UNIVERSIDAD ANDRES BELLO

**FACULTAD DE INGENIERÍA**

**INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA**



Proyecto Big Data

Seminario de Título 1

**David Córdova Cordero**

**VIÑA DEL MAR – CHILE**

**Mayo 2019**

Contenidos

[Introducción 10](#_Toc10751093)

[Definición de Big data 11](#_Toc10751094)

[Atributos de Big Data 11](#_Toc10751095)

[Análisis de la problemática. 14](#_Toc10751096)

[Técnica de los cinco por qué. 14](#_Toc10751097)

[Diagrama Ishikawa 15](#_Toc10751098)

[Objetivo del proyecto 16](#_Toc10751099)

[Descripción del problema 16](#_Toc10751100)

[Objetivo del proyecto 17](#_Toc10751101)

[Objetivo general 17](#_Toc10751102)

[Objetivos específicos 17](#_Toc10751103)

[Diagrama de alto nivel 19](#_Toc10751104)

[Arquitectura de solución 19](#_Toc10751105)

[Solución ideal 21](#_Toc10751106)

[Limitaciones 21](#_Toc10751107)

[Restricciones 22](#_Toc10751108)

[Alternativas de solución 22](#_Toc10751109)

[Procedimientos manuales 22](#_Toc10751110)

[Cambios en procedimientos actuales 22](#_Toc10751111)

[Alternativas disponibles en el mercado. 23](#_Toc10751112)

[Solución propuesta 23](#_Toc10751113)

[Situación Futura 23](#_Toc10751114)

[Factibilidad operativa 24](#_Toc10751115)

[Factibilidad técnica 24](#_Toc10751116)

[Factibilidad Económica 25](#_Toc10751117)

[Historias de Usuario 25](#_Toc10751118)

[Requerimientos de alto nivel 25](#_Toc10751119)

[Requerimientos Sistema: 25](#_Toc10751120)

[Requerimientos Funcionales: 26](#_Toc10751121)

[Requerimientos No Funcionales: 26](#_Toc10751122)

[Delimitación 26](#_Toc10751123)

[El desarrollo del proyecto incluye: 27](#_Toc10751124)

[Plan de proyecto 29](#_Toc10751125)

[Dirección de proyecto 29](#_Toc10751126)

[Tabla de roles 31](#_Toc10751127)

[Posteriormente, se desarrollan varios documentos: 31](#_Toc10751128)

[Scrum 31](#_Toc10751129)

[Product backlog del sprint 32](#_Toc10751130)

[Sprint backlog 33](#_Toc10751131)

[Metodología de desarrollo 33](#_Toc10751132)

[Metodología iterativo incremental 34](#_Toc10751133)

[Planificación 34](#_Toc10751134)

[Plan de entregables 35](#_Toc10751135)

[Plan de pruebas 36](#_Toc10751136)

[Gestión de la configuración 37](#_Toc10751137)

[Gestión de línea Base 37](#_Toc10751138)

[Control de versiones 39](#_Toc10751139)

[Gestión de cambios 39](#_Toc10751140)

[Ambiente de desarrollo, pruebas y producción 41](#_Toc10751141)

[Hardware 42](#_Toc10751142)

[Herramientas de construcción 42](#_Toc10751143)

[Ambiente de prueba 42](#_Toc10751144)

[Ambiente de producción 43](#_Toc10751145)

[Plan de gestión de riegos 43](#_Toc10751146)

[Riesgos que se presentaron en el área técnica del proyecto(Contingencia) 47](#_Toc10751147)

[Arquitectura de un ambiente de big data 49](#_Toc10751148)

[Arquitectura de Big Data 49](#_Toc10751149)

[Arquitectura de un ambiente de Big data 49](#_Toc10751150)

[Ingreso de datos 50](#_Toc10751151)

[Gestión de datos 50](#_Toc10751152)

[Tiempo real de procesamiento 50](#_Toc10751153)

[Análisis de datos 50](#_Toc10751154)

[Bases de datos NoSQL 51](#_Toc10751155)

[Distribuido 51](#_Toc10751156)

[Escalabilidad horizontal 51](#_Toc10751157)

[Construido para grandes volúmenes 51](#_Toc10751158)

[Modelos de datos no relacionales 52](#_Toc10751159)

[No hay definiciones de esquema 52](#_Toc10751160)

[Hadoop y big data 52](#_Toc10751161)

[Ventajas 52](#_Toc10751162)

[Arquitectura hadoop 52](#_Toc10751163)

[Hadoop 53](#_Toc10751164)

[Arquitectura hadoop 53](#_Toc10751165)

[MapReduce 53](#_Toc10751166)

[Función Map 53](#_Toc10751167)

[Función Reduce 54](#_Toc10751168)

[Proceso MapReduce 54](#_Toc10751169)

[HDFS 54](#_Toc10751170)

[Chukwa 55](#_Toc10751171)

[Diagrama de Chukwa 55](#_Toc10751172)

[55](#_Toc10751173)

[Sqoop 56](#_Toc10751174)

[Flume 56](#_Toc10751175)

[Flujo de datos 56](#_Toc10751176)

[Avro 57](#_Toc10751177)

[Hive 57](#_Toc10751178)

[Pig 58](#_Toc10751179)

[Pig posee las siguientes características: 58](#_Toc10751180)

[Facilidad de programación 58](#_Toc10751181)

[Oportunidades de optimización 58](#_Toc10751182)

[Extensibilidad 58](#_Toc10751183)

[Para la implementación de las herramientas 58](#_Toc10751184)

[Ciclo de vida iterativo 59](#_Toc10751185)

[Ciclo de vida incremental 59](#_Toc10751186)

[Modelo de desarrollo 59](#_Toc10751187)

[Modelo Iterativo y acoplamiento al proyecto 59](#_Toc10751188)

[Ventaja 60](#_Toc10751189)

[Desventajas 60](#_Toc10751190)

[Requerimientos de Hardware y software 60](#_Toc10751191)

[Pasos que seguir en la instalación de hadoop y sus herramientas 61](#_Toc10751192)

[Modelo Iterativo-Incremental en la instalación y uso de hadoop 61](#_Toc10751193)

[Fase de Análisis 62](#_Toc10751194)

[Diseño **¡Error! Marcador no definido.**](#_Toc10751195)

[Codificación 62](#_Toc10751196)

[1.-Instalación Java 63](#_Toc10751197)

[2.-Validación Java 63](#_Toc10751198)

[3.-Instalar OpenJdk 64](#_Toc10751199)

[5.-Descargar Hadoop 64](#_Toc10751200)

[6.-Guardar archivo Hadoop 65](#_Toc10751201)

[7.-Descomprimir archivo Hadoop 65](#_Toc10751202)

[8.-Configurar Java Home 66](#_Toc10751203)

[9.-Configurar el archivo core 67](#_Toc10751204)

[10.-Configuración archivo hdfs 67](#_Toc10751205)

[11.-Configuración archivo mapred 68](#_Toc10751206)

[12.-Instalación ssh 69](#_Toc10751207)

[13.-Configuración ssh 69](#_Toc10751208)

[14.-Inicio script ssh 70](#_Toc10751209)

[15.-Configuración PATH hadoop 70](#_Toc10751210)

[16.-Formato del namenode 1](#_Toc10751211)

[17.-Iniciar hadoop 1](#_Toc10751212)

[18.-Interfaz web namenode 2](#_Toc10751213)

[19.-Interfaz web Job Tracker 3](#_Toc10751214)

[Pruebas e integración 3](#_Toc10751215)

[Conclusión de Sprint 1 4](#_Toc10751216)

Resumen

En el presente documento se tiene como meta entregar una propuesta de solución a la problemática que presentan todas aquellas empresas que manejen grandes volúmenes de datos.

Hoy en día gran cantidad de empresas de empresas lo largo de todo chile tiene el problema del mal uso de sus datos ya sea por la escasez de tecnología en sus sistemas, por desconocimiento de las herramientas en el mercado y/o por el uso inadecuado de herramientas.

La materia de los datos necesita ser procesados para ser convertidos en información importante para que ayude a tomar decisiones dentro de las empresas y es lo que hoy en día se presenta como una escasez o como falta de información al respecto de estas tecnologías.

Dado a lo anterior, se propone realizar una investigación respecto a las plataformas que soporten estos grandes volúmenes de datos llamados Big Data, y dar conocer el funcionamiento de estos, lo importantes que puede llegar a ser la buena elección de unas de estas herramientas al interior de las empresas y por ultimó como ayudan en la toma de decisiones para resolver grandes problemas.

CAPÍTULO 1: INTRODUCCIÓN

# Introducción

Con el constante crecimiento de información en cada uno de los aspectos más relevantes mundialmente como sociedad, comercio y ciencia, se vuelve necesario un cambio con respecto al manejo de la información, que hasta hace poco tiempo se venía implementando a partir de bases de datos relacionales. Si bien este esquema funcionaba para ambientes que almacenaban teras de información con datos puntuales y que a través de consultas SQL se podía buscar la información requerida, se vuelve dispendioso y costoso al momento de analizar un mayor volumen de información, en ambientes donde anualmente se almacenan petas de todo tipo de datos, incluyendo aquí los no estructurados, como archivos de video y audio; un ejemplo de esto es “Google, que recibe a diario trillones de bytes, con el objetivo de ofrecer muchos de los servicios que actualmente se conocen como el mismo motor de búsqueda y Google Earth.

El manejo de grandes cantidades de información conlleva a pensar en la implementación de herramientas que permitan administrar y gestionar este tipo de datos no estructurados y semiestructurados, en la búsqueda de patrones concurrentes para la toma de decisiones. Actualmente se pueden encontrar tecnologías como Hadoop, MapReduce y bases de datos NoSQL, que se pueden implementar en la creación de un ambiente Big Data.

De esta manera, el proyecto busco la implementación de una arquitectura para crear un ambiente Big Data en la Universidad Andrés Bello de viña del mar, teniendo en cuenta aspectos importantes como el software y el hardware que se debía utilizar para realizar dicha labor, de igual manera todos los procedimientos que implicaba empezar a utilizar bases de datos no relacionales. Todo esto se ve reflejado a través de dos guías, donde se podrá encontrar de una manera detallada los pasos para la implementación de un ambiente Big Data, sus elementos y procesos para ingresar y consultar información a través de un ejemplo específico, con el propósito de diseñar estrategias y buscar patrones que permitan una buena gestión de la información.

# Definición de Big data

Big data es una referencia a aquellos sistemas de información que manejan conjuntos de datos de gran volumen, de alta velocidad, de veracidad, de valor y de gran variedad de recursos, que demandan formas rentables e innovadoras de procesamiento de la información para mejorar la comprensión y la toma de decisiones.

Big data es la solución al crecimiento exponencial de los datos, en el momento en que se hace difícil su administración con respecto al almacenamiento, procesamiento y acceso.

De esto se puede obtener beneficios como:

* Optimizar el cálculo y la precisión algorítmica para reunir, analizar, enlazar y comparar conjuntos de grandes datos.
* Identificar patrones para la toma de decisiones en los ámbitos económico, social, técnico y legal.

La mayoría de las definiciones que se pueden encontrar de Big data están enfocadas al volumen de los datos, al almacenamiento de dicha información, de esto se puede concluir que el volumen importa pero que también existen otros atributos importantes de Big data, estos son: La velocidad, la veracidad, la variedad y el valor. Estos cinco aspectos constituyen una definición comprensiva y además destruyen el mito acerca de que Big data se trata únicamente del volumen. A cada uno de estos aspectos se le atribuyen las siguientes características:

# Atributos de Big Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Volumen** | **Velocidad** | **Variedad** | **Veracidad** | **Valor** |
| Almacenamiento En terabytes | Por lotes | Estructurado | Integridad y Autenticidad | Estadísticas |
| Registros | Tiempo Cercano | No estructurado | Origen y  Reputación | Eventos |
| Transacciones | Tiempo Real | Multi-factor | Disponibilidad | Correlaciones |
| Tablas y  Archivos | Procesos | Probabilística | Responsabilidad | Hipótesis |

Tabla : 5 Vs

Con el paso del tiempo, las empresas han fomentado la creación de nuestras estrategias para la toma de decisiones, dando un importante lugar al análisis predictivo, ya que con esto se han podido determinar diversos tipos de patrones entre la sociedad, generando como consecuencia gran cantidad de beneficios consistentes en la innovación, investigación y desarrollo de nuevas soluciones.

La generación de estos datos en los últimos años ha venido creciendo de manera inmensurable y se proyecta a seguirlo, por tal motivo es que Big data se convertirá en uno de los principales aspectos a tener en cuenta dentro de los ámbitos comerciales, científicos y social, todo debido al impacto económico e innovador que este ha representado.

Lo anterior se puede ver reflejado en una recopilación de estadísticas nombrada “A comprehensive list of Big data statistics”, donde se exponen algunos puntos del por qué la importancia de Big data, algunos de ellos son:

* Actualmente en el mundo digital existe 2.7 zeta bytes de datos.
* El gobierno de estadounidense invierte cerca de 200 millones de dólares en investigación sobre Big data.
* La red social Facebook almacena, registra y analiza diariamente 30 pata bytes de datos, 94% de los usuarios de Hadoop realiza análisis de grandes volúmenes de información que antes no se podía analizar

Descifrar el genoma humano tardo cerca de 10 años, actualmente ese proceso se puede realizar en una semana.

CAPÍTULO 2: FUNDAMENTACIÓN

# Análisis de la problemática.

## Técnica de los cinco por qué.

Por qué implementar una plataforma de soporte proyectos Big Data

1. ¿Por qué los datos de las empresas son ilocalizables e inutilizables?

Por que manejan demasiado volumen de datos y no utilizan las herramientas y plataformas correctas acorde a los datos que maneja la empresa

1. ¿Por qué que es importante usar las herramientas y plataformas adecuadas para el buen uso de los datos?

Por qué las empresas manejan distintos tipos de datos, los que son; estructurados, semi estructurados y no estructurados y deben tener las herramientas necesarias para el manejo de estos tipos de datos para que así se comuniquen de la mejor forma con las plataformas.

1. ¿Por qué utilizar plataformas que den soporte a proyecto big data?

Porque dependen de que datos use la empresa dependerá de que tecnología se use para desarrollar una plataforma que soporte los datos que maneja la empresa ya sean datos estructurado como no estructurados.

1. ¿Por qué una plataforma sería la solución a la entrega de información tiempo real?

Por qué una plataforma que trabaje con la tecnología adecuada podrá brindar y almacenar datos como el usuario estime conveniente y así entregarle información a tiempo real.

1. ¿Por qué big data suele ser un problema para las empresas que no utilizan herramientas para interactuar con estas?

Por qué las big datas son inmensos volúmenes de datos que al momento de solicitarlo las personas se demora un tiempo considerable en localizar los datos para poder ocuparlos, si se ocupa una herramienta que trabaje con big data y una plataforma que interactúe con esta herramienta se podría localizar los datos acortando el tiempo de localización de los datos para poder analizarlos y poder ocuparlos como información esencial para la empresa.

La utilización de herramientas para manejar los datos y que interactúen con las plataformas de soporte de big data es un avance considerable hacia el tiempo de respuesta que una empresa debe tener para las buenas tomas de decisiones y para corregir errores considerables dentro de esta.

# Diagrama Ishikawa

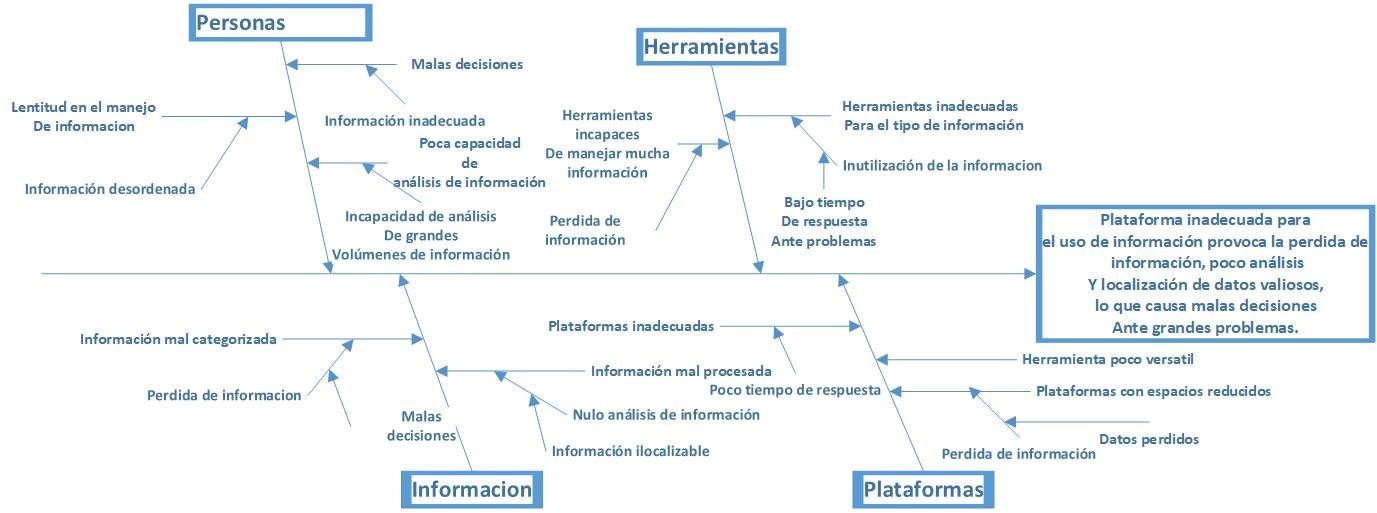


Ilustración : Diagrama de Ishikawa

En el diagrama de Ishikawa se presentan las principales causas de porque hay un rechazo a la plataforma

# Objetivo del proyecto

## Descripción del problema

Big data es un concepto nuevo que se está desarrollando hace pocos años y que solo las grandes empresas han iniciado a implementarlo, la documentación que se encuentra sobre este tema es poca y la mayoría se encuentra en distintos idiomas, especialmente en inglés, las implementación que se han hecho son aisladas y hay pocas guías paso a paso que permita dar soluciones a los errores y fallos que se presentan y que permita realizar la integración de las herramientas open source que se puedan encontrar.

Existen organizaciones y empresas que no pueden pagar o tener una consultoría adecuada sobre el manejo de las nuevas tendencias Big Data, que les permita seleccionar las herramientas adecuadas y la forma de configurar un ambiente para adaptarlo a su modelo de negocio.

Big data es la combinación de viejas y nuevas tecnologías que ayudan a las empresas a obtener una mejor visión del procesamiento de sus informaciones. También puede conceptualizar como la capacidad de manejar un inmenso volumen de datos que se generan de forma caótica, que, a la velocidad y temporización correctas, permite el análisis a tiempo real y la definición de las acciones asociadas necesarias.

Los datos no funcionan y no se les está sacando el mayor partido porque no se están usando bien debido a que los datos no están en el lugar donde deben estar. Según sus estimaciones de experto el 80% de los datos de las empresas no están almacenados de buena forma o no están siendo utilizados por las herramientas y plataformas adecuada lo que provoca la pérdida de información y pérdida de tiempo en dar respuestas a problemas de las empresas.

Según expertos solo el 54 % de las empresas hoy en día tienen localizados sus datos, mientras que el restante porcentaje tiene el gran problema de pérdidas de datos.

## Objetivo del proyecto

El objetivo de este proyecto es dar a conocer estas grandes herramientas tecnologías y brindar apoyo a las plataformas ya existentes de manejo de datos. Hay una enorme cantidad de datos e los que queremos procesar y extraer información, en donde no se puede utilizar metodologías tradicionales ya que todas las imágenes, bases de datos, correos, Facebook, Twitter, sensores, etc. Son información estructurada como no estructurada es decir información que podemos extraer de datos totalmente dispares o bien de nuestra base de datos tradicional todo esto lo procesaremos y lo meteremos dentro de un entorno big data, para poder procesar esa información y extraer la información que necesitamos, dado que las tecnologías tradicionales no pueden hacer frete a esta cantidad de información es necesario utilizar nuevas tecnólogas, es por lo siguiente que se utilizara informática distribuida, se distribuirán datos y procesos que es en lo que se basa big data hadoop.

## Objetivo general

Diseñar y desarrollar una guía para la implementación de un ambiente Big data en la universidad Andrés Bello, la cual ayude a la instalación y a la puesta en marcha de la herramienta hadoop y sus herramientas derivadas open source, la cual permita el manejo de altos volúmenes de datos para así dar a conocer y enseñar el manejo de este ambiente, para incremental en al menos un 60% su manejo.

## Objetivos específicos

* Identificar la tecnología, herramientas de software y requerimientos de hardware necesarios para la implementación de un ambiente de Big data.
* Realizar el montaje, configuración e integración de las herramientas para la construcción del ambiente de Big data
* Diseñar el caso de estudio que se utilizara para realizar las pruebas del ambiente configurado
* Implementar la herramienta hadoop y sus servicios derivados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Objetivo específicos | Situación Actual | Objetivo especifico | Situación esperada | Métricas | Criterio de éxito |
| 1 | Desconocimiento de herramientas y software Open Source que trabajen en conjunto con hadoop. | Identificar la tecnología, herramientas de software y requerimientos de hardware necesarios para la implementación de un ambiente de Big data. | Generar un documento de memoria, el cual contenga la información del ambiente big data. | Documentos con información relevantes | Documento con información relevante >= 100% |
| 2 | Manejo de volúmenes de información de forma manual en discos | Realizar el montaje, configuración e integración de las herramientas para la construcción del ambiente de Big data | Realizar la elevación de los servicios de la herramienta hadoop | Elevación de servicios – no elevación de los servicios | Elevación de servicios =5 |
| 3 | Sin casos de estudios | Diseñar el caso de estudio que se utilizara para realizar las pruebas del ambiente configurado | Realizar casos de estudios para las pruebas de ambiente | Casos de estudios | >=60% de casos de estudios |
| 4 | Sin implementación de la herramientas | Implementar la herramienta hadoop y sus servicios derivados | Montaje de la herramienta hadoop con todos los servicios derivados open source | N° de servicios | >=3 servicios |

Tabla 2: Métricas de Objetivos específicos

La importancia de desarrollar el proyecto radica en apoyar el aprendizaje de las nuevas tendencias en la gestión de la información.

Así mismo el proyecto contribuye y apoya al análisis más detallado a los distintos casos que se puedan presentar y que no puedan ser resueltos mediante bases de datos relacionales.

## Diagrama de alto nivel

A continuación, se presenta el diagrama de alto nivel de la solución para la plataforma de soporte de proyecto big data.

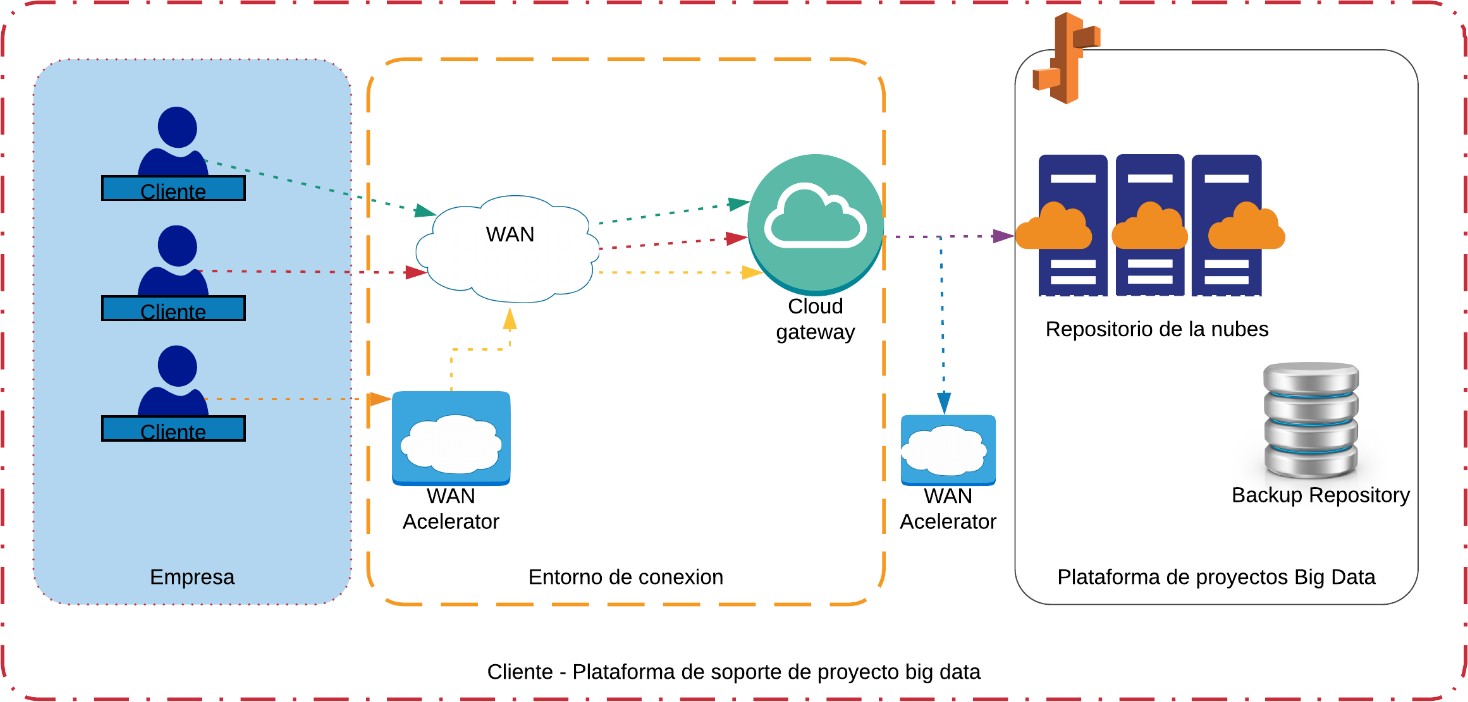


Ilustración 2: Diagrama de alto nivel

## Arquitectura de solución

El sistema es capaz de captar información de fuentes externas con el fin de convertirla en datos de valor mediante técnicas de Procesado del Lenguaje Natural en combinación con tecnologías de procesado y almacenamiento Big Data*.* La información utilizada es obtenida de redes sociales, canales RSS, bases de datos de patentes, artículos científicos, etc, donde los usuarios alimentan al sistema de conocimiento, añadiendo de forma continua nuevas fuentes. De esta manera, se crea una base de conocimiento que escala en dimensiones y que proporciona información para poder conocer el mercado y los competidores próximos a nuestro entorno de una forma automatizada.

La arquitectura está formada por un sistema Cloud Computing compuesto por tres capas: infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS)

* La primera capa del modelo cloud, Infraestructura como servicio (IaaS), proporciona una gestión bajo demanda de todos los recursos de hardware disponibles en un clúster, lo que facilita, mediante técnicas de virtualización, una rápida disponibilidad de los recursos necesarios para las tareas de computación.
* En la capa de Plataforma como Servicio (PaaS) se ofrece un framework de desarrollo formado por un ecosistema Big Data compuesto por diversas herramientas basadas en el paradigma mapreduce , que proporcionan la capacidad de procesar grandes cantidades de información en paralelo.

La última capa del modelo cloud, Software as a Service (SaaS), proporciona al usuario final una aplicación web donde se ofrece toda la información que ha sido generada por el sistema de forma gráfica y sintetizada. La información también se puede obtener mediante interfaces de ejecución e interfaces de consulta a través de un API REST para poder ofrecer información de valor desde otro tipo de dispositivos o aplicaciones de terceros.

La adquisición de información se realiza mediante capaz modulares de extracción, transformación y cargas (ETL), encargados de capturar información de diferentes recursos (redes sociales, páginas web, documentos corporativos y científicos, patentes). La arquitectura está preparada para la captura de información, tanto estructurada como no estructurada, donde herramientas del ecosistema hadoop juegan un papel fundamental tanto en fase de análisis y procesamiento del lenguaje natural como en fase de predicción y clasificación.

* + Apache Pig: Plataforma de alto nivel para crear programas MapReduce utilizados en

Hadoop

* + Apache hive: Infraestructura de almacenamiento de datos construidas sobre hadoop para proporcionar agrupación, consulta, y análisis de datos.
  + Apache mahout:Es un proyecto para producir implementaciones gratuitas de algoritmo de aprendizajes automáticos distribuidos enfocados principalmente en las áreas de filtrado. Mahout también proporciona bibliotecas de java para operaciones matemáticas comunes enfocadas en algebra lineal y estadística

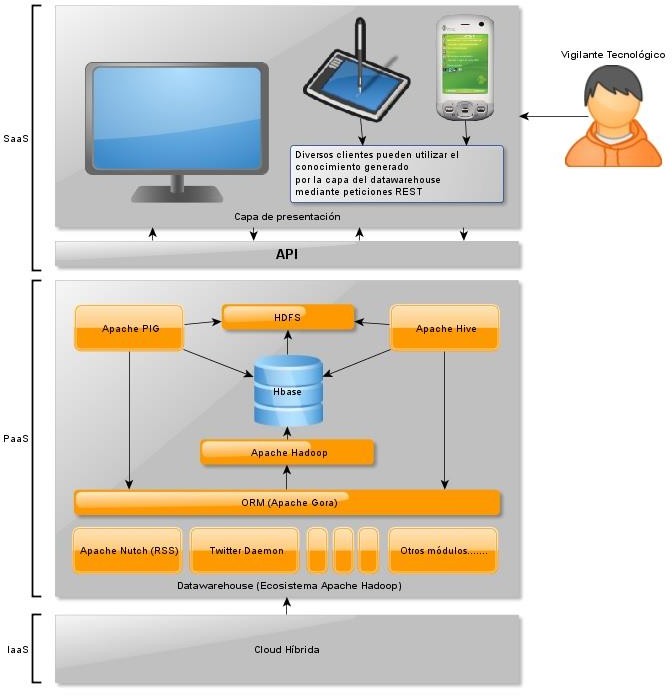


Ilustración 3: Arquitectura de solución hadoop

# 

## Solución ideal

El alcance de la solución propone generar una guía para la implementación del ambiente big data, para así realizar la creación de una nueva plataforma open source que permita el manejo de datos y pueda mostrar a las personas como funciona el ambiente, esto permitirá tener un ambiente de pruebas para esta herramienta y observar las mejoras que trae su implementación.

## Limitaciones

* Los usuarios deben tener acceso a internet para ocupar la plataforma.
* Deben utilizar un browser o explorador web para acceder a la plataforma.
* No envía notificaciones
* Los datos perdidos en el equipo no es responsabilidad de la plataforma.

## Restricciones

* La plataforma deberá mantener a la vista los datos almacenados.
* Los usuarios tendrán una plataforma sin diseño gráfico.
* La plataforma solo muestra como son manejados los datos.

# Alternativas de solución

## Procedimientos manuales

Como primera alternativa, se considera continuar con el proceso manual, disminuir el uso de datos y aumentar los usuarios a cargos de los datos para que estos sean localizados.

Para el análisis de información, categorizar los datos, dando mayor prioridad a los datos no estructurados, para así manejarlos con mayor tiempo y poder hacer uso de ellos.

## Cambios en procedimientos actuales

Si bien es cierto existen plataformas que trabajan con las big datas, muchas de las plataformas no dan a conocer por que es importante el usarlas y cómo usarlas. Es de suma importancia el mostrar cómo se utilizan las nuevas plataformas para que los futuros clientes confíen en ellas y observen los buenos cambios que generan.

Se debe implementar una guía para dar a conocer las instalaciones del ambiente hadoop, para dar a conocer cómo trabaja las herramientas, para así implementar un ambiente de pruebas, esto permitirá que los usuarios puedan usar de mejor forma estos servicios.

## Alternativas disponibles en el mercado.

Algunas de las alternativas disponibles en el mercado y que entregan una plataforma con una gran variedad de sistemas son: Amazon web Service y Microsoft Azure, que son dos de las grandes plataformas que se dedican a prestar estos servicios de soporte de proyecto big data.

Amazon web engloba una gran cantidad de servicios para poder realizar distintos tipos de actividades en la nube. Desde almacenamiento a la gestión de instancias, imágenes virtuales, desarrollo de aplicaciones móviles, etc.

Microsoft Azure es una nube publica de pago por uso que permite compilar, implementar y administrar rápidamente aplicaciones en una red global de datacenters de Microsoft.

Estos servicios ya importantes en el mercado trabajan con herramientas hadoop y sus servicios derivados, solo muestran al usuario un entorno de diseño para que ellos trabajen con sus datos, pero no muestra cómo pueden crear y usar en su completitud estas herramientas.

## Solución propuesta

Si bien las aplicaciones que dan soporte a los grandes volúmenes de datos ya son bastante completas, y la mejora que se puede implementar son bastantes reducidas, pero aun así existen. Estas pueden ser:

Realizar e implementar una guía para la instalación de hadoop en el ambiente big data, con ello viene la incorporación de la instalación del ambiente hadoop y sus servicios derivados, para así dar un entorno de pruebas y manejo de datos dentro de la universidad Andrés bello.

Ofrecer multiplataforma seguras monitoreadas y rastreables.

## Situación Futura

Se debe realizar una investigación en profundidad para saber que herramientas le darán un mayor potencial a la plataforma, esto permitirá extender la guía de implementada para el ambiente haddoop, para tener un entorno más extenso con servicios open source.

En la situación futura lo ideal es la combinación de distintas herramientas que operen en conjunto para entregar al usuario cliente una plataforma rápida, segura y amigable a la vista del cliente para que este al momento de dar uso de ella pueda sacar todo el provecho y tomas las mejores decisiones antes errores en el menor tiempo posible.

## Factibilidad operativa

La factibilidad operativa comprende una determinación de la probabilidad de que un nuevo sistema se use como se supone, aprovechando los beneficios que entrega a todos los usuarios clientes.

* + Operación garantizada
  + Uso garantizado

El correcto funcionamiento de la plataforma dependerá siempre a la capacidad de los empleados para poder utilizarlo, la utilización de herramientas que entreguen apoyo en las labores de trabajo permite encontrar los fallos y dar a conocer las necesidades de una forma más fácil y amigable. Basándose en las cuantiosas investigaciones realizadas y la presencia de plataformas que ya realizan este trabajo es factible operacionalmente realizar la plataforma o implementar mejoras en las ya existentes.

## Factibilidad técnica

La factibilidad técnica consiste en realizar una evaluación de las herramientas tecnológicas que dan soporte a la plataforma big data y la posibilidad de usar las mismas herramientas para mejorar la plataforma y/o combinar nuevas herramientas que funcionen en conjunto con las antiguas.

Hoy en día las plataformas más grandes tanto Amazon web service como Microsoft Azure ocupan herramientas distintas para poder realizar sus labores, pero con las investigaciones realizadas se corrobora que ambas empresas si pueden implementar herramientas que la competencia ocupa para combinarlas con las que utilizan, todo esto para dar mejora a sus sistemas, u/o realizar nuevas plataformas para abarcar más soluciones, ya que ambas Ocupan repositorios en la nube, tienen conexiones similares y ambas trabajan de la misma forma, solo se debe tener en cuenta al momento de elegir la herramienta que tipo de datos soporta y maneja. Dado lo comentado anteriormente es factible técnicamente.

## Factibilidad Económica

La Factibilidad Económica es de vital importancia, dado que la utilización de nuevas herramientas conlleva otros costos los cuales se deben solventar, como la adquisición de servidores o alojamientos de datos en la nube es por eso que se debe tener claro si se tienen los recursos para poder financiar el uso de nuevas herramientas y nubes para albergar la plataforma.

# Historias de Usuario

* HU1.- Investigar sobre una plataforma y herramienta que trabajen en conjunto y que permiten que múltiples usuarios puedan estar conectados al mismo tiempo para realizar distintas operaciones en una misma instancia de tiempo
* HU2.- Dar a conocer cómo se acopla Big data con las herramientas hadoop
* HU3.- Mostrar funcionamiento y como se procesan los datos con las herramientas hadoop y sus herramientas adicionales.
* HU4.- Realizar las pruebas del ambiente configurado.

# Requerimientos de alto nivel

## Requerimientos Sistema:

* RS1 Permitir que múltiples usuarios pueden realizar diversas operaciones en una misma instancia de tiempo.
* RS2 Permitir el almacenamiento de datos
* RS3 Permitir modificación datos Ingresado.
* RS4 Permitir eliminar datos Ingresado.
* RS5 Permitir la realización de búsquedas de datos
* RS6 Permitir detener los servicios del ambiente hadoop cuando se estime conveniente

## Requerimientos Funcionales:

* RF1. El usuario debe ser capaz de leer la guía de instalación de la herramienta hadoop y sus herramientas compatibles para usar
* RF2. EL usuario debe ser capaz de iniciar la aplicación virtualbox para llegar al ambiente big data
* RF3. La plataforma debe mostrar los servicios elevados para la utilización de la herramienta big data
* RF4. El usuario debe poder acceder a las plataformas de los servicios elevados y ver su interfaz de plataforma
* RF5. El usuario podrá realizar pruebas en el ambiente hadoop.

## Requerimientos No Funcionales:

* RNF1. Accesibilidad: La plataforma no debe ser difícil de utilizar por los distintos tipos usuarios.
* RNF2 Lenguaje: EL lenguaje utilizado en la plataforma debe ser lo más natural que se pueda
* RNF3. Disponibilidad: La plataforma hadoop debe estar operativo y funcional cada vez que un usuario lo disponga.
* RNF4. Seguridad: La plataforma debe respaldar los datos de manera confiable en un repositorio físico.
* RNF5. Pruebas: Se podrán realizar pruebas a los servicios elevados para ver el buen funcionamiento del ambiente hadoop

## Delimitación

* Espacio. El espacio para la realización del proyecto será en la Universidad Andrés Bello de viña del mar, teniendo en cuenta que se busca implementar un ambiente de pruebas y desarrollo para Big Data.
* Tiempo. El planteamiento, desarrollo e implementación del proyecto será realizado durante 4 meses y medio.
* Alcance. El proyecto culmina con el diseño y montaje del ambiente de Big Data, teniendo en cuenta que las herramientas y tecnologías serán analizadas y definidas como parte del proyecto. Las herramientas que se van a implementar serán open source, permitiendo así que se puedan descargar y trabajar sin alguna restricción de licencia.

## El desarrollo del proyecto incluye:

* Un caso de estudio que permitirá realizar las pruebas necesarias del ambiente implementado.
* Guías paso a paso sobre la instalación, configuración e integración de cada una de las herramientas requeridas para la construcción del ambiente.
* Montaje de la herramienta hadoop con sus respectivas herramientas open source.

CAPITULO 2: MATERIALES Y METODOS

# Plan de proyecto

En este apartado se revisará el plan de proyecto, las metodologías utilizadas y la planificación estimada a fin de semestre

## Dirección de proyecto

Para dirigir este proyecto se utilizarán dos metodologías, la cuales trabajarán de forma separada para lograr la realización del proyecto.

**La metodología ágil Scrum,** se utilizará para la documentación y la forma de trabajo del proyecto, este ofrece un marco de buenas prácticas que favorece la adaptación los cambios, mitigación de riesgos, retroalimentación continua, etc.

Esta metodología trabaja con periodos de tiempos llamados Sprint, estos duran de 1 a 4 semanas

**FASES DEL SCRUM**

**INICIA**

**PLANEA Y ESTIMAR**

**IMPLEMENTAR**

**REVISION Y REROSPECTIVA**

**ANZAMIENTO**

Ilustración 4: Fases de SCRUM

SCRUM como cualquier metodología cuenta con diferentes fases, y cada fase con distintos procesos, las cual se especifican a continuación:

Iniciar es la primera fase de la metodología, donde se forman los equipos, se identifican y/o asignan los roles de los stakeholder (Personas afectadas por el proyecto, ya sea el equipo de desarrollo, socios, patrocinadores, etc.).

Los roles principales son:

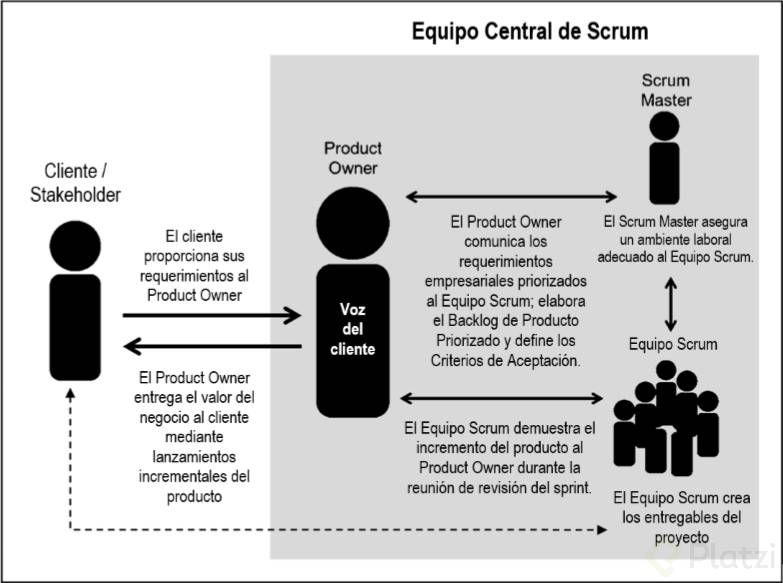


Ilustración 5: Equipo central de SCRUM

* **Scrum master:**Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el ProductOwner para maximizar el rol.
* **Product owner (PO):** Representante de lso accionistas y clientes que usan el software. Se focaliza en la parte de negocio y él es responsable del ROI del proyecto (entregar un valor superior al dinero invertido). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historiasa incorporar en el Product Backlogy las reprioriza de forma regular.
* **Equipo de SCRUM:** Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint.

## Tabla de roles

|  |  |
| --- | --- |
| Rol | Responsabilidad |
| Cliente | **Sarita Gonzales** |
| Product Owner | **Sarita Gonzales** |
| Scrum Master | **David Córdova** |
| Equipo Scrum | **David Córdova** |

Tabla 3: Tabla de roles

## Posteriormente, se desarrollan varios documentos:

Documento de visión del proyecto: define el alcance y el objetivo de alto nivel de un programa, producto o proyecto. Una declaración clara del problema, una propuesta de solución y las características de alto nivel de un producto ayudan a establecer expectativas y reducir riesgos.

## Scrum

Lista de pendientes del producto (Product Backlog): es una lista de historias de usuario. Las historias de usuario, son pequeños documentos que identifican al usuario, su necesidad y finalidad, y estas fueron extraídas y pulidas de la épica, separando cada historia por requerimiento agrupándolas según necesidad.

El plan de lanzamiento y otros documentos opcionales.

En la fase planear y estimar, a partir de documentos de la fase anterior, se elaboran varios documentos los cuales servirán para estimar y planificar el proyecto. De la(s) épica(s) se elaboran las Historias de usuario (User stories), las cuales deben ser aprobadas por el Product Owner, para luego ser revisadas por el Scrum master junto con el equipo Scrum y estimar cuanto esfuerzo es necesario para desarrollar la funcionalidad descrita en cada historia y el valor que le da al negocio.

Para la estimación de esfuerzo se utiliza la técnica puntos de historia o puntos de función, la técnica puntos de historia consiste en estimar cuanto tiempo demora un solo miembro del equipo en desarrollar la historia en cuestión, un punto de historia es igual a una jornada de trabajo, si la historia es muy grande se utiliza Planning Poker. La técnica puntos de función, consiste en asignar puntos a la historia según la complejidad de los datos que se maneja y de los procesos que se realizan sobre ellos. Después de esto las historias de usuario son priorizadas y asignadas a sprint según el valor que aportan al negocio y el esfuerzo necesario. Cada historia de usuario es revisada y se identifican las tareas necesarias para cumplir con los requisitos de la historia. Como este es un proceso iterativo, se crea una lista de pendientes del sprint, es decir, historias de usuario o tareas que quedaron pendientes.

En la fase de implementación, al final de cada sprint, el equipo Scrum presenta un entregable. Un entregable es una o varias funcionalidades de software, el cual permite validar con el cliente la visión del proyecto y recibir una retroalimentación.

En esta fase, se llevan a cabo reuniones de equipo diarias (Daily Standup), las cuales permiten discutir el progreso o el enfoque del proyecto y por último llevar un mantenimiento del Product Back log, esta práctica no se utilizará, ya que el equipo de desarrollo cuenta con un solo miembro.

Respecto a la fase de revisión y retrospectiva, no se utilizará la práctica convocar Scrum de Scrum ya que es sólo un equipo de desarrollo, esto es realizar un seguimiento de su respectivo progreso, de los impedimentos del equipo y de las dependencias entre los equipos.

Reunión de revisión de sprint, en esta reunión el Scrum mastery el equipo Scrum se reúnen con el Product Owner, revisan el sprint y su entregable para retrospectiva del sprint y validan los criterios de aceptación y la aprobación del entregable.

Por último, el Scrum master y el equipo Scrum realizan una reunión de retrospectiva del sprint, donde discuten que hicieron bien, mal y que pueden mejorar. Esta práctica tampoco se utilizará ya que el Scrum master y el equipo Scrum son la misma persona.

## Product backlog del sprint

El producto backlog será realizado con los requerimientos funcionales del sistema.

Las prioridades son el orden en que se ejecutaran los requerimientos.

En el cuadro de historias de usuario se podrá visualizar con las historias de usuario están ligadas a los requerimientos.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requerimiento | Prioridad | Historias de Usuario |
| RQ1 | El usuario debe ser capaz de leer la guía de instalación de la herramienta hadoop y sus herramientas compatibles para usar | 1 | H.U 1 |
| RQ2 | EL usuario debe ser capaz de iniciar la aplicación virtualbox para llegar al ambiente big data | 2 | H.U 1 |
| RQ3 | La plataforma debe mostrar los servicios elevados para la utilización de la herramienta big data | 3 | H.U 2 |
| RQ4 | El usuario debe poder acceder a las plataformas de los servicios elevados y ver su interfaz de plataforma | 4 | H.U 3 |
| RQ5 | El usuario podrá realizar pruebas en el ambiente hadoop | 5 | H. U 4 |

Tabla : Product Backlog

## Sprint backlog

El sprint backlog son las tareas que se realizarán en este sprint 1, las cuales se podrán observar en la siguiente imagen.

Se puede apreciar en la imagen el cumplimiento de las tareas, en caso de incumpliendo de tareas, se documentará que se realizaran el próximo sprint con prioridades altas, para asi luego dar paso a las tareas programadas para el sprint.

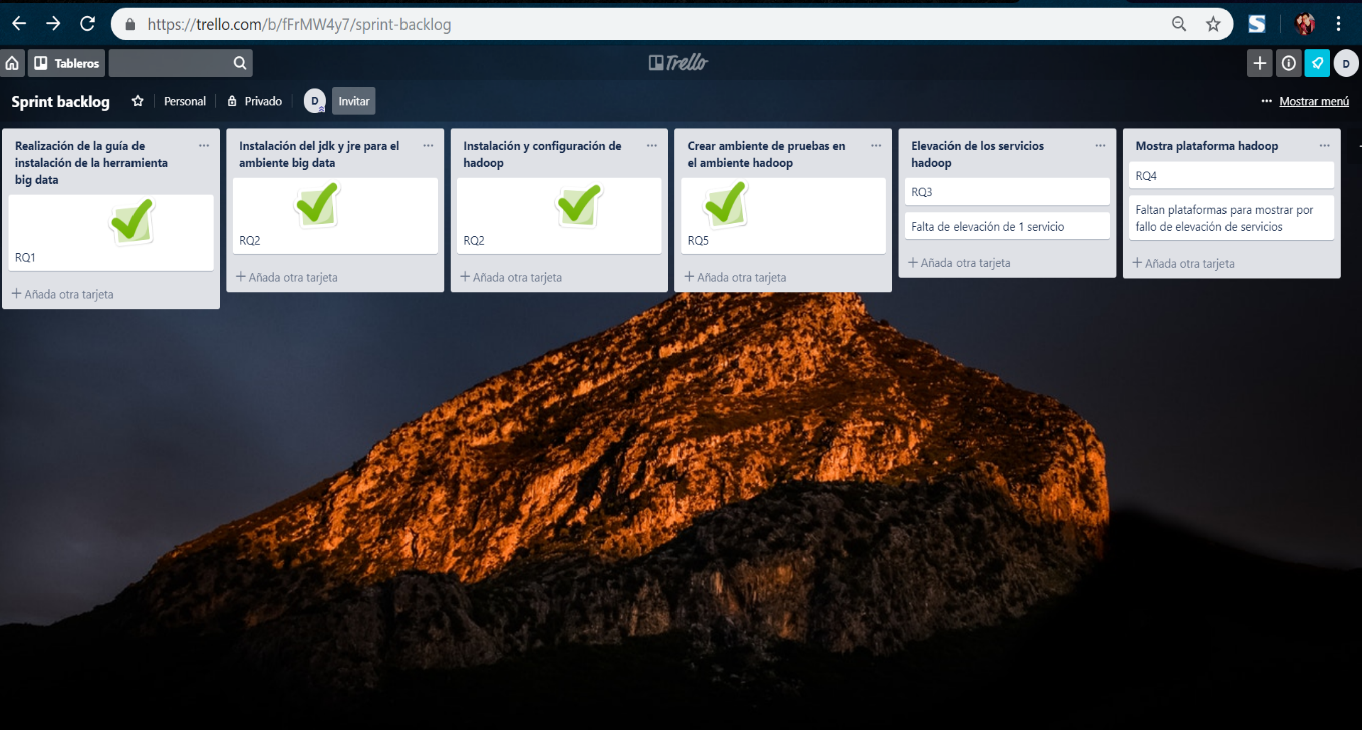


Ilustración : Sprint backlog

## Metodología de desarrollo

La metodología de desarrollo que se utilizara para crear los entregables de cada Sprint será la iterativo-incremental, dado lo anterior, en cada Sprint se hace un trabajo similar, a un tipo cascada de ahí su nombre iterativo, para crear un entregable funcional que sería el incremento.

## Metodología iterativo incremental

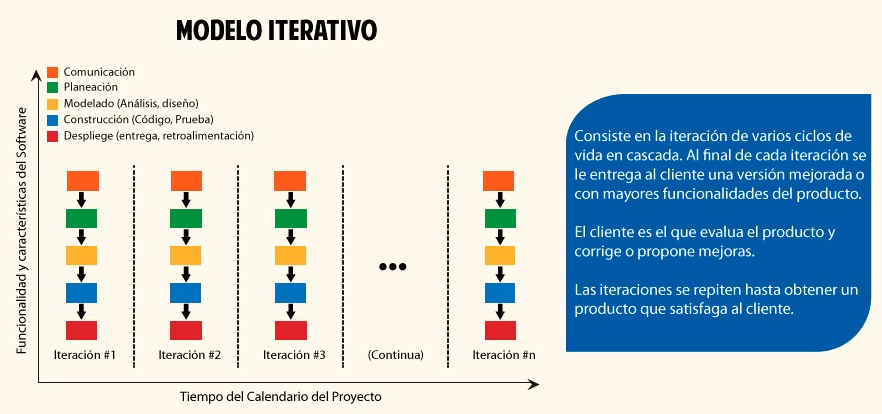
****

Ilustración : Metodología Iterativo-incremental

## Planificación

Para llevar a cabo la planificación del proyecto se utilizará la herramienta Microsoft Project 2016. Esta herramienta permite visualizar el Sprint con sus respectivas historias de usuario que serán desarrolladas, y a su vez, se podrán observar las tareas que requiere cada historia, mostrando como se distribuyen a través del tiempo.

El proyecto tendrá una duración de 19 semanas aproximadamente a partir del 12 de marzo del 2019. Y para la fase de desarrollo de implementación de herramientas, se estima una duración de 14 semanas, a partir del 15 de abril del presente año, con el comienzo del primer sprint del proyecto, dando fin al proyecto de titulo la semana del 15 de julio.

La fase de desarrollo para el proyecto se dividirá en 2 sprint

* 2 Sprint de 4 semanas (8 semanas)

Las horas de esfuerzo necesarias para cada sprint serán medidas con la técnica de estimación puntos de historia, donde cada punto de historia equivale a una jornada de trabajo dedicada solamente a completar la tarea.

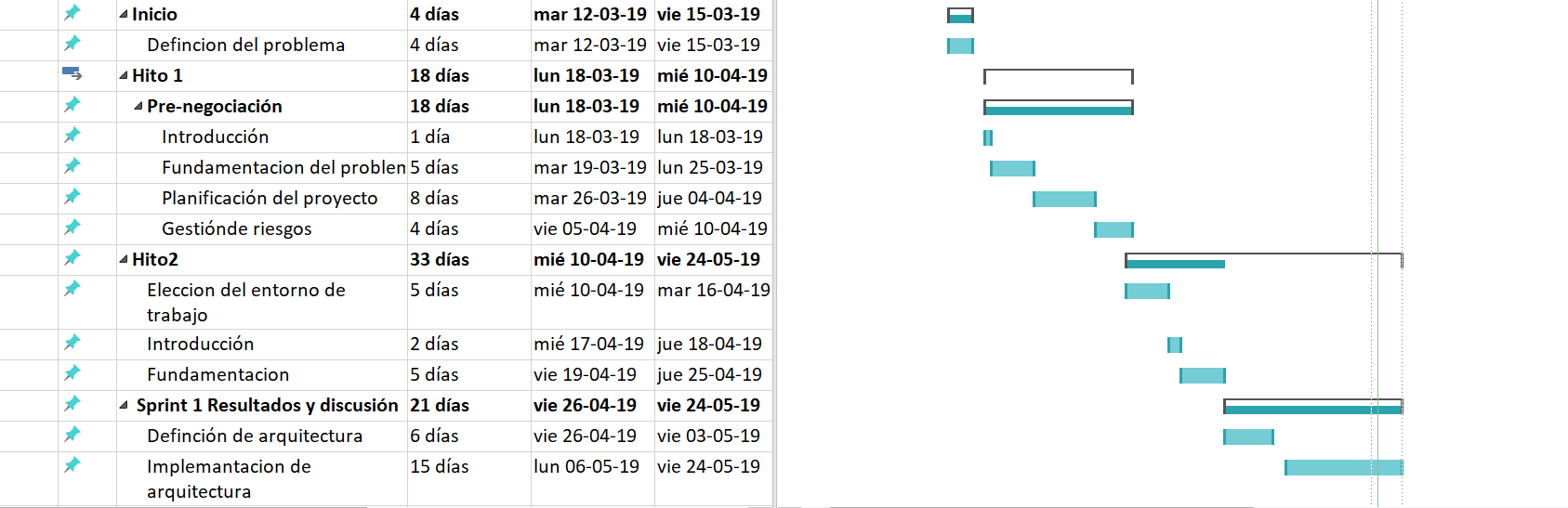


Ilustración 8: Carta gantt Hito1 y hito2

El desempeño del equipo será evaluado con las siguientes métricas:

* Velocidad de tickets: Cantidad de tickets/t
* Velocidad de historias de usuario: Cantidad H.u/t
* Velocidad de problemas resueltos: Cantidad de problemas/t

## Plan de entregables

Cada Sprint terminará con un entregable que será liberado en el ambiente de pruebas definido más adelante. Al final del proyecto el entregable será liberado al ambiente de producción de la universidad siendo probado previamente en equipos de la universidad.

Los entregables se comenzarán a emitir a partir del segundo Sprint, en la cual estos estarán ligados a lo que será el proyecto en sí.

A continuación, se podrá observar los entregables para el Sprint 1 y para el Sprint 2, en donde se define que versión y documentos serán liberados al final de cada Sprint, y en qué ambiente estarán disponibles.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SPRINT | FECHA | VERSIÓN | DOCUMENTOS | AMBIENTE | RESPONSABLES |
| 1 | 26/04/2019 | V 0.1 | LANZAMIENTO NOTAS, MANUAL DE INSTALACIÓN DE AMBIENTE | DESAROOLLO Y PRUEBAS | DAVID CÓRDOVA |
| 2 | S/F | V 0.2 | LANZAMIENTO NOTAS, MANUAL DE INSTALACIÓN DE AMBIENTE | DESARROLLO Y PRUEBAS | DAVID CÓRDOVA |

Tabla 5: Plan de entregables

## Plan de pruebas

Para el primer Sprint, se tendrán pruebas de aceptación y unitarias.

Las pruebas de aceptación serán definidas por el Producto Owner al principio del Sprint. Al final de cada Sprint, el Producto Owner utilizara estos criterios para verificar si los entregables completados, este puede aceptar o rechazar los entregables y sus historias de usuario. Si los entregables es aceptados por el Producto Owner, se consideran como entregables aprobados, de lo contrarios se tomarán medidas de mitigación por parte del Producto Owner, en las cuáles serán las decisiones de dejar tareas pendientes para los siguientes Sprint como prioridad.

Para las pruebas unitarias, estas serán desarrolladas y ejecutada en ambiente Linux, específicamente en S.O Ubuntu versión 18.04, se comprobarán el levantamiento de los servicios mediante la consola de Linux y los comandos utilizados respectivamente para las conexiones.

El plan de pruebas se generará un reporte de pruebas en el cual se podrá observar los siguiente:

* Fecha de ejecución de la prueba
* Tipo de prueba realizada
* Versión del sistema

A continuación, se podrá observar la tabla del reporte:(Los datos mostrados en la siguiente tabla son datos de ejemplos)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID PRUEBA | H.U | NOMBRE | DESCRIPCION | ENTRADA | RESULTADO | RESULTADO | Estado |
| P.0.1 | H.U.4 | Prueba de código | Pruebas de código de montaje de la arquitectura | Revisión de códigos funcionando | Códigos sin función | Código funcional | Aprobado |
| P.0.2 | H.U.4 | Prueba de conexión | Pruebas de levantamiento de conexiones | Entorno con conexión | -Ssh aceptado  -Local host aceptado | Conexión aprobadas | Aprobado |

Tabla 6: Tabla de reportes

El estado de cada prueba realizada a las historias de usuario, puede ser aprobado o rechazado.

El estado aprobado da a conocer que la prueba se ha aprobado sin ningún inconveniente, por otra parte, el rechazado, da a conocer que la prueba tubo fallas las cuales no se acercan al resultado esperado.

## Gestión de la configuración

Para la gestión de la configuración del proyecto Big Data, será divida en dos partes, gestión de línea bases y control de versiones, los cuales serán manejadas con sistemas para llevar un orden y seguridad.

## Gestión de línea Base

Cada vez que se por finalizado una etapa del proyecto será liberada una línea base del proyecto.

Cada Sprint que se dé por finalizado y este sea aprobado por el Scrum Master, será libera una línea base.

Para las presentaciones de proyecto y entregas de hitos, se anexará toda la información relacionada al hito y se generará una línea base del proyecto.

Durante el desarrollo del proyecto, para las presentaciones antes la comisión de profesores, para la evaluación de los avances del proyecto, se liberará una línea base del proyecto.

Cada línea base que posea el proyecto, será documentada en el repositorio local del ordenador y luego subido mediante commit y puch a la plataforma GitHub.

A continuación, se podrá observar una imagen con el tags en GitHub.

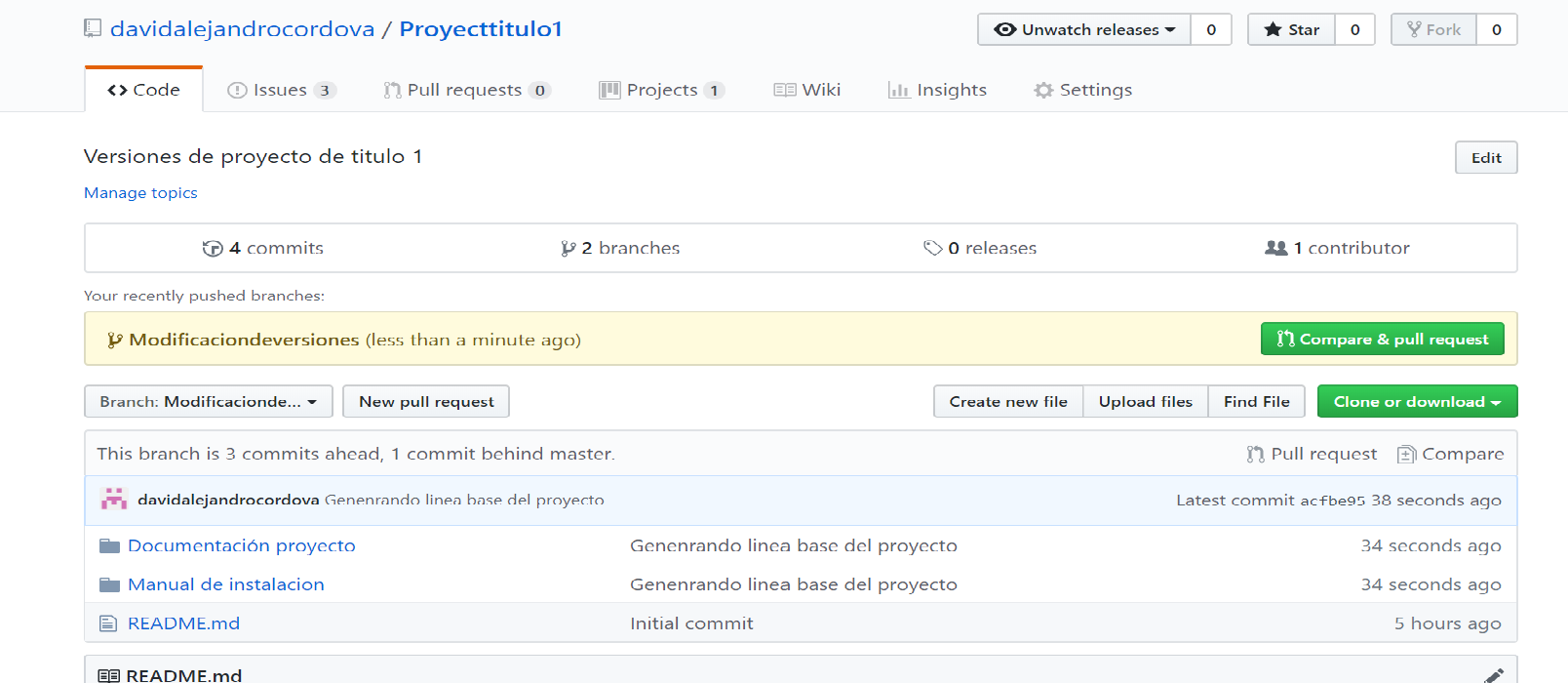


Ilustración : Tags, Línea base

Para el control de archivos y datos del proyecto, se utilizarán branches, en la cual se tendrá una rama master en donde se guardarán los datos como respaldo antes fallas, y una rama modificacióndeversiones, en la cual se guardarán los mismos datos que la rama master, pero a diferencia de que en esta se podrán realizar cambios, y en caso de ser correctos los cambios unir ambas ramas.

A continuación, se podrá observar la rama modificable de nombre modificacionesdeversiones:

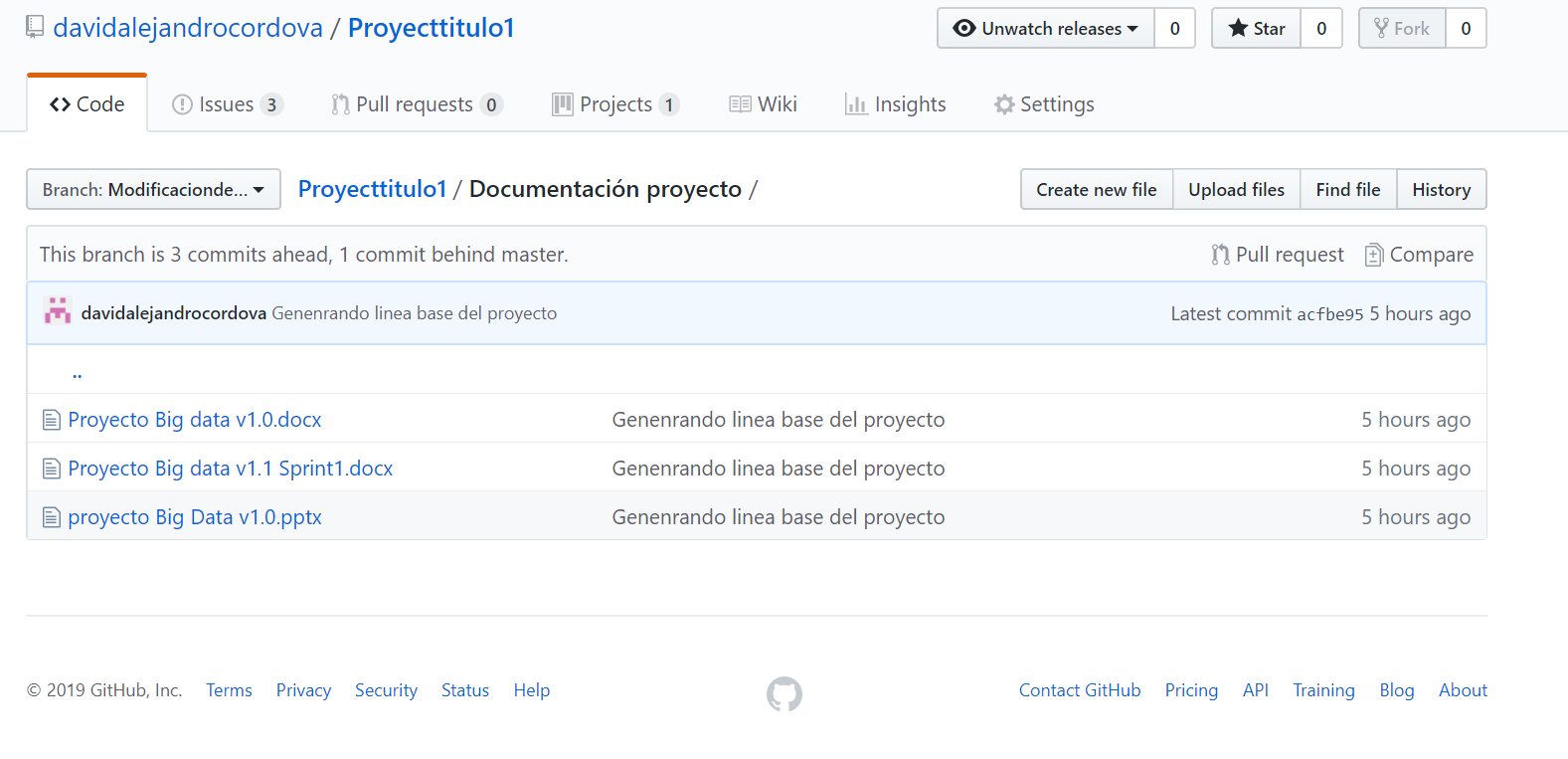


Ilustración : Branches modificaciondeversiones

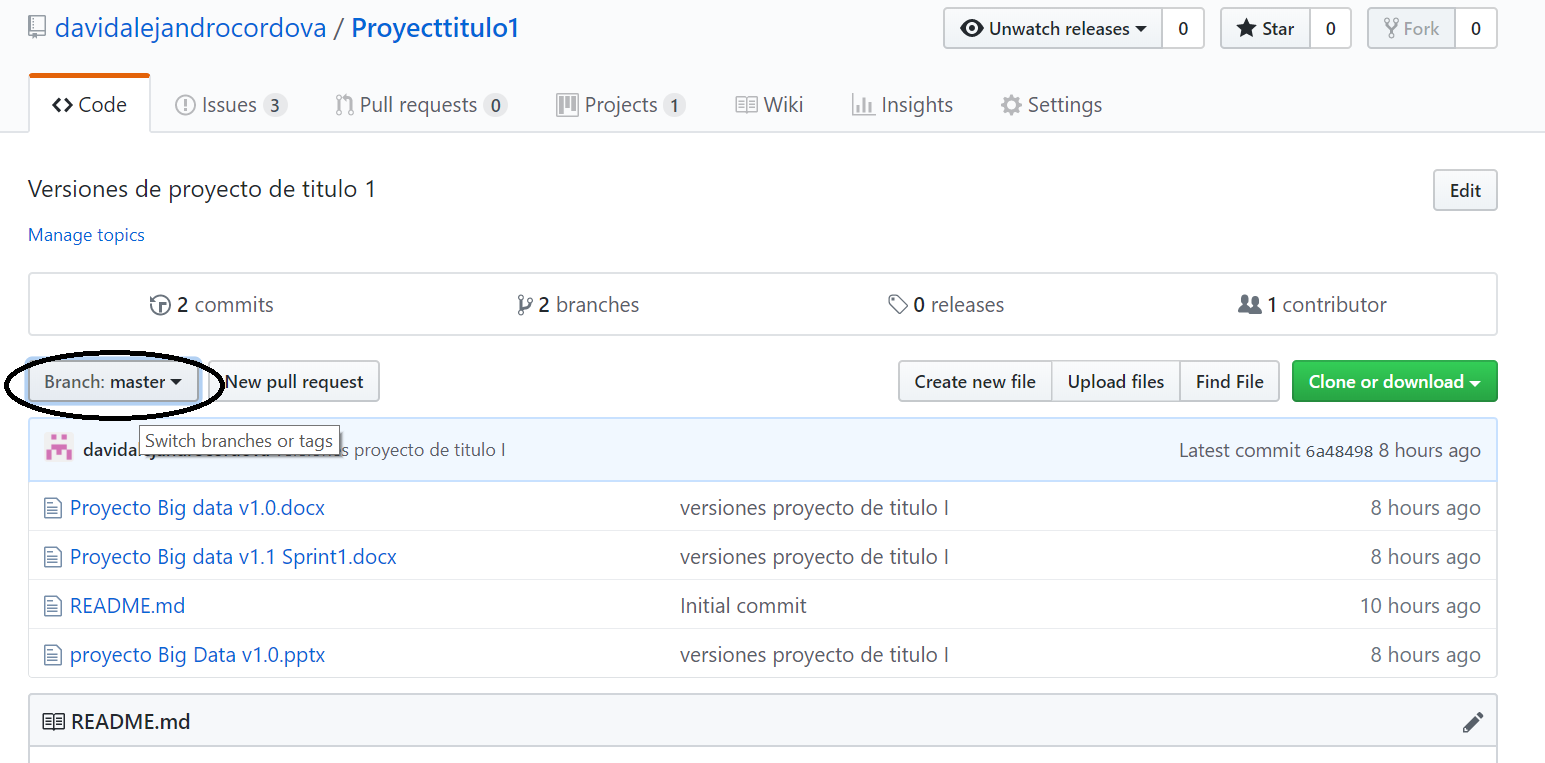
A

Ilustración 11: Branches Master

continuación, se podrá observar la rama master, la cual se mantendrá intacta independiente si las versiones de las otras ramas son modificadas.

Para documentar las instalaciones del ambiente Linux en la máquina virtual, se ha creado un documento con el paso a paso de la instalación de la versión del sistema Linux y luego la instalación paso a paso de hadoop y sus conexiones pertinentes para elevar sus servicios.

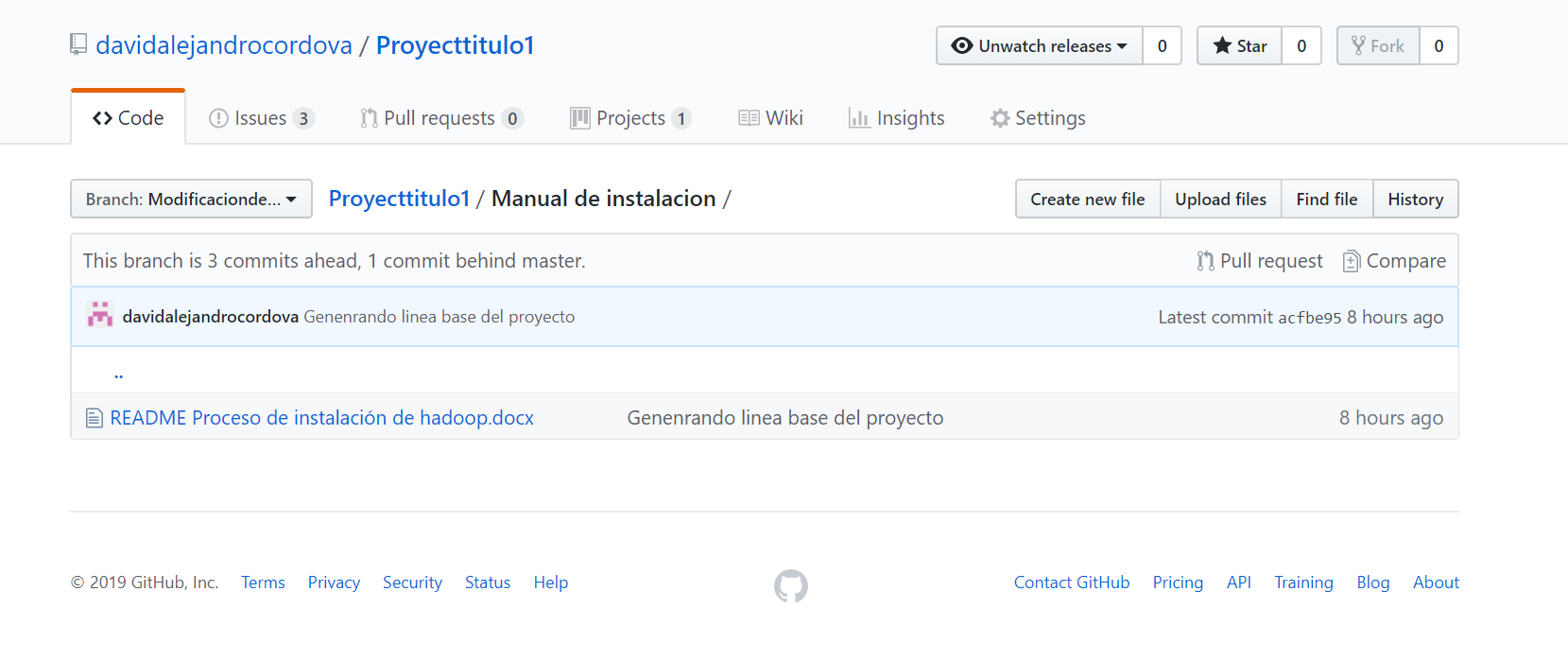
En la siguiente imagen se podrá observar en donde se almacenará este documento paso a paso:

Ilustración 12: Instalación de ambiente Linux

## Control de versiones

Todos los documentos creados serán controlados por versiones de desarrollo, los documentos que sean creados por primera vez serán asignados con la versión “1.0, como se muestra en las imágenes anteriores, cada vez que un documento sea modificado, a este nuevo documento con los cambios se generara una nueva versión, ejemplo “nombre v1.1” y así sucesivamente.

En caso de crear nuevos documentos pasara a la versión v2.0

Todas las versiones generadas se almacenarán en un repositorio local al interior de un ordenador, al cual se conectará con GitHub, el cual se encargará de llevar el registro de las versiones.

## Gestión de cambios

El proceso de gestión de cambios será realizado tanto por la herramienta Trello, que nos permitirá gestionar tickets guardarlos y cerrarlos, y la plataforma GitHub, esta brindara más funcionalidades a la gestión de cambios, permite generar ticket cada vez que surjan problemas, cada ticket pertenecerá a un “Milestone” en el cual se define un periodo de tiempo, este se utilizara para el sprint que se ejecutara en este hito.

El Mileston 1 corresponderá a la fase de inicio del proyecto, el Milestone 2 al Sprint 1 y el Milestone 3 al Sprint 2.

El ciclo de vida para la creación de tickets son los siguientes:

* Nuevos tickets: El ticket ha sido creado y especificado sus características
* Asignado a un desarrollador: El ticket ha sido asignado a un desarrollador (En este caso el equipo solo cuenta con 1 solo desarrollador)
* Cerrado: El ticket se cierra una vez resuelto
* Re-abierto: El ticket ha sido re-abierto por problemas sin solucionar

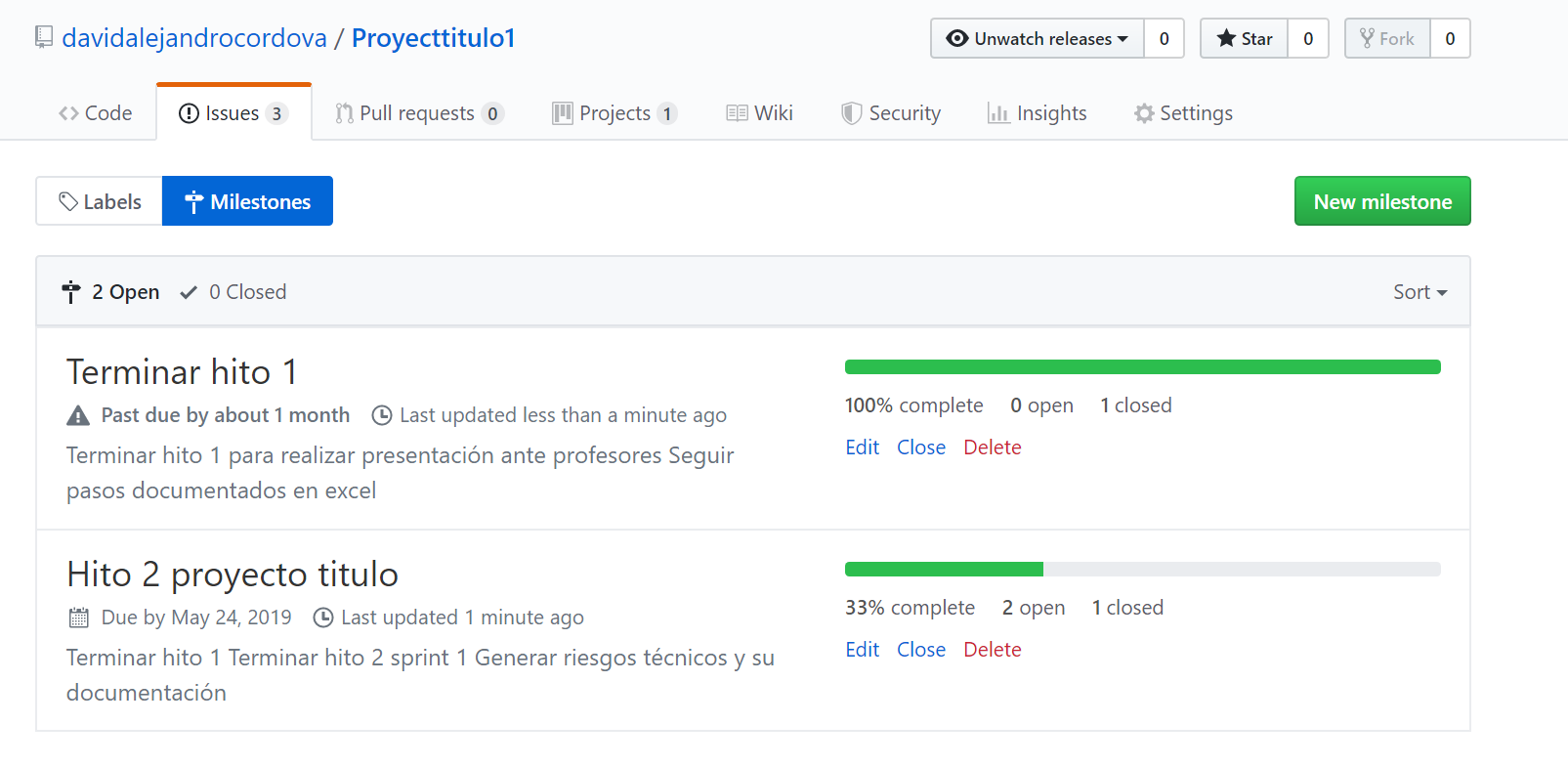
Se puede observar los Milestone creados para el proyecto de título:

Ilustración 13: Milestone para proyecto de titulo

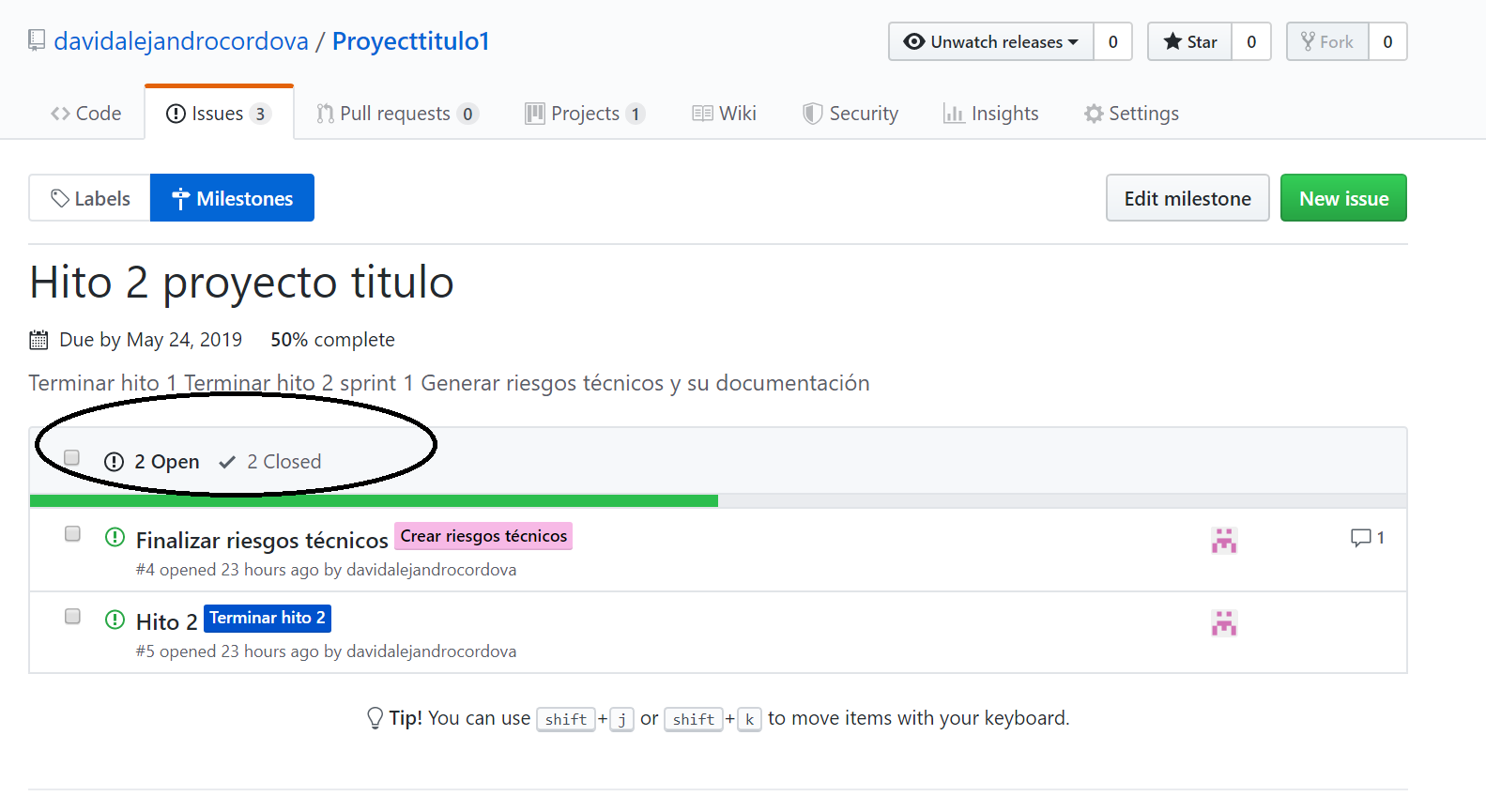
****En la siguiente imagen se puede observar el Milestone creado para el hito 2, en la cual se encuentra a un 50% completo, ya que este posee 4 etiquetas donde 2 ya han sido terminadas con éxito y los dos restantes aún quedan abiertas para realizar modificaciones y comentarios de lo faltante.

Ilustración : Milestone de Hito 2

Para ceder a la plataforma en donde se controla toda la gestión de la confirmación, podre hacer mediante el siguiente enlace:

**https://github.com/davidalejandrocordova/Proyecttitulo1.git**

## Ambiente de desarrollo, pruebas y producción

El ambiente para la investigación e implementación de estas herramientas se realizarán en un computador personal del desarrollador.

## Hardware

|  |  |
| --- | --- |
| **Modelo** | Asus ultrabook UX430U |
| **Procesador** | Intel core i5 8va generación gráficos Nvidia Mx150 |
| **Memoria** | 8GB de Ram |
| **S.O** | Windows 10 64 Bits |

Tabla : Hardware

## Herramientas de construcción

|  |  |
| --- | --- |
| **Tipo** | **Herramienta** |
| **Virtualización** | VirtualBox v6.0.8 |
| **Sistema operativo** | Linux Ubuntu versión 16.04, 64 bits |
| **Herramientas de desarrollo** | JDK versión 11 y jre |
| **Framework a trabajar** | Hadoop |
| **Editor de texto** | Gedit y nano |

Tabla 8: Herramientas de construcción

Hadoop es un framework OpenSource que se montara en la máquina virtual de Oracle Ubuntu, se realizarán las instalaciones de las herramientas de desarrollo de Jdk y jre, estas son necesarias para poder realizar los levantamientos de los servicios, para escribir los códigos necesarios para la instalación y configuración de hadoop, se utilizaran los editores de texto gedit y nano, estos se complementaran el uno al otro en caso de problemas.

## Ambiente de prueba

Para el ambiente de pruebas del este proyecto, se utilizará un programa para la virtualización de virtualbox de Oracle, y S.O Linux Ubuntu, el cual mediante el terminal se verificarán las líneas de comandos escritas para realizar la conexión, gedit y nano serán los editores de texto para verificar los códigos pertinentes para las conexiones correctas a la herramienta hadoop y sus derivados.

En la siguiente imagen se podrá observar una imagen del ambiente de prueba y desarrollo:

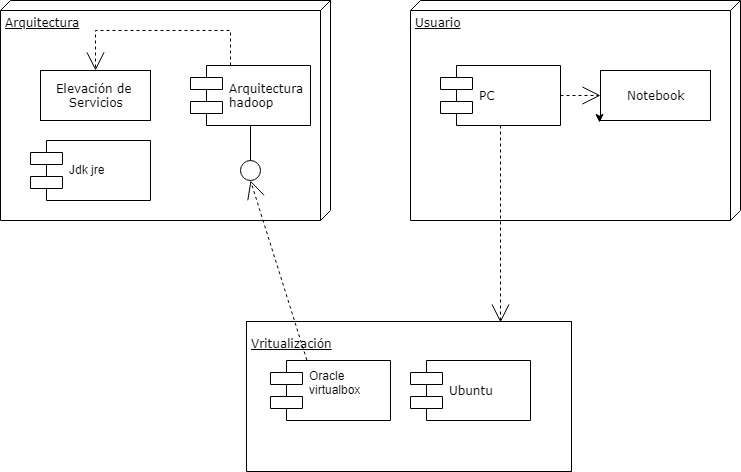


Tabla 9: Ambiente de prueba y desarrollo

## Ambiente de producción

El ambiente de producción estará montado en la virtualización de Ubuntu, en donde se realizarán la instalación de la arquitectura para ambiente Linux.

## Plan de gestión de riegos

El proceso de gestión de riesgos consiste en la evaluación de todos los riesgos que afecten el desarrollo del proyecto semanalmente.

Un riesgo no necesariamente tiene un impacto negativo, los riesgos se clasifican como oportunidades cuando su impacto es positivo y amenazas cuando el mismo es negativo. E este plan se concentra en las amenazas y se han clasificado en tres categorías:

Riesgos Administrativos(RP): Son aquellas situaciones cuyo disparador depende de circunstancias propiamente ligadas al área del proyecto.

Riesgos Técnicos(RT): Son aquellas situaciones cuyo disparador depende de circunstancias propiamente ligadas al área administrativa.

Riesgos externos(RE): Son aquellas actividades donde el disparador del riesgo no depende de los departamentos o áreas de la organización, sino de un externo como un proveedor de bienes o servicios.

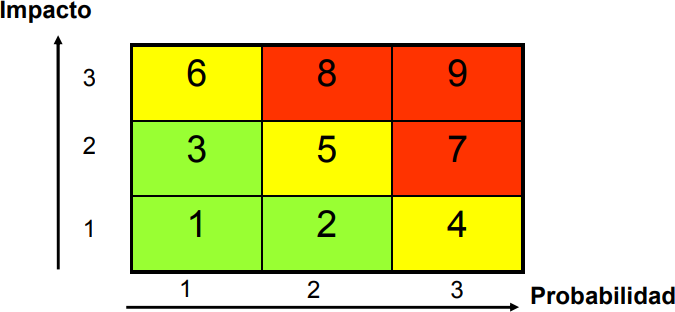
Para lograr priorizar los riesgos identificados e utiliza una tabla de prioridad por impacto

Tabla : Tabla valorización de riesgo

**Porcentaje de amenaza = (Probabilidad \* impacto del riesgo) \*100**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Riesgos | Probabilidad | Impacto | % de amenazas | Mitigación | Contingentes |
| RT1 | Desconocimientos de herramientas y tecnología a usar | 0.5 | 0.5 | 25 | Estudiar los manuales de instalación y funcionamiento de estas herramientas |  |
| RT2 | Problemas de versión de S.O | 0.5 | 0.4 | 20 | -Investigar problemas de compatibilidad | Cambiar versión del sistema operativo  Cambiar sistema operativo |
| RT3 | Problemas de versión y compatibilidad con herramientas de java | 0.3 | 0.5 | 15 | Analizar problemas de versiones de compatibilidad con S.O | Cambiar versión de Jdk y jre para la compatibilidad con S.O |
| RT4 | Problemas de código de instalación de herramientas hadoop | 0.2 | 0.4 | 8 | Analizar y estudiar el código de las herramientas | Cambiar tipo de codificación de las herramientas |
| RP1 | Problemas con la entrega de área técnica del proyecto | 0.7 | 0.7 | 49 | Estudiar la solución para solucionar área técnica | Documentar y agregar área técnica al siguiente Sprint |
| RP2 | Problemas con las reuniones con el cliente permite el buen manejo del proyecto | 0.2 | 0.1 | 2 | Encontrar la forma de comunicación con el cliente para seguir con el proyecto | Comunicarse mediante llamadas de teléfono y/o redes sociales |
| RP3 | Mala planificación del sprint del hito 3 | 0.4 | 0.3 | 12 | Planificar el sprint con más riesgos | Re planificar sprint |
| RP4 | Mala estimación de sprint | 0.2 | 0.5 | 10 | Usar técnicas de validación validadas | Re estimar los sprint |
| RP5 | Plataforma de gestión de la configuración no está disponible | 0.3 | 0.4 | 12 | Estudiar otra plataforma para la realización de gestión de la configuración y/o realizarla manualmente | Utilizar trello para la gestión de la configuración manualmente |
| RP6 | Plataforma de control de versiones no disponibles | 0.3 | 0.4 | 12 | Estudiar otra plataforma para la realización de control de versiones | Utilizar trello para la gestión de la de control de versiones |
| RP7 | Ausencia del product Owner no permite el avance semanal normal | 0.2 | 0.2 | 4 | Estudiar la conexión mediante alguna plataforma para la comunicación con el product owner | Comunicarse mediante alguna plataforma de video llamada con el product owner |
| RP8 | Dificultad en la instalación de las herramientas de hadoop | 0.3 | 0.3 | 9 | Revisar los manuales de instalaciones de hadoop | Aumentar las horas de trabajo para la revisión del manual |
| RE1 | Problemas con la conexión de internet | 0.1 | 0.2 | 2 | Analizar la red y buscar soluciones al problema de red | Conectarse a modem inalámbrico, o a redes móviles. |
| RE2 | Problemas de con la corriente eléctrica | 0.2 | 0.1 | 2 | Buscar el restablecimiento de la energía eléctrica mediante llamados telefónicos al proveedor | Cambiar de establecimiento de abastecimiento de electricidad, o aumentar horas de trabajo del proyecto en los siguientes días |
| RE3 | Robo del material de proyecto | 0.1 | 0.1 | 1 | Buscar respaldo del proyecto realizado en la nube | Aumentar horas de trabajo en el proyecto |

Tabla 11: Riesgos del proyecto

### Riesgos que se presentaron en el área técnica del proyecto(Contingencia)

Se habían presentados riesgos en el sprint 1 del hito 2, en el cual es riesgo técnico no se pudo completar por completo.

Como se documentó, el riesgo del sprint 1 entra como prioridad al sprint 2, en la cual se logra dar con la solución del proyecto, mediante la medida de contingencia, la cual fue cambiar y modificar el código de instalación de hadoop, gracias a esto los servicios de hadoop se encuentran elevados, por lo cual la instalación de hadoop ha sido todo un éxito.

CAPITULO 3: RESULTADOS Y DISCUSION

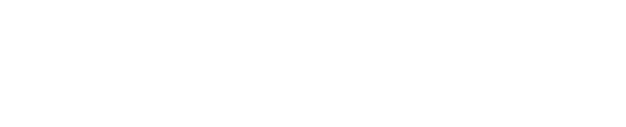
# Arquitectura de un ambiente de big data

## Arquitectura de Big Data

La gestión y procesamiento de Big Data es un problema abierto y vigente que puede ser manejado con el diseño de una arquitectura de 5 niveles, la cual está basada en el análisis de la información y en el proceso que realizan los datos para el desarrollo normal de las transacciones. A continuación, se pueden ver los niveles que contienen un ambiente Big Data y la forma en que se relacionan e interactúan entre ellos:

### Arquitectura de un ambiente de Big data

Ilustración 15: Arquitectura big data



**Ingreso de datos**

Integración, Formato de datos

**Gestión de datos**

Transformación, Manipulación, Correlación

**Análisis de datos**

Modelos de datos, Métricas de calculo

**Tiempo Real de procesamiento**

Alertas y Triggers

**Fuente de Datos**

Redes sociales, Transacciones de Bancos, Google

HDFS y Avro

Motores de bases de datos NoSQL

Hive y Pig

Repositorio de Big Data

Acceso a los datos

Hadoop con herramientas como: Chukwa, Sqoop y Flume

### Ingreso de datos

El Ingreso de datos es el procedimiento de obtener e importar información para su posterior uso o almacenamiento en una base de datos. Consiste en coleccionar datos de muchas fuentes con el objetivo de realizar un análisis basado en modelos de programación

### Gestión de datos

La administración de datos es el desarrollo y ejecución de arquitecturas, políticas, prácticas y procedimientos con el fin de gestionar las necesidades del ciclo de vida de información de una empresa de una manera eficaz. Es un enfoque para administrar el flujo de datos de un sistema a través de su ciclo de vida, desde su creación hasta el momento en que sean eliminados. La administración de Big data es la forma en que se organizan y gestionan grandes cantidades de datos, tanto de información estructurada como no estructurada para desarrollar estrategias con el fin de ayudar con los conjuntos de datos que crecen rápidamente, donde se ven involucrados terabytes y hasta peta bytes de información con variedad de tipos.

### Tiempo real de procesamiento

Es un proceso que automatiza e incorpora el flujo de datos en la toma de decisiones, este aprovecha el movimiento de los datos para acceder a la información estática y así lograr responder preguntas a través de análisis dinámicos. Los sistemas de procesamiento de flujo se han construido con un modelo centrado que funciona con datos estructurados tradicionales, así como en aplicaciones no estructuradas, como vídeo e imágenes.

El procesamiento de flujos es adecuado para aplicaciones que tiene tres características: calcular la intensidad (alta proporción de operaciones de E/S), permitir paralelismo de datos y por último la capacidad de aplicar los datos que se introducen de forma continua.

### Análisis de datos

Es el proceso de examinar grandes cantidades de datos para descubrir patrones ocultos, correlaciones desconocidas y otra información útil. Esta información puede proporcionar ventajas competitivas y resultar en beneficios para el negocio, como el marketing para generar mayores ingresos.

El objetivo principal del análisis de datos es ayudar a las empresas a tomar mejores decisiones de negocios al permitir a los científicos y otros usuarios de la información analizar grandes volúmenes de datos transaccionales, así como otras fuentes de datos que puedan haber quedado sin explotar por la inteligencia del negocio convencional.

### Bases de datos NoSQL

Con la aparición del término NoSQL en los 90s y su primer uso en el 2009 por Eric Vans, se pretende dar una solución a las problemáticas planteadas anteriormente, dando una posibilidad de abordar la forma de gestionar la información de una manera distinta a como se venía realizando.

Para dar una definición adecuada de las bases de datos NoSQL se pude tener en cuenta las siguientes características:

### Distribuido

Sistemas de bases de datos NoSQL son a menudo distribuidos donde varias máquinas cooperan en grupos para ofrecer a los clientes datos. Cada pieza de los datos se replica normalmente durante varias máquinas para la redundancia y alta disponibilidad.

### Escalabilidad horizontal

A menudo se pueden añadir nodos de forma dinámica, sin ningún tiempo de inactividad, lo que los efectos lineales de almacenamiento logran capacidades de procesamiento general.

### Construido para grandes volúmenes

Muchos sistemas NoSQL fueron construidos para ser capaz de almacenar y procesar enormes cantidades de datos de forma rápida.

### Modelos de datos no relacionales

Los modelos de datos varían, pero en general, no son relacional. Por lo general, permiten estructuras más complejas y no son tan rígida que el modelo relacional.

### No hay definiciones de esquema

La estructura de los datos generalmente no se define a través de esquemas explícitos que las bases de datos manejan. En su lugar, los clientes almacenan datos como deseen, sin tener que cumplir con algunas estructuras predefinidas.

# Hadoop y big data

Hadoop es un sistema de código abierto que se utiliza para almacenar, procesas y analizar grandes volúmenes de datos.

## Ventajas

* Aísla a los desarrolladores de todas las dificultades presentes en la programación paralela.
* Cuenta con un ecosistema que sirve de gran ayuda al usuario, ya que permite distribuir el fichero en nodos, que no son otra cosa que ordenadores con commodity-hardware.
* Es capaz de ejecutar procesos en paralelo en todo momento.
* Dispone de módulos de control para la monitorización de los datos.
* Presenta una opción que permite realizar consultas.

También potencia la aparición de distintos add- ons, que facilitan el trabajo, manipulación y seguimiento de toda la información que en él se almacena.

## Arquitectura hadoop

Dentro de la arquitectura de un ambiente Big Data se pueden utilizar diferentes herramientas, cada una de estas cumple un papel importante para la implementación. A continuación, se realiza una descripción de cada una de las tecnologías que están incluidas en la arquitectura.

## 

## Hadoop

Es un framework que permite el procesamiento distribuido de grandes conjuntos de datos a través de grupos de ordenadores que utilizan modelos de programación simple. Está diseñado para detectar y controlar los errores en la capa de aplicación.

Apache Hadoop tiene dos componentes centrales, el almacenamiento de archivos llamado Hadoop Distributed File System (HDFS), y la infraestructura de programación llamada MapReduce, como se muestra en la Figura.

## Arquitectura hadoop

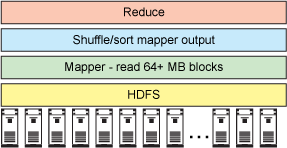


Ilustración : Arquitectura hadoop

## MapReduce

El modelo de programación MapReduce se basa en dos funciones llamadas Map y Reduce. La entrada a dicho modelo es un conjunto de pares clave/valor y la salida es otro conjunto de pares clave/valor.

### Función Map

A partir del conjunto de pares clave/valor de entrada se genera un conjunto de datos intermedios. La función Map asocia claves idénticas al mismo grupo de datos intermedios. Cada grupo de datos intermedios estará formado por una clave y un conjunto de valores, por lo tanto, estos datos intermedios van a ser a su vez la entrada de la función de Reduce.

### Función Reduce

La fase de Reduce se encargará de manipular y combinar los datos provenientes de la fase anterior para producir a su vez un resultado formado por otro conjunto de claves/valores.

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.

La siguiente figura da un ejemplo del flujo de datos en un proceso sencillo de MapReduce.

### Proceso MapReduce

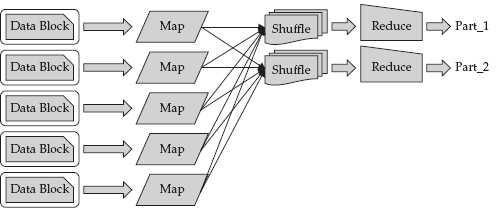


Ilustración : Arquitectura MapReduce

## HDFS

Es el sistema de ficheros distribuido utilizado por Hadoop. Por lo tanto, está especialmente diseñado para cumplir con las necesidades propias de Hadoop. Las dos ideas principales de HDFS es por un lado que sea un sistema de ficheros que facilite una alta escalabilidad tolerante a fallos. Por otro lado, Hadoop necesita que los problemas que se estén intentando solucionar involucren un gran número de datos. HDFS debe garantizar un alto rendimiento de datos para que Hadoop sea capaz de procesar.

A continuación, se puede observar la arquitectura que presenta HDFS con sus respectivos componentes:

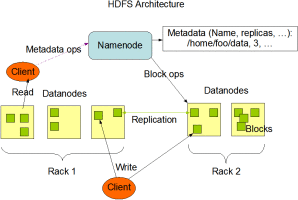


Ilustración : Arquitectura Hdfs

## Chukwa

Es un sistema de recopilación de datos de código abierto para el seguimiento de grandes sistemas distribuidos. Se construye en la parte superior del sistema de archivos distribuido Hadoop (HDFS) y Map/Reduce. Chukwa también incluye un conjunto de herramientas flexibles para la visualización, seguimiento y análisis de resultados de los datos recogidos.

La siguiente figura muestra la interacción de Chukwa con las otras tecnologías y la función que tiene dentro de los procesos de datos

### Diagrama de Chukwa

### 

Ilustración : Arquitectura chukwa

## Sqoop

Es una herramienta diseñada para transferir datos entre Hadoop y bases de datos relacionales. Sqoop importa los datos de un sistema de gestión de bases de datos relacionales (RDBMS) como MySQL u Oracle al sistema de archivos distribuido Hadoop (HDFS), donde transforma los datos y luego los exporta de nuevo a un RDBMS.

Sqoop automatiza la mayor parte de este proceso, basándose en la base de datos para describir el esquema de importación de los datos. Sqoop utiliza MapReduce para importar y exportar los datos, lo que proporciona el funcionamiento en paralelo, así como tolerancia a fallos.

## Flume

Es un servicio distribuido, confiable y disponible para recolectar, agregar y mover grandes cantidades de datos de registro eficientemente. Cuenta con una arquitectura simple y flexible basada en transmisión de flujos de datos. Es robusto y tolerante a fallos con los mecanismos de fiabilidad, conmutación por error y los mecanismos de recuperación. Se utiliza un modelo de datos extensible simple que permite una aplicación analítica en línea, el cual se puede ver en la siguiente figura.

### Flujo de datos

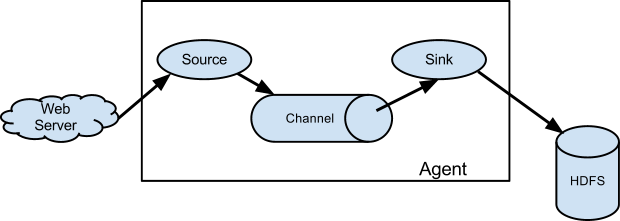


Ilustración : Flujo de datos flume

## Avro

Es un sistema de serialización de datos, contiene lo siguiente:

* Estructuras de datos.
* Formato de datos binario.
* Un archivo contenedor para almacenar datos persistentes.
* Llamada a procedimiento remoto (RPC).
* Fácil integración con lenguajes dinámicos. La generación de código no está obligado a leer y escribir archivos de datos ni a utilizar o implementar protocolos RPC.

Cuando los datos de Avro se almacenan en un archivo, su esquema se almacena con él, para que los archivos se puedan procesar posteriormente con cualquier programa. Si el programa de lectura de los datos espera un esquema diferente esto puede ser fácilmente resuelto, ya que ambos esquemas están presentes.

## Hive

Es la infraestructura de almacenamiento de datos construida sobre Apache Hadoop para proporcionar el resumen de datos, consultas ad-hoc y análisis de grandes conjuntos de datos. Proporciona un mecanismo para proyectar en la estructura de los datos en Hadoop y consultar los datos utilizando un lenguaje similar a SQL llamado HiveQL (HQL). Hive facilita la integración entre Hadoop y herramientas para la inteligencia de negocios y la visualización. Hive permite al usuario explorar y estructurar los datos, analizarlos y luego convertirla en conocimiento del negocio.

Estas son algunas de las características de Hive:

* Cientos de usuarios pueden consultar simultáneamente los datos utilizando un lenguaje familiar para los usuarios de SQL.
* Los tiempos de respuesta son típicamente mucho más rápido que otros tipos de consultas sobre el mismo tipo de conjuntos de datos.

Controladores JDBC y ODBC, aplicaciones para extraer datos. Hive permite a los usuarios leer los datos en formatos arbitrarios, usando SerDes y entrada / Formatos de salida.

## Pig

Es una plataforma para el análisis de grandes conjuntos de información que consiste en un lenguaje de alto nivel para la expresión de los programas de análisis de datos, junto con la infraestructura necesaria para la evaluación de estos programas. La propiedad más importante es que su estructura es susceptible de paralelismo, que a su vez les permite manejar grandes conjuntos de datos.

En la actualidad, la capa de infraestructura de Pig se compone de un compilador que produce secuencias de programas de Mapa-Reduce, para el que ya existen implementaciones paralelas a gran escala (por ejemplo, el Hadoop subproyectos).

### Pig posee las siguientes características:

#### Facilidad de programación

Es trivial para lograr la ejecución paralela de tareas de análisis simples. Las tareas complejas compuestas de múltiples transformaciones de datos relacionados entre sí, están codificados explícitamente como secuencias de flujo de datos, lo que hace que sean fáciles de escribir, entender y mantener.

#### Oportunidades de optimización

La forma en que se codifican las tareas permite que el sistema pueda optimizar su ejecución de forma automática, lo que permite al usuario centrarse en la semántica en lugar de la eficiencia.

#### Extensibilidad

Los usuarios pueden crear sus propias funciones para realizar el procesamiento de propósito especial

# Para la implementación de las herramientas

Para la implementación de alguna herramienta tecnológica en el proyecto, se utilizará la metodología tradicional iterativo-incremental.

## ****Ciclo de vida iterativo****

Responde a la alta incertidumbre del proyecto realizando iteraciones, que no son más que una división del proyecto en fases cíclicas en las que el proyecto va avanzando progresivamente. A cada uno de estos ciclos se le denomina **iteración** y al inicio de cada una de ellas debe planificarse el trabajo a realizar en la misma.

Este ciclo de vida permite ir detallando el plan conforme avanza el proyecto y se va conociendo más sobre el mismo (disminuye la incertidumbre).

## ****Ciclo de vida incremental****

Es una particularización del anterior, mediante la cual cada ciclo que se realiza va obteniendo una porción de producto, servicio o resultado completa. A cada porción generada en una iteración se le denomina **incremento**.

Es decir, vamos produciendo porciones del resultado del proyecto que están acabadas al 100% e iteramos hasta tener todas las porciones, esto es, todo el resultado esperado.

El entregable se produce a través de una serie de iteraciones que sucesivamente añaden funcionalidad dentro de un marco de tiempo predeterminado. El entregable contiene la capacidad necesaria y suficiente para considerarse completo sólo después de la iteración final.

# Modelo de desarrollo

## Modelo Iterativo y acoplamiento al proyecto

Este modelo Busca reducir el riesgo que surge entre la necesidad del cliente y el producto final por procesos equívocos y confusos durante la etapa de recogida de requisitos

Este consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada iteración evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente.

Este tipo de modelo se utiliza en proyectos en los que los requisitos no están claros por parte del usuario, por lo que se hace la creación de distinta prototipos para presentarlos y conseguir la conformidad del cliente.

Este modelo servirá a este proyecto ya que la investigación de la tecnología big data y sus herramientas es algo totalmente nuevo y muy grande en el sentido de la implementación, los requerimientos de estas irán cambiando a medida que pase tiempo, ya que se irán incorporando nuevos requerimientos para ir completando y definiendo la investigación de este gran recurso de datos.

## Ventaja

Una de las principales ventajas de este modelo tradicional y que se acopla bastante bien al proyecto que estamos investigando, es que no hace falta que los requerimientos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir refinando en cada una de las iteraciones en el tiempo.

## Desventajas

Las desventajas de este modelo, es que al no tener los requerimientos bien definidos del comienzo del proyecto la arquitectura del proyecto puede fallar y la implementación de la tecnología, ya que este proyecto dependerá netamente del mejor uso de herramientas tecnológicas que se acoplen de la mejor forma a las plataformas.

## Requerimientos de Hardware y software

* Sistema Operativo: Ubuntu 18.04.
* Tipo de sistema operativo: 64 bits.
* JDK: Java 11.0.0
* JRE: Java
* Memoria RAM: 2 GB.
* Espacio libre en disco: 1 GB (Esto incluye únicamente la instalación, se puede requerir más dependiendo del tamaño de la data a procesar).

## Pasos que seguir en la instalación de hadoop y sus herramientas

* Instalar Hadoop con el fin de configurar un clúster y lograr incluir todas las herramientas en el mismo nodo.
* Instalar Hive (Framework para consultas SQL).
* Integrar Hive con Hadoop.
* Instalar Hbase dentro del cluster de Hadoop (motor NoSQL).
* Integrar Hbase con Hive.

# Modelo Iterativo-Incremental en la instalación y uso de hadoop

El modelo iterativo-incremental, es la metodología usada para el desarrollo técnico del proyecto, es la metodología que más se adapta por su forma iterativa de ir mostrando avances.

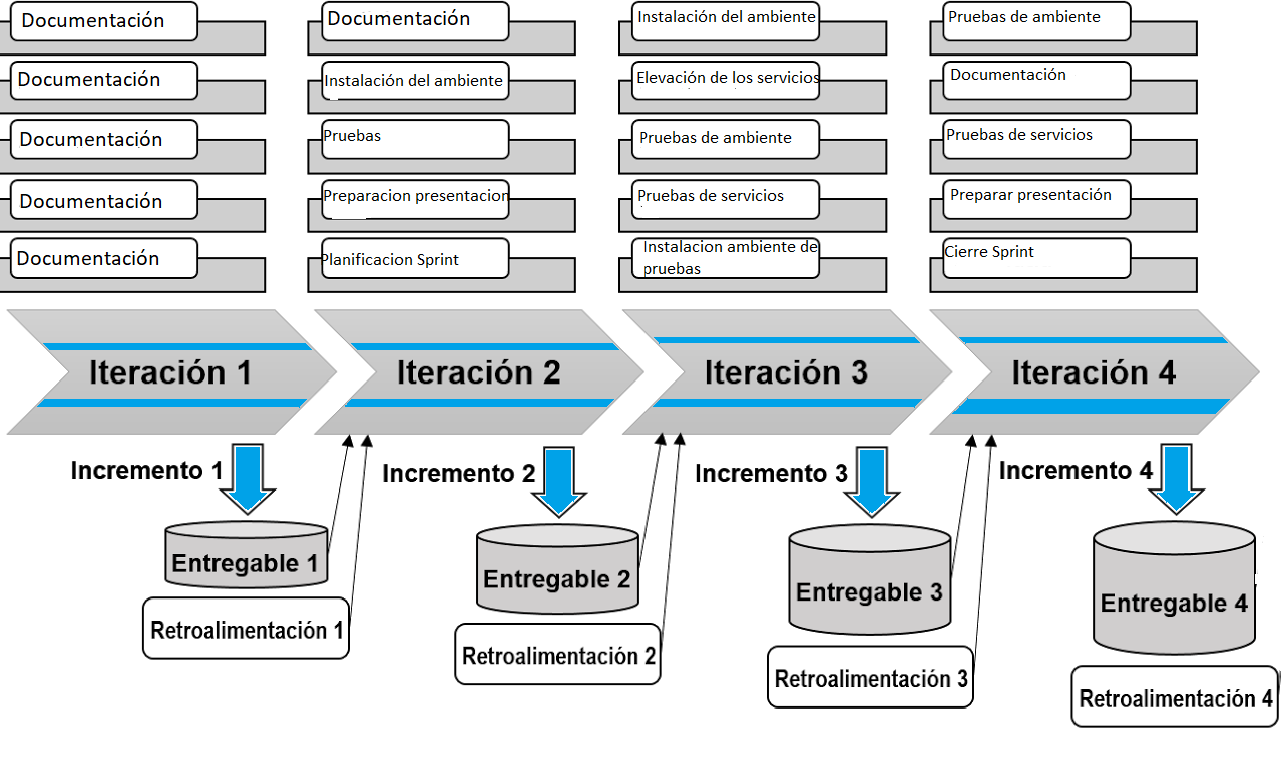
Se mostrará en el siguiente apartado como trabajan el proceso incremental:

Ilustración 21: Proceso Incremental

En el siguiente apartado se mostrará y dará evidencia del trabajo iterativo del proyecto en el Sprint 1.

Pruebas e integración

Codificación

Diseño

Análisis

Ilustración 22: Proceso iterativo

Las actividades se encadenan en una mini-cascada con un alcance limitado por los objetivos de la iteración.

## Fase de Análisis

En este apartado se realiza el análisis del alcance del proyecto, de los objetivos específicos y general. Se analiza el proceso actual sin la tecnología a utilizar, las oportunidades de mejoras y como se debería realizar la implementación de la tecnología.

## Planificación

Para este apartado, se toma en cuenta toda la investigación relacionada con la tecnología, se toman en cuenta los diagramas de alto nivel para tomar una decisión y documentar hasta que proceso llegará el sprint que se estará trabajando.

## Diseño

En el sprint 1, se llegará hasta la elevación de los siguientes servicios:

Estos permitirán ver el diseño e interfaz de la herramienta hadoop.

* NameNode
* SecondaryNameNode
* DataNode
* JobTracker
* Tracker
* 31978 ResourceManager

Cada uno de estos servicios nos permite la visualización de una plataforma que nos permite el ver cómo se trabaja con los datos.

## Codificación

Este es un proyecto de investigación, desarrollo e implementación, en la cual en este apartado se realizará la respectiva instalación y codificación para la instalación del ambiente hadoop y elevar sus servicios. En el siguiente apartado se podrá visualizar como se realiza dicha instalación y como mediante incremento se puede llegar a un producto final, en este caso el producto final son las plataformas que dan evidencia de la elevación de los servicios e hadoop.

Para la instalación y pruebas de hadoop se ha utilizado el modelo iterativo incremental, en el cual se trabajará con iteraciones las cuales irán mostrando cambios a medida que se codifica la instalación de la herramienta, mientras más servicios se levanten de las herramientas, esta permitirá mostrar interfaces gráficas.

A continuación, se mostrará el modelo iterativo- incremental en la instalación de la herramienta hadoop.

Antes de empezar con la instalación de Hadoop se debe tener en la máquina virtual Java, por lo tanto, se instala Java 6 desde el centro de Software de Ubuntu, como se muestra en la siguiente figura:

### 1.-Instalación Java

Ilustración 23: Instalación Java

Verificar que java quedo instalado probando desde la terminal con el comando java. Debe arrojar la información que se muestra en la siguiente figura:

### 2.-Validación Java

Ilustración 24: Validación java

Instalar OpenJdk11 desde la terminal, con el comando **sudo apt-get install openjdk-11-jdk**. Cuando aparezca el mensaje ¿Desea continuar [s/n]?, escribir s como se muestra en la figura N°16, dar enter y seguir con la instalación.

### 3.-Instalar OpenJdk

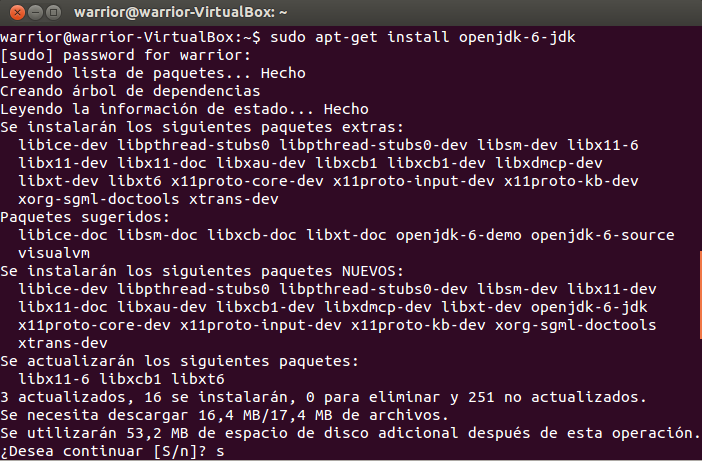


Ilustración 25: Instalación de OpenJdk 11

Descargar la última versión de Hadoop en la página de Apache [**http://apache.mirrors.tds.net/hadoop/common/**](http://hadoop.apache.org/releases.html), escoger la carpeta stable, como se muestra en la figura N°18 y allí escoger el archivo con extensión .tar.gz.

### 5.-Descargar Hadoop



Ilustración 26: Descarga de hadoop v2.9.0

Guardar el archivo en la carpeta Documentos, como se ilustra en la siguiente imagen:

### 6.-Guardar archivo Hadoop

Ilustración : Guardar archivo hadoop

Descomprimir el archivo desde la terminal a través del comando: $ tar xzf hadoop-1.2.1.tar.gz.Luego mover el archivo descomprimido a una carpeta de nombre hadoop a través del comando mv, como se muestra en la siguiente figura:

### 7.-Descomprimir archivo Hadoop



Ilustración 28: Comandos para descomprimir archivo hadoop

### 8.-Configurar Java Home

Ilustración 29: Configuración java Home

A continuación, se realiza la modificación de los archivos para ejecutar hadoop en un modo pseudo-distribuido, esto permitirá que las demás herramientas se puedan integrar correctamente.

Cambiar la configuración del archivo core-site.xml que se encuentra en la carpeta conf, como se muestra por la siguiente información:

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

### 9.-Configurar el archivo core

Ilustración 30: Configuración Archivo core de hadoop

Cambiar la configuración del archivo hdfs-site.xml que se encuentra en la carpeta conf, como se muestra por la siguiente información:

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

### 10.-Configuración archivo hdfs

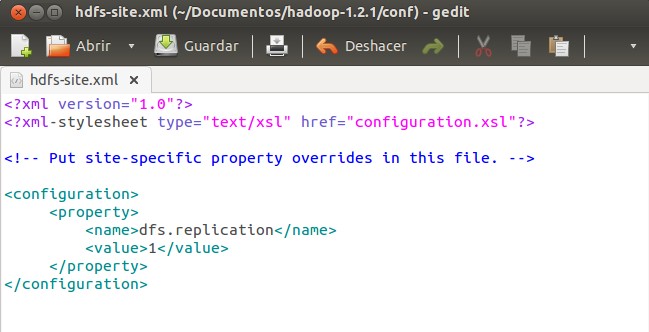


Ilustración : Configuración de hdfs

Cambiar la configuración del archivo mapred-site.xml que se encuentra en la carpeta conf, como se muestra por la siguiente información:

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>localhost:9001</value>

</property>

</configuration>

### 11.-Configuración archivo mapred

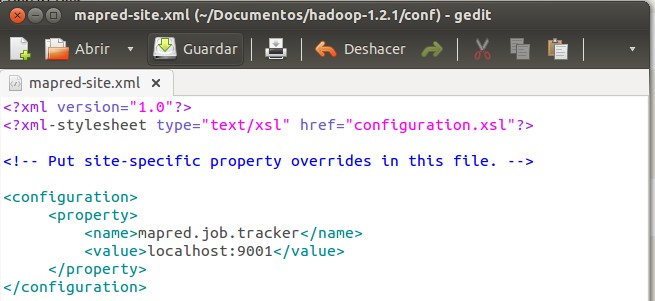


Ilustración : Configuración archivo Mapred

Instalar en la terminal el complemento ssh para poder utilizar los scripts de Hadoop a través del siguiente comando: $ sudo apt-get install openssh-server openssh-client. Cuando aparezca el mensaje ¿Desea continuar [s/n]?, como aparece en la figura N°25, se escoge la opción “s” y se sigue la instalación.

### 12.-Instalación ssh

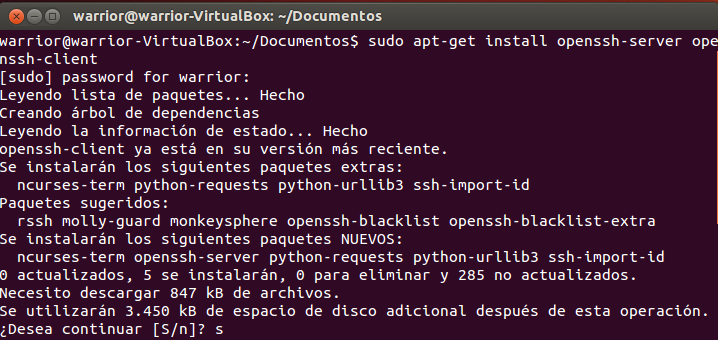


Ilustración : Instalación ssh

Configurar el complemento ssh, como se muestra en la figura N°26, a través de los siguientes comandos:

$ ssh-keygen -t rsa -P "".

$ cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys.

### 13.-Configuración ssh

Ilustración : Configuración ssh

Iniciar el script ssh como se muestra en la siguiente figura:

### 14.-Inicio script ssh

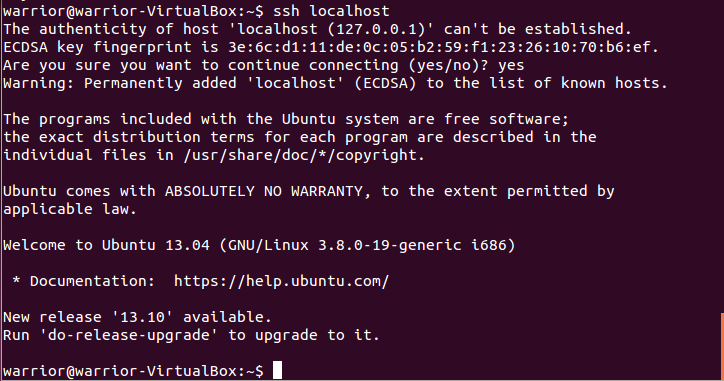


Ilustración 35: Inicio de Script

Crear la variable HADOOP\_HOME en el path como se muestra en la figura N°28, a través de los siguientes comandos:

$ export HADOOP\_HOME=/Documentos/hadoop/bin (dirección de descarga).

$ export HADOOP\_IN\_PATH=/Documentos/hadoop/bin.

### 15.-Configuración PATH hadoop

Ilustración 36: Configuración de Path

Dar formato al namenode, como se muestra en la figura N°29, desde la ruta de la carpeta bin a través del siguiente comando.

$./hadoop namenode –format

### 16.-Formato del namenode



Ilustración 37: Formato del servicio namenode

Iniciar los daemons de hadoop, ejecutando el archivo start-all.sh que se encuentra en la carpeta bin, a través del siguiente comando: $ ./start-all.sh.

Se debe ver la información que aparece en la siguiente figura:

### 17.-Iniciar hadoop

Ilustración : Iniciar Hadoop

Comprobar los servicios que están arriba a través del comando jps, debe mostrar algo como lo siguiente:

14799 NameNode

14977 SecondaryNameNode

15183 DataNode

15596 JobTracker

15897 TaskTracker

31978 ResourceManager

Comprobar la conexión a través de http://localhost:50070, arrojara la siguiente imagen.

### 18.-Interfaz web namenode

Comprobar la conexión de job tracker a través de http;//localhost:50030, debe mostrar la siguiente imagen.

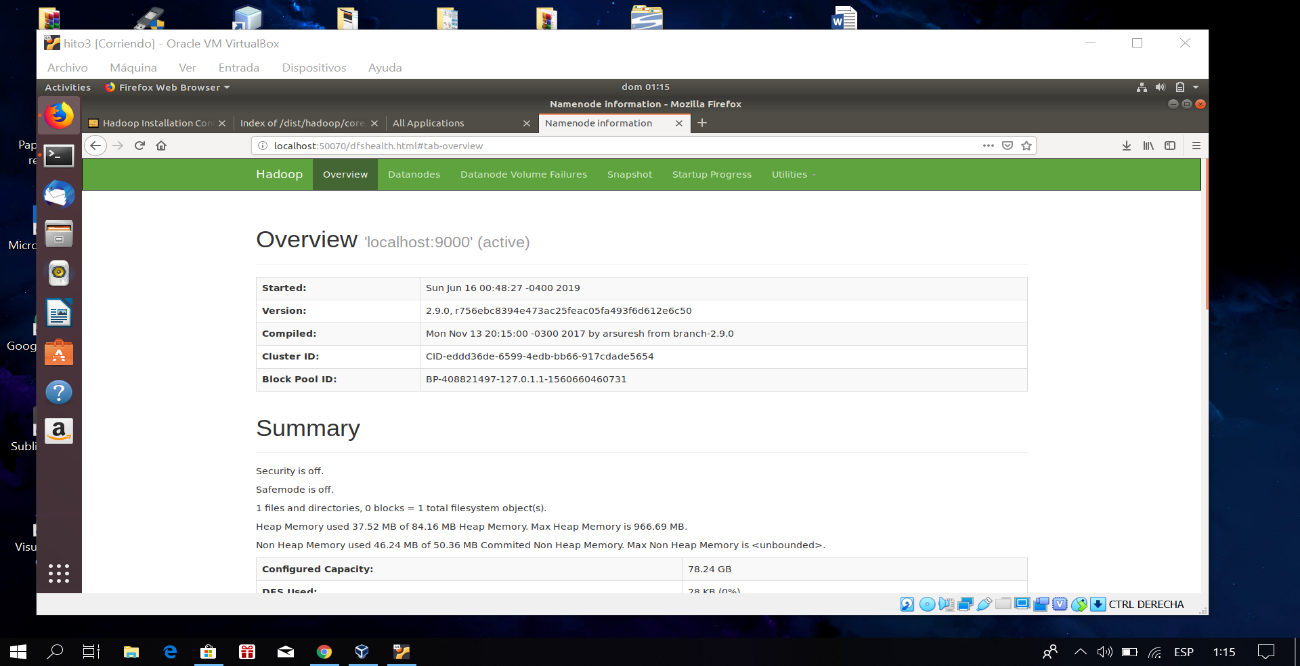


Ilustración 39: Interfaz servicio Namenode

### 

### 19.-Interfaz web Job Tracker



Ilustración : Interfaz servicio Job tracker

Como se observa, para poder mostrar la interfaz gráfica de hadoop, se tienen que seguir pasos para lograr la instalación exitosa de hadoop, en la cual, mediante modelo iterativo-incremental, mediante iteraciones se podrá llegar al objetivo planeado.

# Pruebas e integración

Las pruebas de integración se realizarán visualizando los códigos digitalizados para la instalación del ambiente hadoop, mediante el comando JPS se podrá visualizar si los servicios de hadoop fueron elevados correctamente, si no fueron elevados correctamente, se realizarán las pruebas necesarias para poder hacer el levantamiento de los servicios.

En el sprint 1, se presentaron fallos en la elevación del servicio **31978 ResourceManager,** el cual muestra una de las interfaces más importante de la instalación hadoop, ya que muestra los datos guardados, se realizaron todas las pruebas pertinentes para su elevación, no pudiéndose realizar su elevación se tomará la medida de contingencia de los riesgos del proyecto dejando dicha tarea para el siguiente sprint, el cual entrará como prioridad para luego avanzar con las demás tareas.

# Conclusión

Hadoop no es algo que lleve pocos años implementados, pero su complejidad al memento de instalarlo y la falta de guías de ayuda para poder implementarlo y entender sus herramientas, hace que las empresas no se interesen en usarlos u/o no comprendan del todo las funcionalidades que este puede dar a las empresas que manejan grandes volúmenes de datos y que requieran procesarlos y tenerlos seguros.

Es por eso que en este proyecto se ha realizado una guía de instalación de las herramientas hadoop y herramientas para manejar datos, para realizar la instalación de estas herramientas en forma de prueba se ha tenido problemas para poder completar al 100% su instalación, por las faltas de guías en idioma español y falta de ambientes de pruebas que muestren cómo funcionan las herramientas que interactúan con hadoop.

Es por eso que el sprint 1 mostrara la instalación de hadoop, para elevar todos su servicios y plataformas con interfaces, en el sprint 2 se manejaran datos, para que se pueda observar como las herramientas interactúa con datos y lo fundamental que es implementar estas herramientas.