variables en regresión lineal



Fuente Autores.

Pero ya en palabras más castizas lo que se busca con esta técnica es básicamente que por medio de un paso atrás sea posible obtener respuestas futuras o proyecciones futuras, cabe indicar que el uso de dicha técnica no garantiza que el resultado es el más acertado y que gracias al uso de esta técnica se tendrá la solución a los problemas, lo que en realidad busca es dar posibles proyecciones para tratar de adelantarse a posibles problemas posteriores y así

dar una solución positiva y rápida pero no quiere decir que no tendrán otra clase de problemas.

3.3.4 Machine Learning

El Machine learning o aprendizaje automático se generó como una rama de las muchas que existen en la inteligencia artificial, y es la capacidad de aprender automáticamente por medio de los datos que se les va suministrando, gracias a esos datos el machine learning aprende a distinguir por medio de patrones soluciones a problemas de gran magnitud. Este utiliza una capacidad autómata que se ve reflejada en distintos motores de búsqueda, medicina, fraude bancario, reconocimiento de voz, robótica entre otros.

Esta tecnología al servicio de todos requiere una gran capacidad de almacenamiento y cálculo (proceso), pueda que hace unos años atrás esto hubiese sido un contratiempo, pero en la actualidad y gracias al cloud computing ya es posible la utilización de dicho servicio, esto porque ya el almacenamiento y los procesamientos no están limitados en las maquinas que estén destinadas para la realización de procedimientos en minería de datos, ahora con el cloud computing es posible crear unas estructuras escalables las cuales llegan a ser muy útiles para las necesidades que se tienen en los proyectos y todo esto con un menor costo ya que es posible almacenar información en la “nube”.

En la actualidad hay tres grandes empresas las cuales suministran herramientas basadas en el machine learning estas son; IBM Bluemix, BigML, Amazon ML, dichas empresas proveen el mejor servicio para las organizaciones que manejan grandes cantidades de información, y además son las que más apuestan a que por medio del machine learning las organizaciones tendrían un éxito y una mayor organización en la información que manejan las empresas acogidas con dicho servicio[69].

Este método utiliza algoritmos de árboles de decisión, pero no los tradicionales tales como ID3, C4.5, o CART, ya que estos algoritmos no pueden ser utilizados en grandes volúmenes de información, es por esto que machine learning utiliza el método llamado top-Down. Este tipo de árbol de decisión también tiene sus complicaciones, pero es el efectivo para manejar volúmenes amplios de información.

Se puede decir que este tipo de aprendizaje es una buena herramienta para el manejo de una alta cantidad de información, basada en algoritmos especiales los cuales buscan una optimización en tiempo para resolver problemas con la información, y además proporcionar una buena base de entrenamiento de maquina mejorando la eficiencia en el manejo de grandes datos.

3.3.5 Contenedor de aplicaciones

Un contenedor es simplemente un proceso para el sistema operativo, que internamente contiene la aplicación que se quiere ejecutar y todas sus dependencias. La aplicación contenida solamente tiene visibilidad sobre el sistema de ficheros virtual del contenedor y utiliza indirectamente el kernel del sistema operativo principal para ejecutarse[70]. Los contenedores de aplicaciones pueden ser maquinas físicas o también máquinas virtuales las cuales tienen como tarea tener un grupo de aplicaciones y mantenerlas en funcionamiento.

Un contenedor de aplicaciones tiene que cumplir con un estándar, cuya normatividad existente para las plataformas informáticas establecidas para su uso en cualquier parte del mundo, la función más especifica que tienen estos contenedores es poder plegarse y solucionar la problemática del espacio, un aliado para el buen funcionamiento de los contenedores son las máquinas virtuales, ya que se puede realizar un paralelismo entre ellos, los dos fueron creados específicamente para ejecutarse en grandes cantidades de información. Claro está que estos dos aliados tienen diferencias, la más resaltada o referenciada es que la máquina virtual tiene como necesidad la de contener todo el sistema operativo, mientras que el contenedor lo que hace es aprovechar el sistema operativo en el que se está ejecutando el mismo.

En el trabajo de implementación que hace parte de este trabajo de grado, se quiere es utilizar chef server como contenedor de las aplicaciones o los servicios que se tienen como objetivo de instalación, esto por medio de unas rutinas que nos permitirán aplicarlas en otras máquinas de una manera específica y rápida, ya después que se tengan instaladas dichas aplicaciones lo que se busca es que gracias a este contenedor se pueda realizar una tarea de servicio, seguimiento y a la vez mantenimiento.

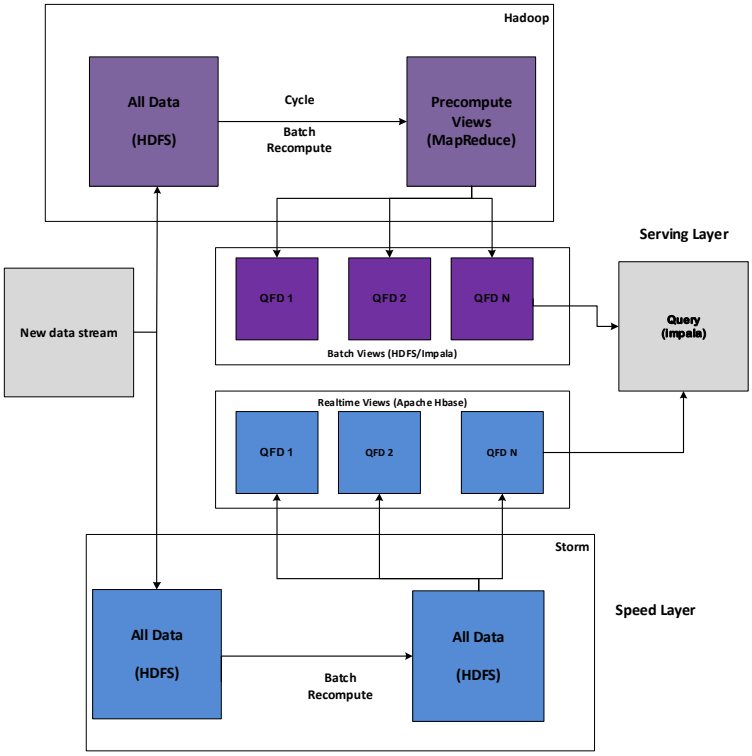
3.4 Arquitecturas de big data

3.4.1 Arquitectura lambda.

La arquitectura lambda está definida como un conjunto de elementos que pueden ser útiles en sistemas gigantescos de manejo de la información o lo conocido actualmente como big data, lo que la arquitectura lambda busca es solucionar las necesidades de los sistemas robustos de información y que a su vez sea tolerante a fallos, es decir que solucione las incidencias que se encuentren en los procesos, con el ánimo de tener claridad y entendimiento en las funciones y las diferencias que se tenga en cada capa de esta arquitectura. La cual está conformada por tres capas: (Batch Layer, Serving Layer, Speed Layer) que se

describen de forma general a continuación. En la primera capa se encuentran los procesos de almacenamiento y acceso a los datos. La segunda es la encargada de administrar la indexación y la exposición de vistas, las cuales pueden ser buscadas por medio de Query´s. Y por último en la tercera capa se observa que su función es garantizar la eficacia en tiempo real de los procesos para que todo lo anterior se efectué de una manera óptima, que permita obtener beneficios sobre los procesos que cada organización lleve a cabo. Para este fin es necesario adentrarse más al detalle de esta arquitectura la cual se observa en la figura16 donde se encuentra como se compone cada capa y de qué manera estas se relacionan, dándonos pie a la profundización de cada una de ellas.

Figura 16. Capas de la arquitectura lambda



Fuente Autores.

Batch Layer (Capa por Lotes) (Apache Hadoop)

Como se había indicado en la figura16 esta capa se encarga de los siguientes procesos:

- Almacenar en HDFS EBAl data set maestro que es inmutable y constantemente crece. Esto permite soportar operaciones de extracción e importación de información que luego será almacenada.

- Crear vistas arbitrarias desde este data set vía MapReduce (Hive, Pig,…). Esta computación se planifica y conforme llegan nuevos datos se agregan a las vistas en la siguiente iteración. Cada generación puede llevar horas[71].

Serving Layer (Capa de Servir) (Cloudera Impala)

Es la capa que se encarga de ordenar los datos recopilados de acuerdo a lo que sea requerido y a su vez exponerlo para que se pueda ver por medio de consultas, las cuales son realizadas por el usuario, para dar claridad se cita el siguiente ejemplo.

Para exponer las vistas con impala lo único a hacer es crear una tabla en el Metastore de HIVE que apunte a los ficheros HDFS y el usuario ya podrá consultar vía SQL. Lo anterior es lo que realiza un usuario por medio de Hive que es un servicio empleado en Big Data[72].

Speed Layer (Capa de velocidad) (Storm, Apache HBase)

Con las dos capas anteriores se puede tener una Arquitectura Big Data completa, aunque no se satisface los requisitos de Tiempo Real (Near Real Time) esto se debe a que MapReduce es un proceso Batch y puede llevarse horas en crear las vistas y propagarlas a la Serving Layer. Aquí aparece la Speed Layer. En esencia esta Capa hace lo mismo que la Batch Layer: ya que computa Vistas cuando llegan los datos. Al realizar este proceso se puede compensar alta latencia, ya que esta entrega actualizaciones de información haciendo uso de información recientemente actualizada en la fuente de datos creando vistas en tiempo real atreves de algoritmos incrementales[73].

Esta Capa usa un modelo incremental donde las vistas son secuenciales, también permite exponer las Vistas para que puedan ser consultadas con las Vistas Batch para conseguir el resultado completo. Se requiere tanto lectura como escritura random (Aleatorio), es recomendado HBase, que permite que Storm actualice continuamente las vistas en tiempo real y a su vez puede ser consultado por Impala para visualizarlas con las Vistas Batch.

3.4.2 Arquitectura kappa

La arquitectura kappa se obtiene de fusionar una capa de la arquitectura lambda, esto se puede observar según la figura17 donde se detallan sus componentes. La arquitectura Kappa nace de la iniciativa y de la motivación de evitar el mantenimiento de dos bases de datos separadas para las capas de lote y velocidad[74], que se pueden ver en la arquitectura lambda. Para lograr esto se

plantea como idea principal manejar el procesamiento de datos en tiempo real y el reprocesamiento continuo, haciendo uso de un procesamiento de flujo, lo que permite fusionar una capa y se obtiene la arquitectura kappa.

La estructura de Kappa está conformada por dos, la primera es la capa de procesamiento de flujo y la segunda se le llama la capa de servicio. A continuación se describen cada una de estas y de qué manera operan:

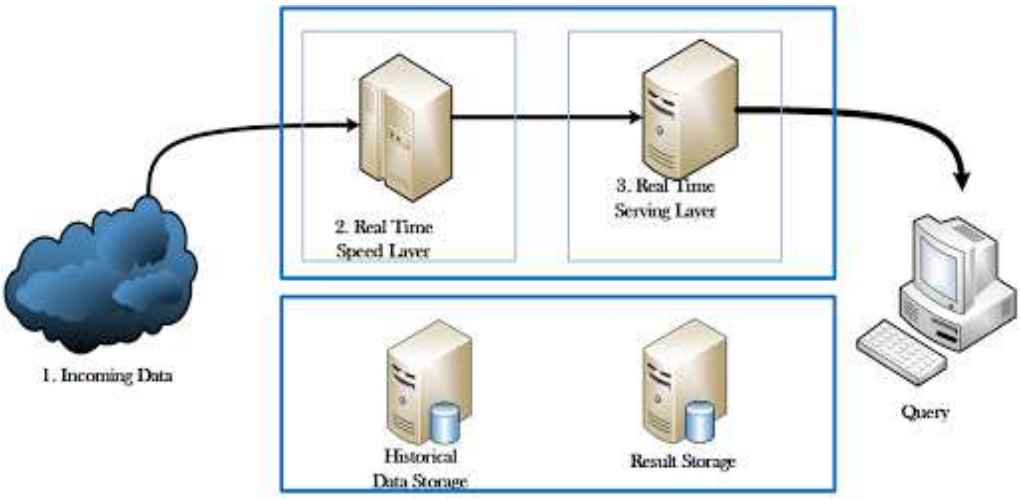
Capa de Procesamiento de Flujo

La capa de procesamiento de Flujo, es la que ejecuta los trabajos de procesamiento de la secuencia, normalmente esta ejecuta un solo trabajo de procesamiento de flujo, el cual permite habilitar el procesamiento de datos en tiempo real. Este proceso solo se lleva a cabo cuando es necesario efectuar algún cambio en el código del trabajo de procesamiento. Teniendo de esta manera en paralelo otro trabajo de procesamiento de flujo para lograr esto.

Capa de Servicio

La capa de servicio es la encargada de entregar los resultados, al igual que en la arquitectura lambda, esta capa es la encargada de permitir la consulta de los mismos por medio de Query's. En la figura17 se visualiza la manera en la que se conforma y se implementa la arquitectura kappa.

Figura 17. Arquitectura Kappa



Fuente Autores.

Finalmente lo anterior se resume en la tabla 5, donde se encuentran las principales características, de las arquitecturas mencionadas Kappa y Lambda.

Paradigma de procesamiento	Lambda trasmisión + lote	Kappa trasmisión
Re-procesamiento del paradigma	Cada ciclo de lote	Sólo cuando el código cambia
Consumo de recursos	función = consulta	Algoritmos incrementales, corriendo en deltas
Confiabilidad	Lote es confiable, la transmisión es aproximada	transmisión con consistencia (exactamente una vez)

Tabla 5 comparación arquitecturas lambda y kappa fuente autores.

3.4.3 Arquitectura Zeta

Zeta Architecture es una construcción arquitectónica empresarial de alto nivel que permite procesos empresariales simplificados y define una forma escalable para aumentar la velocidad de integración de datos en el negocio[75], es considerada la próxima generación, que está enfocada para ser trabajada en arquitecturas empresariales, trabajando por medio de archivos distribuidos y dando un almacenamiento en tiempo real utilizando un motor de ejecución y a su vez un motor de contención buscando una arquitectura de solución por medio de un óptimo procesamiento logrando gestionar recursos dinámicos y globales.

Esta arquitectura está compuesta por 7 componentes interconectados de alto nivel, En la figura18 se observa la manera en la que se conforma y se implementa la arquitectura zeta.

Figura 18. Arquitectura zeta

Fuente Autores.

Sistema de archivos distribuido: haciendo uso de un sistema de archivos distribuidos compartido, todas las aplicaciones pueden leer y escribir de manera centralizada, lo que permite simplificar la arquitectura.

Almacenamiento en tiempo real: Consiste en la manera de respaldar las aplicaciones empresariales haciendo uso de bases de datos en tiempo real.

Modelo de Computación (Motor de Ejecución): Dentro de la arquitectura empresarial se deben satisfacer necesidades y requerimientos, lo que hace necesario tener motores y modelos potencialmente diferentes para satisfaces las necesidades del negocio.

Contenedores de Despliegue: Es importante contar con un enfoque estandarizado para implementar software.

Arquitectura de la solución: Se enfoca en la solución de un problema específico, esto quiere decir que puede haber una o más aplicaciones diseñadas para dar la solución total al problema planteado.

Aplicaciones Empresariales: Simplifica aplicaciones mediante la entrega de los componentes necesarios, para realizar todos los objetivos planteados en la arquitectura.

Gestión Dinámica de los Recursos: Permite la asignación dinámica de recursos, con el objetivo de que la organización se adapte de manera eficaz a cualquier tarea[76].

3.5 Marco Conceptual

- **Almacenamiento distribuido:** Es una colección de datos que pertenecen lógicamente a un sólo sistema, pero se encuentra físicamente esparcido en varios "sitios" de la red. Un sistema de base de datos distribuidos se compone de un conjunto de sitios, conectados entre sí mediante algún tipo de red de comunicaciones[77].

- **Base de datos:** Colección compartida de datos relacionados desde el punto de vista lógico, Bases de datos junto con una descripción de esos datos (metadatos), diseñada para satisfacer las necesidades de información de una organización[78].

- **Clúster:** Es un grupo de computadoras que están interconectadas y funcionan como una sola unidad de procesamiento[79].

- **Computación en la nube – (cloud computing):** Es un concepto tecnológico (buzzwords) cada vez más utilizado. Las organizaciones ven en esta tecnología la solución a muchos problemas, económicos o de infraestructuras tecnológicas [80].

- **Consulta:** Búsqueda de información en una fuente de documentación para aprender una cosa o para aclarar una duda.

- **ERP:** (ERP), del inglés Enterprise Resource Planning, es lo que en español conocemos como Software de gestión integrada, y se define como un grupo de módulos conectados a una única base de datos. El ERP es un paquete de software que permite administrar todos los procesos operativos de una empresa, integrando varias funciones de gestión en un único sistema; en otras palabras, representa la “columna vertebral” de una empresa[81].

- **Exabyte:** Unidad de medida informática simbolizada como "EB". Un exabyte equivale a 1024 petabytes. El orden de las unidades de almacenamiento es el siguiente: Byte, Kilobyte, Megabyte, Gigabyte, Terabyte, Petabyte, Exabyte, Zettabyte, YottaBytee, Brontobyte. La unidad de medida de la medida Exabyte es tan grande, que no se utiliza para medir la capacidad de los dispositivos de almacenamiento de datos. Incluso la capacidad de almacenamiento de los mayores nubecentros de almacenamiento se mide en petabytes, que es una fracción de un exabyte[82].

- **Grafos:** Un grafo es un conjunto, no vacío, de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas (edges en inglés) que pueden ser orientados o no. Típicamente, un grafo se representa

mediante una serie de puntos (los vértices) conectados por líneas (las aristas) [83].

- **Integración de datos:** es una combinación de procesos técnicos y de negocio que se utilizan para combinar datos de diferentes fuentes para convertirlos en información útil y valiosa [84].
- **Inteligencia de negocio**: se refiere al proceso de convertir datos en conocimiento y conocimiento en acciones para crear la ventaja competitiva del negocio[85].
- **Internet de las cosas (IOT):** Internet de las Cosas (I o T, por sus siglas en inglés) se refiere a una red de objetos cotidianos interconectados, singularmente identificables y con representaciones virtuales; en una estructura similar al internet[86].
- **IPV4:** Es un protocolo de Internet, el sistema de identificación que ésta utiliza para enviar información entre dispositivos – este sistema asigna una serie de cuatro números – cada uno de los cuales está comprendido entre 0 y 255 – a cada dispositivo. IPv4 sólo permite aproximadamente 4.000 millones de direcciones, una cifra baja en comparación con lo que necesita Internet[87].
- **Metadatos:** Desde un punto de vista informático los metadatos se consideran un conjunto de reglas incluidas en las aplicaciones de manejo de información geográfica que describen la estructura interna de los esquemas de datos[88].
- **Paradigma:** El concepto de paradigma se utiliza en la vida cotidiana como sinónimo de “ejemplo” o para hacer referencia en caso de algo que se toma como “modelo digno de seguir”. En principio se tenía en cuenta en el campo, tema, ámbito, entre otros..., gramatical (para definir su uso en un cierto contexto) y se valoraba desde la retórica (para hacer mención a una parábola o fábula). A partir de la década de 1960, los alcances de la noción se ampliaron y paradigma comenzó a ser un término común en el vocabulario científico y en expresiones etimológicas cuando se hacía necesario hablar de modelos de conocimiento aceptados por las comunidades científicas[89].
- **Proceso paralelo:** Un proceso paralelo es aquel que se realiza al mismo tiempo que otro, siendo ejecutados ambos de modo simultáneo. Cuando hablamos de procesos paralelos en un ordenador, nos referimos a aquellos procesos que se ejecutan y/o procesan a la vez, ante poniéndose a los procesos lineales o secuenciales, que serán ejecutados de uno en uno[90].

- **Zettabyte:** Un zettabyte es un trillón de gigabytes. Para ser un poco más claros y usando términos más comunes, mil megabytes (MB) equivalen a 1 gigabyte (GB); mil gigabytes equivalen a 1 terabyte (TB); mil terabytes equivalen a un petabyte (PT), mil petabytes equivalen a 1 exabyte (EB) y mil exabytes (EB) equivalen a 1 zettabyte (ZB) [91].

4. METODOLOGÍA.

4.1 Servicios y tipos de servicio

En primera instancia se realiza la definición de los servicios a implementar, para llevar a cabo la selección se realiza un comparativo entre 16 herramientas, de las cuales se seleccionan 8 a implementar teniendo en cuenta parámetros se usabilidad, facilidad de implementación y siempre manteniendo el enfoque en herramientas open source.

4.2 Descripción General

Para desarrollar este proyecto se tendrán en cuenta 4 fases que se enumeran a continuación:

- Investigación y análisis de herramientas existentes.
- Diseño del modelo a implementar.
- Implementación de la herramienta seleccionada y Construcción de manuales de usuario y configuración.
- Pruebas y Resultado a la herramienta implementada.

4.3 Fase 1: Investigación y análisis de herramientas existentes.

- Se realizará el análisis a herramientas existentes opensource que se enfoquen en autogestión y autoconfiguración de servicios.
- Se evaluará cada herramienta seleccionando las más adecuada de acuerdo a sus características.
- La fase finaliza con la selección de la herramienta a implementar la cual constara de 8 servicios orientados a proyectos de big data.

4.4 Fase 2: Diseño de modelo de infraestructura a implementar

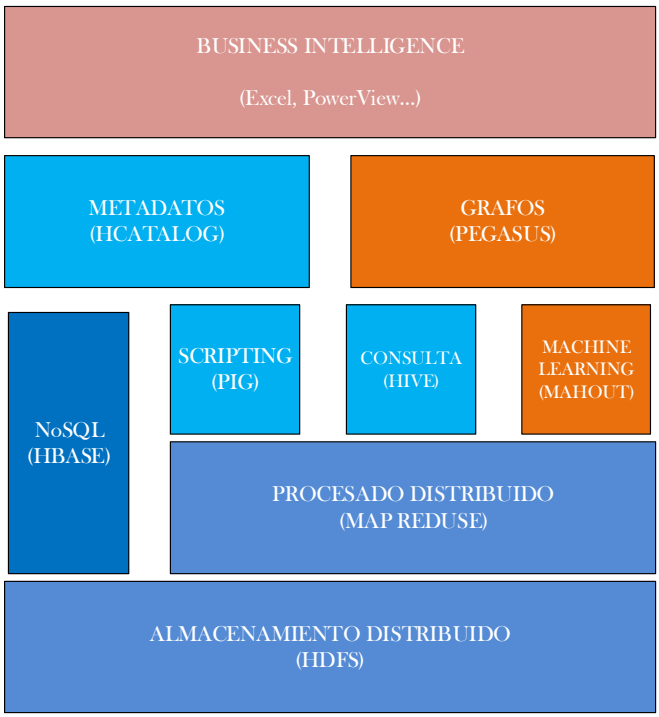
Luego de tener definidos los servicios y la herramienta a implementar se realizar el prototipo del montaje a realizar.

El diseño a implementar se basará en los siguientes componentes:

- Indexación de datos
- Almacenamiento de datos
- Procesamiento de datos
- Visualización de datos

El diseño propuesto se basa en figura 19 donde se identifican los principales componentes de un ambiente orientado a big data.

Figura 19. Capas de servicios en proyectos orientados a big data



Fuente Autores

4.5 Fase 3: Implementación de la herramienta seleccionada y construcción de manuales de usuario y configuración.

Para esta fase se realiza el despliegue la herramienta de autogestión y auto implementación en el clúster de la universidad. Posteriormente se generarán las rutinas para ser aplicadas en los 20 equipos ubicados en los laboratorios de la universidad.
Paralelamente se generarán los manuales de instalación y configuración de cada uno de los servicios implementados.

4.6 Fase 4: Pruebas y resultado a la herramienta implementada.

Posterior a la implementación de la herramienta se generarán pruebas de funcionamiento de cada uno de los servicios implementados enfocado en la optimización de tiempos y recursos al ser aplicados. Finalmente se realizará informe de resultados a las pruebas efectuadas tomando como indicador los tiempos de optimización de cada servicio.

5. DESARROLLO DEL PROYECTO

5.1 SEGUIMIENTO DE ACTIVIDADES

Se sabe que los proyectos orientados a Big Data, son desarrollados para obtener grandes resultados en temas que exigen capacidades de almacenamiento, procesamiento, distribución y visualización de gran cantidad de información en las organizaciones. Es por esto, que el objetivo principal a trazar en este proyecto es la Implementación de una herramienta open source de autogestión y autoconfiguración para el levantamiento de servicios orientados a este tipo de proyectos. Para esto, en primera instancia se desarrolla la investigación y clasificación entre 16 servicios gratuitos, de los cuales se seleccionarán 8 servicios; teniendo en cuenta parámetros de evaluación como usabilidad, eficiencia y eficacia. Una vez estos son seleccionados se implementan a partir del diseño de la arquitectura propuesta, basándose en arquitecturas existentes como lambda, zetta, y kappa.

Posterior a esto se instalan 8 servicios seleccionados en la etapa de investigación, los cuales cumplen con las características que deben tener los ambientes orientados al manejo masivo de información, para este proyecto se realiza el montaje de los siguientes servicios:

- Casandra
- Graphana
- Hadoop
- Hive
- Mongo DB
- OpenMPI
- Spark
- Tableau

Luego de realizar la instalación de cada uno de estos servicios de forma tradicional, se evalúa el tiempo y la dificultad que consume cada montaje para posteriormente integrar a Chef Server, que es la herramienta seleccionada que permite la autogestión y autoconfiguración de los servicios elegidos. Para llevar a cabo esto, es necesario desarrollar rutinas que permitan que estos servicios se auto instalen, de manera tal que al finalizar el proyecto se comparan los resultados obtenidos, entre las instalaciones tradicionales y las realizadas desde Chef Server; midiendo el nivel de optimización que esta genera para este tipo de

implementaciones orientadas a ecosistemas de Big Data. Se realizarán pruebas donde se demuestre el nivel de optimización presentado al implementar estos servicios de manera masiva, a través de Chef Server.

Finalmente se generan conclusiones y se entregan los archivos de apoyo para la replicación de este proyecto, los cuales consisten en la entrega de los siguientes anexos:

- Manual de Instalación de cada uno de los servicios implementados
- Manual de instalación de Chef Server
- Manual de Configuración de Chef Server
- Manual de usuario para Chef Server
- Pruebas de Laboratorio
- Script de Rutinas Desarrolladas Para la autogestión de servicios

Al finalizar el proyecto se entregarán las lecciones aprendidas y las recomendaciones para futuros proyectos enfocados en el estudio de Big Data.

Como se sabe, existe en la actualidad una variedad de servicios gratuitos los cuales pueden ser útiles, y se pueden aplicar en los distintos proyectos que realizan las organizaciones, tanto locales como globales, siendo organizaciones estatales, privadas o de investigación. Para este proyecto el propósito es poder evitar que en las organizaciones se malgaste tiempo en llevar a cabo una investigación tratando de encontrar cuales de estos servicios podrían ser útiles para llevar acabo de una manera organizada y rápida los objetivos de dichos procesos. Logrando así una optimización en el tiempo y otorgando el invertir este tiempo en otras tareas específicas, dando a su vez eficiencia en los proyectos, ya que gracias a nuestra herramienta se tendrán servicios específicos para todo lo que con lleva sacar adelante un proyecto basado en big data.

Todo lo anterior buscando la mayor eficiencia posible, obtenido también velocidad y eficacia para los proyectos futuros que se tengan en las organizaciones, es por eso que la investigación nos arrojara como resultado la garantía de que estos 8 servicios, estarán en la capacidad de obtener información específica y certera para llevar a cabo una tarea que satisfaga las necesidades que los proyectos, mostrando y arrojando buenos resultados para lograr los objetivos del mismo. Todo gracias a que dichos servicios escogidos son en la actualidad los más usados, eficientes, eficaces y porque no decirlo óptimos que se encuentran en la web a la mano y de manera gratuita, obteniendo como conclusión una herramienta la cual busca armar un gran equipo de trabajo totalmente organizado y rápido en la solución de las necesidades existentes. Para certificar que nuestra herramienta será efectiva y garantice la buena funcionalidad.

5.2 DESARROLLO DE ACTIVIDADES

Con base en la metodología planteada, se llevara a cabo el desarrollo del proyecto, el cual inicia con la etapa de investigación y finaliza con la implementación de las herramientas seleccionadas, para llegar a este objetivo se han desarrollado las siguientes actividades que permiten el buen término del mismo.

Investigación de las herramientas

El estudio de las herramientas consistió en el levantamiento del estado del arte acerca de herramientas usadas en ambientes Orientados a Big Data.

Una vez realizado el estado del arte, se procede a realizar la evaluación de cada una de las herramientas, teniendo en cuenta parámetros significativos, los cuales fueron consolidados como resultado de la investigación realizada. De esta manera se construyen matrices de evaluación, las cuales se presentan a continuación:

Matrices de evaluación de herramientas

El objetivo de las matrices desarrolladas, es elegir las herramientas a implementar, de acuerdo a parámetros que se eligieron durante el desarrollo de la investigación, para este caso se decidió clasificar las matrices de acuerdo a la taxonomía del Big Data, evaluando las herramientas de cada uno de los reinos que la compone, clasificándolas de la siguiente manera:

- Herramientas de Recolección
- Herramientas de Almacenamiento
- Herramientas de Procesamiento
- Herramientas de Visualización

De acuerdo a lo anterior a continuación se presentan las matrices desarrolladas para la evaluación de las herramientas.

Herramientas de Recolección

Dentro de esta matriz, se hace énfasis en cuanto a las herramientas que permiten la captura de datos que pertenecen de acuerdo a la taxonomía propuesta al primer reino del ecosistema de Big Data. Para evaluar estas herramientas se tienen en cuenta los siguientes parámetros:

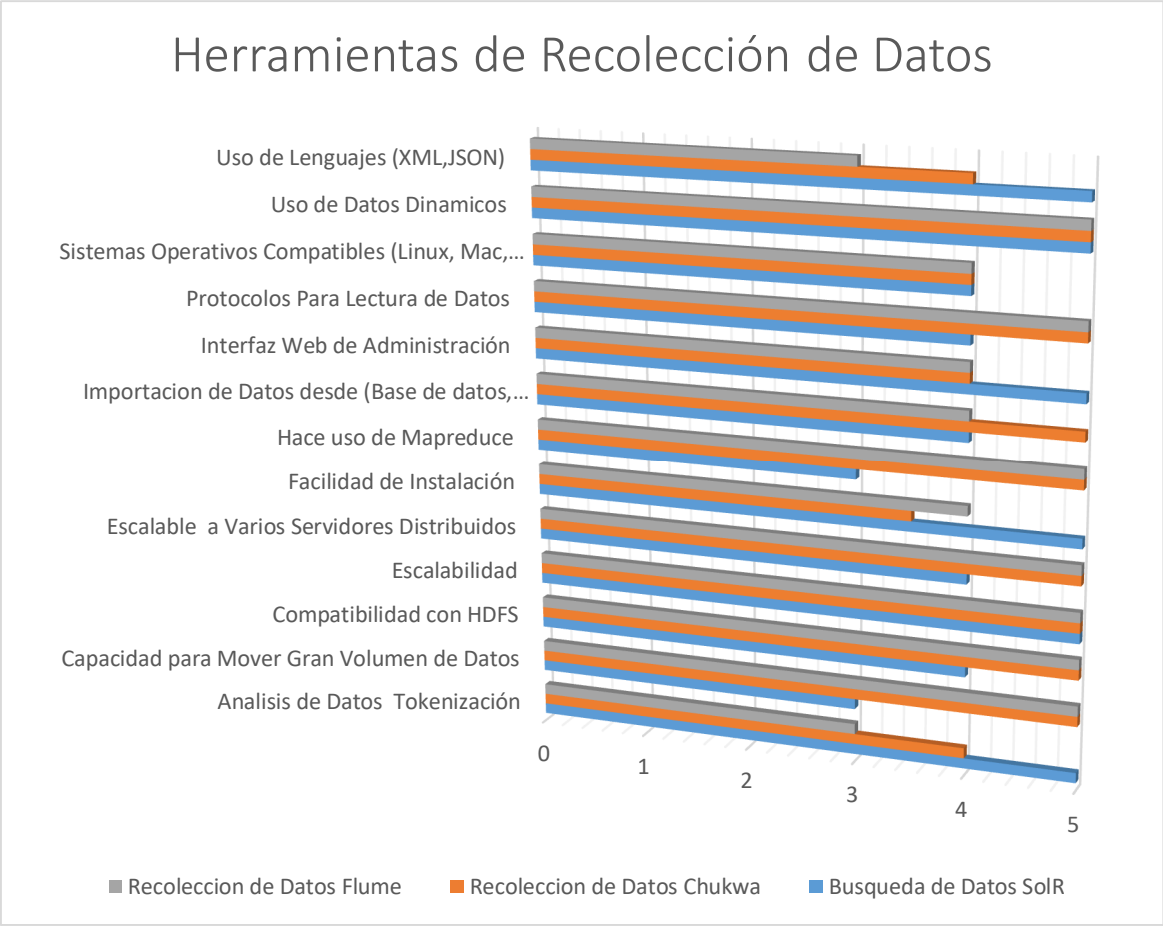
- Análisis de datos tokenización
- Capacidad para mover gran volumen de datos
- Compatibilidad con hdfs
- Escalabilidad
- Escalable a varios servidores distribuidos
- Facilidad de instalación

- Uso de Map Reduce
- Importación de datos de diversas fuentes
- Interfaz web de administración
- Protocolos para lectura de datos
- Sistemas operativos compatibles
- Uso de datos dinámicos
- Uso del lenguaje XML, JSON

Recolección de Datos Clasificación de Herramientas de Recolección de Datos			
	Búsqueda de Datos	*Recolección de Datos*	
Atributos	**SolR**	**Chukwa**	**Flume**
Análisis de Datos Tokenización	**5**	**4**	**3**
Capacidad para Mover Gran Volumen de Datos	**3**	**5**	**5**
Compatibilidad con HDFS	**4**	**5**	**5**
Escalabilidad	**5**	**5**	**5**
Escalable a Varios Servidores Distribuidos	**4**	**5**	**5**
Facilidad de Instalación	**5**	**3,5**	**4**
Hace uso de MapReduce	**3**	**5**	**5**
Importación de Datos desde (Base de datos, Mail, Archivos de texto Enriquecido)	**4**	**5**	**4**
Interfaz Web de Administración	**5**	**4**	**4**
Protocolos Para Lectura de Datos	**4**	**5**	**5**
Sistemas Operativos Compatibles (Linux, Mac, Windows)	**4**	**4**	**4**
Uso de Datos Dinámicos	**5**	**5**	**5**
Uso de Lenguajes (XML,JSON)	**5**	**4**	**3**
TOTALES	**56**	**59,5**	**57**

Tabla 6 Matriz de Evaluación para Herramientas de Recolección de datos fuente autores.

Figura 20 Resultado de Herramientas de Recolección de datos



Fuente Autores

Herramientas de Procesamiento de datos

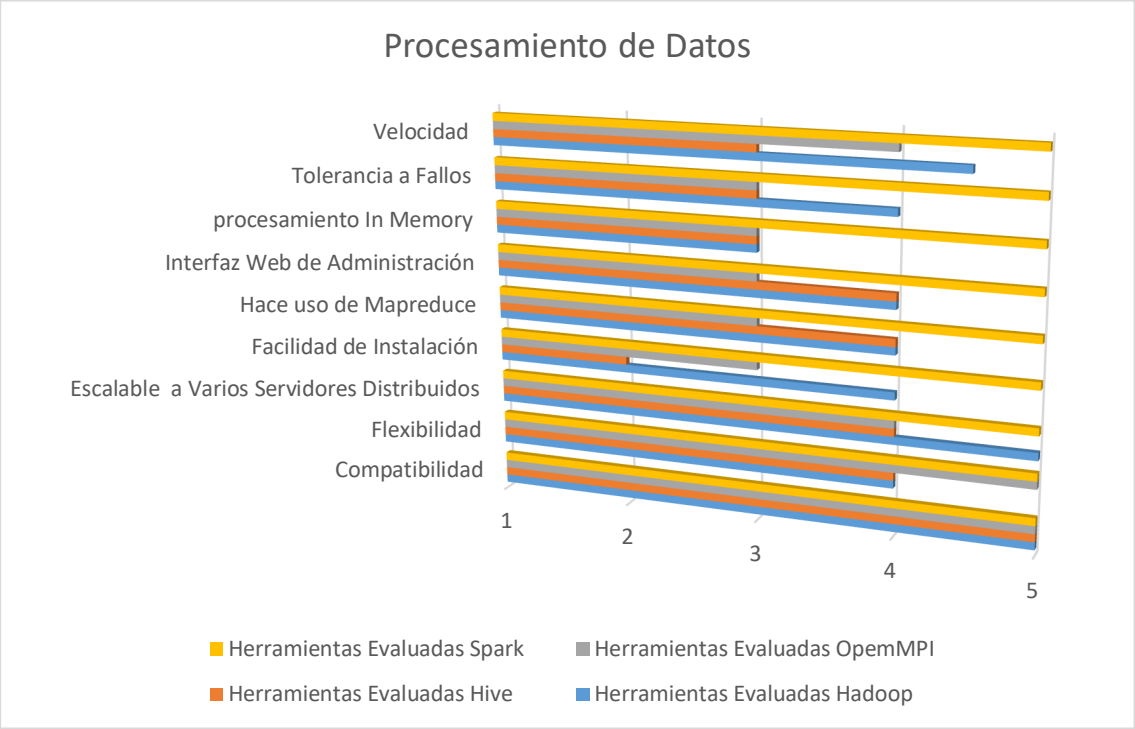
Dentro de esta matriz, se realiza énfasis en cuanto a las herramientas del procesamiento de datos, teniendo en cuenta lo siguientes parámetros:

- Compatibilidad
- Flexibilidad
- Escalable a varios servidores distribuidos
- Facilidad de instalación
- Uso de Map Reduce
- Interfaz Web de administración
- Procesamiento in memory
- Tolerancia a fallas
- Velocidad

Tabla 7 Matriz de Evaluación para Herramientas de Procesamiento fuente autores.

Procesamiento de Datos Clasificación de Herramientas de Procesamiento de Datos				
	Herramientas Evaluadas			
Atributos	**Hadoop**	**Hive**	**OpenMPI**	**Spark**
Compatibilidad	**5**	**5**	**5**	**5**
Flexibilidad	**4**	**4**	**5**	**5**
Escalable a Varios Servidores Distribuidos	**5**	**4**	**4**	**5**
Facilidad de Instalación	**4**	**2**	**3**	**5**
Hace uso de Map reduce	**4**	**4**	**3**	**5**
Interfaz Web de Administración	**4**	**4**	**3**	**5**
procesamiento In Memory	**3**	**3**	**3**	**5**
Tolerancia a Fallos	**4**	**3**	**3**	**5**
Velocidad	**4,5**	**3**	**4**	**5**
TOTALES	**37,5**	**32**	**33**	**45**

Figura 21 Resultado de Herramientas de Procesamiento



Fuente Autores

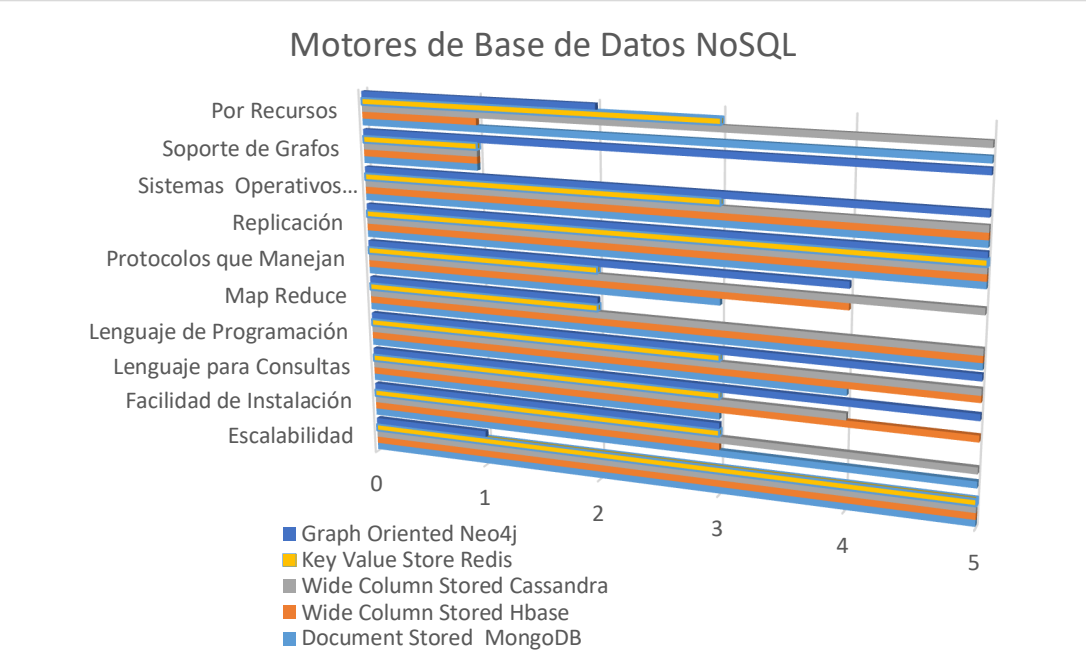
Con base en estos parámetros y para el caso a implementar, se tienen en cuenta los pilares del Big Data (Velocidad, Volumen, Variedad) y se le asignó un puntaje

a cada parámetro en un rango de 1-5, siendo 1 el menos importante y 5 el más importante, arrojando los siguientes resultados.

Tabla 8 Matriz de Evaluación para Herramientas de Almacenamiento fuente autores.

MOTORES DE BASE DE DATOS NoSQL Clasificación de Base de Datos					
Atributos	*Document Stored*	*Wide Column Stored*		*Key Value Store*	*Graph Oriented*
	MongoDB	**Hbase**	**Cassandra**	**Redis**	**Neo4j**
Escalabilidad	**5**	**5**	**5**	**5**	**1**
Facilidad de Instalación	**5**	**3**	**5**	**3**	**3**
Lenguaje para Consultas	**3**	**5**	**4**	**3**	**5**
Lenguaje de Programación	**4**	**5**	**5**	**3**	**5**
Emplean Map Reduce	**5**	**5**	**5**	**2**	**2**
Protocolos de Comunicación	**3**	**4**	**5**	**2**	**4**
Replicación	**5**	**5**	**5**	**5**	**5**
Sistemas Operativos Compatibles	**5**	**5**	**5**	**3**	**5**
Soporte de Grafos	**1**	**1**	**1**	**1**	**5**
Por Exigencia de Recursos	**5**	**1**	**5**	**3**	**2**
TOTALES	**41**	**39**	**45**	**30**	**37**

Figura 22 Resultado de Herramientas de Almacenamiento



Fuente Autores

Herramientas de Almacenamiento

Dentro de esta matriz, se realiza énfasis en cuanto a las herramientas de visualización, teniendo en cuenta lo siguientes parámetros:

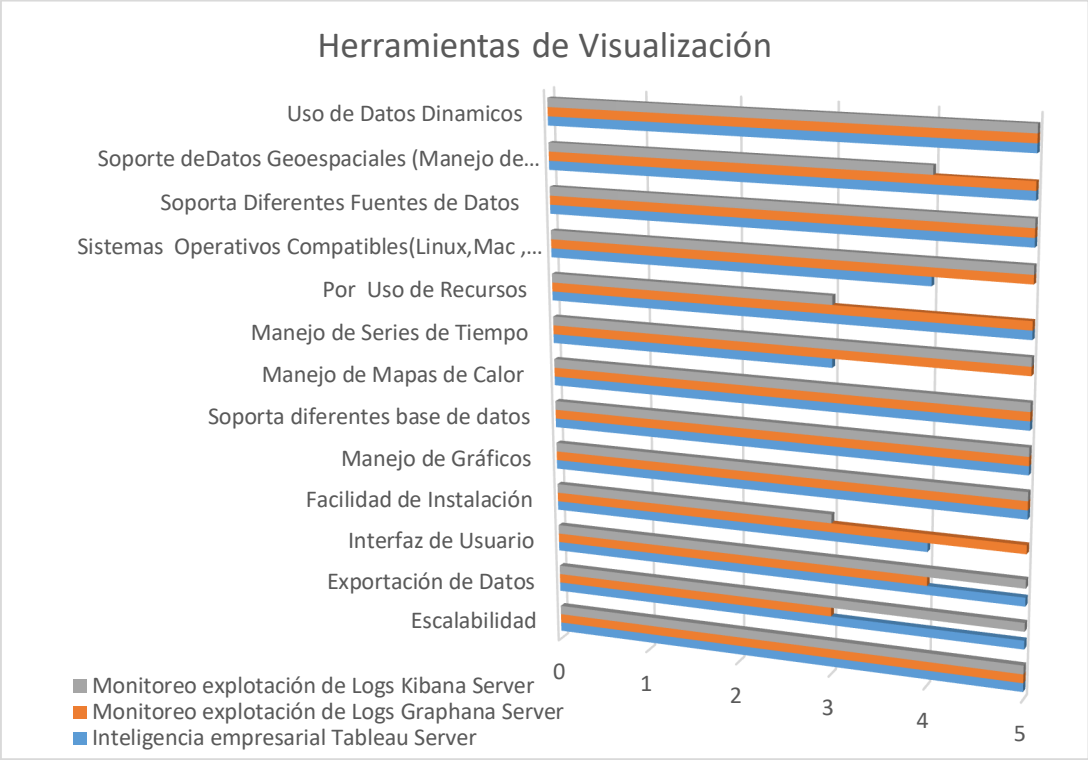
- Escalabilidad
- Exportación de Datos
- Facilidad de instalación
- Interfaz de Usuario
- Manejo de Gráficos
- Manejo de Mapas de Calor
- Manejo de Series de Tiempo
- Por Uso de Recursos
- Sistemas Operativos Compatibles
- Soporta diferentes Fuentes de Datos
- Soporta Datos Geoespaciales
- Uso de Datos Dinámicos

Con base en estos parámetros y para el caso a implementar, se tienen en cuenta los pilares del Big Data (Velocidad, Volumen, Variedad) y se le asignó un puntaje a cada parámetro en un rango de 1-5, siendo 1 el menos importante y 5 el más importante, arrojando los siguientes resultados.

Tabla 7 Matriz de Evaluación para Herramientas de Visualización fuente autores.

HERRAMIENTAS DE VISUALIZACION Clasificación de Herramientas de Visualización			
	Inteligencia empresarial	*Monitoreo explotación de Logs*	
Atributos	**Tableau Server**	**Graphana Server**	**Kibana Server**
Escalabilidad	**5**	**5**	**5**
Exportación de Datos	**5**	**3**	**5**
Interfaz de Usuario	**5**	**4**	**5**
Facilidad de Instalación	**4**	**5**	**3**
Manejo de Gráficos	**5**	**5**	**5**
Soporta diferentes base de datos	**5**	**5**	**5**
Manejo de Mapas de Calor	**5**	**5**	**5**
Manejo de Series de Tiempo	**3**	**5**	**5**
Por Uso de Recursos	**5**	**5**	**3**
Sistemas Operativos Compatibles(Linux, Mac , Windows)	**4**	**5**	**5**
Soporta Diferentes Fuentes de Datos	**5**	**5**	**5**
Soporte de Datos Geoespaciales (Manejo de mapas, coordenadas)	**5**	**5**	**4**
Uso de Datos Dinámicos	**5**	**5**	**5**
TOTALES	**61**	**62**	**60**

Figura 23 Resultado de Herramientas de Visualización



Fuente Autores

5.3 RESULTADO DE ACTIVIDADES

Al realizar la investigación, se obtiene como resultado las siguientes herramientas a implementar.

RESULTADOS DE SERVICIOS DE BIG DATA SERVICIOS ESCOJIDOS PARA INSTALACION POR MEDIO DE HARREMIENTA DE AUTO GESTION
Cassandra
Graphana
Hadoop
Flume
MongoDB
Open MPI
Spark
Tableau

Tabla 9 Resultados para los servicios a implementar fuente autores

Los resultados de este trabajo, no solo se enfocan en la evidencia que se obtiene al comparar como los servicios implementados, a través de chef server permiten optimizar los tiempos en la implementaciones de los mismos. También al ser un trabajo de investigación es importante resaltar los conocimientos y la curva de aprendizaje que se obtuvo durante el desarrollo de este proyecto. De esta manera se obtuvieron los siguientes resultados:

- La curva de aprendizaje adquirida durante el desarrollo del proyecto y necesaria para la instalación de cada una de las herramientas.

- La optimización de tiempos y recursos que usualmente consumen configuración de cada una de las herramientas, que componen un ambiente de Big Data.

- Se establece un comparativo entre la instalación e implementación de cada uno los servicios seleccionados, haciendo uso de la herramienta implementada.

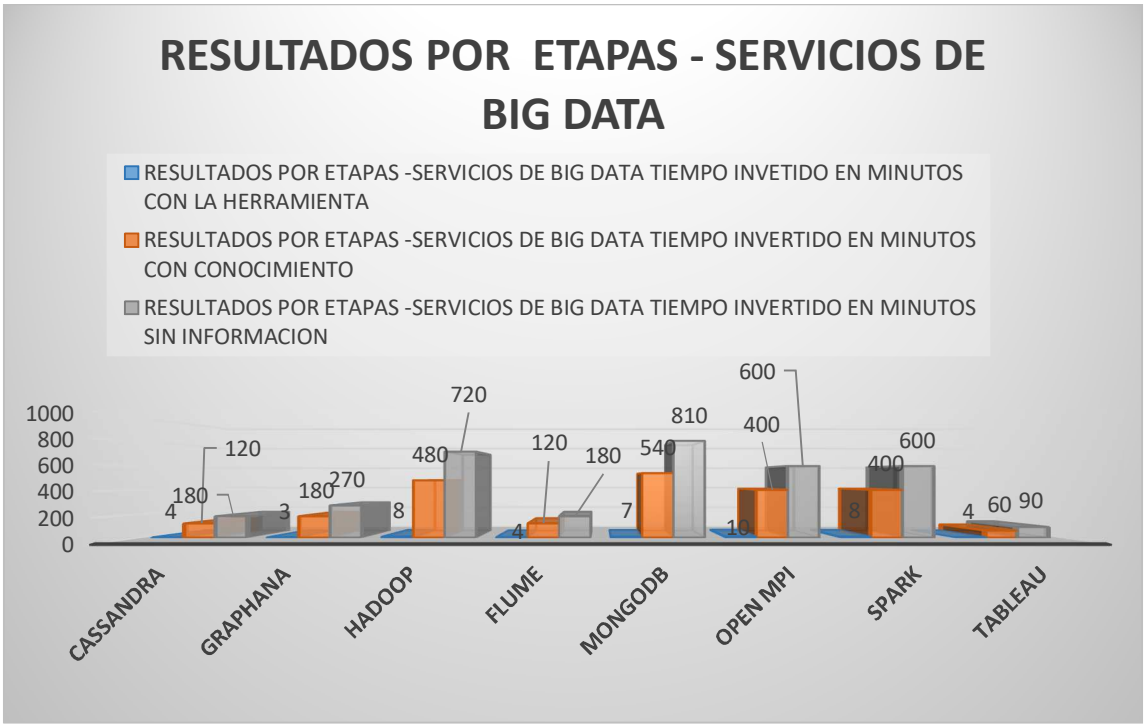
- Se realizan manuales de instalación y configuración de cada una de las herramientas, para la replicación de este trabajo desde un aspecto práctico y como aporte para investigaciones futuras.

Estos parámetros permiten determinar que las herramientas de autogestión y autoconfiguración, facilitan las escalabilidad y mantenimiento de los ecosistemas de big data y constituyen una forma de optimizar tiempos y recursos tecnológicos claves en las áreas de tecnologías de las organizaciones. A continuación se encuentran la recopilación de los resultados luego de realizar este proyecto.

RESULTADOS POR ETAPAS -SERVICIOS DE BIG DATA			
SERVICIOS	TIEMPO INVERTIDO EN MINUTOS CON LA HERRAMIENTA	TIEMPO INVERTIDO EN MINUTOS CON CONOCIMIENTO	TIEMPO INVERTIDO EN MINUTOS SIN INFORMACION
Cassandra	4	120	180
Graphana	3	180	270
Hadoop	8	480	720
Flume	4	120	180
MongoDB	7	540	810
Open MPI	10	400	600
Spark	8	400	600
Tableau	4	60	90

Tabla 10 Toma de tiempos, implementación de herramientas Fuente autores

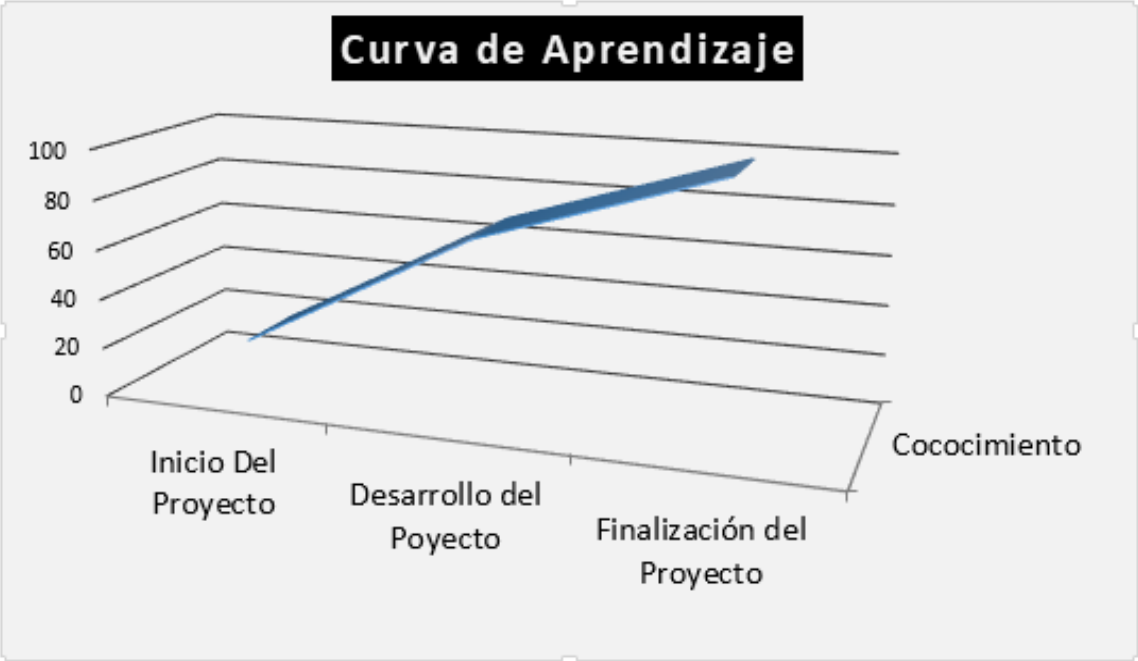
Figura 24 Tiempos empleados Vs, Nivel de conocimiento de herramientas



Fuente Autores

Teniendo en cuenta la gráfica anterior, se puede observar que la curva de aprendizaje adquirida durante el desarrollo del proyecto, permitió disminuir los tiempos de implementación a medida que se afianzaban y se obtenía mayor información de cada herramienta. Una manera de plasmar a curva de aprendizaje, se puede encontrar en la gráfica que se muestra a continuación.

Figura 24 Curva de aprendizaje



Fuente Autores

Sin duda alguna, desde nuestro punto de vista personal, al desarrollar este proyecto, consideramos muy importante la curva de aprendizaje que obtuvimos, en un tema del cual no se tiene mucho conocimiento, dándole el componente de investigación a este proyecto.

6 ENTREGABLES DEL PROYECTO

6.1 MANUAL DE IMPLEMENTACION DE HERRAMIENTA DE AUTOGESTION

- Documento Implementación de Herramientas de Autogestión para el levantamiento de servicios en proyectos de Big Data

6.2 MANUALES DE INSTALACION

- Manuales de Instalación y configuración de las herramientas usadas para la implementación de los servicios.

7. CONCLUSIONES.

- Al realizar esta investigación, se encontró que existen herramientas que facilitan la administración en las áreas TI, lo cual permite la optimización de los recursos y permite que las áreas de TI contribuyan al plan estratégico de una compañía u organización.

- Al hacer uso de herramientas de automatización, se disminuye el desgaste y el impacto que tiene implementar una infraestructura en Big Data, tanto en tiempo como en costos, lo cual permite a las compañías el escalamiento dinámico de las áreas de TI.

- El uso de Big Data es una tendencia a nivel Global y de manera especial en la actualidad en Colombia, por tanto adquirir conocimiento para este tipo de herramientas, permite explotar las cualidades que tiene este nuevo concepto de la ingeniería, maximizando el uso de los datos a nivel estratégico en las organizaciones.

- El uso de herramientas de Big Data, permite diseñar estrategias apoyadas en el estudio de los datos para la toma de decisiones.

8. RECOMENDACIONES

- Es importante tener claro el diseño de la infraestructura a desarrollar, debido a que se deben tener en cuenta las exigencias de recurso de cada una de las herramientas a implementar.

- Es muy importante documentarse bien en el manejo de las herramientas, también antes de enfrentar este tipo de proyectos, recordar línea de comandos de Linux, puesto que estos proyectos, la mayoría se basan en este sistema operativo.

- Es necesario el conocimiento básico del lenguaje de programación de Ruby para la generación de las rutinas a desplegar.

9. TRABAJOS FUTUROS

- Creación de una aplicación que permita orquestar los servicios orientados a big data enlazándola a través de la herramienta de autogestión todo por medio de chef server.

- Investigar e implementar más servicios los cuales puedan ser útiles en las necesidades explicitas que se tengan en las organizaciones.

- Realizar una nueva herramienta que sea útil para que se puedan desplegar en otros sistemas operativos.

- Lograr que los servicios instalados y los que se quieran incorporar a futuro sean de alta disponibilidad.

10. BIBLIOGRAFIA

[1] C. R. Blanco, “SQL básico, Página 2 Qué es SQL.”

[2] D. El Mundo, “Buscar en el sitio,” pp. 1–7, 2014.

[3] L. miguel Armendariz, “SOBRE EL â€TM CÓDIGO ABIERTO â€TM (OPEN SOURCE),” no. c. pp. 2003–2006, 2006.

[4] “¿Qué es Business Intelligence?,” 2012. [Online]. Available: http://www.sinnexus.com/business_intelligence/. [Accessed: 15-Nov-2017].

[5] D. Evans, “The Internet of Things - How the Next Evolution of the Internet is Changing Everything,” *CISCO white Pap.*, no. April, pp. 1–11, 2011.

[6] “Sistema de SaaS (Software as a Service) para centros educativos.”

[7] J. C. C. Romani, “El concepto de tecnologías de la información. Benchmarking sobre las definiciones de las TIC en la sociedad del conocimiento,” *Zer - Rev. Estud. Comun.*, vol. 14, no. 27, pp. 285–318, 2009.

[8] E. Alegría, A. Adrián, C. Demartini, P. E. Catrilef, and E. P. Zuleta, “NAT y su relación con IPv6.”

[9] M. -Castilla León -Galicia -Levante, “Hacemos que las piezas encajen www.trc.es.”

[10] A. L. Guillen, “Una introducción al ecosistema Hadoop.”

[11] L. Al, “Bases de datos NoSQL,” 2011.

[12] “Curso librerias Web 2.0.”

[13] J. Gutiérrez, “Qué es un framewor¿k web?,” p. 1, 2006.

[14] “INTRODUCCIÓN AL LENGUAJE HTML.”

[15] G. Lehey and <grog@freebsd Org>, “Qué es BSD Tabla de contenidos.”

[16] J. A. A. M. V. Universitat de Barcelona. Facultat de Biblioteconomia i Documentació., *BiD : textos universitaris de biblioteconomia i documentació.* Facultat de Biblioteconomia i Documentació, Universitat de Barcelona, 1999.

[17] “IBM Almacenamiento en cloud | Cloud Business | IBM España.” [Online]. Available: https://www.ibm.com/cloud-computing/es-es/infrastructure/object-storage/?S_PKG=AW&cm_mmc=Search_Google-_-Leadership+Agenda_DIGITAL+INBOUND-_-ES-_-+ibm++services_Broad_AW&cm_mmca1=000009PY&cm_mmca2=10002507&mkwid=7995ce9a-6448-43f8-ad17-245d7cd64589%257C594%25. [Accessed: 19-Oct-2017].

[18] J. Schroeck, Michael; Shockley, Rebecca; Smart, “Analytics: el uso de big data en el mundo real,” *IBM. Inf. Ejec.*, p. 22, 2012.

[19] “Historia De Las Bases De Datos Timeline | Preceden.” [Online]. Available: https://www.preceden.com/timelines/48236-historia-de--las-bases-de-datos. [Accessed: 19-Oct-2017].

[20] A. Muñoz Chaparro, *Oracle 11g SQL curso práctico de formación*. RC Libros, 2011.

[21] W. R. Neuman and Y. J. I. N. Park, “Tracking the Flow of Information into the Home : An Empirical Assessment of the Digital Revolution in the United States , 1960 – 2005 University of Michigan,” vol. 6, pp. 1022–1041, 2012.

[22] D. Boyd and K. Crawford, “Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon,” *Inf. Commun. Soc.*, vol. 15, no. 5, pp. 662–679, Jun. 2012.
[23] D. J. Power, “A Brief History of Decision Support Systems,” 2014.
[24] T. Especial De Grado, P. Alfonso, and P. Muñoz, “Desarrollo de una Arquitectura Big Data para Registros Mercantiles.”
[25] “Big Data – Historia cronológica | Winshuttle.” [Online]. Available: https://www.winshuttle.es/big-data-historia-cronologica/. [Accessed: 19-Oct-2017].
[26] M. Cox and D. Ellsworth, “Application-controlled demand paging for out-of-core visualization,” *Proceedings. Vis. ’97 (Cat. No. 97CB36155)*, no. July, p. 235–244, 1997.
[27] “Cómo el Internet de las cosas transformará su cadena de suministro - Winshuttle Spanish.” [Online]. Available: https://www.winshuttle.es/blog/como-el-internet-de-las-cosas-transformara-su-cadena-de-suministro/. [Accessed: 19-Oct-2017].
[28] “How Much Information Is ‘Too Much Information’? - NYTimes.com.” [Online]. Available: https://mobile.nytimes.com/blogs/learning/2012/02/17/how-much-information-is-too-much-information/?referer=. [Accessed: 19-Oct-2017].
[29] J. A. Carrillo Ruiz *et al.*, “Big Data en los entornos de Defensa y Seguridad,” *Inst. Español Estud. Estratégicos*, vol. 1, p. 124, 2013.
[30] M. Olson, “HADOOP: Scalable, Flexible Data Storage and Analysis,” *IQT Q.*, vol. 1, no. 3, pp. 14–18, 2010.
[31] D. Reinsel *et al.*, “John F. Gantz, Project Director A Forecast of Worldwide Information Growth Through 2010,” 2007.
[32] “BIG DATA Y OPEN DATA: EL UNIVERSO DIGITAL DE DATOS timeline | Timetoast timelines.” [Online]. Available: https://www.timetoast.com/timelines/big-data-y-open-data-el-universo-digital-de-datos. [Accessed: 21-Oct-2017].
[33] R. Bryant, R. Katz, and E. Lazowska, “Big-Data Computing: Creating Revolutionary Breakthroughs in Commerce, Science and Society,” *Comput. Res. Assoc.*, pp. 1–15, 2008.
[34] W. R. Neuman, Y. J. Park, and E. Panek, “Info Capacity| Tracking the Flow of Information into the Home: An Empirical Assessment of the Digital Revolution in the U.S. from 1960–2005,” *Int. J. Commun.*, vol. 6, no. January, p. 20, 2012.
[35] “A special report on managing information ı,” 2010.
[36] S. Rogers, “Top 10 Trends in Business Intelligence and Analytics for 2011,” pp. 1–3, 2011.
[37] J. Rivera and R. van der Meulen, “Gartner Says 4.9 Billion Connected ‘Things’ Will Be in Use in 2015,” *Gart. - Newsroom*, pp. 9–10, 2014.
[38] D. Borthakur, “HDFS architecture guide,” *Hadoop Apache Proj. http//hadoop apache …*, pp. 1–13, 2008.
[39] R. Barranco, “¿Qué es Big Data?,” *18-06-2012*, 2012. [Online]. Available: https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/. [Accessed: 21-Oct-2017].
[40] A. McAfee and E. Brynjolfsson, “Big Data. The management revolution,”

Harvard Buiness Rev., vol. 90, no. 10, pp. 61–68, 2012.
[41] Y. Demchenko, C. De Laat, and P. Membrey, “Defining architecture components of the Big Data Ecosystem,” *2014 Int. Conf. Collab. Technol. Syst. CTS 2014*, no. March 2015, pp. 104–112, 2014.
[42] P. C. Science, “Spark,” *INNS Conference on Big Data*, pp. 121–131, 2015.
[43] “Encuentro mundial de NATs.” .
[44] “¿Qué es CAOBA? - Alianza Caoba MinTic, Conciencias, universidades.” [Online]. Available: http://alianzacaoba.co/que-es-caoba/. [Accessed: 21-Oct-2017].
[45] Y. Demchenko, C. De Laat, and P. Membrey, “Defining Architecture Components of the Big Data Ecosystem-Reviewed .pdf.” .
[46] Chukwa, “Chukwa - Welcome to Apache Chukwa,” 2016. [Online]. Available: http://chukwa.apache.org/. [Accessed: 21-Oct-2017].
[47] “¿Qué es Kibana? – Un poco de Java.” [Online]. Available: https://unpocodejava.com/2012/10/25/que-es-apache-flume/. [Accessed: 21-Oct-2017].
[48] R. Serrat Morros, “Big Data : análisis de herramientas y soluciones,” pp. 10–18, 2013.
[49] “IBM Knowledge Center - Rangos de direcciones privadas.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SSPT3X_4.1.0/com.ibm.swg.im.infosphere.biginsights.analyze.doc/doc/hbaseConcepts.html. [Accessed: 21-Oct-2017].
[50] L. Joyanes, *Big data - análisis de grandes volúmenes de datos en organizaciones*. 2013.
[51] MongoDB, “Reinventando la gestión de datos | MongoDB,” 2016. [Online]. Available: https://www.mongodb.com/es. [Accessed: 21-Oct-2017].
[52] Neo4j, “Neo4j, the world’s leading graph database - Neo4j Graph Database.” [Online]. Available: https://neo4j.com/. [Accessed: 21-Oct-2017].
[53] N. Jiménez Barquín, “Big Data: Hadhoop,” 2014.
[54] M. K. Islam and A. Srinivasan, *Apache Oozie*. 2015.
[55] M. Marquis, “RedisConf 2017 is near,” 2017. [Online]. Available: https://redis.io/.
[56] J. L. Reyes-Ortiz, L. Oneto, and D. Anguita, “Big data analytics in the cloud: Spark on Hadoop vs MPI/OpenMP on Beowulf,” in *Procedia Computer Science*, 2015, vol. 53, no. 1, pp. 121–130.
[57] M. (University of C. Al-Fares, A. (University of C. Loukissas, and A. (University of C. Vahdat, “A scalable, commodity data center network architecture,” *Sigcomm*, pp. 63–74, 2008.
[58] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” *Proc. 6th Symp. Oper. Syst. Des. Implement.*, pp. 137–149, 2004.
[59] B. Y. J. Dean and S. Ghemawat, “MapReduce: a flexible data processing tool,” *Commun. ACM*, vol. 53, no. 1, pp. 72–77, 2010.
[60] R. Wesley, M. Eldridge, and P. T. Terlecki, “An analytic data engine for visualization in tableau,” *Proc. 2011 Int. Conf. Manag. data - SIGMOD ’11*, p. 1185, 2011.

[61] J. Pérez Díaz and C. P. Solà, “Plataforma para analizar la red Bitcoin,” 2017.
[62] “Graphite.” [Online]. Available: http://graphiteapp.org/. [Accessed: 23-Oct-2017].
[63] G. Hernández Coca, “Tipos de Modelos en Investigación de Operaciones,” 2011.
[64] C. Espino, “Trabajo de Fin de Grado ‘ Análisis predictivo : técnicas y modelos utilizados y aplicaciones del mismo - herramientas Open Source que permiten su uso ,’” p. 65, 2017.
[65] “Modelos predictivos: reforzando el valor de una buena decisión.” [Online]. Available: https://blog.es.logicalis.com/analytics/modelos-predictivos-reforzando-el-valor-de-una-buena-decision. [Accessed: 21-Oct-2017].
[66] “Aprovecha los modelos predictivos de Big Data: tres de cada cuatro predicciones son correctas | Centro de Innovación BBVA.” [Online]. Available: http://www.centrodeinnovacionbbva.com/noticias/aprovecha-los-modelos-predictivos-de-big-data-tres-de-cada-cuatro-predicciones-son. [Accessed: 21-Oct-2017].
[67] “full-text.”
[68] J. Ignacio and M. Alberdi, “La piedra angular del Análisis Predictivo.”
[69] “Machine Learning, ¿qué aporta al Big Data? - CICE.” [Online]. Available: https://www.cice.es/noticia/machine-learning-big-data/?gclid=CjwKCAjwo4jOBRBmEiwABWNaMXjJgzeyB83aPbfM5GhjLXxMg8dZ4PHAX_YwFyUOKjvGOlz9U97NxxoCoHAQAvD_BwE. [Accessed: 21-Oct-2017].
[70] “Asir 14-15,” 2015.
[71] L. M. Gracia, “Arquitectura Lambda: Principios de Arquitectura para Sistemas Big Data en Tiempo Real,” p. 3, 2013.
[72] V. Astakhov and M. Chayel, “Lambda Architecture for Batch and Real-Time Processing on AWS with Spark Streaming and Spark SQL,” no. May, pp. 1–12, 2015.
[73] J. Careaga, “Arquitectura Lambda vs Arquitectura Kappa ¿Cuál es el mejor enfoque para implementar un ambiente de trabajo para procesar big data?”
[74] J. Forgeat, “Data processing architectures – Lambda and Kappa,” *Ericsson Research Blog*, 2015. [Online]. Available: https://www.ericsson.com/research-blog/data-processing-architectures-lambda-and-kappa/. [Accessed: 21-Oct-2017].
[75] J. Scott, “Zeta Architecture: Hexagon is the new circle,” 205AD. [Online]. Available: https://www.oreilly.com/ideas/zeta-architecture-hexagon-is-the-new-circle.
[76] “NEXT GENERATION ENTERPRISE MODELLING,” p. 2016, 2016.
[77] J. R. C. Rodríguez, “Base de datos distribuidos,” pp. 1–14, 2014.
[78] Departamento de ciencias de la computación e I.A, “Introducción a las bases de datos,” *Univ. Granada*, 2013.
[79] V. R. M. O. Carlos Enrique Rodas Gálvez, Álvaro Daniel Castillo Carrera, Miguel Enrique Guerra Connor, “CLUSTER Curso: Sistemas Operativos II Plataforma: Linux - OpenSuse CLUSTER,” *Univ. San Carlos Guatemala Fac. Ing. Esc. Ciencias y Sist.*
[80] L. Joyanes Aguilar, “COMPUTACIÓN EN LA NUBE. Notas para una

estrategia española en cloud computing," *Rev. del Inst. Español Estud. Estratégicos*, vol. 1, no. 1, pp. 89–112, Apr. 2012.
[81] "Qué es un ERP | Encuentra tu solución ERP en España." [Online]. Available: https://www.elegirerp.com/erp/que-es-un-erp. [Accessed: 21-Oct-2017].
[82] "Zettabyte." [Online]. Available: http://www.tecnologiahechapalabra.com/datos/eventos/articulo.asp?i=5896. [Accessed: 21-Oct-2017].
[83] L. Euler, G. Kirchhoff, F. Guthrie, K. Appel, and W. Haken, "Teoría de grafos w w w ib ro sZ . c."
[84] "Integración de datos: Concepto e importancia en la empresa actual." [Online]. Available: https://www.powerdata.es/integracion-de-datos. [Accessed: 23-Oct-2017].
[85] Ernst&Young, "Inteligencia de Negocio (BI)," p. 2, 2013.
[86] M. Internet and Q. O. Cosa, "EL 'INTERNET DE LAS COSAS.'"
[87] "IPV4-IPV6 | CERTIFICACIÓN DE SISTEMAS OPERATIVOS." [Online]. Available: https://gcvc09.wordpress.com/ipv4-ipv6/. [Accessed: 23-Oct-2017].
[88] "Definiciones de metadatos."
[89] "Teoría de Lenguajes de Programación: Paradigmas." [Online]. Available: http://tlp-lcc-umt.blogspot.com.co/2017/04/paradigmas.html. [Accessed: 23-Oct-2017].
[90] "PCPI Cheste - Procesamiento en paralelo (SLI, Crossfire)." [Online]. Available: https://informaticapcpicheste.wikispaces.com/Procesamiento+en+paralelo+(SLI,+Crossfire). [Accessed: 23-Oct-2017].
[91] "¿Qué es un zettabyte?" [Online]. Available: http://www.parentesis.com/noticias/ciencias/Sabes_que_es_un_zettabyte. [Accessed: 23-Oct-2017].

11. ANEXOS

MANUAL PARA LA INSTALACION DE SOFTWARE

CHEF

CHEF SERVER VERSION 12

Tabla de contenido

Introducción

Este manual busca dar las pautas necesarias para realizar la instalación de Chef Server junto con sus 3 componentes principales, Chef Server, Chef Workstation y Chef Node .Los cuales componen la infraestructura básica de este software de automatización. A continuación se encuentra la versión y sistema operativo para el cual está diseñado este manual

Sistema Operativo	Centos 7 (colocar toda la versión)
Chef Server	Chef Server Versión 12

Prerrequisitos de Instalación

Para realizar la instalación de chef server sed deben tener en cuenta los siguientes prerrequisitos para evitar inconvenientes en el momento de su instalación.

- Se debe tener configurado el Hostname de la maquina sobre la cual se va a realizar la instalación.
- Se debe tener configurado el DNS de la maquina en la cual se va a realizar la instalación.
- Se debe instalar el siguiente paquete.

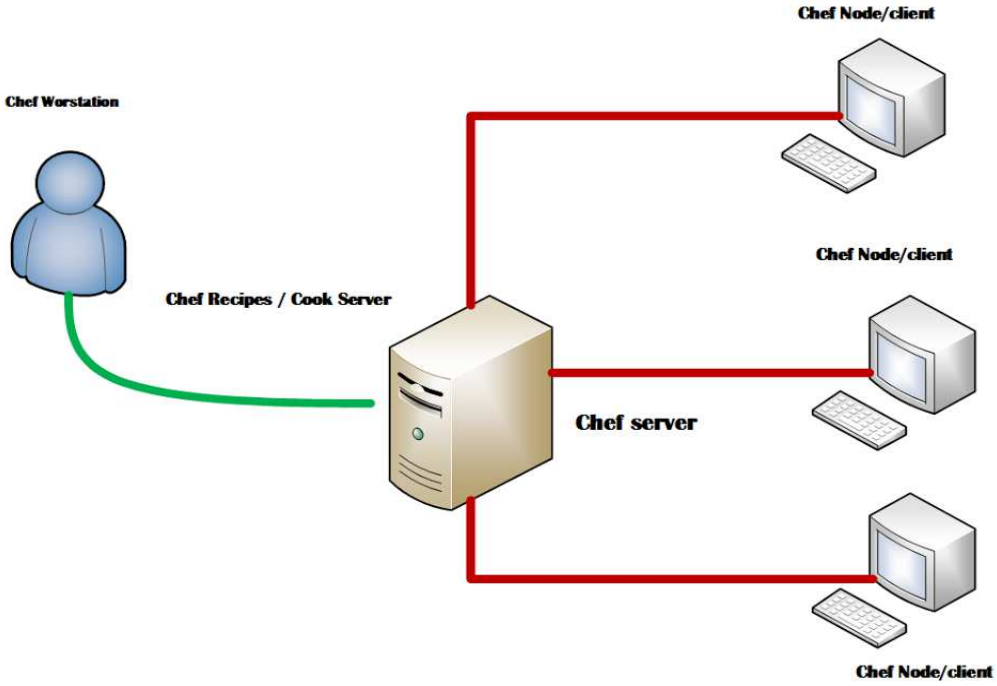
```
# Yum install -y wget curl
```

Componentes de Chef

Chef está formado por tres componentes, que permiten generar el despliegue de su potencial, dentro de chef server se identifican los siguientes roles:

- Chef Workstation: Donde se Desarrollan las rutinas a implementar
- Chef Server: Es el encargado de desplegar las rutinas en los clientes
- Chef Client: Permite la comunicación con el Server para que este lo pueda gestionar.

Cada uno de los roles anteriormente mencionados cumplen una función específica y son necesarios los tres roles para conformar la infraestructura de chef y el despliegue de la misma, como se observa en la siguiente imagen.

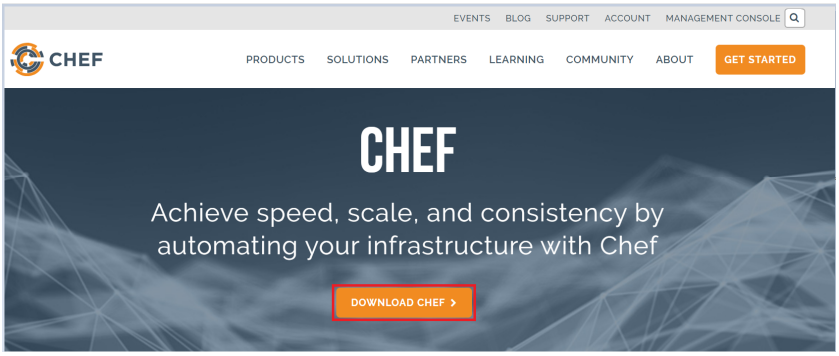


Instalación de chef Server

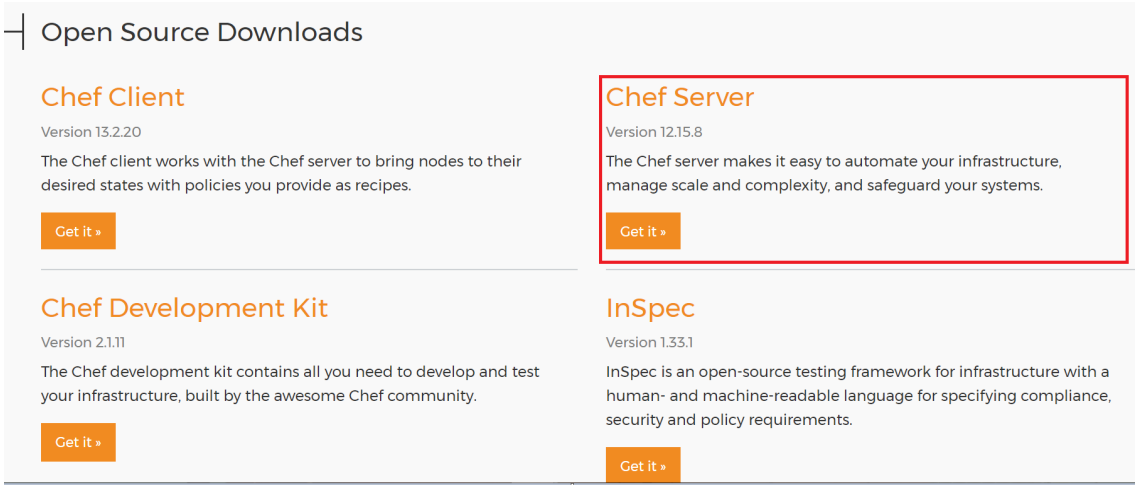
a. Descargar chef server

Para descargar chef dirigirse a la siguiente URL: https://www.chef.io/chef/

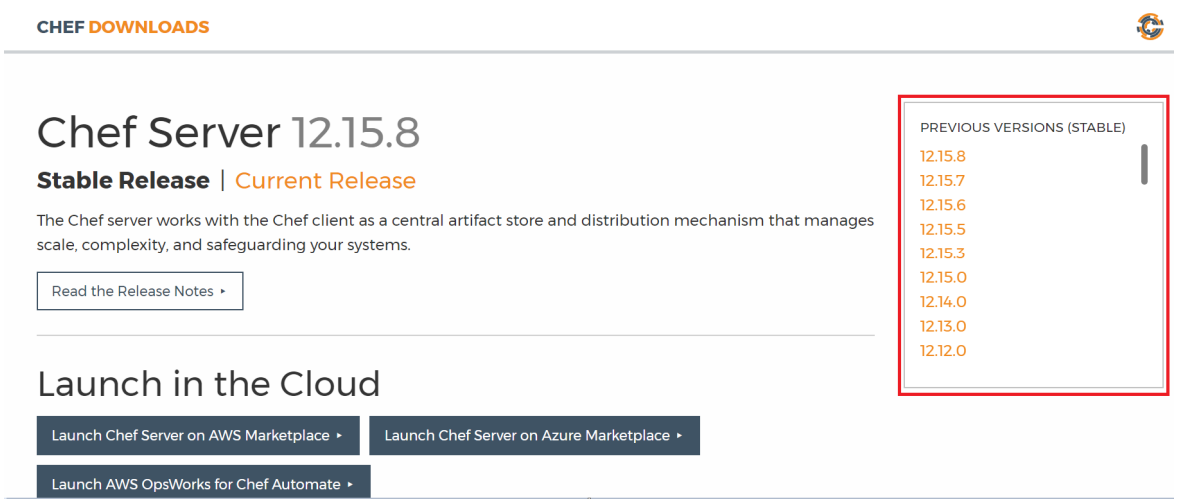
Una vez en la página dar clic en el botón Download Chef



Al dar clic en el botón se despliega la página donde se observan las fuentes de descargas, para ello seleccionar la que dice chef server y dar clic en Obtener.



Una vez se da clic en obtener, se presenta la siguiente página donde se observan las versiones disponibles para descargas.



Seleccionar la Versión 12.15.8 y dar clic en este enlace, una vez en la página a la que se redirecciona el enlace, se visualiza un listado que contiene de acuerdo a la distribución de Linux existente, para el caso de Centos se debe seleccionar aquella que se encuentre dentro de **Red Hat Enterprise Linux 7** y teniendo en cuenta la arquitectura, que para este caso es **X86_64.** Para instalación en Linux se puede realizar obteniendo la URL del paquete.

Una vez con esta URL podemos realizar el próximo paso dentro de la instalación.

b. Instalación del paquete descargado

Para realizar la instalación del paquete descargado usar el siguiente comando, reemplazando la URL por la que se ajuste a la necesidad de la distribución de Linux.

```
# rpm -ivh https://opscode-omnibus-
packages.s3.amazonaws.com/el/6/x86_64/chef-server-11.0.8-1.el6.x86_64.rpm
```

Al instalar el paquete se puede a entrar a desarrollar la configuración del Chef Server.

c. Configuración de Chef Server

Para realizar la configuración de Chef Server, hacer uso del siguiente comando.

```
# chef-server-ctl reconfigure
```

Al ejecutar el comando anterior se configurarán los componentes requeridos incluyendo el Erchef, RabbitMQ, PostgresSQL y todos los cookbook de los cuales hace uso en su versión 12. Una vez se realiza la

configuración es importante verificar que el Hostname se encuentra correctamente seteado. Para ello se ejecuta el comando.

```
# Hostname
```

d. Verificando instalación de chef server

Para verificar la instalación del servidor, basta con ejecutar el siguiente comando

```
# chef-server-ctl test
```

Recomendación: Es importante detener el servicio de Apache antes de ejecutar esta prueba.

Luego de estar seguros de la parametrización del hostname, se puede realizar una prueba para observar si se despliega o no el servido de Chef en el navegador. En este caso se puede explorar a través de la URL del servidor de Chef la cual se compone:

```
# Https: // FQDN-OR-IP-OF-CHEF-SERVER
```

Recomendación: Para ingresar al servidor una vez este se despliegue hacer uso del siguiente login de chef por defecto para ingresar.

- Username: admin
- Contraseña: p@ssword1

Instalación Chef Server Workstation

Para la instalación de este componente, es necesario ejecutar este comando, el cual es compatible para ambientes de Linux.

```
curl -L https://www.opscode.com/chef/install.sh | bash
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Curre
                 Dload  Upload   Total   Spent    Left  Speed
101  6790  101  6790    0     0   3826      0  0:00:01  0:00:01 --:--:-- 12
Downloading Chef  for el...
Installing Chef
warning: /tmp/tmp.KnyQTnqz/chef-.x86_64.rpm: Header V4 DSA/SHA1 Signature,
Preparing...                ########################################### [100%]
1:chef                      ########################################### [100%]
Thank you for installing Chef!
```

Una vez finalizado la ejecución del comando anterior se recomienda verificar si el cliente del chef

Ha quedado instalado para ello, se ejecuta el siguiente comando.

```
# chef-client -v
```

Una vez se ejecuta el comando, se tendrá como respuesta la versión del chef cliente que se está Usando, esto indicara que el chef client está correctamente instalado.

a. Crear el Directorio para Chef

Es necesario crear el directorio para chef, el cual es usado para almacenar archivos importantes de Chef, los cuales permiten su uso. A continuación encuentra los archivos que se almacenaran en este directorio.

- Knife.rb
- ORGANIZATION-validator.pem
- USER.pem

b. Claves de Certificación

Para que el servidor Chef funciones, es necesario copiar las llaves de certificación en la carpeta del usuario de la estación de trabajo, para esto se ejecuta el siguiente comando.

```
$ Mkdir ~ / .chef
$ Scp root @ chef-servidor: / etc / chef-server / admin.pem ~ / .chef
$ Scp root @ chef-servidor: / etc / chef-server / chef-validator.pem ~ /
```

c. Configuración del Cliente

Una vez se tengan las llaves de certificación se puede entrar a revisar la configuración del cliente, para ello se realiza la configuración haciendo uso del comando Nife de la siguiente forma.

```
$ Knife configure -i
  ¿Sobreescribir /root/.chef/ knife.rb ? (S / N) y
  Por favor ingrese la URL del servidor del chef: [ https://test.example.
  Por favor ingrese un nombre para el nuevo usuario: Root] knife-user1
  Por favor ingrese el nombre de administrador existente: [admin] Enter
  Por favor ingrese la ubicación de la clave privada del admin existente:
  Introduzca el nombre de cliente de validación: [chef-validator]
  Introduzca la ubicación de la clave de validación: [/ etc / chef-server
  Por favor ingrese la ruta al repositorio de un cocinero (o deje en blar
  Creación de un usuario de API inicial ...
  Introduzca una contraseña para el nuevo usuario:
  Usuario creado [knife-user1]
  Archivo de configuración escrito en /root/.chef/ knife.rb
```

d. Comprobando la instalación.

Al igual que para instalar el servidor de Chef, también se tiene un comando para comprobar la instalación de la Workstation , mostrando el listado de clientes de Knife así como el listado de usuarios ,basta con ejecutar el comando que se muestra a continuación.

Lista de Clientes

```
$ knife client list
  chef-validator
  chef-webui
```

Lista de Usuarios

```
$ knife user list
  admin
  knife-user1
```

Instalación Chef Node

Una vez realizadas configuraciones del Chef server como las del Workstation iniciamos la instalación del chef Node, se observara que el proceso de instalación es similar al de los dos componentes anteriormente explicados en este manual. Para la instalación del Chef Node es necesario ejecutar el siguiente comando.

```
curl -L https://www.opscode.com/chef/install.sh | bash
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Cu
                 Dload  Upload   Total   Spent    Left  Speed
101  6790  101  6790    0     0   3826      0  0:00:01  0:00:01 --:--:--
Downloading Chef  for el...
Installing Chef
warning: /tmp/tmp.KnyQTnqz/chef-.x86_64.rpm: Header V4 DSA/SHA1 Signatur
Preparing...                ########################################### [10
1:chef                      ########################################### [10
Thank you for installing Chef!
```

Una vez realizada la instalación del chef se procede con crear el directorio Chef.

a. Crear el Directorio para Chef

Para crear el directorio chef, se emplea el siguiente comando.

```
# mkdir /etc/chef
```

b. Claves de Certificación

Para que el servidor Chef funciones, es necesario copiar las llaves de certificación en la carpeta del usuario del nodo de trabajo la cual se encuentra en /etc/chef para esto se ejecuta el siguiente comando. Para llevar a cabo se ejecuta la siguiente instrucción.

```
# scp root@chef-server:/etc/chef-server/chef-validator.pem /etc/chef
```

e. Iniciando el Chef Cliente.

Iniciar sesión en el cliente de Chef y ejecutar el siguiente comando para registrar un cliente con Chef Server:

```
# Chef-client -S https: // FQDN-OR-IP-OF-CHEF-SERVIDOR -K / etc /
chef / chef-validator.pem
```

f. Crear archivo cliente.

Una vez se ha verificado el cliente, es necesario crear un archivo llamado "client.rb" en la siguiente ruta "/etc/chef". Para generar este archivo se debe ejecutar el siguiente comando.

```
# Vi / etc / chef / client.rb
  Log_level: información
  Log_location STDOUT
  Chef_server_url 'https: // FQDN-OR-IP-OF-CHEF-SERVER'
```

g. Comprobar registro del Nodo.

Para comprobar que el nodo ha quedado registrado se debe ejecutar el siguiente comando.

```
# knife node list
```

Si dentro de este listado se muestra el hostname del nodo instalado, indicara que el nodo ha quedado instalado de manera correcta.

h. Ejecutar el Chef Cliente.

El objetivo es determinar si el Cookbook respectivo se direcciona al nodo instalado, para esto se emplea el siguiente comando.

```
# chef-client
# chef-client -l debug (In case if you want to debug)
```

Una vez verificado esto, se puede inicializar el chef cliente e indicar el tiempo en el cual se sincronizara con el servidor, para este caso se hace uso de un tiempo 3600 segundos.

```
# Chef-client -i 3600
```

MANUAL PAR LA INSTALACION DE SOFTWARE

APACHE FLUME VERSION 1.8.0

Tabla de contenido

Introducción

Este manual tiene como objetivo, indicar las instrucciones necesarias para realizar la instalación de apache Flume, que permite recolectar información y una herramienta altamente confiable y muy usada en ambientes de Big Data.

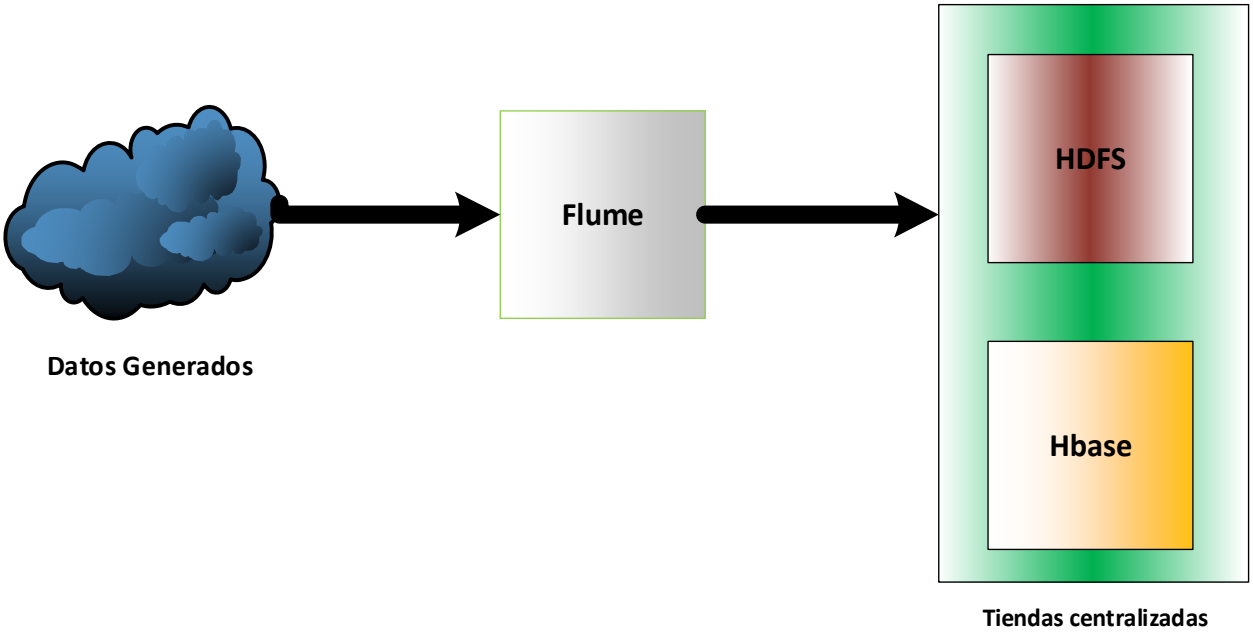
Sistema Operativo	Centos 7
Flume	Versión

Componentes de Flume

Chef está formado por tres componentes que conforman su arquitectura, estos componentes se enumeran a continuación:

- Nube: Es donde se ubican los datos de trasmisión.
- Flume: herramienta confiable y distribuida
- HDFS: Función de almacenamiento de información.

Cada uno de los componentes anteriormente mencionados permite articular la arquitectura de Flume y son necesarios para que este funcione, esto se puede observar en la siguiente imagen.



Prerrequisitos de Instalación

Para realizar la instalación de Flume se deben tener en cuenta los siguientes prerrequisitos para evitar inconvenientes en el momento de su instalación.

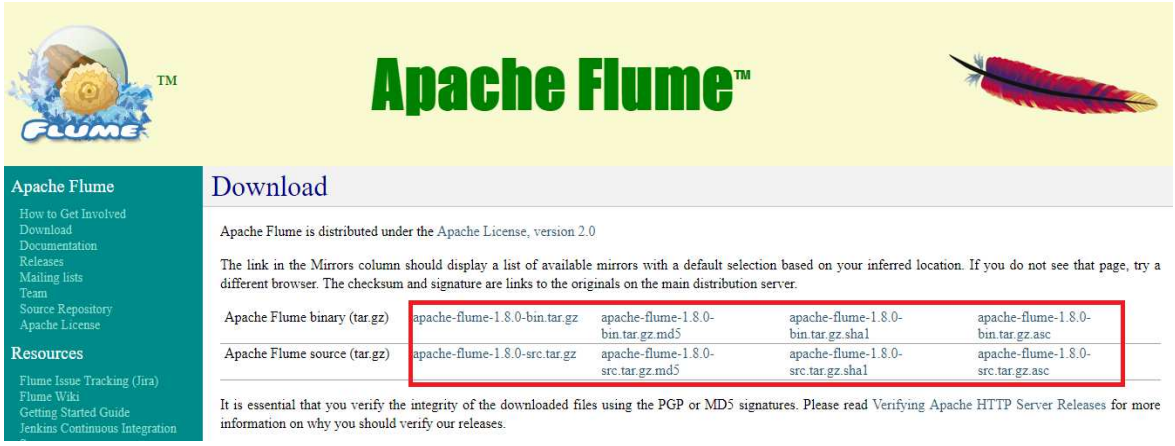
Se debe tener un entorno de java instalado en el sistema, adicionalmente se debe tener instalado un entorno de Hadoop para que Flume no tenga problemas en su instalación.

Instalación de Flume

e. Descarga de Flume

En primer lugar, descargue la última versión del software Apache Flume del sitio web https://flume.apache.org/

Abra el sitio web. Haga clic en el enlace de **descarga** en el lado izquierdo de la página de inicio. Te llevará a la página de descarga de Apache Flume.



f. Elección de versión y descargue

En la página de Descargas, puede ver los enlaces para archivos binarios y fuente de Apache Flume. Haga clic en el enlace apache-flume-1.6.0-bin.tar.gz

Se le redirigirá a una lista de réplicas donde puede comenzar la descarga haciendo clic en cualquiera de estos duplicados. De la misma manera, puedes descargar el código fuente de Apache Flume haciendo clic en apache-flume-1.6.0-src.tar.gz

g. Crear un directorio

Cree un directorio con el nombre Flume en el mismo directorio donde se instalaron los directorios de instalación de Hadoop, HBase y otro software (si ya ha instalado alguno) como se muestra a continuación.

```
$ mkdir Flume
```

h. Extraer archivos

Extraiga los archivos tar descargados como se muestra a continuación.

```
$ cd Downloads/
$ tar zxvf apache-flume-1.6.0-bin.tar.gz
$ tar zxvf apache-flume-1.6.0-src.tar.gz
```

Extraer directorios

Mueva el contenido del archivo apache-flume-1.6.0-bin.tar al directorio Flume creado anteriormente como se muestra a continuación. (Supongamos que hemos creado el directorio Flume en el usuario local llamado Hadoop).

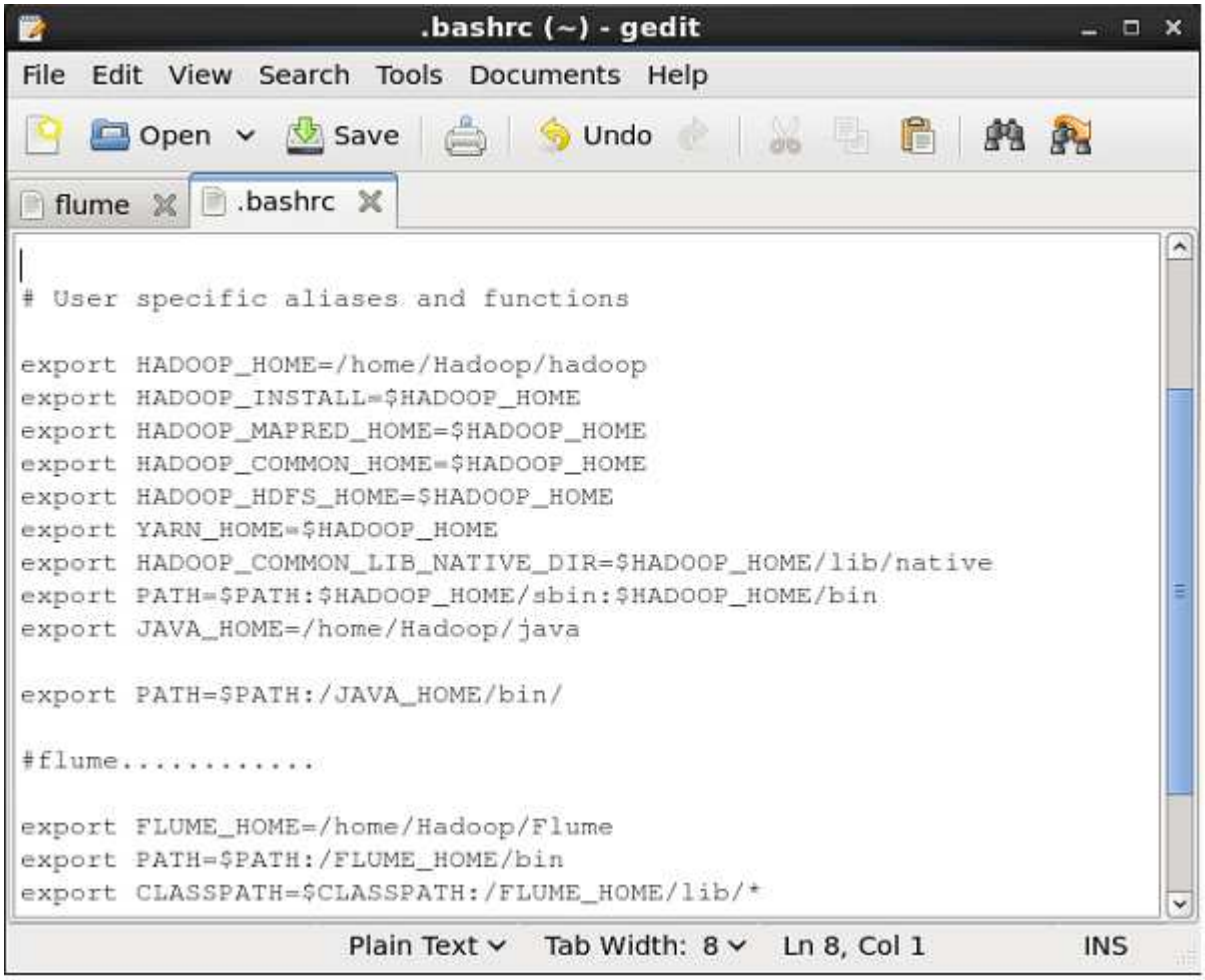
```
$ mv apache-flume-1.6.0-bin.tar/* /home/Hadoop/Flume/
```

Configuración del canal

Para configurar Canal de flujo, tenemos que modificar tres archivos a saber, flume-env.sh, flumeconf.properties, y bash.rc.

Establecer el camino / Classpath

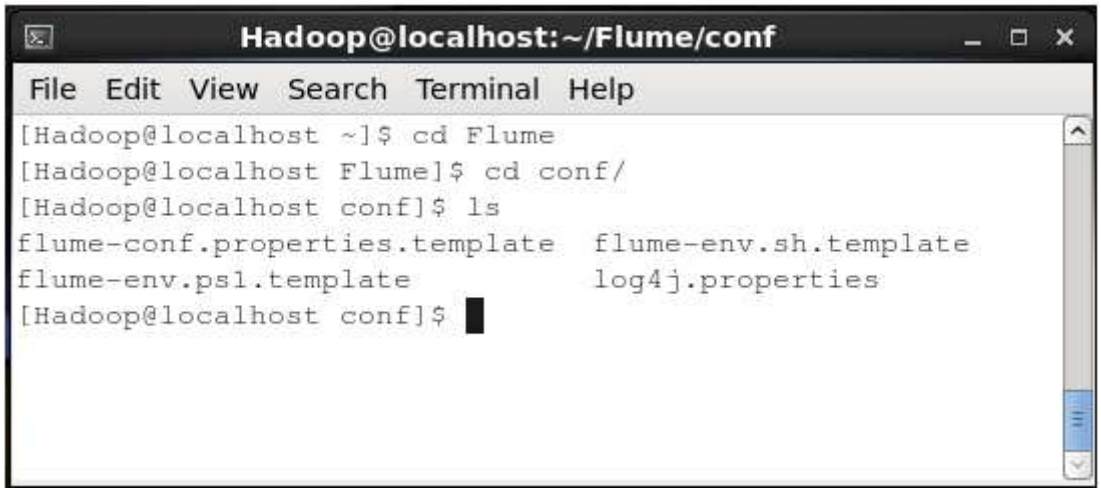
En el archivo .bashrc, configure la carpeta de inicio, la ruta y la ruta de clase para Flume como se muestra a continuación.



a. Carpeta de conf

Si abre la carpeta conf de Apache Flume, tendrá los siguientes cuatro archivos:

- flume-conf.properties.template,
- flume-env.sh.template,
- flume-env.ps1.template, y
- log4j.properties.



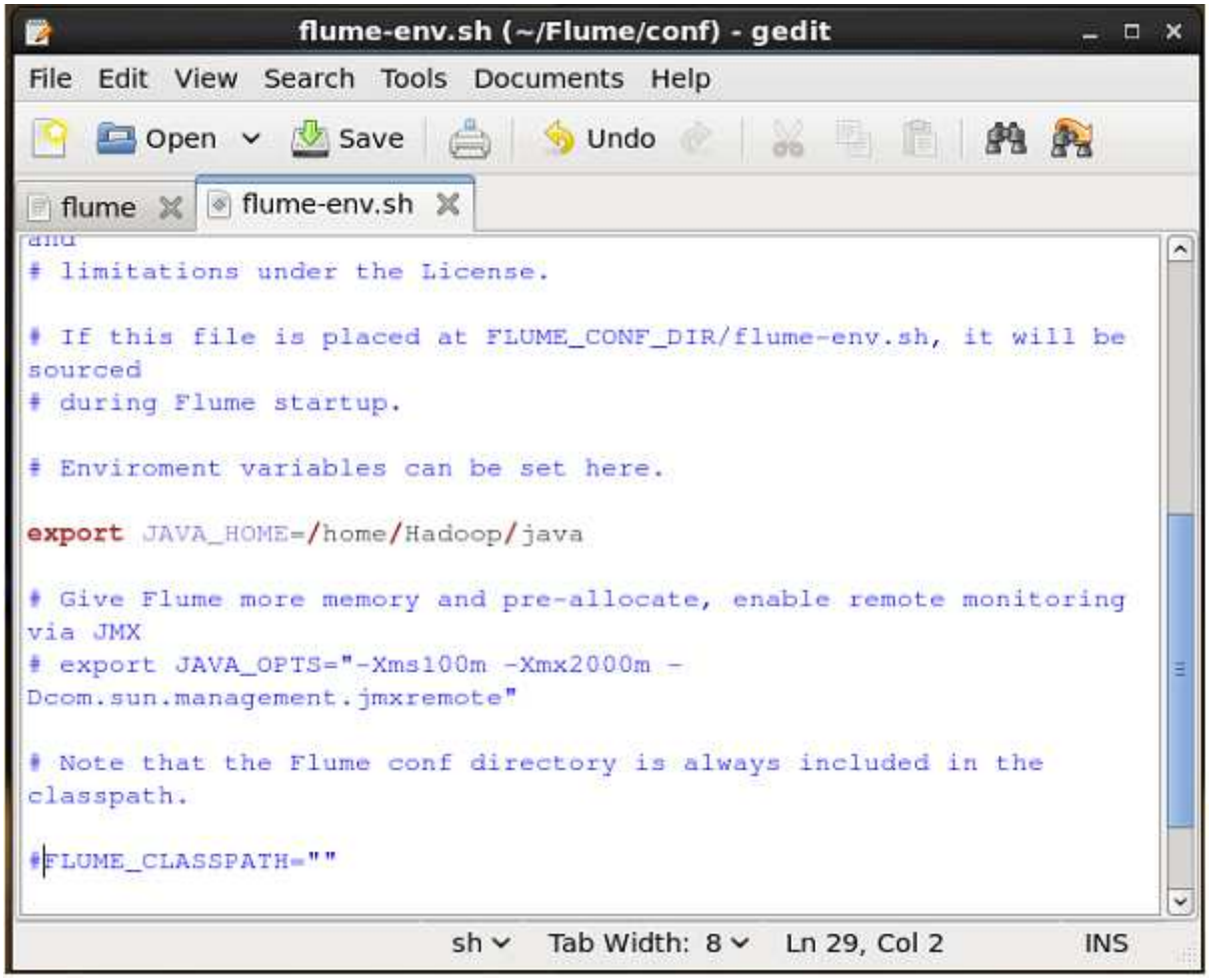
Ahora cambie el nombre

- archivo flume-conf.properties.template como flume-conf.properties y

- flume-env.sh.template como flume-env.sh

flume-env.sh

Abra el archivo flume-env.sh y configure el JAVA_Home en la carpeta donde se instaló Java en su sistema.



b. Verificando la instalación

Verifique la instalación de Apache Flume navegando por la carpeta bin e ingresando el siguiente comando.

```
$ ./flume-ng
```

Si ha instalado con éxito Flume, recibirá un aviso de ayuda de Flume como se muestra a continuación.

Después de instalar Flume, debemos configurarlo utilizando el archivo de configuración que es un archivo de propiedades Java con pares clave-valor. Necesitamos pasar valores a las claves en el archivo.

En el archivo de configuración de Flume, necesitamos.

- Nombre los componentes del agente actual.
- Describe / Configura la fuente.
- Describe / Configure el fregadero.
- Describe / Configura el canal.
- Une la fuente y el receptor al canal.

Usualmente podemos tener múltiples agentes en Flume. Podemos diferenciar a cada agente usando un nombre único. Y usando este nombre, tenemos que configurar cada agente.

c. Nombrando los componentes

En primer lugar, debe nombrar / enumerar los componentes, como las fuentes, los sumideros y los canales del agente, como se muestra a continuación.

```
agent_name.sources = source_name
agent_name.sinks = sink_name
agent_name.channels = channel_name
```

Flume admite varias fuentes, receptores y canales. Se enumeran en la tabla que encuentra a continuación.

Fuentes	Canales	Fregaderos
▫ Avro Source ▫ Fuente de ahorro ▫ Fuente de Exec ▫ Fuente JMS ▫ Fuente de directorio de Spooling ▫ Twitter 1% firehose Fuente ▫ Fuente de Kafka ▫ Fuente de NetCat ▫ Fuente del generador de secuencia ▫ Fuentes de Syslog ▫ Fuente Syslog TCP ▫ Multiport Syslog TCP Source ▫ Fuente Syslog UDP ▫ Fuente HTTP	▫ Canal de memoria ▫ Canal JDBC ▫ Canal Kafka ▫ Canal de archivos ▫ Canal de memoria derramable ▫ Canal de Pseudo Transacción	▫ Fregadero HDFS ▫ Hive Hundir ▫ Logger Hundir ▫ Fregadero de Avro ▫ Fregadero de ahorro ▫ Fregadero IRC ▫ Archivo Roll Sink ▫ Fregadero nulo ▫ HBaseSink ▫ AsyncHBaseSink ▫ MorphlineSolrSink ▫ ElasticSearchSink ▫ Kite Dataset Sink ▫ Kafka Hundir

Puedes usar cualquiera de ellos. Por ejemplo, si está transfiriendo datos de Twitter usando la fuente de Twitter a través de un canal de memoria a un receptor HDFS, y el nombre del agente id TwitterAgent, entonces.

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS
```

Después de enumerar los componentes del agente, debe describir la (s) fuente (s), sumidero (s) y canal (es) proporcionando valores a sus propiedades.

Descubriendo la fuente

Cada fuente tendrá una lista separada de propiedades. La propiedad denominada "tipo" es común a todas las fuentes y se utiliza para especificar el tipo de la fuente que estamos utilizando.

Junto con el "tipo" de propiedad, es necesario proporcionar los valores de todas las propiedades requeridas de una fuente en particular para configurarlo, como se muestra a continuación.

```
agent_name.sources. source_name.type = value
agent_name.sources. source_name.property2 = value
agent_name.sources. source_name.property3 = value
```

Por ejemplo, si consideramos el origen de Twitter, a continuación se muestran las propiedades a las que debemos proporcionar valores para configurarlo.

```
TwitterAgent.sources.Twitter.type = Twitter (type name)
TwitterAgent.sources.Twitter.consumerKey =
TwitterAgent.sources.Twitter.consumerSecret =
TwitterAgent.sources.Twitter.accessToken =
TwitterAgent.sources.Twitter.accessTokenSecret =
```

Descubriendo el canal

Flume proporciona varios canales para transferir datos entre fuentes y sumideros. Por lo tanto, junto con las fuentes y los canales, es necesario describir el canal utilizado en el agente.

Para describir cada canal, debe establecer las propiedades requeridas, como se muestra a continuación.

```
agent_name.channels.channel_name.type = value
agent_name.channels.channel_name. property2 = value
agent_name.channels.channel_name. property3 = value
```

Por ejemplo, si consideramos el canal de memoria, a continuación se muestran las propiedades a las que debemos proporcionar valores para configurarlo.

```
TwitterAgent.channels.MemChannel.type = memory (type name)
```

Iniciar un agente Flume

Después de la configuración, debemos iniciar el agente de Flume. Se hace de la siguiente manera:

```
$ bin/flume-ng agent --conf ./conf/ -f conf/twitter.conf
Dflume.root.logger=DEBUG,console -n TwitterAgent
```

Donde

- **agent** - Comando para iniciar el agente de Flume
- **--conf, -c <conf>** - Usa el archivo de configuración en el directorio conf
- **-f <archivo>** - Especifica una ruta de archivo de configuración, si falta
- **--name, -n <name>** - Nombre del agente de twitter
- **-D propiedad = valor** - Establece un valor de propiedad del sistema Java.

MANUAL PARA LA INSTALACION DE SOFTWARE

MONGODB

Tabla de contenido

Introducción

Este manual tiene como objetivo, indicar las instrucciones necesarias para realizar la instalación MongoDB motor de base de datos no relacional, que permite almacenar información y es muy usada en ambientes de Big Data. A continuación se encuentra la versión y sistema operativo para el cual está diseñado este manual.

Sistema Operativo	Centos 7
MongoDB	Versión

Componentes de MongoDB

Chef está formado por tres componentes que conforman su arquitectura, estos componentes se enumeran a continuación:

- Mongod: Es el núcleo de la Base de datos
- Mongos: Controlador de Particionamiento
- GridFS: Función que gestiona el almacenamiento.

Cada uno de los componentes anteriormente mencionados permite articular la arquitectura de MongoDB y son necesarios para que este funcione, esto se puede observar en la siguiente imagen.

Prerrequisitos de Instalación

Para realizar la instalación de MongoDB se deben tener en cuenta los siguientes prerrequisitos para evitar inconvenientes en el momento de su instalación.

Se debe preferiblemente crear un usuario que tenga Sudo privilegios, es decir un usuario que no es root, pero que tiene privilegios sobre el sistema.

Instalación de MongoDB

i. Adicionar Repositorio Para MongoDB

Es necesario adicionar el repositorio, ya que no existe dentro de los repositorios predeterminados para CentOS, sin embargo MongoDB mantiene un repositorio dedicado, para agregarlo.

Desde la consola con el editor vi de Linux se crea un archivo de tipo .repo para usar la utilidad yum para realizar la instalación, esto se realiza a través del siguiente comando:

```
$ sudo vi /etc/yum.repos.d/mongodb-org.repo
```

Luego de crear el archivo, visitar la sección de instalación para Red Hat de la documentación disponible en la página de MongoDB, donde se encuentra el repositorio para ser agregado para la última versión que se encuentre disponible. En este caso encontramos el siguiente repositorio:

```
[mongodb-org-3.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc
```

Una vez agregadas las líneas guardar y cerrar el archivo, finalmente para verificar que el repositorio ya está agregado, ingresar el siguiente comando:

```
$ yum repolist
```

Una vez se lanza la instrucción, esta arrojara el siguiente resultado, confirmando que el repositorio de MongoDB ya se ha instalado.

```
Output
. . .
repo id                          repo name
base/7/x86_64                    CentOS-7 - Base
extras/7/x86_64                  CentOS-7 - Extras
mongodb-org-3.2/7/x86_64         MongoDB Repository
updates/7/x86_64                 CentOS-7 - Updates
```

j. Instalación de MongoDB

Para realizar la instalación de MongoDB a través del repositorio creado, se hace uso del comando yum, como se observa a continuación.

```
$ sudo yum install mongodb-org
```

Una vez se ejecuta el comando en consola, este va a preguntar que si desea realizar la instalación del paquete y sus dependencias, `Is this ok [y/N]:` para inicializar la instalación de MogoDB es necesario digitar “YES”, para que este inicie el proceso de instalación.

k. Iniciar el Servicio de MongoDB

Luego de instalar MongoDB, es necesario inicializar el servicio, para esto se ejecuta el siguiente haciendo uso del Systemctl, como se observa a continuación.

```
$ sudo systemctl start mongod
```

Systemctl también permite Detener, Reiniciar un servicio haciendo uso de **reload** o **stop** o **status**, el cual permite ver el estado de un servicio. Una vez MongoDB es iniciado, debe aparecer en la ventana de consola, un mensaje como este:

```
Output

. . .
[initandlisten] waiting for connections on port 27017
```

I. Verificando Instalación de MongoDB

Para verificar la instalación, basta con ejecutar el siguiente comando

```
$ mongo
```

De esta manera si ha quedado instalado correctamente, se puede acceder al Shell de MongoDB una forma de confirmar que el Shell funciona es ejecutar el comando de ayuda `> db.help()` el cual debe arrojar lo siguiente:

```
Output
DB methods:
    db.adminCommand(nameOrDocument) - switches to 'admin' db, and runs command
    db.auth(username, password)
    db.cloneDatabase(fromhost)
    db.commandHelp(name) returns the help for the command
    db.copyDatabase(fromdb, todb, fromhost)
    db.createCollection(name, { size : ..., capped : ..., max : ... } )
    db.createUser(userDocument)
    db.currentOp() displays currently executing operations in the db
    db.dropDatabase()
```

De esta manera concluye la instalación de MongoDB, una buena práctica, debido a que MongoDB es una aplicación de base de datos, es recomendable asegurar que el servicio siempre esté disponible.

Configurar MongoDB Para que Inicie con el Sistema.

Lo primero que se debe hacer es verificar el estado de inicio de servicio de MongoDB, a través del siguiente comando.

```
$ systemctl is-enabled mongod; echo $?
```

Para este comando, se definen dos salidas si la salida es Cero confirma que el servicio está habilitado, si el resultado es Uno confirma que el servicio no está habilitado. Por tanto la salida que se espera al ingresar este comando es:

```
Output

. . .
enabled
0
```

Importar Conjunto de Datos en MongoDB

A continuación se encuentran instrucciones para descargar un conjunto de datos de muestra, ya que MongoDB no incluye datos dentro de sus bases de datos de prueba , para esto se importa un documento JSON que contiene una colección de restaurantes. Para esto ejecutamos el siguiente comando para ubicar un directorio de escritura.

```
$ cd /tmp
```

Ahora se hace uso del comando **curl** el cual hace el enlace del archivo JSON con MongoDB.

```
$ curl -LO https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.js
```

Una vez se descarga el archivo JSON se hace uso del **mongoimport** comando que insertar los datos en la base de datos de prueba, este comando tiene los parámetros de **db,** que define qué base se usa, el parámetro de **collection,** que indica donde se almacenara la información en la base de datos y finalmente el parámetro **file,** que indica sobre que archivo realizar el proceso de importación, finalmente en consola se debe tener un comando similar a:

```
$ mongoimport --db test --collection restaurants --file /tmp/primer-dataset.json
```

La salida al ejecutar el comando, es la confirmación de la importación del data set JSON esto se muestra en consola de la siguiente forma:

```
Output
connected to: localhost
imported 25359 documents
```

Una vez importado el data set, es posible realizar consultas en MongoDB, para comprobar que los datos han sido cargados, para esto se inicia el Shell de MongoDB, con el comando **mongo**, en este caso de manera predeterminada, se está trabajando con la base de datos de prueba de MongoDB, y se puede realizar una consulta de búsqueda realizando la siguiente instrucción.

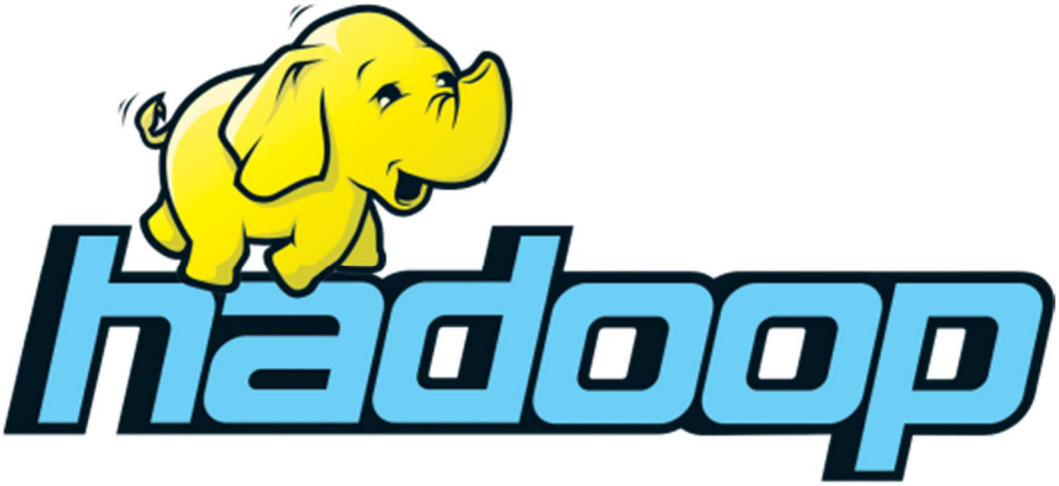
```
> db.restaurants.find().limit( 1 ).pretty()
```

Al realizar esta consulta, el resultado que se espera obtener es el siguiente.

```
Output
{
    "_id" : ObjectId("57e0443b46af7966d1c8fa68"),
    "address" : {
        "building" : "1007",
        "coord" : [
            -73.856077,
            40.848447
        ],
        "street" : "Morris Park Ave",
        "zipcode" : "10462"
    },
    "borough" : "Bronx",
    "cuisine" : "Bakery",
    "grades" : [
        {
            "date" : ISODate("2014-03-03T00:00:00Z"),
            "grade" : "A",
            "score" : 2
        },
        {
            "date" : ISODate("2013-09-11T00:00:00Z"),
            "grade" : "A",
            "score" : 6
        },
        {
            "date" : ISODate("2013-01-24T00:00:00Z"),
            "grade" : "A",
            "score" : 10
        },
        {
            "date" : ISODate("2011-11-23T00:00:00Z"),
            "grade" : "A",
            "score" : 9
        },
        {
            "date" : ISODate("2011-03-10T00:00:00Z"),
            "grade" : "B",
            "score" : 14
        }
    ],
    "name" : "Morris Park Bake Shop",
    "restaurant_id" : "30075445"
}
```

De esta manera termina el proceso de instalación de MongoDB.

MANUAL PARA LA INSTALACION DE SOFTWARE



Hadoop - Configuración de entorno

Versión 2.4.1

Tabla de contenido

Introducción

Este manual busca dar las pautas necesarias para realizar la instalación del entorno de Hadoop. Los cuales componen la infraestructura básica de este software de automatización. A continuación se encuentra los requisitos que se deben tener para lograr llevar acabo la instalación de este servicio.

Sistema Operativo	Centos 7
Hadoop	Hadoop Versión 2.4.1

Prerrequisitos de Instalación

Para realizar la instalación de Hadoop se deben tener en cuenta los siguientes prerrequisitos para evitar inconvenientes en el momento de su instalación.

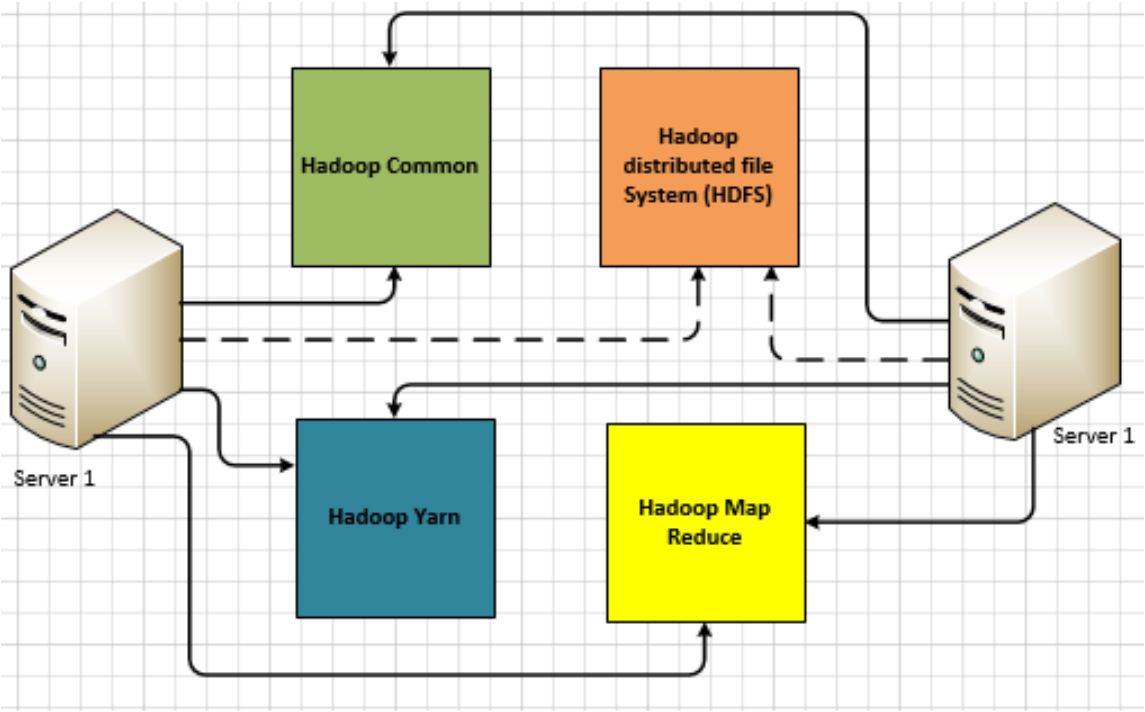
- Es necesario crear un usuario por separado para Hadoop para aislar el sistema de archivos Hadoop del sistema de archivos Unix.
- Se debe tener configurado el Open jdk (Java) en la máquina, de acuerdo a la arquitectectura de la misma.

Componentes de Hadoop

Está formado por cuatro componentes, que permiten generar el despliegue de su entorno, los cuales se mencionan a continuación y muestran sus interacciones en la figura.

- Hadoop Common: Contiene librerías y coord. de Hadoop
- Hadoop Distributed file (HDFS): Es el sistema de archivos distribuidos de Hadoop.
- Hadoop Yarn: es la tecnología de administración de clústeres para Hadoop.
- Hadoop Map Reduce: Es el núcleo de Hadoop, contiene un framework que proporciona un sistema de procesamiento de datos de una forma paralela y distribuida.

Cada uno de los componentes anteriormente mencionados cumple una función específica y son necesarios que los cuatro trabajen en simultáneo para el buen funcionamiento del entorno.

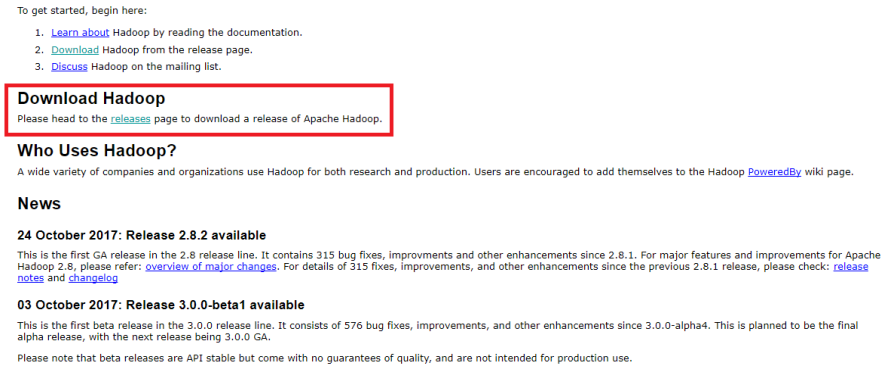


Instalación de Hadoop

m. Descargar Hadoop

Para descargar Hadoop dirigirse a la siguiente URL: http://hadoop.apache.org/

Una vez en la página dar clic en el botón Download Hadoop


To get started, begin here:

1. Learn about Hadoop by reading the documentation.
2. Download Hadoop from the release page.
3. Discuss Hadoop on the mailing list.

Download Hadoop

Please head to the releases page to download a release of Apache Hadoop.

Who Uses Hadoop?

A wide variety of companies and organizations use Hadoop for both research and production. Users are encouraged to add themselves to the Hadoop PoweredBy wiki page.

News

24 October 2017: Release 2.8.2 available

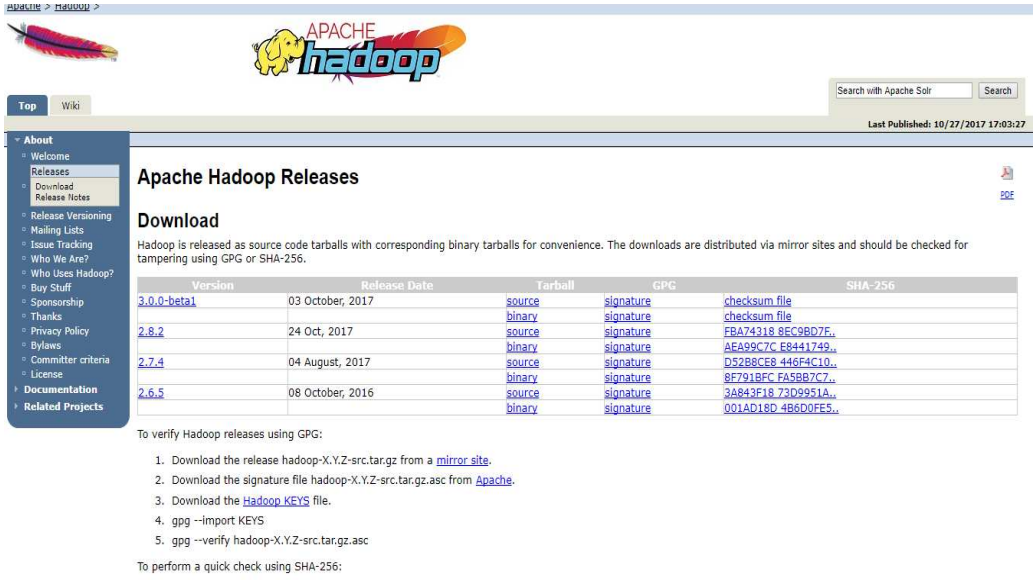
This is the first GA release in the 2.8 release line. It contains 315 bug fixes, improvments and other enhancements since 2.8.1. For major features and improvements for Apache Hadoop 2.8, please refer: overview of major changes. For details of 315 fixes, improvements, and other enhancements since the previous 2.8.1 release, please check: release notes and changelog

03 October 2017: Release 3.0.0-beta1 available

This is the first beta release in the 3.0.0 release line. It consists of 576 bug fixes, improvements, and other enhancements since 3.0.0-alpha4. This is planned to be the final alpha release, with the next release being 3.0.0 GA.

Please note that beta releases are API stable but come with no guarantees of quality, and are not intended for production use.

Al dar clic en el botón se despliega la página donde se observan las fuentes de descargas, para ello seleccionar la que dice releases y dar clic en la versión que se tiene interés.



Seleccionar la Versión y dar clic en source, donde ya se podrá observar el link para su descarga.

n. Configuración de SSH y generación de claves

La configuración de SSH es necesaria para realizar diferentes operaciones en un clúster, como iniciar, detener, distribuir las operaciones del Shell daemon. Para autenticar a diferentes usuarios de Hadoop, es necesario proporcionar pares de claves públicas / privadas para un usuario de Hadoop y compartirlas con diferentes usuarios.

Los siguientes comandos se usan para generar un par de valores clave usando SSH. Copie las claves públicas de id_rsa.pub en authorized_keys y proporcione al propietario permisos de lectura y escritura para el archivo authorized_keys, respectivamente.

```
$ ssh-keygen -t rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

o. Configuración de Hadoop luego de su descarga

Luego de ser Descargado el paquete de Hadoop se debe extraer Hadoop 2.4.1 de la base del software Apache utilizando los siguientes comandos.

```
$ su
password:
# cd /usr/local
# wget http://apache.claz.org/hadoop/common/hadoop-2.4.1/
hadoop-2.4.1.tar.gz
# tar xzf hadoop-2.4.1.tar.gz
# mv hadoop-2.4.1/* to hadoop/
# exit
```

p. Modos de operación de Hadoop

Una vez que haya descargado Hadoop, puede operar su clúster Hadoop en uno de los tres modos admitidos:

- Modo local / autónomo: después de descargar Hadoop en su sistema, de forma predeterminada, se configura en un modo independiente y se puede ejecutar como un único proceso de Java.

- Modo pseudo distribuido: es una simulación distribuida en una sola máquina. Cada daemon de Hadoop, como hdfs, hilo, MapReduce, etc., se ejecutará como un proceso java independiente. Este modo es útil para el desarrollo.

- Modo completamente distribuido: este modo se distribuye por completo con un mínimo de dos o más máquinas como un clúster. Nos toparemos con este modo en detalle en los próximos capítulos.

Instalación de Hadoop modo completamente distribuido

En esta parte del tutorial se explica la configuración del Hadoop completamente distribuido, Para esto se utiliza tres sistemas; uno que ara la función de maestro y los otros dos que harán la función de esclavos cada sistema con su respectiva dirección ip.

- Hadoop Master: 192.168.1.xxx (Hadoop-master)
- Esclavo de Hadoop: 192.168.1.xxx (Hadoop-slave-1)
- Esclavo de Hadoop: 192.168.1.xxx (Hadoop-slave-2)

i. Mapeo de los nodos

Debe editar el archivo de **hosts** en **/ etc /** folder en todos los nodos, especifique la dirección IP de cada sistema seguido de sus nombres de host.

```
# vi /etc/hosts
enter the following lines in the /etc/hosts file.
192.168.1.109 hadoop-master
192.168.1.145 hadoop-slave-1
192.168.56.1 hadoop-slave-2
```

j. Configurar el inicio de sesión basado en clave

Configure ssh en cada nodo para que puedan comunicarse entre sí sin necesidad de una contraseña.

```
# su hadoop
$ ssh-keygen -t rsa
$ ssh-copy-id -i ~/.ssh/id_rsa.pub tutorialspoint@hadoop-master
$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop_tp1@hadoop-slave-1
$ ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop_tp2@hadoop-slave-2
$ chmod 0600 ~/.ssh/authorized_keys
$ exit
```

k. Instalando Hadoop

En el servidor maestro, descargue e instale Hadoop usando los siguientes comandos.

```
# mkdir /opt/hadoop
# cd /opt/hadoop/
# wget http://apache.mesi.com.ar/hadoop/common/hadoop-1.2.1/hadoop-1.2.0.tar.gz
# tar -xzf hadoop-1.2.0.tar.gz
# mv hadoop-1.2.0 hadoop
# chown -R hadoop /opt/hadoop
# cd /opt/hadoop/hadoop/
```

l. Configurando Hadoop

Debe configurar el servidor de Hadoop realizando los siguientes cambios como se indica a continuación.

Core-site.xml

Abra el archivo core-site.xml y edítelo como se muestra a continuación.

```
<configuration>
   <property>
      <name>fs.default.name</name>
      <value>hdfs://hadoop-master:9000/</value>
   </property>
   <property>
      <name>dfs.permissions</name>
      <value>false</value>
   </property>
</configuration>
```

hdfs-site.xml

Abra el archivo hdfs-site.xml y edítelo como se muestra a continuación.

```
<configuration>
   <property>
      <name>dfs.data.dir</name>
      <value>/opt/hadoop/hadoop/dfs/name/data</value>
      <final>true</final>
   </property>

   <property>
      <name>dfs.name.dir</name>
      <value>/opt/hadoop/hadoop/dfs/name</value>
      <final>true</final>
   </property>

   <property>
      <name>dfs.replication</name>
      <value>1</value>
   </property>
</configuration>
```

mapred-site.xml

Abra el archivo mapred-site.xml y edítelo como se muestra a continuación.

```
<configuration>
   <property>
      <name>mapred.job.tracker</name>
      <value>hadoop-master:9001</value>
   </property>
</configuration>
```

hadoop-env.sh

Abra el archivo hadoop-env.sh y edite JAVA_HOME, HADOOP_CONF_DIR y HADOOP_OPTS como se muestra a continuación.

Nota: Establezca JAVA_HOME según la configuración de su sistema.

```
export JAVA_HOME=/opt/jdk1.7.0_17 export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
```

m. Instalación de Hadoop en servidores esclavos

Instale Hadoop en todos los servidores esclavos siguiendo los comandos dados.

```
# su hadoop
$ cd /opt/hadoop
$ scp -r hadoop hadoop-slave-1:/opt/hadoop
$ scp -r hadoop hadoop-slave-2:/opt/hadoop
```

n. Configurando Hadoop en servidor maestro

Abra el servidor maestro y configúrelo siguiendo los comandos dados.

```
# su hadoop
$ cd /opt/hadoop/hadoop
```

Configurando nodo maestro

```
$ vi etc/hadoop/masters
hadoop-master
```

Configurando el nodo esclavo

```
$ vi etc/hadoop/slaves
hadoop-slave-1
hadoop-slave-2
```

Formato de nodo de nombre en Hadoop master

```
# su hadoop
$ cd /opt/hadoop/hadoop
$ bin/hadoop namenode -format
```

```
11/10/14 10:58:07 INFO namenode.NameNode: STARTUP_MSG: /************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = hadoop-master/192.168.1.109
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 1.2.0
STARTUP_MSG: build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-
STARTUP_MSG: java = 1.7.0_71 ************************************************
.....
.....
..... 11/10/14 10:58:08 INFO common.Storage: Storage directory /opt/h
SHUTDOWN_MSG: /************************************************************ SHUTDOWN
```

o. Iniciando los servicios de Hadoop

El siguiente comando es iniciar todos los servicios de Hadoop en Hadoop-Master.

```
$ cd $HADOOP_HOME/sbin
$ start-all.sh
```

MANUAL PARA LA INSTALACION DE SOFTWARE

Instalación y ejecución de clúster MPI

Tabla de contenido

Introducción

Este manual busca dar las pautas necesarias para realizar la instalación del entorno del clúster MPI. Los cuales son necesarios para el entorno componen la infraestructura básica de este software de automatización. A continuación se encuentra los requisitos que se deben tener para lograr llevar acabo la instalación de este servicio.

Sistema Operativo	Centos 7
Clúster MPI	MPI

Prerrequisitos de Instalación

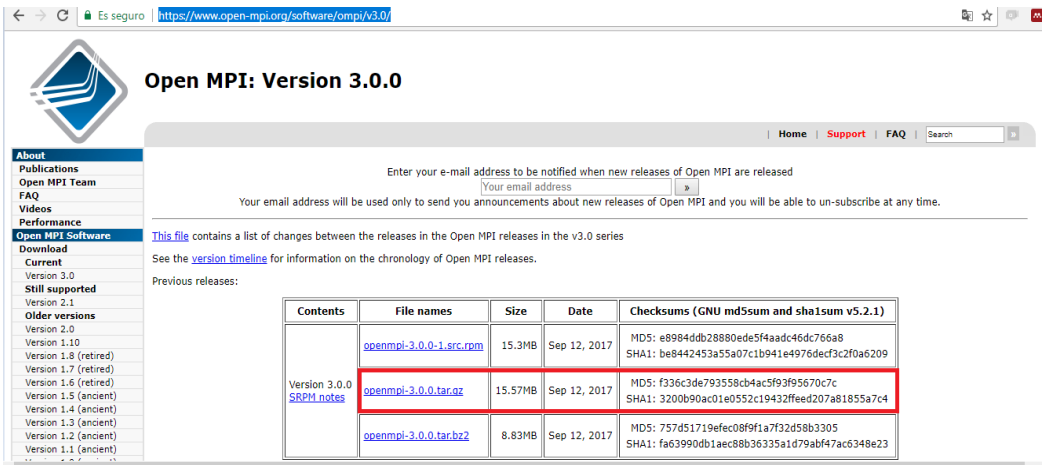
Si no ha instalado MPICH2 en cada una de las máquinas, siga los pasos que se detallan a continuación.

Instalación de Clúster MPI

q. Descargar MPI

Para descargar Open MPI dirigirse a la siguiente URL: https://www.open-mpi.org/software/ompi/v3.0/

Una vez en la página dar clic en el botón Download



Al dar clic en file names tal y como se muestra en la imagen se debe automáticamente descargar tu open MPI.

Configuración de host archivo

Tendrá que comunicarse entre las computadoras y no desea escribir las direcciones IP cada cierto tiempo. En cambio, puede dar un nombre a los distintos nodos de la red con los que desea comunicarse. Hosts El sistema operativo del dispositivo usa el archivo para mapear nombres de host a direcciones IP.

```
$ cat /etc/hosts

127.0.0.1       localhost
172.50.88.34    client
```

El cliente aquí es la máquina que le gustaría hacer sus cálculos con. Del mismo modo, haz lo mismo master en el cliente.

Crear un nuevo usuario

Aunque puede operar su clúster con su cuenta de usuario existente, le recomendaría crear una nueva para mantener nuestras configuraciones simples. Permítanos crear un nuevo usuario mpiuser. Crea nuevas cuentas de usuario con el mismo nombre de usuario en todas las máquinas para mantener las cosas simples.

```
$ sudo adduser mpiuser
```

Sigue las indicaciones y estarás bien. No use el useradd comando para crear un nuevo usuario ya que eso no crea un hogar separado para los nuevos usuarios.

Configuración del SSH

Sus máquinas hablarán a través de la red a través de SSH y compartirán datos a través de NFS, sobre lo cual hablaremos un poco más adelante.

```
$ sudo apt-get install openssh-server
```

Y justo después de eso, inicie sesión con su cuenta recién creada

```
$ su - mpiuser
```

Como el ssh servidor ya está instalado, debe poder iniciar sesión en otras máquinas ssh username@hostname, donde se le pedirá que ingrese la contraseña de username. Para permitir un inicio de sesión más fácil, generamos claves y las copiamos a la lista de otras máquinas authorized_keys.

```
$ ssh-keygen -t dsa
```

También puede generar claves RSA. Pero nuevamente, es totalmente tu decisión. Si quieres más seguridad, ve con RSA. De lo contrario, DSA debería estar bien. Ahora, agregue la clave generada a cada una de las otras computadoras. En nuestro caso, la máquina cliente.

```
$ ssh-copy-id client #ip-address may also be used
```

Realice el paso anterior para cada una de las máquinas cliente y su propio usuario (servidor local).

Esto se configurará openssh-serverpara que se comunique de forma segura con las máquinas del cliente. Sshtodas las máquinas una vez, para que se agreguen a su lista de known_hosts. Este es un paso muy simple pero esencial que falla y que sin contraseña sshserá un problema.

Ahora, para habilitar ssh sin contraseña,

```
$ eval `ssh-agent`
$ ssh-add ~/.ssh/id_dsa
```

Ahora, suponiendo que haya agregado correctamente sus claves a otras máquinas, debe poder iniciar sesión en otras máquinas sin ningún aviso de contraseña.

```
$ ssh client
```

Instalación de Hadoop modo completamente distribuido

En esta parte del tutorial se explica la configuración del Hadoop completamente distribuido, Para esto se utiliza tres sistemas; uno que ara la función de maestro y los otros dos que harán la función de esclavos cada sistema con su respectiva dirección ip.

- Hadoop Master: 192.168.1.xxx (Hadoop-master)
- Esclavo de Hadoop: 192.168.1.xxx (Hadoop-slave-1)
- Esclavo de Hadoop: 192.168.1.xxx (Hadoop-slave-2)

Configuración de NFS

Usted comparte un directorio a través de NFS en el maestro que el cliente monta para intercambiar datos.

p. Servidor NFS

Instale los paquetes requeridos por.

```
$ sudo apt-get install nfs-kernel-server
```

Ahora, (suponiendo que aún esté conectado mpiuser), cloud creamos una carpeta con el nombre que compartiremos en la red.

```
$ mkdir cloud
```

Para exportar el cloud directorio, crea una entrada en/etc/exports

```
$  cat /etc/exports
/home/mpiuser/cloud *(rw,sync,no_root_squash,no_subtree_check)
```

Aquí, en lugar de *usted, puede especificar específicamente la dirección IP a la que desea compartir esta carpeta. Pero esto hará que nuestro trabajo sea más fácil.

- RW: Esto es para habilitar la opción de lectura y escritura. ro es para solo lectura.
- sincronización: esto aplica los cambios en el directorio compartido solo después de que se hayan confirmado los cambios.
- NO_SUBTREE_CHECK: esta opción evita la verificación del subárbol. Cuando un directorio compartido es el subdirectorio de un sistema de archivos más grande, NFS realiza escaneos de cada uno de los directorios superiores para verificar sus permisos y detalles. Deshabilitar la verificación de subárbol puede aumentar la confiabilidad de NFS, pero reducir la seguridad.
- NO_ROOT_SQUASH: Esto permite que la cuenta de root se conecte a la carpeta.

Después de haber realizado la entrada, ejecute lo siguiente.

```
$ exportfs -a
```

Ejecute el comando anterior, cada vez que realice un cambio en /etc/exports.

Si es necesario, reinicie el NFS servidor.

```
$ sudo service nfs-kernel-server restart
```

q. NFS cliente

Instale los paquetes requeridos.

```
$ sudo apt-get install nfs-common
```

Crea un directorio en la máquina del cliente con el samename cloud

```
$ mkdir cloud
```

Y ahora, monta el directorio compartido como

```
$ sudo mount -t nfs master:/home/mpiuser/cloud ~/cloud
```

Para verificar los directorios montados

```
$ df -h
Filesystem                          Size  Used Avail Use% Mounted on
master:/home/mpiuser/cloud  49G   15G   32G  32% /home/mpiuser/cloud
```

Para que el montaje sea permanente y no tenga que montar manualmente el directorio compartido cada vez que reinicie el sistema, puede crear una entrada en su tabla de sistemas de archivos, es decir, un /etc/fstab archivo como este:

```
$ cat /etc/fstab
#MPI CLUSTER SETUP
master:/home/mpiuser/cloud /home/mpiuser/cloud nfs
```

Ejecutar programas MPI

Por consideración, tomemos un programa de muestra, que viene junto con el paquete de instalación MPICH2 mpich2/examples/cpi. Tomaremos este ejecutable e intentaremos ejecutarlo de forma paralela.

O si desea compilar su propio código, cuyo nombre, por ejemplo mpi_sample.c, es, lo compilará como se indica a continuación, para generar un ejecutable mpi_sample.

```
$ mpicc -o mpi_sample mpi_sample.c
```

Primero copie su ejecutable en el directorio compartido cloud o mejor aún, compile su código dentro del directorio compartido de NFS.

```
$ cd cloud/
$ pwd
/home/mpiuser/cloud
```

Para ejecutarlo solo en tu máquina, lo haces

```
$ mpirun -np 2 ./cpi # No. of processes = 2
```

Ahora, para ejecutarlo dentro de un clúster

```
$ mpirun -np 5 -hosts client,localhost ./cpi
#hostnames can also be substituted with ip addresses.
```

O especifique lo mismo en un archivo de host

```
$ mpirun -np 5 --hostfile mpi_file ./cpi
```

MANUAL PARA LA INSTALACION DE SOFTWARE

Spark - Configuración de entorno

Tabla de contenido

Introducción

Este manual busca dar las pautas necesarias para realizar la instalación de apache Spark; que es una tecnología de computación en clúster. Está basada en Hadoop y se diseñó para resistir una amplia cantidad de información. A continuación se encuentra los requisitos que se deben tener para lograr llevar acabo la instalación de este servicio.

Sistema Operativo	Centos 7
Spark	Apache Spark

Prerrequisitos de Instalación

Para realizar la instalación de Spark se deben tener en cuenta los siguientes prerrequisitos para evitar inconvenientes en el momento de su instalación.

- Se debe tener configurado el Open jdk (Java) en la máquina.
- En lo posible se debe tener un intérprete de Python.

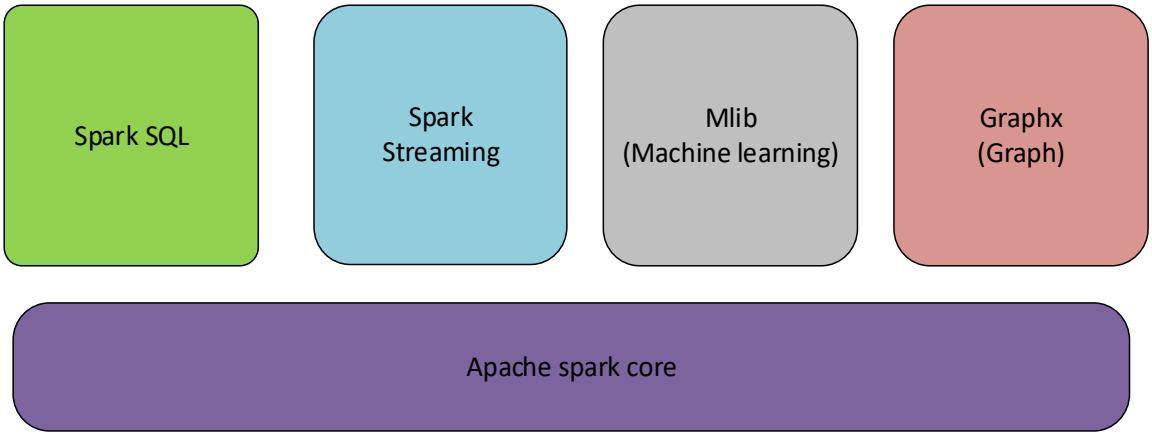
Componentes de Spark

Está formado por cinco componentes, que permiten generar el despliegue de su entorno, y el buen funcionamiento para el procesamiento de grandes cantidades de información. Y también tiene ventajas para su uso los cuales se mencionan a continuación.

Velocidad: Spark ayuda a ejecutar una aplicación en el clúster Hadoop, hasta 100 veces más rápido en la memoria y 10 veces más rápido cuando se ejecuta en el disco. Esto es posible al reducir el número de operaciones de lectura / escritura en el disco. Almacena los datos intermedios de procesamiento en la memoria.

Admite varios idiomas: Spark proporciona API integradas en Java, Scala o Python. Por lo tanto, puede escribir aplicaciones en diferentes idiomas. Aparece Spark con 80 operadores de alto nivel para consulta interactiva.

Análisis avanzado: Spark no solo es compatible con 'Mapa' y 'reducir'. También admite consultas SQL, transmisión de datos, aprendizaje automático (ML) y algoritmos de gráficos. Cada uno de los componentes anteriormente mencionados

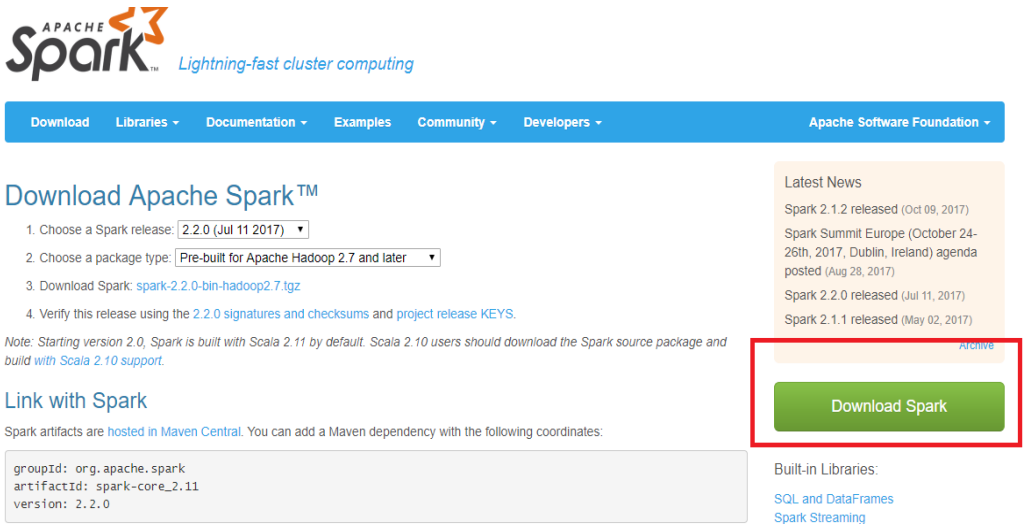


Instalación de Spark

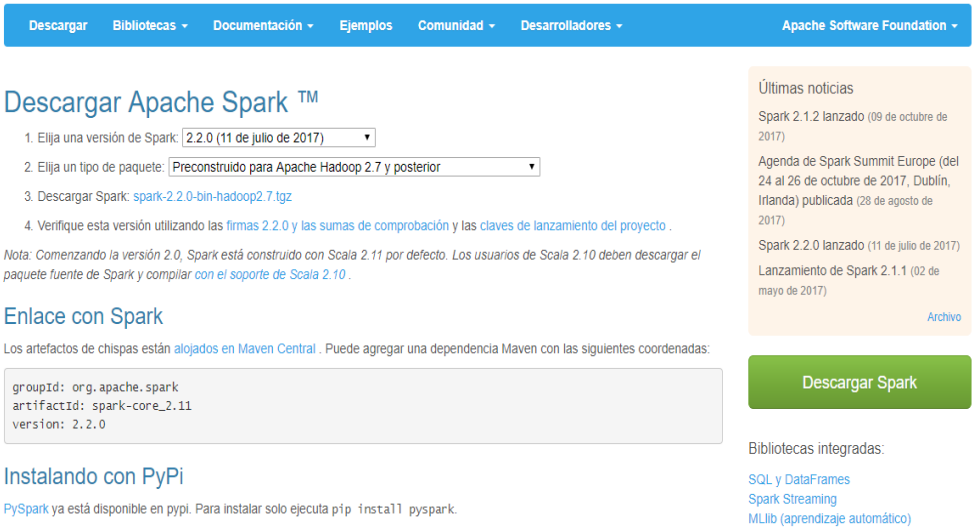
r. Descargar apache Spark

Para descargar Spark se debe dirigir a la siguiente URL: https://spark.apache.org/downloads.html

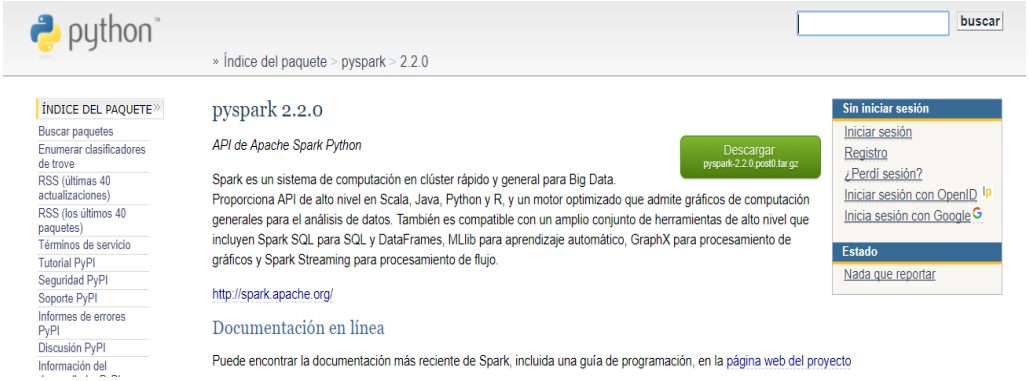
Una vez en la página dar clic en el botón Download Spark



Al dar clic en el botón se despliega la página donde se observan las fuentes de descargas, para ello seleccionar la versión que se tiene interés.



Seleccionar la Versión y dar clic en source, donde ya se podrá observar el link para su descarga.



Verificar la instalación de Scala

Debería implementar el lenguaje Scala para implementar Spark. Entonces, verifiquemos la instalación de Scala usando el siguiente comando.

```
$scala -version
```

Si Scala ya está instalado en su sistema, puede ver la siguiente respuesta:

```
Scala code runner version 2.11.6 -- Copyright 2002-2013, LAMP/EPFL
```

En caso de que no tenga instalado Scala en su sistema, continúe con el siguiente paso para la instalación de Scala.

Descarga Scala

Descargue la última versión de Scala visitando el siguiente enlace Descargar Scala. Para este tutorial, estamos usando la versión scala-2.11.6. Después de la descarga, encontrará el archivo .tar de Scala en la carpeta de descargas.

Instalando Scala

Extraiga el archivo tar de Scala

Escriba el siguiente comando para extraer el archivo tar de Scala.

```
$ tar xvf scala-2.11.6.tgz
```

r. Mover los archivos del software Scala

Utilice los siguientes comandos para mover los archivos del software Scala al directorio respectivo (/ usr / local / Scala).

```
$ su -
Password:
# cd /home/Hadoop/Downloads/
# mv scala-2.11.6 /usr/local/scala
# exit
```

s. Establecer PATH para Scala

Use el siguiente comando para configurar PATH para Scala.

```
$ export PATH = $PATH:/usr/local/scala/bin
```

t. Verificación de la instalación de Scala

Después de la instalación, es mejor verificarlo. Use el siguiente comando para verificar la instalación de Scala.

```
$scala -version
```

Si Scala ya está instalado en su sistema, puede ver la siguiente respuesta:

```
Scala code runner version 2.11.6 -- Copyright 2002-2013, LAMP/EPFL
```

Descargar apache Spark

Descargue la última versión de Spark visitando el siguiente enlace Descargar Spark. Para esto puede usar la versión que se prefiera. Después de descargarlo, encontrará el archivo Spark tar en la carpeta de descargas.

Instalación de Spark

Para esto es necesario rea lisar los siguientes pasos para su instalación.

a. Extracción de Spark tar

El siguiente comando para extraer el archivo Spark tar.

```
$ tar xvf spark-1.3.1-bin-hadoop2.6.tgz
```

b. Moviendo archivos de software Spark

Los siguientes comandos para mover los archivos del software Spark al directorio respectivo (/ usr / local / Spark).

```
$ su -
Password:

# cd /home/Hadoop/Downloads/
# mv spark-1.3.1-bin-hadoop2.6 /usr/local/spark
# exit
```

c. Configurando el ambiente para Spark

Agregue la siguiente línea al archivo ~ /.bashrc. Significa agregar la ubicación, donde se encuentra el archivo de software de chispa a la variable PATH.

```
export PATH = $PATH:/usr/local/spark/bin
```

Use el siguiente comando para obtener el archivo ~ / .bashrc.

```
$ source ~/.bashrc
```

Verificar la instalación de Spark

Escriba el siguiente comando para abrir el shell Spark.

```
$spark-shell
```

Si Spark se instala con éxito, encontrará la siguiente salida.

```
Spark assembly has been built with Hive, including Datanucleus jars on classpath
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
15/06/04 15:25:22 INFO SecurityManager: Changing view acls to: hadoop
15/06/04 15:25:22 INFO SecurityManager: Changing modify acls to: hadoop
15/06/04 15:25:22 INFO SecurityManager: SecurityManager: authentication disabled;
   ui acls disabled; users with view permissions: Set(hadoop); users with modify permissior
15/06/04 15:25:22 INFO HttpServer: Starting HTTP Server
15/06/04 15:25:23 INFO Utils: Successfully started service 'HTTP class server' on port 4329
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.4.0
      /_/

Using Scala version 2.10.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_71)
Type in expressions to have them evaluated.
Spark context available as sc
scala>
```

MANUAL PARA LA INSTALACION DE GRAFANA

GRAFANA VERSION 4.6.2

Tabla de contenido

Introducción

Este manual tiene como objetivo, indicar las instrucciones necesarias para realizar la instalación Grafana, herramienta de visualización de datos, que se especializa en tareas de Monitoreo y Explotación de logs, muy usada en ambientes de Big Data par la monitorización de recursos. A continuación se encuentra la versión y sistema operativo para el cual está diseñado este manual.

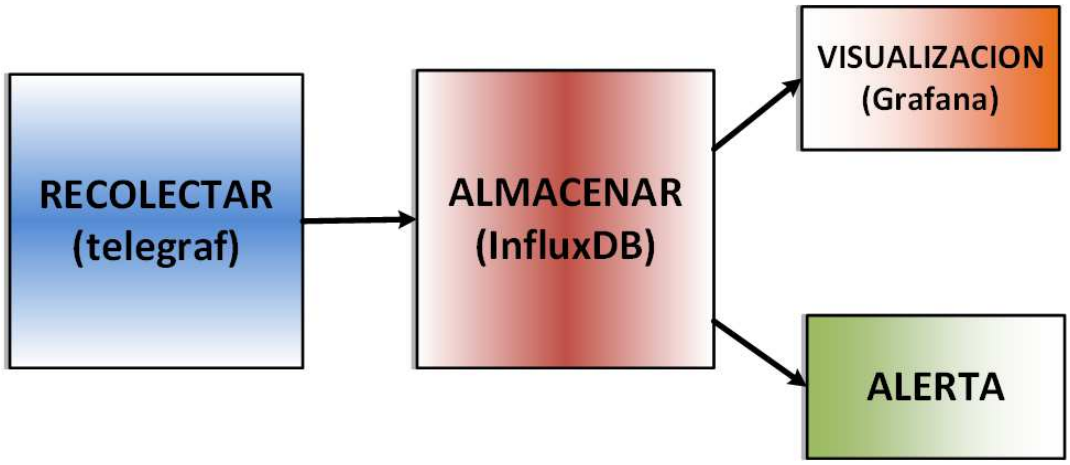
Sistema Operativo	Centos 7
Grafana	Versión 4.6.2

Componentes de Grafana

Grafana al ser una herramienta de visualización, enfocada en monitorización de recursos basa su arquitectura en 4 componentes que debe tener este tipo de herramientas, que se encuentran a continuación:

- Recolección: Telegraf, es el componente que se encarga de la recolección de datos en una serie de tiempos , esto permite capturar eventos de diversas fuentes de Datos
- Almacenamiento: Permite almacenar lo datos recolectados por Telegraf, en este caso el motor de base de datos de InfluxDB.
- Visualización: Es la capa de cara al usuario, en este caso está representada por la interfaz gráfica de Grafana, la cual realiza consultas a la base de datos y permite visualizar los mismos, a través de un Dashboard (Tablero de control) generando métricas, que permiten el control de recursos.
- Generación de Alertas: Son parametrizadas a través de Grafana y permiten delimitar parámetros que permitan notificar comportamientos sobre los recursos que se administran.

Cada uno de los componentes anteriormente mencionados permite articular la arquitectura de MongoDB y son necesarios para que este funcione, esto se puede observar en la siguiente imagen.



Prerrequisitos de Instalación

Para realizar la instalación de Grafana es necesario que el firewall este configurado para aceptar tráfico por los puertos 3000 y 8083, a continuación se encuentra el comando que debe ejecutarse para configurar el firewall.

```
firewall-cmd --permanent --zone=public --add-port=8086/tcp
firewall-cmd --permanent --zone=public --add-port=8083/tcp
firewall-cmd --reload
```

Si se quiere configurar el Puerto 3000, se reemplaza el número de puerto y se vuelve a ejecutar el comando.

Instalación de InfluxDB

s. Adicionar Repositorio Para InfluxDB

Es necesario adicionar el repositorio, ya que no existe dentro de los repositorios predeterminados para Centos, sin embargo InfluxDB mantiene un repositorio dedicado, para agregarlo.

Desde la consola basta con ejecutar el siguiente comando para agregar el repositorio de InfluxDB

```
cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo
[influxdb]
name = InfluxDB Repository - RHEL \$releasever
baseurl = https://repos.influxdata.com/rhel/\$releasever/\$basearch/stable
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdb.key
EOF
```

t. Instalación de InfluxDB

Luego de crear el archivo, se debe guardar y cerrar, luego de esto se procede a realizar la instalación, usando el siguiente comando:

```
sudo yum install influxdb
```

u. Iniciar el Servicio de InfluxDB

Al instalar InfluxDB, se puede inicializar el servicio que este instala, para esto se hace uso del siguiente comando:

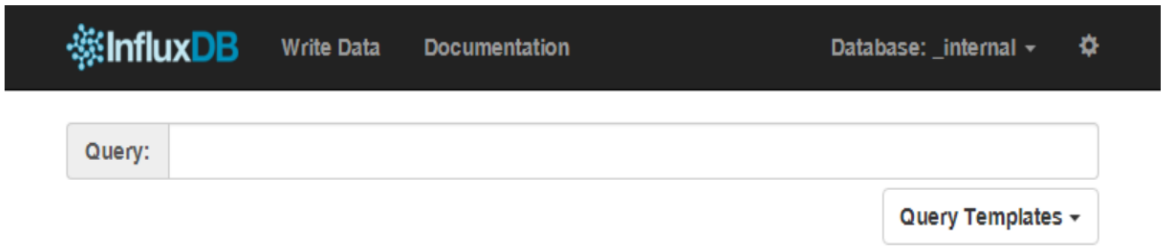
```
sudo /bin/systemctl start  influxdb.service
```

Systemctl también permite Detener, Reiniciar un servicio haciendo uso de **reload** o **stop** o **status**, el cual permite ver el estado de un servicio.

Una buena práctica, es asegurar que el servicio de InfluxDB siempre esté disponible, para esto se debe habilitar el servicio con el inicio del sistema, haciendo uso del siguiente comando:

```
sudo /bin/systemctl enable influxdb.service
```

Una vez iniciado el servicio, desde consola se puede realizar la petición http para lanzar la interfaz web de InfluxDB como se muestra a continuación. De esta manera finaliza la instalación de InfluxDB



Instalación de Telegraf

a. Adicionar Repositorio Para Telegraf

Es necesario adicionar el repositorio, ya que no existe dentro de los repositorios predeterminados para Centos, sin embargo Telegraf mantiene un repositorio dedicado, para agregarlo.

Desde la consola basta con ejecutar el siguiente comando para agregar el repositorio de Telegraf.

```
cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo
[influxdb]
name = InfluxDB Repository - RHEL \$releasever
baseurl = https://repos.influxdata.com/rhel/\$releasever/\$basearch/stable
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdb.key
EOF
```

b. Instalación de Telegraf

Después de crear el archivo, se debe guardar y cerrar, posteriormente se procede a realizar la instalación, usando el siguiente comando:

```
sudo yum install telegraf
```

c. Iniciar el Servicio de Telegraf

Al instalar Telegraf, se puede inicializar el servicio que este instala, para esto se hace uso del siguiente comando:

```
systemctl start Telegraf
```

Systemctl también permite Detener, Reiniciar un servicio haciendo uso de **reload** o **stop** o **status**, el cual permite ver el estado de un servicio.

Una buena práctica, es asegurar que el servicio de InfluxDB siempre esté disponible, para esto se debe habilitar el servicio con el inicio del sistema, haciendo uso del siguiente comando:

```
Sudo /bin/systemctl enable telegraf.service
```

Una vez iniciado el servicio, se culmina la instalación de Telegraf.

Instalación de Grafana

a. Adicionar Repositorio Para Grafana

Es necesario adicionar el repositorio, ya que no existe dentro de los repositorios predeterminados para Centos, sin embargo Grafana mantiene un repositorio dedicado, para agregarlo.

Desde la consola basta con ejecutar el siguiente comando para agregar el repositorio de Grafana.

```
cat <<EOF | sudo tee /etc/yum.repos.d/grafana.repo
[grafana]
name=grafana
baseurl=https://packagecloud.io/grafana/stable/el/6/$basearch
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packagecloud.io/gpg.key https://grafanarel.s3.amazonaws.com/RPM-GPG-KEY-grafana
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
EOF
```

b. Instalación de Grafana

Después de crear el archivo, se debe guardar y cerrar, posteriormente se procede a realizar la instalación, usando el siguiente comando:

```
sudo yum install grafana
```

c. Iniciar el Servicio de Grafana

Al instalar Grafana, se puede inicializar el servicio que este instala, para esto se hace uso del siguiente comando:

```
sudo /bin/systemctl start grafana-server.service
```

Systemctl también permite Detener, Reiniciar un servicio haciendo uso de **reload** o **stop** o **status**, el cual permite ver el estado de un servicio.

Una buena práctica, es asegurar que el servicio de Grafana siempre esté disponible, para esto se debe habilitar el servicio con el inicio del sistema, haciendo uso del siguiente comando:

```
sudo /bin/systemctl enable grafana-server.service
```

Una vez iniciado el servicio, desde consola se puede realizar la petición http://ip:3000 para lanzar la interfaz web de Grafana como se muestra a continuación. De esta manera finaliza la instalación de Grafana.