



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

**Desarrollo de interfaces para la configuración,
administración y monitoreo del Clúster Live-Hadoop**

Trabajo Especial de Grado presentado ante la ilustre

Universidad Central de Venezuela por:

Br. Iván Narváez

Br. Jhoselin Bracho

Tutores:

Prof. Jesús Lares

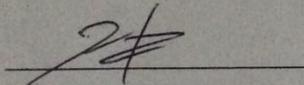
Prof. José Sosa

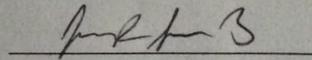
Acta

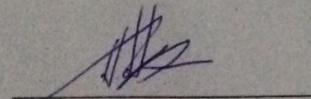
Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por los bachilleres Jhoselín Ginette Bracho Velásquez , portador de la Cédula de Identidad V-19.933.329 y Iván Ignacio Narváez Martínez , portador de la Cédula de Identidad V-19.932.693, con el título: **“Desarrollo de interfaces para la configuración, administración y monitoreo del Clúster Live-Hadoop”**, a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

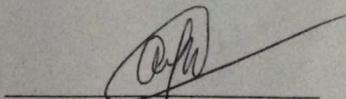
Leído el trabajo por cada uno de los miembros del jurado, se fijó el día 24 de octubre, a las 10:00am, para que sus autores lo defiendan en forma pública, en el aula PAIII de la Escuela de Computación, Facultad de Ciencias de la Universidad Central de Venezuela, lo cual realizaron mediante una presentación oral de su contenido, luego de lo cual respondieron satisfactoriamente las preguntas formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlos con una nota de 19 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas a los veinticuatro (24) días del mes de octubre del año dos mil dieciséis (2016), dejándose también constancia de que actuó como Coordinador del Jurado el Profesor Tutor Jesús Lares.


Prof. Jesús Lares (Tutor Firmante)


Prof. José Sosa (Tutor)


(Jurado) Héctor Navarro


(Jurado) Carlos Acosta

Agradecimientos y Dedicatoria

Por la Br. Jhoselin Bracho

Principalmente quiero agradecer a Dios por haberme dado la fortaleza y la sabiduría para persistir en esta carrera y haber guiado mis pasos en cada momento de esta etapa que no ha sido fácil pero que ha representado una de las etapas más maravillosas e importantes de mi vida.

A mis padres a quienes en especial dedico este trabajo, por haberme apoyado en todo momento, por sus consejos, sus valores y la motivación constante. Todo este trabajo ha sido posible gracias a ustedes. Los amo.

A mis hermanas Jhoicar Bracho y Odilys Velásquez, por sus consejos, comprensión, amor, ayuda en los momentos difíciles, y por ayudarme con los recursos necesarios para estudiar a lo largo de toda la carrera.

A mi amigo Julio Caicedo, gracias por la ayuda y paciencia que me brindó en esta etapa que fue uno de los momentos donde más lo necesité.

A mis tutores Jesús Lares y José Sosa por guiarnos en el desarrollo de este proyecto.

Y finalmente aunque no menos importante a la Universidad Central de Venezuela por ser más que una institución, un lugar de refugio e inspiración, testigo y cómplice de nuestro crecimiento personal y profesional. Orgullosa siempre de ser ucevista.

Agradecimientos y Dedicatoria

Por el Br. Iván Narváz

Gracias a Dios por colocar en mi camino todas las oportunidades que me llevaron al logro de mi más anhelada meta.

A mi familia por el apoyo incondicional y las palabras de aliento en los momentos más difíciles, a ellos especialmente dedico este trabajo.

Gracias a mis amigos por hacer de la carrera más que una meta una experiencia de vida llena de buenos momentos.

A nuestros profesores por inspirarnos clase tras clase, por ser ejemplo de vocación y compromiso.

Gracias a todos aquellos que durante esta carrera fueron de gran apoyo y me ayudaron a crecer como profesional y como persona.

Universidad Central De Venezuela

Facultad De Ciencias

Escuela De Computación

Diseño y desarrollo de interfaces para la configuración, administración y monitoreo del Clúster Live-Hadoop.

Autores: Br. Jhoselin Ginette Bracho Velásquez

Br. Ivan Ignacio Narváez Martínez

Tutores: Prof. Jesús Lares

Resumen

Para hacer procesamientos con grandes volúmenes de datos (Big Data) es necesario tener un clúster. Montar un clúster no es una tarea sencilla, existe un sistema operativo desarrollado en la Universidad Central de Venezuela por los profesores José R. Sosa y Jesús Lares que monta un clúster Beowulf listo para procesar y almacenar grandes volúmenes de datos. Este sistema operativo lleva por nombre Live Hadoop y presenta ciertas limitaciones que hacen que resuelva un conjunto limitado de problemas de Big Data. Algunas de las limitaciones que presenta actualmente es a la hora de iniciar, configurar y monitorear dicho clúster ya que estas tareas se realizan mediante scripts por consola, lo que hace todo el proceso más lento, tedioso y complicado y que tenga que ser aprovechado solo por usuarios expertos.

En consecuencia el objetivo de este Trabajo Especial de Grado consistió en desarrollar una Aplicación Web para llevar a cabo cada uno de estos procesos de instalación, configuración, administración y monitoreo cuyas interfaces nos permitan hacer estas tareas más fáciles de usar y entender, y así poder dar un mejor y mayor uso a este clúster y llegar a muchos más usuarios finales

Para el desarrollo de esta aplicación se empleó una adaptación de la metodología Programación Extrema (XP), incluyendo el uso de tecnología tales como: HTML, CSS, PHP, Python entre otras.

Índice

Resumen	5
Introducción	12
Capítulo I: Planteamiento del Problema	14
1.1 Planteamiento del Problema	14
1.2 Justificación.....	15
1.2.1 ¿Por qué es un problema?.....	15
1.2.2 ¿Para quién es un problema?.....	15
1.2.3 ¿Desde cuándo es un problema?	16
1.3 Objetivos de la Investigación.....	16
1.3.1 Objetivo General.....	16
1.3.2 Objetivos Específicos.....	16
1.4 Alcance de la Aplicación.....	16
1.5 Potenciales Usuarios.....	17
Capítulo II: Marco Teórico.....	18
2.1 Live CD	18
2.2 Apache Hadoop	18
2.3 Live Hadoop	20
2.4 Aplicaciones Web.....	22
2.5 Arquitectura Cliente-Servidor	24
2.6 Tecnologías Web	26
2.6.1 Tecnologías del lado del Cliente	26
2.6.1.1 Lenguaje de Marcas de Hipertexto (HTML).....	27
2.6.2 Tecnologías del lado del Servidor	33
2.6.3 Otras Tecnologías.....	35
Capítulo III: Marco Metodológico	37
3.1 Metodologías tradicionales de desarrollo de software	37
3.2 Metodologías para el desarrollo ágil	38
3.3 Diferencias entre las metodologías	40
3.4 Metodología de Desarrollo del Software	41
3.4.1 Roles	42
3.4.2 Planificación.....	43
3.4.3 Diseño	44

3.4.4 Codificación	44
3.4.5 Pruebas	45
Capítulo IV: Marco Aplicativo	46
4.1 Iteración 0: Planificación y Diseño	48
4.2 Iteración 1: Módulo Descargar.....	58
4.3 Iteración 2: Módulo Configurar	61
4.4 Iteración 3: Módulo Administrar y Monitorear	68
4.5 Iteración 4: Pruebas	73
Capítulo V: Resultados	79
5.1 Descargar Sistema Operativo Live Hadoop.....	79
5.2 Configurar Clúster Live Hadoop	80
5.3 Administrar y Monitorear Clúster Live Hadoop	84
Conclusiones	86
Recomendaciones	87
Bibliografía.....	88
Anexos	90
Cuestionario de Usabilidad	90

Índice de Figuras

Ilustración 1: Sistema Operativo Live Hadoop	22
Ilustración 2: Ejemplos Páginas Web.....	23
Ilustración 3: Arquitectura Cliente-Servidor	25
Ilustración 4: Tecnologías Web.....	26
Ilustración 5: Ejemplo Código HTML.....	28
Ilustración 6: Ejemplo Código CSS	29
Ilustración 7: Comparación gráfica del modelo tradicional de aplicacion web y del nuevo modelo propuesto por AJAX	30
Ilustración 8: Ejemplo Código JQuery	32
Ilustración 9: Ejemplo Código PHP	34
Ilustración 10: Roles del método XP	41
Ilustración 10: Roles del método XP	42
Ilustración 11: Fases del Método XP adaptado.....	43
Ilustración 12: Ejemplo Tabla descriptiva fase parificación	44
Ilustración 13: Tormenta de ideas.....	50
Ilustración 14: Caso de Uso Descargar Sistema Operativo Live Hadoop - Nivel 0	52
Ilustración 15: Caso de Uso Configurar Clúster Live Hadoop - Nivel 0.....	52
Ilustración 16: Caso de Uso Administrar y Monitorear Clúster Live Hadoop - Nivel 0	53
Ilustración 17: Caso de Uso Descargar Sistema Operativo Live Hadoop - Nivel 1	54
Ilustración 18: Caso de Uso Configurar Clúster Live Hadoop - Nivel 1.....	55
Ilustración 19: Caso de Uso Administrar y Monitorear Clúster Live Hadoop - Nivel 1	56
Ilustración 20: Diseño Módulo Descargar	60
Ilustración 21: Extracto de código Módulo Descargar	60
Ilustración 22: Diseño Módulo Configurar Paso1	64
Ilustración 23: Diseño Módulo Configurar Seleccionar IP.....	64
Ilustración 24: Diseño Módulo Configurar Pagina Fin	65
Ilustración 25: Extracto de código Módulo Configurar Paso1	66
Ilustración 26: Extracto de código Módulo Configurar Seleccionar IP	66
Ilustración 27: Extracto de código Módulo Configurar archivo js	67
Ilustración 28: Diseño Módulo Administrar y Monitorear – Ingresar	70
Ilustración 29: Diseño Módulo Administrar y Monitorear - General.....	70
Ilustración 30: Diseño Módulo Administrar y Monitorear - Nodo.....	71
Ilustración 31: Extracto de código Módulo Administrar y Monitorear - Ingresar ..	71
Ilustración 32: Extracto de código Módulo Administrar y Monitorear – General..	72
Ilustración 33: Extracto de código Módulo Administrar y Monitorear archivo js ..	72
Ilustración 34: Resultados Item 1	74
Ilustración 35 : Resultados Item 2.....	74
Ilustración 36: Resultados Item 3.....	75
Ilustración 37: Resultados Item 4.....	75
Ilustración 38: Resultados Item 5.....	76

Ilustración 39: Resultados Item 6.....	76
Ilustración 40: Resultados Item 7.....	77
Ilustración 41: Resultados Item 8.....	77
Ilustración 42: Pagina inicio - Módulo Descargar	79
Ilustración 43: Pagina de descargas - Módulo Descargar	80
Ilustración 44: Pagina crear - Módulo Descargar	80
Ilustración 45: Pagina inicio - Módulo Configurar	81
Ilustración 46: Pagina encendido - Módulo Configurar	81
Ilustración 47: Pagina remoto - Módulo Configurar	82
Ilustración 48: Página manual - Módulo Configurar.....	82
Ilustración 49: Página seleccionar IP - Módulo Configurar	83
Ilustración 50: Página final - Módulo Configurar.....	83
Ilustración 51: mensaje final - Módulo Configurar	84
Ilustración 52: Página general- Módulo Administrar y Monitorear	84
Ilustración 53: Nodos - Módulo Administrar y Monitorear	85

Índice de Tablas

Tabla 1: Diferencia entre metodologías	40
Tabla 2: Planificación - Iteración 0	48
Tabla 3: Historias de Usuario - iteración 0	49
Tabla 4: Requerimientos Funcionales de Descargas	51
Tabla 5: Requerimientos Funcionales de Configuración	51
Tabla 6: Requerimientos Funcionales de Administración y Monitoreo	51
Tabla 7: Requerimientos No Funcionales	52
Tabla 8: Planificación de la Iteración 1	58
Tabla 9: Historias de Usuario - Iteración 1	59
Tabla 10: Planificación de la Iteración 2	62
Tabla 11: Historias de Usuario - Iteración 2	64
Tabla 12: Planificación de la Iteración 3	68
Tabla 13: Historias de Usuario – Iteración 3	69

Introducción

En el mundo de Big Data es necesario tener un clúster para poder almacenar o procesar grandes volúmenes de datos. Hacer que un conjunto de computadoras estén unidas entre sí por una red y que se comporten como si fuesen una única computadora no es tarea fácil, requiere de fuertes conocimientos en redes y en el área de sistemas operativos en general.

Actualmente existen un conjunto de sistemas operativos que montan clústeres Beowulf basándose en el concepto de Live-CD, es decir, clústeres con la característica especial de ser una infraestructura de forma AdHoc o temporal, sobre un conjunto de computadoras que no necesariamente estén pre-configurados para ello. Sobre estas computadoras se corre un sistema operativo totalmente en memoria, se configura el clúster con dicho sistema operativo y cuando se apague el computador ninguna de su información en disco se ve alterada. Sin embargo, los sistemas operativos existentes que montan clúster Beowulf basándose en un arranque por Live-CD no están preparados para Big Data.

Motivado por todo lo anterior surgió una iniciativa de los profesores José R. Sosa y Jesús Lares de la Universidad Central de Venezuela de crear un sistema operativo que montara un clúster Beowulf basándose en el concepto de Live-CD y que esté preparado para Big Data. Este sistema operativo se llama Live Hadoop.

Live Hadoop se encuentra todavía en una etapa de desarrollo y no ofrece interfaces gráficas a la hora de configurar, monitorear y administrar un clúster lo que limita considerablemente la comunidad de usuarios que pueden aprovechar esta herramienta.

Es por ello que el propósito fundamental de este Trabajo Especial de Grado será desarrollar una Aplicación Web que nos permita más fácilmente manejar los procesos nombrados anteriormente y de esta manera expandir la comunidad de usuarios que puedan utilizar este instrumento.

A continuación describiremos cómo se encuentran estructurados cada uno de los capítulos del Trabajo Especial de Grado:

En el capítulo I: Planteamiento del problema. En este capítulo se describe el planteamiento del problema, identificando la situación actual y la problemática existente. Además se exponen los objetivos generales, específicos y el alcance para dar solución al problema planteado.

El capítulo II: Marco conceptual. Se describen todos los fundamentos teóricos en los cuales se basó la investigación para armar la solución final así como las diversas tecnologías empleadas para el desarrollo de la Aplicación Web, tanto del lado del cliente como del lado del servidor.

El capítulo III: Marco metodológico. Se describe la metodología de desarrollo de software utilizada en la realización de este proyecto y las tecnologías de desarrollo.

El capítulo IV: Marco aplicativo. Se realiza la descripción general de la Aplicación Web, las principales interfaces, se expone las pruebas que se realizaron al sitio web y los resultados que se obtuvieron.

Finalmente, se presentan las conclusiones y recomendaciones, las fuentes consultadas y anexos.

Capítulo I: Planteamiento del Problema

En este capítulo se presenta el problema o necesidad, justificación e importancia de crear interfaces que nos permitan acceder de manera más rápida y sencilla al momento de querer configurar y/o administrar un clúster que corra en Live Hadoop los objetivos específicos, el alcance de la aplicación y los potenciales usuarios.

1.1 Planteamiento del Problema

Hoy en día, para poder procesar grandes volúmenes de datos, es necesario poseer un clúster. Específicamente un clúster basado en Apache Hadoop. Sin embargo, desde el punto de vista de investigación, esto se hace difícil ya que el costo para armar una de una de estas infraestructuras es significativo. Si se quiere realizar la validación de un algoritmo, o simplemente para la formación de estudiantes, una solución pudiese ser recurrir a instalaciones de prueba o “SandBox” apoyados en máquinas virtuales. El problema de esto es que este tipo de instalaciones no son capaces de efectuar tareas analíticas de envergadura. Incluso, si se posee un clúster de máquinas que pueden ser usadas para este propósito, el proceso de configuración y levantamiento de un ecosistema Hadoop no es una tarea sencilla ya que, se necesitan tener una serie de conocimientos y tiempo para llevar este proceso a cabo, además que, la instalación de estas herramientas puede causar que no se le dé otro uso a estas máquinas.

Dado esto, surgió una iniciativa de los profesores José R. Sosa y Jesús Lares para crear una distribución de Hadoop, llamada Live Hadoop, que mediante el uso del formato LiveCD, permite levantar una infraestructura de pruebas para grandes volúmenes de datos de forma AdHoc o temporal, sobre un conjunto de equipos en red, que no necesariamente estén pre-configurados para ello. Para lograr esto, el sistema funciona enteramente en memoria, con lo cual, no requiere de ningún tipo de instalación en disco. Esto nos permite implementar un clúster Hadoop sobre un conjunto de máquinas reales que pudieran tener otro uso o fin, como por ejemplo una sala de formación, salas de navegación, etc., para hacer cálculos analíticos sin la necesidad de contar con la costosa tecnología que requiere un clúster para computación de alto rendimiento.

A pesar de que este sistema operativo goza de estas bondades, muchas de sus funcionalidades deben ser realizadas mediante la ejecución de scripts por consola, lo que no permite que este software sea utilizado por un usuario final. Dado este problema, se nos ha encargado la tarea de diseñar e implementar las interfaces que les faciliten a estos usuarios el uso de dichas funcionalidades.

Dentro de las posibles interfaces a realizar, se habla de una que nos permita asignar qué programas y qué configuraciones se van a utilizar en cada una de las máquinas que pertenecen al clúster. Otra para la instalación y configuración de del ecosistema Hadoop dentro de las mismas y una tercera que nos deje monitorear tanto los recursos y rendimiento de las maquinas como el de los programas del ecosistema Hadoop.

1.2 Justificación

1.2.1 ¿Por qué es un problema?

Es un problema porque, sin la existencia de una interfaz gráfica para la configuración e instalación del clúster, este proceso se vuelve más difícil y tedioso de realizar, además de que se incrementa la probabilidad de que se cometa un error durante la realización del mismo.

También es necesaria la existencia de una interfaz gráfica para el monitoreo del clúster porque, si no existe una manera fácil de ver todas las estadísticas importantes relacionadas a él, esto complicará la detección de problemas existentes y la creación de posibles optimizaciones que mejoran su rendimiento.

Finalmente es requerida una interfaz gráfica para la administración del clúster, ya que esta permitirá que se hagan cambios al clúster de manera más sencilla y rápida.

1.2.2 ¿Para quién es un problema?

Es un problema para todas las personas que deseen utilizar Live Hadoop, incluyendo a los estudiantes de las cátedras de Ciencia de Datos que lo usaran para el aprendizaje de las mismas.

1.2.3 ¿Desde cuándo es un problema?

Es un problema desde que se creó la herramienta, y se ha vuelto más necesario a medida que se han ido agregando un mayor número de funcionalidades a Live Hadoop.

1.3 Objetivos de la Investigación

1.3.1 Objetivo General

Desarrollar interfaces web que permitan al usuario final mayor facilidad a la hora de querer configurar y/o administrar un clúster que corra Live Hadoop.

1.3.2 Objetivos Específicos

- ✓ Definir los requerimientos funcionales y no funcionales que proveerá el sistema.
- ✓ Definir las herramientas a utilizar.
- ✓ Adaptar la metodología XP para que se adecúe a las características propias del proyecto.
- ✓ Diseñar y desarrollar las interfaces para cada uno de los módulos de la aplicación (configuración, administración y monitoreo).
- ✓ Realizar pruebas de funcionalidad y usabilidad.
- ✓ Documentar el desarrollo y el análisis del Trabajo Especial de Grado.

1.4 Alcance de la Aplicación

- ✓ Descargar imagen ISO del sistema operativo Live Hadoop.
- ✓ Mostrar diferentes opciones para iniciar clúster en Live Hadoop (manual o remoto).
- ✓ Permitir el monitoreo del estado de la memoria RAM del clúster.
- ✓ Permitir el monitoreo del estado del disco duro del clúster.
- ✓ Permitir el monitoreo del estado del CPU del clúster.
- ✓ .Permitir detener o iniciar nodos pertenecientes al clúster.
- ✓ Administrar los diferentes usuarios y el registro de los mismos.

1.5 Potenciales Usuarios

El desarrollo de estas interfaces esta principalmente dirigido a profesores y estudiantes del área de computación de las cátedras de Ciencia de Datos que lo usaran para el aprendizaje de las mismas. Así como también para el público en general que tenga interés en utilizar Live Hadoop.

Capítulo II: Marco Teórico

En este capítulo se exponen las bases teóricas usadas para el desarrollo de este trabajo, contemplando los conceptos y descripciones de las tecnologías utilizadas bajo la arquitectura Cliente/Servidor debido a que la masificación de la Internet ha convertido la Web en una herramienta para acceder y compartir información en todo el mundo a tan sólo un click al hacer uso de un conjunto de tecnologías que aportan características específicas a la estructura, comunicación y presentación de la información.

2.1 Live CD

Son sistemas operativos almacenados en un medio extraíble con un conjunto de aplicaciones que hacen posible su funcionamiento en cualquier ordenador que disponga de los dispositivos (sea CD, DVD o incluso USB).

La principal ventaja de los LiveCDs es la autonomía, no se necesita un sistema operativo base para correrlos, se puede tener una PC sin disco duro y aún así funcionará bien un sistema operativo en un LiveCD. Uno de los usos más comunes para éstos es para cuando se dañan los sistemas operativos bases y se necesitan extraer datos, o bien, comprobar el correcto funcionamiento del hardware.

Los Live CD pueden ser creados con la herramienta Linux-Live. Esta herramienta nos brinda la posibilidad de crear módulos que son combinados con el sistema de archivos del sistema operativo. Se pueden descargar e instalar "módulos" y todo ello sin tocar o necesitar un disco duro pues es en la memoria RAM dónde temporalmente se almacenan y ejecutan.

2.2 Apache Hadoop

Apache Hadoop es un framework que permite el procesamiento de grandes volúmenes de datos a través de clústeres, usando un modelo simple de programación. Además su diseño permite pasar de pocos nodos a miles de nodos de forma ágil. Hadoop es un sistema distribuido basado en una arquitectura Maestro-Esclavo, que utiliza para el almacenamiento de los

datos Hadoop Distributed File System (HDFS) y algoritmos de MapReduce para hacer cálculos.

Al margen de que Hadoop sea un framework con entidad propia, su tremendo éxito ha propiciado que hoy en día haya ligados a él una miríada de proyectos (HBase, Zookeeper, Hive, Pig, Avro, Flume, Oozie, Whirr, Mahout, Fuse, Spark... y un largo etcétera que no deja de crecer) que pueden considerarse una prolongación del mismo, pues son herramientas concebidas para mejorar el procesamiento de los grandes archivos de datos.

De hecho, gracias a su implementación Hadoop sigue siendo la tecnología líder opensource dentro del universo de Big Data, entendido por doble partida, tanto como el proyecto Apache Hadoop, formado por sus dos componentes centrales, como sumándoles la constelación de proyectos que lo complementan.

El rompecabezas que es Hadoop aumenta su número de piezas de forma impar, y aunque todas encajan a la perfección, el resultado no deja de ser sino un conjunto heterogéneo de satélites que giran alrededor de un sol amarillo que, curiosamente, tiene forma de elefante.

Sus lenguajes son también muy diversos, tanto por su programación como porque, en algunos casos, facilita codificar en otros lenguajes, como hace el mencionado Hadoop Streaming, y también Hive, que permite escribir en SQL y hacer que se convierta en un trabajo de Map Reduce.

Pig, por ejemplo, es un entorno de programación de alto nivel para codificar MapReduce que trabaja con el lenguaje PigLatin. O Lucene, una herramienta que se encarga de la indexación de grandes bloques de texto no estructurado, escrita en Java, mientras Hive, pongamos por caso, es un almacén de datos construido sobre Hadoop que trabaja con un lenguaje similar a SQL.

Dentro del ecosistema Hadoop, por lo tanto, es posible codificar en distintos lenguajes y, aunque en él conviven diferentes elementos diseñados en lenguajes de codificación diversos, todos ellos persiguen un mismo objetivo: un eficiente análisis de los grandes datos.

Cuenta con cuatro componentes principales el Hadoop Common, HDFS, YARN y el paradigma MapReduce.

2.3 Live Hadoop

La ciencia de datos es una de las áreas que más se encuentra en desarrollo a nivel mundial debido a sus proyecciones de uso y su impacto en el mundo de las TIC en los próximos años. Para poder usar sus tecnologías y hacer procesamientos con grandes volúmenes de datos (Big Data) es necesario tener un clúster. Específicamente un clúster basado en Apache Hadoop. Se puede proceder a instalar alguna de las distribuciones ofrecidas por las 3 grandes compañías especializadas en el procesamiento de grandes volúmenes de datos las cuales son: Cloudera, Hortonworks y MapR, pero esto requiere la realización de un proceso no trivial de instalación y configuración para poder comenzar.

Otro de los problemas que existe, es que, realizar esto, nos lleva a tener una infraestructura especializada para Hadoop, lo cual nos puede dificultar darle otro uso a este conjunto de nodos. Adicionalmente, desde el punto de vista de investigación, el planteamiento anterior consiste en una tecnología muy costosa porque requiere de grandes infraestructuras. Para la validación de algoritmos, o simplemente formación, resulta más sencillo y económico recurrir a instalaciones de prueba o "SandBox" apoyados en máquinas virtuales. El problema de estas soluciones es que no sirven para efectuar tareas analíticas de envergadura.

Para evitar varios de estos problemas, existen un conjunto de sistemas operativos que montan clústeres Beowulf basándose en el concepto de Live-CD, es decir, clústeres con la característica especial de ser una infraestructura de forma AdHoc o temporal, sobre un conjunto de computadoras que no necesariamente estén pre-configurados para ello y con características heterogéneas. Sobre estas computadoras se corre dicho sistema operativo totalmente en memoria, se realiza un proceso de configuración del sistema operativo, se ejecuta lo que se tenga que ejecutar y cuando se apague el computador ninguna de la información en que estaba originalmente en disco se ve alterada.

Dado todo esto, surgió una iniciativa de los profesores José R. Sosa y Jesús Lares de la Universidad Central de Venezuela de crear un sistema operativo que

montara un clúster Beowulf basándose en el concepto de Live-CD y que esté preparado para Big Data. Este sistema operativo se llama Live Hadoop, por el proyecto de la Fundación Apache que lleva por nombre Hadoop.

El objetivo fundamental del proyecto Live Hadoop fue la creación de una distribución de Hadoop en el formato LiveCD. Esta tecnología permite implementar un clúster Hadoop sobre un conjunto de máquinas que pudieran tener otro uso o fin y que no necesariamente estén pre-configurados para ello, como por ejemplo una sala de formación, salas de navegación o laboratorios.

Actualmente Live Hadoop brinda un ecosistema de Hadoop básico conformado por: Hadoop Common, Hadoop Distributed File System (HDFS), Hadoop Yet Another Resource Negotiator (YARN), HadoopMapReduce, una distribución de R con el componente RHadoop y al igual que el SandBox, está pre-configurado en modo mono-nodo cuando se ejecuta en un solo equipo. A pesar de esto, aún se encuentra limitada para resolver ciertos problemas del área de Big Data y por ello se están llevando a cabo acciones para seguir expandiendo las funcionalidades del Live Hadoop.

Vale la pena mencionar, que existen diversos sistemas operativos similares a Live Hadoop en el sentido de que permiten levantar una infraestructura de pruebas de forma AdHoc o temporal, sobre un conjunto de equipos en red, que no necesariamente estén pre-configurados para ello y con características heterogéneas. Uno muy popular se llama PelicanHPC, el cual permite montar un clúster de alto rendimiento en pocos minutos. El modelo de procesamiento que se puede hacer sobre un clúster montado con PelicanHPC es OpenMPI, si se desea hacer otro tipo de procesamiento se deben realizar las configuraciones pertinentes.

Otro sistema operativo con características similares es MOLA, basado en PelicanHPC. El uso principal de MOLA es para hacer procesamiento paralelo y distribuido de virtual screening. Virtual screening es una técnica computacional usada en el descubrimiento de drogas donde básicamente se busca librerías de pequeñas moléculas para identificar la estructura de aquellas que tienen más probabilidad de unirse a otro tipo de droga, generalmente un receptor de proteína o una enzima.

PelicanHPC y MOLA al igual que Live Hadoop montan un clúster Beowulf pero hasta el momento de la creación de esta investigación no se tiene conocimiento alguno de distribuciones que permitan el procesamiento de grandes volúmenes de datos sin tener que realizar un arduo trabajo de instalaciones y configuraciones.[1]

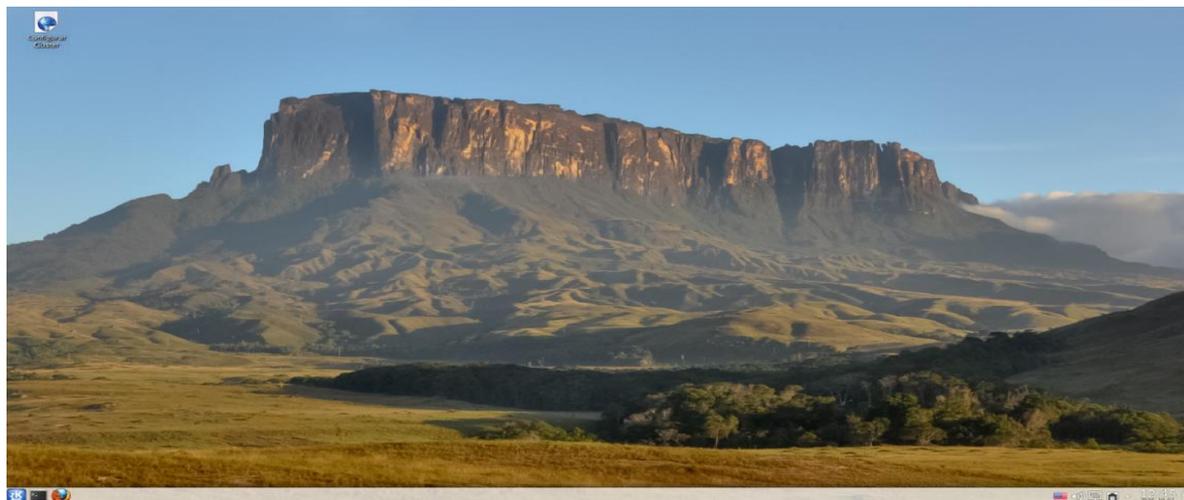


Ilustración 1: Sistema Operativo Live Hadoop

2.4 Aplicaciones Web

Una Aplicación Web es un software diseñado para ayudar al usuario a realizar tareas específicas, que puede ser accedida a través de una conexión de red y en la que todas o algunas de sus partes se descargan de una red cada vez que se ejecuta. Este término puede referirse a las aplicaciones basadas en navegadores que se ejecutan dentro del navegador web del usuario, así como también a las aplicaciones de escritorio o móviles que acceden a la red para obtener información adicional. [2]

Una de las ventajas de las Aplicaciones Web, es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes. Además de tener la posibilidad de ser ejecutadas en múltiples plataformas al emplear tecnologías que permiten su portabilidad; estas tecnologías se dividen en tecnologías del lado del cliente y en tecnologías del lado del servidor, cada una con funciones específicas que serán explicadas en este capítulo. Otra ventaja que poseen las Aplicaciones Web, es

que esta puede ser accedida por uno o más clientes mediante el envío de peticiones a un servidor central en el cual permanece alojada la aplicación, todo esto se logra a través de una conexión a Internet.

Una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.^[3]

Algunos ejemplos



Ilustración 2: Ejemplos Páginas Web

Ventajas de las aplicaciones web:

- **Ahorran costes de hardware y software:** Sólo es necesario usar un ordenador con un navegador web y conectarse a Internet. Las aplicaciones basadas en web usan menos recursos que los programas instalados. Por otra parte las aplicaciones web no requieren canales de distribución como el software tradicional, lo que permite que su precio sea inferior al de los programas instalables.
- **Fáciles de usar:** Las aplicaciones web son muy sencillas de utilizar, sólo necesitará conocimientos básicos de informática para trabajar con ellas. Además, en muchos casos podrá personalizarlas a su gusto y adaptarlas a su forma de trabajo.

- **Facilitan el trabajo colaborativo y a distancia:** Las aplicaciones web pueden ser usadas por varios usuarios al mismo tiempo. Al estar toda la información centralizada no tendrá que compartir pantallas o enviar emails con documentos adjuntos.
- **Escalables y de rápida actualización:** Existe solo una versión de la aplicación web en el servidor. El proceso de actualización es rápido y limpio. Las aplicaciones basadas en web no requieren que el usuario se preocupe por obtener la última versión ni interfieren en su trabajo diario para descargar, instalar y configurar últimas versiones.
- **Provocan menos errores y problemas:** Las aplicaciones web son menos propensas a colgarse y crear problemas técnicos debido a conflictos con hardware, con otras aplicaciones existentes, protocolos o con software personal interno. Todos los usuarios utilizan la misma versión de la aplicación web y los posibles fallos pueden ser corregidos tan pronto son descubiertos.
- **Los datos son más seguros:** Ya no deberá preocuparse de posibles rupturas del disco duro ni de los virus que pueden hacerle perder toda la información. Los proveedores de hosting donde se almacenan las aplicaciones usan granjas de servidores, con altísimas medidas de seguridad, donde guardan los datos de forma redundante y con amplios servicios de backups.^[4]

Usar aplicaciones web ahorra dinero. Empleará mejor su tiempo por no tener que ocuparse de aprender a manejar nuevos programas, ni mantenerlos o hacer copias de seguridad de sus datos y podrá trabajar desde cualquier sitio. Será más eficiente; podrá ganar más y gastará menos.^[5]

2.5 Arquitectura Cliente-Servidor

Para entender cómo funciona este esquema, es necesario describir dos (2) aspectos fundamentales:

Cliente: es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término front-end. El Cliente normalmente maneja todas las funciones relacionadas con la manipulación y

despliegue de datos, además de acceder a los servicios distribuidos en cualquier parte de una red.

Servidor: Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end. Es común que un servidor esté asociado a muchos clientes, es por ello que toma ventaja el proceso de actualización y mantenimiento de la aplicación, ya que los cambios que deban realizarse en el software serán aplicados únicamente en el servidor.

Arquitectura Cliente - Servidores un modelo de computación en el que el servidor, entrega y administra la mayor parte de los recursos y servicios que se consumen por el cliente, estos servicios pueden incluir el acceso de aplicaciones, almacenamiento, intercambio de archivos, entre otros. Este tipo de arquitectura tiene uno o más ordenadores cliente conectados a un servidor central a través de una conexión de red o Internet. [6]



Ilustración 3: Arquitectura Cliente-Servidor

Para la construcción de Aplicaciones Web es necesario buscar soluciones tecnológicas que ayuden al desarrollo, implementación y posterior uso de la aplicación. Para ello se necesita definir que tecnologías se pueden usar tanto para la presentación de la información como para procesar las peticiones de recursos que el cliente solicita.

2.6 Tecnologías Web

Las tecnologías web implican un conjunto de técnicas y recursos que hacen posible desarrollar y a la vez lograr mejores resultados a la hora del desarrollo de aplicaciones web. Estas técnicas permiten la distribución de información en hipertexto o hipermedios enlazados y accesibles a través de Internet. [7]

Las tecnologías web se dividen en dos grandes grupos, tecnologías del lado del cliente y tecnologías del lado del servidor. A continuación se mencionan y se describen cada una de ellas.



Ilustración 4: Tecnologías Web

2.6.1 Tecnologías del lado del Cliente

Las tecnologías del lado del cliente se refieren a aquellos componentes de software que permiten a un cliente realizar operaciones a través de un navegador web que se ejecuta en una computadora y que se conecta a un determinado servidor. Este comportamiento es posible dado que los navegadores se encargan de la visualización de Aplicaciones Web al interpretar sus documentos, estos documentos pueden estar escritos en uno o más lenguajes. A continuación se describen las tecnologías del lado del cliente utilizadas para el desarrollo de esta aplicación.

2.6.1.1 Lenguaje de Marcas de Hipertexto (HTML)

El lenguaje de marcado de hipertexto por sus siglas en inglés HTML (HyperText Markup Language) Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.^[8]

Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos. Para la escritura de este lenguaje, se crean etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades (el contenido en sí mismo y sus atributos).

Por otra parte, cabe destacar que el HTML permite ciertos códigos que se conocen como scripts, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los scripts que pueden agregarse, los más conocidos y utilizados son JavaScript y PHP.^[9]

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Example</title>
5     <link rel="stylesheet" href="styl
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <nav>
12      <a href="one/">One</a>
13      <a href="two/">Two</a>
14      <a href="three/">Three</a>
15    </nav>

```

Ilustración 5: Ejemplo Código HTML

2.6.1.2 Hoja de Estilo en Cascada (CSS)

Hoja de estilo en cascada o CSS por sus siglas en inglés Cascading Style Sheets es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML. El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.^[10]

El CSS fue diseñado para separar el contenido de un documento de su presentación, permite definir elementos como colores, fuentes, márgenes, etc. Esta separación de código y aspecto permite mejorar la accesibilidad, provee flexibilidad, mejor control de las características de presentación, permite que múltiples páginas compartan un formato común, reduce la complejidad y la repetición de estructuras, etc.

El archivo que contiene la hoja de estilo con lenguaje CSS suele tener la extensión “.css”. De todas maneras el documento HTML puede incluir el diseño CSS integrado en el archivo .html (es decir, no se necesita un archivo externo).

Las hojas de estilo en CSS permiten dar formatos generales y particulares a las etiquetas (tags) HTML. Por ejemplo, CSS (utilizando su sintaxis) puede decir “todos los elementos marcados como párrafos deben llevar la fuente de nombre Arial y tener tamaño 15”.

El CSS también permite que una página web se adapte en su presentación al dispositivo que accede (PC, un celular, la impresora como destino, dispositivo táctil), a la resolución de pantalla o al tipo de visitante que visualiza la web, etc.^[11]

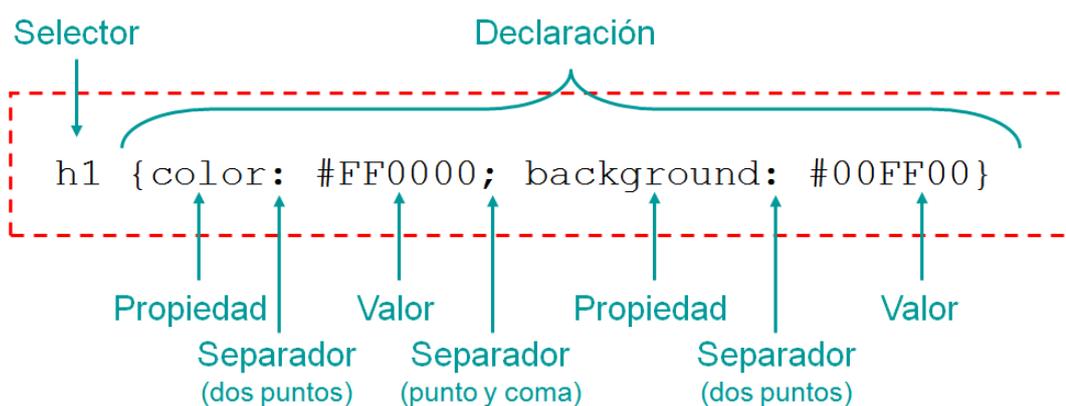


Ilustración 6: Ejemplo Código CSS

2.6.1.3 AJAX

AJAX, acrónimo de *JavaScript asíncrono* y *XML* (*Asynchronous JavaScript And XML*, por sus siglas en ingles), que como se indica en sus siglas está basado en el lenguaje *JavaScript*, lenguaje de script orientado a objetos que brinda a los desarrolladores la capacidad de crear páginas web dinámicas; es un conjunto de técnicas de programación para el desarrollo Aplicaciones Web interactivas. Estas técnicas involucran la capacidad de hacer sin problemas una actualización en la Aplicación Web o en una sección de ella con una petición que realiza el servidor, sin necesidad de refrescar la página completa. ^[12]

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.^[13]

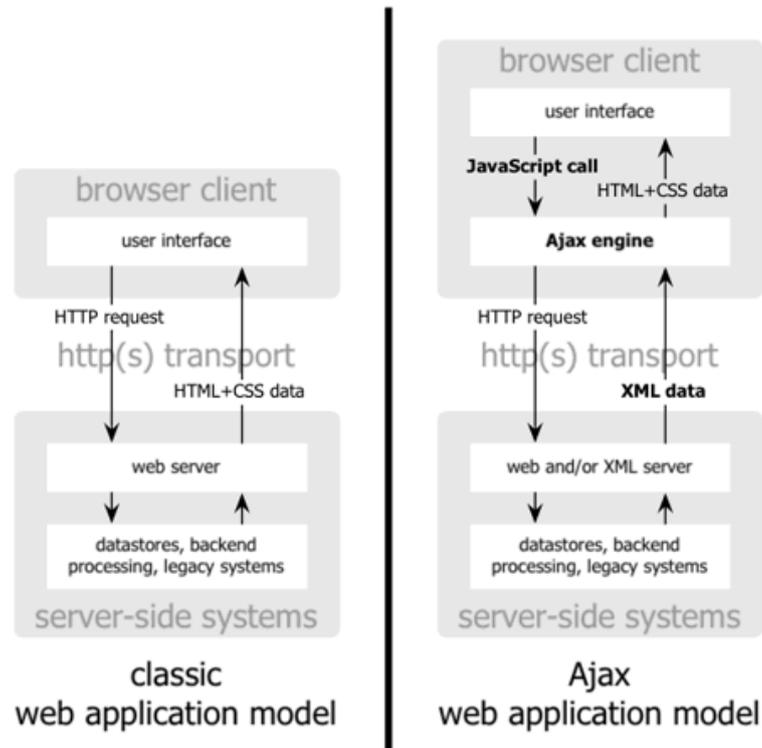


Ilustración 7: Comparación gráfica del modelo tradicional de aplicación web y del nuevo modelo propuesto por AJAX

Dentro de los principales beneficios que conlleva el uso de *AJAX* destacan:

- *AJAX* cubre una amplia gama de tecnologías web que se utilizan para iniciar una *Aplicación Web* mientras se está comunicando con el servidor en segundo plano. Esto beneficia al usuario, ya que no interfiere o interrumpe la página Web que él o ella está utilizando.
- *JavaScript* no es el único lenguaje de script del lado del cliente que utiliza la programación *AJAX*; *VBScript* y otros idiomas tienen este tipo de funcionalidad, pero *JavaScript* es el más popular.
- Existen tecnologías del lado del servidor que son interpretadas por el servidor, para las cuales este lenguaje es independiente ya que es compatible con cualquier tipo de lenguaje de programación web de lado del servidor como *PHP*, *ASP.NET*, *JSP*, entre otros.

- A pesar de lo que su nombre indica, *AJAX* no tiene que correr de una manera asincrónica (en segundo plano), ni tampoco tiene que utilizar *XML*. De hecho, se utiliza más a menudo la notación de objetos de *JavaScript*.

Para este proyecto se hace importante el uso de *AJAX* principalmente porque durante el uso de la aplicación es necesario enviar peticiones al servidor sin interrumpir las acciones realizadas por el usuario, permitiéndole así realizar múltiples tareas sin necesidad de mantenerse refrescando la aplicación.

2.6.1.4 JavaScript

JavaScript, como se mencionaba previamente en la definición de *AJAX*, es un lenguaje ligero e interpretado, orientado a objetos, más conocido como el lenguaje de script que brinda a los desarrolladores la capacidad de crear páginas web dinámicas al interactuar con el usuario; siendo *ECMAScript* el *JavaScript* estándar, y a partir de 2012, todos los navegadores modernos soportan completamente *ECMAScript* 5.1. Los navegadores más antiguos soportan por lo menos *ECMAScript* 3.^[14]

Una de las principales ventajas de JavaScript es que permite reconocer eventos, es decir, si un usuario realiza una acción, ésta puede ser identificada y proceder a realizar algún proceso en específico, por ejemplo, el cambio del valor de un formulario, o la pulsación de un enlace.

El código JavaScript de los manejos de eventos puede ser interno, es decir, que está embebido en el código HTML de la página web encerrado entre etiquetas `<script></script>`, o externo en donde las instrucciones JavaScript están en un archivo con extensión `.js` separado del archivo con código HTML y luego se incluye en este último.

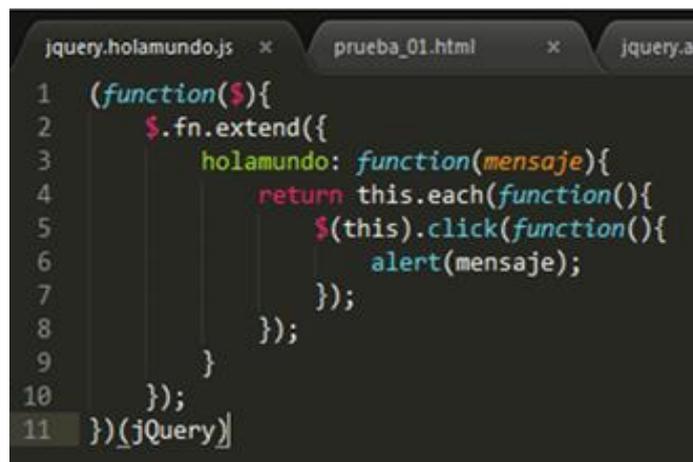
2.6.1.5 JQuery

JQuery es una librería escrita en lenguaje JavaScript que simplifica la interacción con documentos HTML, provee manejo de eventos, animaciones, manipula el DOM e interactúa con *AJAX*. JQuery está diseñado para cambiar la manera de escribir en JavaScript.^[15]

JQuery se especializa en permitir al desarrollador seleccionar los elementos en una página, determinar que navegador se está usando y cuáles son sus capacidades; además uno de sus principales atractivos son los efectos visuales que produce: sombras, fondos, entre otros.

Dentro de los principales beneficios que conlleva el uso de *JQuery* destacan:

- Utiliza sintaxis muy parecida a css.
- Funciona con series de elementos.
- Permite manipular series de elementos y modificarlas con una simple línea de código.
- Te ayuda a concentrarte en el resultado final.
- Es muy fácil de expandir, ya que cuenta con gran cantidad de plug-ins que se pueden utilizar o hasta crear uno propio.
- Compatible con todos los navegadores modernos.



```
1 (function($){
2   $.fn.extend({
3     holamundo: function(mensaje){
4       return this.each(function(){
5         $(this).click(function(){
6           alert(mensaje);
7         });
8       });
9     }
10  });
11 })(jQuery)
```

Ilustración 8: Ejemplo Código JQuery

2.6.1.6 JQuery UI

JQuery UI es una librería de componentes que aporta un conjunto de plugins, widgets y efectos visuales para la creación de aplicaciones web. Cada componente o módulo se desarrolla de acuerdo a la filosofía de JQuery (encuentra algo, manipúlalo).^[16]

JQuery UI ofrece acceso a un conjunto completo de controles de interfaces de usuario (widgets), permite añadir comportamientos complejos a los

elementos (interacciones), y cuenta con una API que brinda transiciones animadas para dichos elementos (efectos).

2.6.2 Tecnologías del lado del Servidor

Las tecnologías del lado del servidor son aquellas tecnologías que son usadas para brindar soporte a los requerimientos del lado del cliente y que son implementadas directamente en el servidor en el contexto del desarrollo de una aplicación web.

El servidor realiza operaciones para generar respuestas a las peticiones del usuario cuando la información o funciones a las que éste quiere acceder están o no disponibles en el lado del cliente.

El propósito principal de las tecnologías que se encuentran del lado del servidor es que permiten modificar el contenido de una Aplicación Web de forma online al crear una interfaz dinámica entre el cliente y el servidor, mediante el uso de lenguajes de programación como PHP, el cual será descrito a continuación.

2.6.2.1 PHP

PHP, acrónimo de Hypertext Preprocessor, es un lenguaje interpretado de alto nivel embebido en páginas HTML, lo que lo hace realmente fácil de utilizar, y ejecutado en el servidor.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores.^[17]

En principio, permite establecer una conexión con la base de datos, a través de lo cual es posible, por ejemplo, presentar en pantalla datos personales del cliente cuando éste lo requiera. Pero la característica más importante de PHP es que permite modificar dinámicamente el contenido de una página, lo cual resulta esencial hoy en día. Puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix, Microsoft Windows, Mac OS X, RISC OS y probablemente otros más, admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros, de modo

que con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.^[18]

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.



Ilustración 9: Ejemplo Código PHP

2.6.2 2 Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos, orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Mientras que R ha sido tradicionalmente el lenguaje de programación elegido por los científicos de datos, poco a poco va cediendo terreno a Python. Si bien hay varias razones para el cambio, tal vez la más importante es que Python es de uso general y comparativamente más fácil de aprender, mientras que R sigue siendo un entorno de programación bastante complejo de dominar. En un mundo cada vez más dependiente de los datos y con “hambre” de científicos de datos, lo fácil gana.

Python es fácil para los analistas de aprender y usar, pero lo suficientemente potente como para hacer frente a los problemas más difíciles en prácticamente cualquier dominio. Se integra bien con la infraestructura de TI existente, y es muy independiente de la plataforma. Entre las lenguas modernas, la agilidad y la productividad de las soluciones basadas en Python son legendarias. Empresas de todos los tamaños y en todos los están utilizando Python para manejar su negocio y gestionar sus datos.

Python está avanzando dentro de la empresa tradicional, sus bibliotecas para la manipulación y análisis de datos lo convierten en un gran ajuste para el auge de Big Data.

2.6.3 Otras Tecnologías

La evolución ha tenido como resultado que hay una gran cantidad de tecnologías, librerías, herramientas y estilos arquitectónicos para desarrollar una aplicación web y actualmente existen tecnologías que permiten facilitar esta tarea como desarrollador de software.

2.6.3.1 Git

Git es un sistema orientado al desarrollo de aplicaciones que permite un control distribuido de versiones de código haciendo énfasis en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Git es un software diseñado por Linus Torvalds en el año 2005. Se ha convertido desde entonces en un sistema gratuito de control de versiones con funcionalidad plena para cualquier desarrollador.^[20]

La principal ventaja de esta herramienta es que cada directorio en *Git* es un repositorio con grandes capacidades de rastreo de versiones y manejo de historia, que no depende de un acceso a una red específica ni a un servidor central, características que se hicieron necesarias durante el desarrollo de la aplicación, ya que en múltiples ocasiones el equipo de desarrollo implementaba funcionalidades por separado, las cuales debían ser integradas al código fuente, esta herramienta se encargaba de realizar ese trabajo con el manejo de versiones y la integración de cambios realizados.

Entre las características más relevantes de esta herramienta se encuentran:

- Fuerte apoyo al desarrollo distribuido, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones y fusión de cambios. Cada programador cuenta con una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales.
- Los almacenes de información pueden publicarse por *HTTP*, *FTP*, *rsync* o mediante un protocolo nativo, ya sea a través de una conexión *TCP/IP* simple o a través de cifrado *SSH*.
- Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos.

Capítulo III: Marco Metodológico

En este capítulo procederemos a definir en qué consiste una metodología de desarrollo tradicional y ágil, describiremos cuál fue la metodología utilizada para el desarrollo del sitio web y cada una de las características y sus componentes.

3.1 Metodologías tradicionales de desarrollo de software

Las metodologías tradicionales se caracterizan por recibir especificaciones precisas de los requisitos y el modelado del sitio web, tienen mayor énfasis en la planificación y control del proyecto.

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software con la finalidad de conseguir un software más eficiente, por esto hacen énfasis en la planificación total de todo el trabajo que se vaya a realizar para que al momento en el que el trabajo esté completamente detallado se comienza con el desarrollo del software. Este método se centra específicamente en el control del proceso a través de un riguroso detalle de cada uno de los roles, actividades, artefactos y herramientas para el modelado y la documentación detallada.

Las metodologías tradicionales no se adaptan de manera adecuada a los cambios, es por esta razón que no son métodos adecuados al momento de trabajar en un entorno sensible a cambios o donde los requisitos no puedan predecirse.

Existen varias metodologías tradicionales, de las cuales se pueden mencionar las siguientes:

- RUP (*Rational Unified Procces*)
- MSF (*Microsoft Solution Framework*)
- *Win-Win Spiral Model*
- *Iconix*

3.2 Metodologías para el desarrollo ágil

Las metodologías ágiles son un conjunto de técnicas que permiten la gestión de un proyecto, con el objetivo de evitar las metodologías tradicionales ya que se enfocan en los desarrolladores y en los resultados del software. La metodología de desarrollo ágil o modelado ágil (*Agile Modeling, AM*), fue propuesta por Scott Ambler (uno de los principales “agilistas” en el mundo), como un conjunto de valores, principios y prácticas para modelar software que pueda ser aplicados a un proyecto de desarrollo de una forma efectiva y ligera, que complementan a las diferentes metodologías o procesos de desarrollo, ágiles o no, de manera de hacerlas adaptables a requerimientos cambiantes del cliente y lograr un proceso acorde a las necesidades de cada proyecto.

Este un marco de trabajo conceptual de la ingeniería del software que promueve las iteraciones en el desarrollo, cada iteración está constituida por una planificación, análisis de los requerimientos, diseño, codificación, revisión y documentación. Los métodos ágiles enfatizan las comunicaciones entre los desarrolladores a través de la documentación y reuniones, en su gran mayoría estas reuniones se caracterizan por incluir diseñadores de iteraciones, escritores de documentación, ayuda y directores de proyecto. [21]

Los procesos ágiles se fundamentan en diversos principios como:

- La prioridad más alta es satisfacer al cliente con entregas rápidas y continuas del producto de software.
- Los cambios a los requerimientos son aceptados, inclusive durante el desarrollo. Un proceso ágil permite incorporar los cambios a fin de que el cliente posea ventaja competitiva.
- El software se entrega frecuentemente.
- Los clientes y los desarrolladores deben trabajar juntos diariamente hasta la finalización del proyecto.
- El producto de software se desarrolla con personas motivadas, creando un ambiente propicio que de soporte y confianza a los desarrolladores en el éxito del proyecto.
- La manera más eficiente y efectiva de transmitir la información dentro de un equipo de desarrollo es la interacción personal.

- El software que funciona es la principal medida del progreso.
- Un proceso ágil promueve el desarrollo. Los clientes, desarrolladores y usuarios deben mantener un ritmo constante de trabajo.
- Prestar continua atención a la excelencia técnica y realizar buenos diseños.
- La simplicidad permitirá abarcar la cantidad de trabajo del proyecto.
- En una buena arquitectura, los requerimientos y diseños, son producto de un equipo bien organizado.
- En intervalos de tiempo regulares, el equipo de trabajo reflexiona sobre cómo serían más efectivos y ajustan su comportamiento adaptativamente.

Características de un desarrollo ágil:

- Proceso iterativo e incremental
- Mitigación del riesgo mediante iteraciones fijas
- Mejora continua
- Calidad desde el primer día
- Priorización de requerimientos de acuerdo a su valor
- Equipos dedicados y auto-gestionados
- Colaboración continua con el cliente
- Incorporar al cambio
- Prácticas de desarrollo modernas

Dentro de las metodologías ágiles más destacadas se pueden mencionar las siguientes:

- Programación extrema o XP (*Extreme Programmig*)
- Metodología Scrum
- Crystal Clear
- Método de Desarrollo de Sistemas Dinámicos o DSDM (*Dynamic Systems Development Method*)
- Desarrollo Basado en Funcionalidades o FDD (*Feature Driven Development*)
- Desarrollo de Software Adaptable o ASD (*Adaptive Software Development*)

- *XBreed*

3.3 Diferencias entre las metodologías

En el cuadro que se presentara a continuación compararemos las características más importantes de las metodologías ágiles y las metodologías tradicionales.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en eucarísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Tabla 1: Diferencia entre metodologías

3.4 Metodología de Desarrollo del Software

Existen estándares y normativas que determinan los pasos para la creación de un software (ciclo de vida del software) al definir los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso.

El ciclo de vida propuesto por *XP* divide el desarrollo del software en bloques de tiempo, llamado iteraciones, en donde se construyen un conjunto de funcionalidades específicas en un período de tiempo estimado.

Para el desarrollo de la aplicación web se utilizó una adaptación de la metodología de desarrollo ágil de aplicaciones llamada Programación Extrema (*XP*), con el fin de desarrollar un sistema usable, robusto y accesible mediante una programación organizada y con capacidad de respuesta inmediata a cambios en requerimientos durante el proceso de desarrollo. Centra sus procesos en la simplicidad, la comunicación y la satisfacción del usuario; brinda flexibilidad en la definición de los requerimientos del sistema y reduce los tiempos de desarrollo.

El método *XP* se centra en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, por lo que establece varios roles con responsabilidades únicas para así asegurar la premisa de “si desean avanzar en el trabajo, tendrán que hablar entre ustedes”. Además, este método abarca un conjunto de reglas y prácticas que ocurren en el contexto de 4 fases del marco de trabajo: planificación, diseño, codificación y pruebas.^[22]

La propuesta original de éste método contempla un equipo de desarrollo integrado por varios roles con tareas específicas. A continuación se detallan estos roles con sus respectivas asignaciones.

Rol	Tarea
Programador	Producir el código que lleve a cabo las funcionalidades del software y describir las pruebas unitarias del mismo.
Cliente	Escribir las historias de usuario y las pruebas funcionales para validar la implementación del software.
Tester	Ayudar al cliente a escribir las pruebas funcionales, ya que éste las

Ilustración 11: Roles del método XP

Una de las cualidades más destacables del método *XP* es su sencillez, tanto en su aprendizaje como en su aplicación, reduciendo así los costos de implantación en un equipo de desarrollo. Siendo una de las razones por la que en esta investigación se decide utilizar éste método con ciertas modificaciones.

A continuación se describen las adaptaciones del método de desarrollo de software XP aplicadas en este trabajo de acuerdo a los roles utilizados, la estructuración de las fases y las iteraciones realizadas.

3.4.1 Roles

En la propuesta original del método se especifican siete roles distintos, sin embargo, para el desarrollo de esta aplicación se cuenta con un equipo reducido, por lo que es necesario reestructurar la asignación de responsabilidades según los recursos disponibles, generando como resultado los siguientes roles:

Usuario: El usuario o cliente es la persona o ente que impulsa el proyecto, definiéndolo según sus necesidades. Este rol lo lleva a cabo la Facultad de Ciencias, Escuela de Computación.

Programador: El desarrollador o programador trabaja en estrecha colaboración con el usuario para captar sus requerimientos. Éste rol, además de llevar a cabo el análisis de requerimientos, se encarga de la planificación, diseño, programación y ejecución de las pruebas.

Entrenador (Coach): Es la persona encargada de proveer directrices al equipo para garantizar la aplicación de las prácticas XP; en este proyecto, las responsabilidades de este rol son llevadas a cabo por el tutor.

Consultor: Es la persona designada para instruir al equipo con sus conocimientos de surgir algún problema en un tema en específico. Durante el desarrollo de la aplicación, el tutor realizó las acciones asociadas a este rol.^[23]

Además de la repartición de responsabilidades por roles, el método XP abarca un conjunto de buenas prácticas que ocurren en el contexto de cuatro (4) fases, cuya aplicación en esta investigación se ilustra y se explica a continuación.



Ilustración 12: Fases del Método XP adaptado

3.4.2 Planificación

Es el proceso de recolectar los requerimientos funcionales de la aplicación según las necesidades expresadas por el cliente, para así trazar objetivos por tiempo y prioridad que permitan programar dichos requerimientos.

Como parte de la configuración del método XP, en éste trabajo se utilizan tablas descriptivas como artefacto de la fase de planificación

Iteración 0	
Descripción	
Duración	
Historia	Duración

Ilustración 13: Ejemplo Tabla descriptiva fase planificación

3.4.3 Diseño

El método XP recomienda un diseño simple respecto a una presentación compleja porque su objetivo principal es ofrecer una guía de implementación para las historias de usuario, pero este proceso se realiza tanto antes como después del comienzo de la etapa de codificación, debido a que el diseño ocurre de manera continua a medida que se codifica la aplicación. Para esta fase, los entregables son prototipos que permiten visualizar las funcionalidades de la aplicación esbozando su contenido y la disposición física de los elementos.

3.4.4 Codificación

Luego de crear las historias de usuario, y realizar el prototipo correspondiente a cada una de ellas, el desarrollador se centra en lo que debe implementarse, para ello el método XP sugiere que dos programadores trabajen en una misma computadora para complementar los conocimientos y asegurar la calidad de las soluciones. Los entregables en esta fase se refieren a los requerimientos programados hechos componentes de software totalmente funcionales, de los cuales se muestran extractos del código más relevante o las vistas resultantes, que luego, para verificar su correcto comportamiento se realizan pruebas en la siguiente fase.

3.4.5 Pruebas

En esta fase se comprueba que la aplicación desarrollada, o el componente de ella más recientemente desarrollado, se comporte como debería, es decir, se realizan pruebas funcionales para verificar la robustez de cada una de las funcionalidades implementadas. Adicionalmente, se aplican pruebas unitarias para asegurar que un conjunto de funcionalidades actúen correctamente por separado, y para validar el comportamiento esperado de la aplicación como un todo, se realizan pruebas de integración. Además de las pruebas mencionadas para verificar el correcto funcionamiento del software desarrollado, se llevan a cabo pruebas de aceptación para evaluar el grado de satisfacción del cliente con la aplicación.

Todas las fases descritas anteriormente conforman los procesos por los cuales debe pasar un proyecto que se desarrolla según las directrices que proporciona el método XP, y para cumplir con las premisas de adaptabilidad y entregas tempranas de software, promulgadas por la metodología ágil, éste desarrollo debe ser realizado por partes, es decir, por cada módulo o grupo de funcionalidades que comprenda la aplicación, se ejecutan todas las etapas de desarrollo mencionadas en un tiempo determinado, éste proceso se denomina iteración.

Capítulo IV: Marco Aplicativo

En este capítulo se describe el proceso de desarrollo del software, se mencionan las actividades y herramientas utilizadas siguiendo las directrices de la adaptación de la metodología XP, descrita anteriormente. A continuación se describen las distintas iteraciones y tareas llevadas a cabo durante el desarrollo de la aplicación.

El desarrollo de la aplicación se divide en bloques de trabajo para así descomponer las tareas necesarias para llevar a cabo el producto final en partes claramente identificables. Cada una de estas partes se llama iteraciones, las cuales agrupan el desarrollo de un conjunto de funcionalidades que dan respuesta a los siguientes requerimientos:

1. Crear imagen ISO (Descargar Sistema Operativo Live Hadoop)
2. Crear targz (Descargar Sistema Operativo Live Hadoop)
3. Descargar Sistema Operativo Live Hadoop con perfiles pre-seleccionados (Descargar Sistema Operativo Live Hadoop)
4. Disponer Sistema Operativo Live Hadoop en red (Configuración Clúster Live Hadoop)
5. Encender remotamente los nodos pertenecientes al clúster (Configuración Clúster Live Hadoop)
6. Iniciar Clúster Live Hadoop (Configuración Clúster Live Hadoop)
7. Detener Nodo (Administración Clúster Live Hadoop)
8. Detener Clúster (Administración Clúster Live Hadoop)
9. Iniciar Nodo (Administración Clúster Live Hadoop)
10. Iniciar Clúster (Administración Clúster Live Hadoop)
11. Ver estatus nodo (Monitorear Clúster Live Hadoop)
12. Ver estatus Clúster (Monitorear Clúster Live Hadoop)

Estos requerimientos son detallados en Historias de Usuarios a lo largo de las 6 iteraciones que se realizan para la generación de artefactos de manera incremental, tomando en cuenta, para cada iteración, las 4 fases de desarrollo descritas en la sección 3.4. A continuación una descripción breve de las mismas:

- **Iteración 0:** Se refiere a la planificación inicial, y a la preparación del entorno de desarrollo.
- **Iteración 1:** Se planifican las historias de usuario correspondientes a la creación del módulo descarga, generando como entregable el módulo de Descargar Sistema Operativo Live Hadoop.
- **Iteración 2:** Se planifican las historias de usuario correspondientes a la creación del módulo configuración, generando como entregable el módulo de Configuración Clúster Live Hadoop.
- **Iteración 3:** Se planifican las historias de usuario correspondientes a la creación del módulo de administración, generando como entregable el módulo de Administrar y Monitorear Clúster Live Hadoop.
- **Iteración 4:** A partir de la iteración 1, en la fase de pruebas, se llevan a cabo pruebas funcionales y de integración que verifican el correcto comportamiento de las funcionalidades desarrolladas, además de sesiones de reunión con el cliente para obtener su opinión sobre los componentes de software entregados. Razón por la que en esta iteración sólo se aplican pruebas de aceptación para ponderar el grado de satisfacción de los usuarios con el sistema en general.

4.1 Iteración 0: Planificación y Diseño

En esta iteración se conciben las primeras ideas de diseño de interfaces de usuario contemplando el desarrollo de las funcionalidades e interacciones con el cliente y análisis de la aplicación, con la finalidad de llevar a cabo el levantamiento de información contemplando el desarrollo de las funcionalidades e interacciones.

En el desarrollo de esta iteración se aplican las fases de planificación, diseño, codificación y pruebas las cuales se explican a continuación.

Planificación

En esta fase se realizó un análisis detallado con el fin de establecer las directrices a seguir para darle respuesta a los requerimientos del usuario. Se especifican con detalle las historias de usuario involucradas y las reuniones realizadas para esta iteración.

Iteración 0	
Descripción	Elaboración de ideas conceptuales de la aplicación, instalación, configuración del entorno de desarrollo y levantamiento de información.
Duración Iteración	de 12 días
Historia	Duración
Levantamiento de información	24 horas
Preparación del entorno de desarrollo	48 horas
Recolección de material básico para aprendizaje del Lenguaje	48 hora

Tabla 2: Planificación - Iteración 0

Estas tareas planificadas que se muestran en la tabla anterior, corresponden a las necesidades expresadas por el cliente y registradas en Historias de Usuario, tal como lo muestra la tabla 3.

Titulo	Levantamiento de Información
Duración	24 horas
Descripción	Se utiliza una serie de técnicas de indagación, los cuales permiten conocer las necesidades y requerimientos del usuario, con la finalidad de recopilar datos necesarios para darle solución a la problemática planteada.
Titulo	Preparación del entorno de desarrollo
Duración	48 horas
Descripción	Se debe lleva a cabo la configuración del entorno de programación, necesario para el desarrollo de la aplicación. La Configuración del entorno incluye principalmente, la selección de versiones, instalación y configuración de tecnologías como PHP, MySQL y Git.
Titulo	Recolección de material básico para aprendizaje del Lenguaje
Duración	48 horas
Descripción	Se recolecta todo el material posible sobre el lenguaje y las tecnologías a utilizar para el desarrollo de la aplicación, tutoriales, libros, guías y se estudian tanto como sea necesario para poder comenzar el proceso.

Tabla 3: Historias de Usuario - iteración 0

Diseño

En esta etapa se utiliza una serie de técnicas de indagación, los cuales permiten conocer las necesidades y requerimientos del usuario. Algunos de ellos son:

1. **Tormenta de ideas:** Se logra intercambiar ideas entre el equipo de desarrollo para dar solución al problema. Para llevar a cabo esta técnica comenzamos presentando el tema central, que es la configuración y monitoreo de un clúster Live Hadoop, en donde cada integrante del equipo expuso sus ideas y conceptos para tener en un corto periodo de tiempo las bases para el desarrollo de la aplicación de manera creativa.



Ilustración 14: Tormenta de ideas

2. **Requerimientos funcionales y no funcionales:** Mediante reuniones con el cliente, se levantó la información necesaria para establecer los requerimientos funcionales y no funcionales de la aplicación.

Requerimientos Funcionales Descargar Sistema Operativo Live Hadoop

- | |
|--|
| <ul style="list-style-type: none">✓ Descargar imagen ISO del Sistema Operativo Live Hadoop✓ Descargar targz del Sistema Operativo Live Hadoop✓ Escoger los módulos que se quiere que pertenezcan al Sistema Operativo. |
|--|

Tabla 4: Requerimientos Funcionales de Descargas

Requerimientos Funcionales Configuración Clúster Live Hadoop

- | |
|---|
| <ul style="list-style-type: none">✓ Disponer el Sistema Operativo en red.✓ Encender remotamente las maquinas.✓ Iniciar Clúster Live Hadoop. |
|---|

Tabla 5: Requerimientos Funcionales de Configuración

Requerimientos Funcionales Administración y Monitoreo Clúster Live Hadoop
--

- | |
|--|
| <ul style="list-style-type: none">✓ Detener Clúster Live Hadoop.✓ Detener Nodo.✓ Iniciar Nodo.✓ Monitorear estado del Clúster Live Hadoop.✓ Monitorear estado de nodo. |
|--|

Tabla 6: Requerimientos Funcionales de Administración y Monitoreo

Requerimientos No Funcionales

- ✓ Desarrollado bajo la plataforma de PHP.
- ✓ Brindarle a los usuarios una interfaz que les permita realizar las tareas de forma sencilla, basándose en los principios que garanticen la usabilidad de la aplicación.
- ✓ Proveer ayuda a los Usuarios.
- ✓ Garantizar la escalabilidad del sistema.

Tabla 7: Requerimientos No Funcionales

3. **Elaboración de casos de uso:** los casos de uso buscan describir las actividades que deben realizarse para llevar a cabo algún proceso. Sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios.

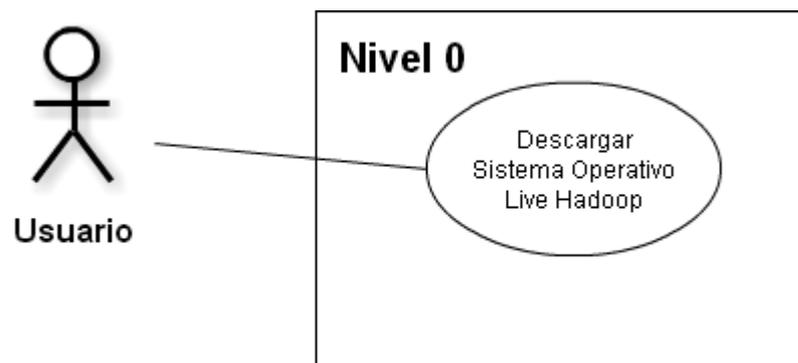


Ilustración 15: Caso de Uso Descargar Sistema Operativo Live Hadoop - Nivel 0

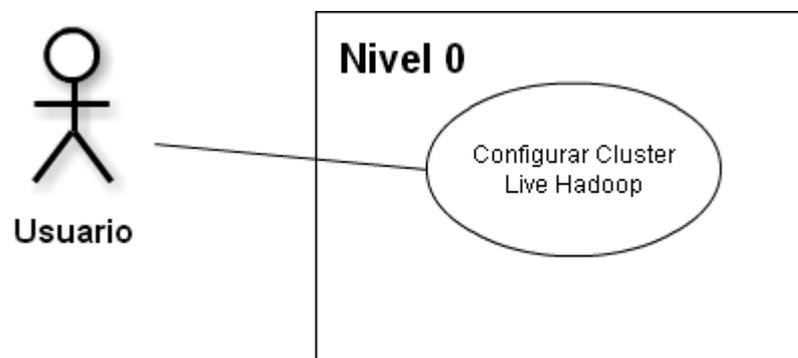


Ilustración 16: Caso de Uso Configurar Clúster Live Hadoop - Nivel 0

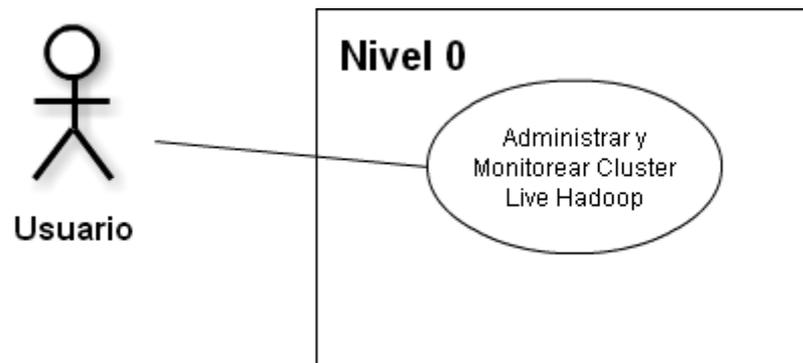


Ilustración 17: Caso de Uso Administrar y Monitorear Clúster Live Hadoop - Nivel 0

Para el módulo de descargar se creó el caso de uso correspondiente para este proceso, de manera de dar una visión más detallada de las funcionalidades de la aplicación. (Ver ilustración 17).

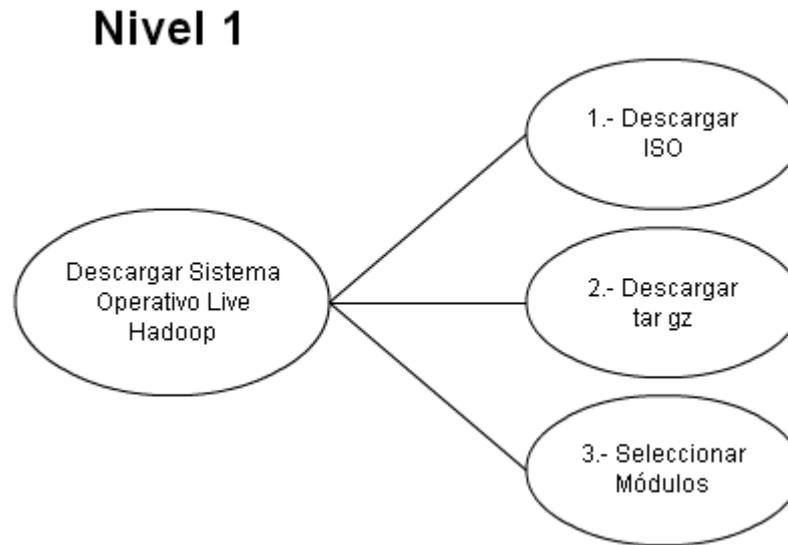


Ilustración 18: Caso de Uso Descargar Sistema Operativo Live Hadoop - Nivel 1

A continuación se describe la figura 17:

1. Descargar ISO: Permite al usuario escoger los módulos que quiere que pertenezcan al sistema operativo y descargar una imagen ISO del sistema operativo Live Hadoop compuesta por dichos módulos.
2. Descargar tar.gz: Permite al usuario escoger los módulos que quiere que pertenezcan al sistema operativo y descargar un archivo .tar.gz que contiene el sistema operativo Live Hadoop compuesto por dichos módulos.
3. Seleccionar Módulos: Permite al usuario seleccionar los módulos que quiere que pertenezcan al sistema operativo para después crear una copia del sistema operativo Live Hadoop con dichos módulos.

Para el módulo de configurar se creó el caso de uso correspondiente para este proceso, de manera de dar una visión más detallada de las funcionalidades de la aplicación. (Ver ilustración 18).

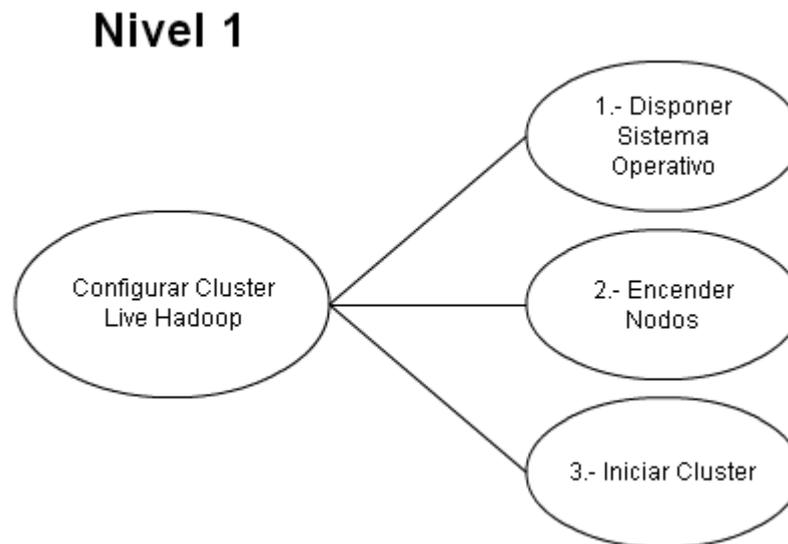


Ilustración 19: Caso de Uso Configurar Clúster Live Hadoop - Nivel 1

A continuación se describe la figura 18:

1. Disponer Sistema Operativo: Permite al usuario disponer en la red el sistema operativo Live Hadoop.
2. Encender Nodos: Permite al usuario encender las maquinas que van a pertenecer al clúster Live Hadoop.
3. Iniciar Clúster: Luego de realizar las configuraciones pertinentes permite al usuario iniciar un clúster Live Hadoop con dichas configuraciones.

Para el módulo de administrar y monitorear se creó el caso de uso correspondiente para este proceso, de manera de dar una visión más detallada de las funcionalidades de la aplicación. (Ver ilustración 19).

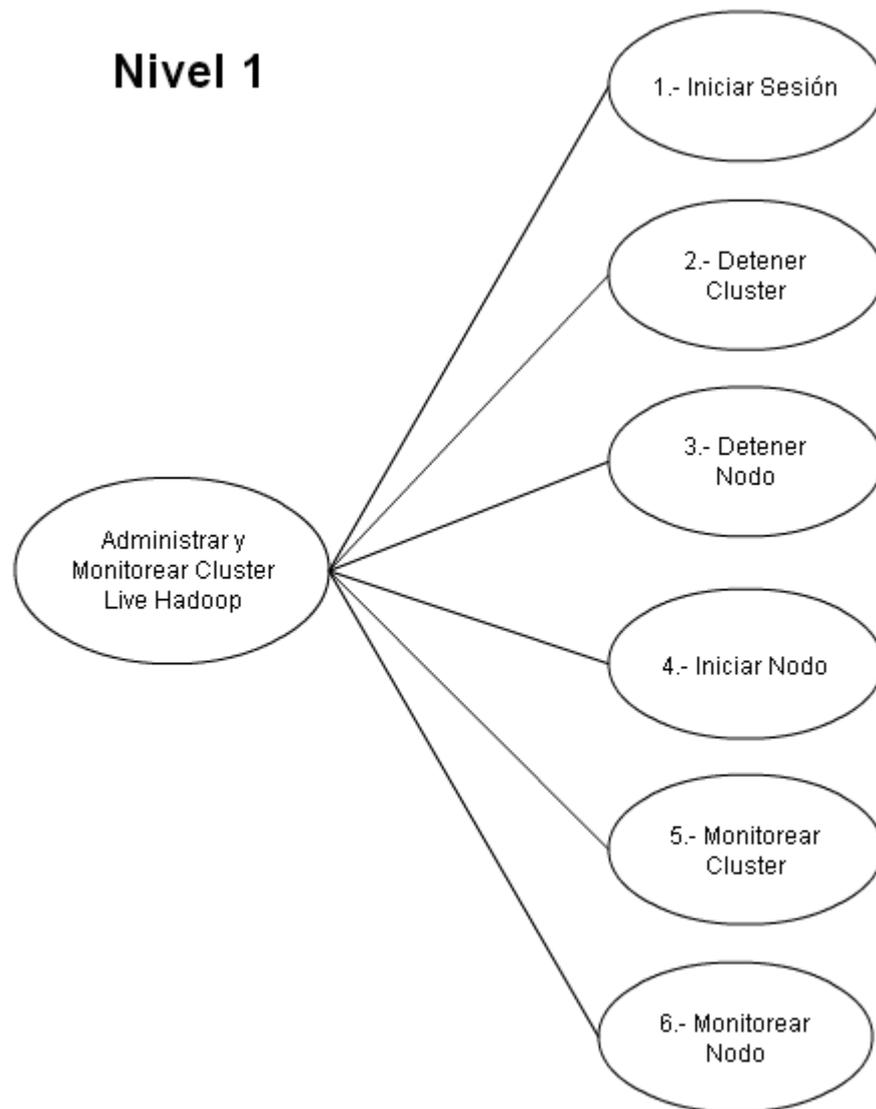


Ilustración 20: Caso de Uso Administrar y Monitorear Clúster Live Hadoop - Nivel 1

A continuación se describe la figura 19:

1. Detener Clúster: Permite al usuario parar el procesamiento que se esté ejecutando en el clúster Live Hadoop.
2. Detener Nodo: Permite al usuario parar el procesamiento que se esté ejecutando en un nodo que pertenezca al clúster Live Hadoop.

3. Iniciar Nodo: Permite al usuario unir un nuevo nodo al clúster Live Hadoop.
4. Monitorear Clúster: Permite al usuario monitorear los aspectos generales del clúster Live Hadoop.
5. Monitorear Nodo: Permite al usuario monitorear el funcionamiento de un nodo perteneciente al clúster Live Hadoop.
4. **Especificaciones Técnicas:** la aplicación WEB para el proceso de gestión de bienes e inventario se desarrolló utilizando las siguientes tecnologías:
 - ✓ HTML
 - ✓ CCS
 - ✓ JavaScript
 - ✓ PHP 5.3.3 sobre el servidor Apache /2.2.16

Una vez configurado el ambiente de desarrollo se procede con el diseño de la base de datos, identificando tablas, campos y tipos de datos. Cabe destacar que es una base de datos de una sola tabla que nos servirá para controlar el acceso al modulo de administrar y monitorear clúster Live Hadoop.

Codificación

En este paso se la estructura general que va a tener la aplicación, se decide donde irá el logo, que colores tendrá la barra principal, el tamaño y posicionamiento de la misma y se codifica esta estructura para ser utilizada en las siguientes iteraciones.

Pruebas

Con el fin de comprobar el cumplimiento de los objetivos planteados para esta iteración, se realizan pruebas para verificar el funcionamiento y comprensión de la estructura creada.

4.2 Iteración 1: Módulo Descargar

Durante esta iteración se desarrolla el módulo que nos dará acceso a la descarga del sistema operativo Live Hadoop, permitiéndonos elegir los módulos que se quiere que lo conformen.

Planificación

En esta fase se especificara con detalle el plan de la iteración, se establecen las tareas a realizar para desarrollar las funcionalidades que permiten satisfacer los requisitos del cliente.

Iteración 1	
Descripción	Desarrollo del módulo Descargar Sistema Operativo Live Hadoop y creación de perfiles preseleccionados.
Duración Iteración	de 10 días
Historia	Duración
Página de inicio	24 horas
Selección de Perfiles	24 horas
Selección otros Módulos	24 horas
Recolección de material de apoyo.	10 horas

Tabla 8: Planificación de la Iteración 1

Las historias de usuario involucradas en esta iteración son las especificadas en la tabla 9.

Título	Página de Inicio
Duración	24 horas
Descripción	La aplicación debe facilitar la navegación de manera que el usuario pueda acceder a las principales opciones ofrecidas por el módulo.
Título	Selección de Perfiles
Duración	24 horas
Descripción	El usuario tiene la opción de escoger perfiles que ya tienen ciertos módulos del sistema operativo Live Hadoop preseleccionados dependiendo de las tareas que se quieran ejecutar.
Título	Selección Otros Módulos
Duración	24 horas
Descripción	El usuario tiene la opción de escoger los módulos existentes del sistema operativo Live Hadoop para que compongan el mismo, dependiendo de las tareas que se quieran ejecutar.
Título	Recolección de material de apoyo
Duración	10 horas
Descripción	Se recolecta todo el material posible para despejar dudas y seguir con el proceso de desarrollo.

Tabla 9: Historias de Usuario - Iteración 1

Diseño

En esta etapa se realiza el prototipo de la interfaz del módulo, que debe presentar todas las funcionalidades antes descritas. (Ver ilustración 21)



Ilustración 21: Diseño Módulo Descargar

Codificación

Para esta iteración se implementa una tabla que permite presentar los perfiles predeterminados, señalando también los módulos que los componen. Además se crea otra sección, de manera de que si el usuario no desea ninguno de los perfiles presentados, el usuario pueda seleccionar y combinar los módulos existentes para crear una copia del sistema operativo Live Hadoop. En la ilustración 22 se muestra parte del código utilizado.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
2 <html>
3 <head>
4 <title>Mi primera página con estilo</title>
5 </head>
6
7 <body>
8
9 <!-- Menú de navegación del sitio -->
10 <ul class="navbar">
11 <li><a href="indice.html">Página principal</a>
12 <li><a href="meditaciones.html">Meditaciones</a>
13 <li><a href="ciudad.html">Mi ciudad</a>
14 <li><a href="enlaces.html">Enlaces</a>
15 </ul>
16
17 <!-- Contenido principal -->
18 <h1>Mi primera página con estilo</h1>
19
20 <p>¡Bienvenido a mi primera página con estilo!
```

Ilustración 22: Extracto de código Módulo Descargar

Pruebas

Para comprobar el cumplimiento de los lineamientos establecidos en esta iteración, se realizan pruebas funcionales aplicadas de forma manual verificando el correcto funcionamiento de las interfaces que posee esta sección.

4.3 Iteración 2: Módulo Configurar

Durante esta iteración se desarrolla el módulo que nos dará acceso a la configuración del clúster Live Hadoop, permitiéndonos decidir la distribución inicial del mismo.

Planificación

En esta fase se especificara con detalle el plan de la iteración, se establecen las tareas a realizar para desarrollar las funcionalidades que permiten satisfacer los requisitos del cliente.

Iteración 2	
Descripción	Desarrollo del módulo Configurar Clúster Live Hadoop e iniciar el mismo.
Duración Iteración	de 20 días
Historia	Duración
Página inicio	24 horas
Página encendido de máquinas	24 horas
Página encendido manual	24 horas

Página encendido remoto	24 horas
Página seleccionar IP	24 horas
Página fin	24 horas
Recolección de material de apoyo.	10 horas

Tabla 10: Planificación de la Iteración 2

Las historias de usuario involucradas en esta iteración son las especificadas en la tabla 11.

Título	Página de Inicio
Duración	24 horas
Descripción	Se muestra el primer paso para configurar un clúster Live Hadoop donde se debe decidir si disponer o no el sistema operativo en la red.
Título	Página encendido de máquinas
Duración	24 horas
Descripción	Se debe decidir si encender las máquinas de forma manual o remota.
Título	Página encendido manual
Duración	24 horas

Descripción	Si se decidió encender las maquinas manualmente entonces debemos asegurarnos de que estén encendidas las máquinas que deseamos integrar al clúster.
Título	Página encendido remoto
Duración	24 horas
Descripción	Si se decidió encender las maquinas remotamente entonces ahora se debe proporcionar las direcciones MAC de las máquinas que queremos encender.
Título	Página seleccionar IP
Duración	24 horas
Descripción	Se debe seleccionar las direcciones IP de las máquinas que se quiere que pertenezcan al clúster.
Título	Página fin
Duración	24 horas
Descripción	Nos da un mensaje si el clúster se inicio correctamente.
Título	Recolección de material de apoyo
Duración	10 horas
Descripción	Se recolecta todo el material posible para despejar dudas y seguir con el

	proceso de desarrollo.
--	------------------------

Tabla 11: Historias de Usuario - Iteración 2

Diseño

En esta etapa se realiza el prototipo de la interfaz del módulo, que debe presentar todas las funcionalidades antes descritas. (Ver ilustraciones 23, 24,25)



Ilustración 23: Diseño Módulo Configurar Paso1



Ilustración 24: Diseño Módulo Configurar Seleccionar IP

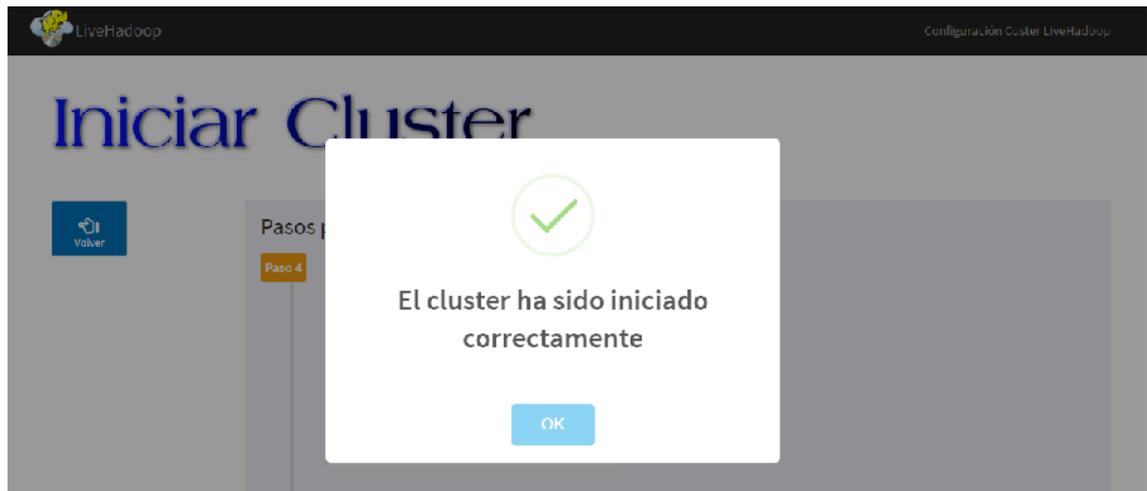


Ilustración 25: Diseño Módulo Configurar Pagina Fin

Codificación

En esta fase se crean los archivos.html para generar cada una de las funcionalidades. Cada uno de los pasos existentes corresponde a una funcionalidad que el usuario puede realizar incluyendo un último paso de confirmación de la solicitud.

En las siguientes ilustraciones se puede apreciar una porción de código que muestra la implementación de las funciones mencionadas anteriormente.

```

1. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
2. ["http://www.w3.org/TR/html4/strict.dtd">
3. <html>
4. <head>
5.   <meta content="text/html; charset=ISO-8859-1"
6.   http-equiv="content-type">
7.   <title>ejemplo1</title>
8. </head>
9. <body>
10. Página de ejemplo<br>
11. <br>
12. <br>
15. <br>
16. <table style="text-align: left; width: 478px; height: 60px;"
17. border="1" cellpadding="2" cellspacing="2">
18.   <tbody>
19.     <tr>
20.       <td>Filal</td>
21.     <td></td>
22.   </tr>
23.   <tr>
24.     <td></td>
25.     <td></td>
26.   </tr>
27. </tbody>

```

Ilustración 26: Extracto de código Módulo Configurar Paso1

```

<!DOCTYPE html>
<html>
  <head>
    <title>YouTube HTML5 Demo</title>
    <link rel="stylesheet" href="/demo/demo.css" type="text/css">
    <script type="text/javascript" src="/demo/demo.js"></script>
  </head>
  <body>
    <header id="global-header">
      <h1 id="global-title">HTML5 Demo</h1>
      <a href="/"></a>
    </header>
    <nav id="global-nav">
      <a href="/" class="nav-button">Home</a>
      <a href="/my_subscriptions?masthead=1" class="nav-button">Subscriptions</a>
      <a href="/browse" class="nav-button">Videos</a>
      <a href="/shows" class="nav-button">Shows</a>
      <a href="/members" class="nav-button">Channels</a>
      <form id="search-form">
        <input id="search-textbox" type="search" name="query">
        <button id="search-button" type="submit" class="blue-button">Search</button>
      </form>
      <button id="upload-button" class="yellow-button" type="button">Upload</button>
    </nav>
    <article id="watch-page">
      <h1>Old Beach Demo</h1>
      <section id="main-content">
        <div id="video-player">
          <video width="640" height="360" src="/demo/google_main.mp4?2" autobuffer>
            <div class="video-fallback">
              <br>You must have an HTML5 capable browser.
            </div>
          </video>
          <div class="controls">

```

Ilustración 27: Extracto de código Módulo Configurar Seleccionar IP

```

view plain copy to clipboard print ?
$(document).ready(function()
{
$.get('myData.xml', function(d){
$('body').append('<h1> Recommended Web Development Books </h1>');
$('body').append('<dl>');

$(d).find('book').each(function(){

var $book = $(this);
var title = $book.attr("title");
var description = $book.find('description').text();
var imageUrl = $book.attr('imageUrl');

var html = '<dt>  </dt>';
html += '<dd> <span class="loadingPic" alt="Loading" />';
html += '<p class="title">' + title + '</p>';
html += '<p> ' + description + '</p>';
html += '</dd>';

$('dl').append$(html));

$('.loadingPic').fadeOut(1400);
});
});
});
</dl>

```

Ilustración 28: Extracto de código Módulo Configurar archivo js

Pruebas

Para comprobar el cumplimiento de los lineamientos establecidos en esta iteración, se realizan pruebas funcionales aplicadas de forma manual verificando el correcto funcionamiento de las interfaces que posee esta sección.

4.4 Iteración 3: Módulo Administrar y Monitorear

En esta iteración se desarrollan las interfaces del módulo que nos dará acceso a la administración del clúster Live Hadoop, permitiéndonos tanto ver el estado en el que se encuentra el clúster como detenerlo totalmente.

Planificación

En esta fase se especifica con detalle el plan de la iteración, se establecen las tareas a realizar para desarrollar las funcionalidades que permiten satisfacer los requisitos del cliente.

Iteración 2	
Descripción	Desarrollo del módulo Administrar y Monitorear Clúster Live Hadoop.
Duración Iteración	de 15 días
Historia	Duración
Página inicio	24 horas
Página general	24 horas
Página nodo	24 horas
Página registrar	24 horas
Recolección de material de apoyo.	10 horas

Tabla 12: Planificación de la Iteración 3

Las historias de usuario involucradas en esta iteración son las especificadas en la tabla 13.

Título	Página de Inicio
--------	------------------

Duración	24 horas
Descripción	Se muestra un formulario donde se debe ingresar el usuario y la contraseña para ingresar al sistema.
Título	Página general
Duración	24 horas
Descripción	Se muestran las distintas especificaciones del clúster completo.
Título	Página nodo
Duración	24 horas
Descripción	Se muestran las distintas especificaciones de cada nodo.
Título	Página registrar
Duración	24 horas
Descripción	Se muestra un formulario para crear un nuevo usuario.
Título	Recolección de material de apoyo
Duración	10 horas
Descripción	Se recolecta todo el material posible para despejar dudas y seguir con el proceso de desarrollo.

Tabla 13: Historias de Usuario – Iteración 3

Diseño

En esta etapa se realiza el prototipo de la interfaz del módulo, que debe presentar todas las funcionalidades antes descritas. (Ver ilustraciones 29, 30, 31)

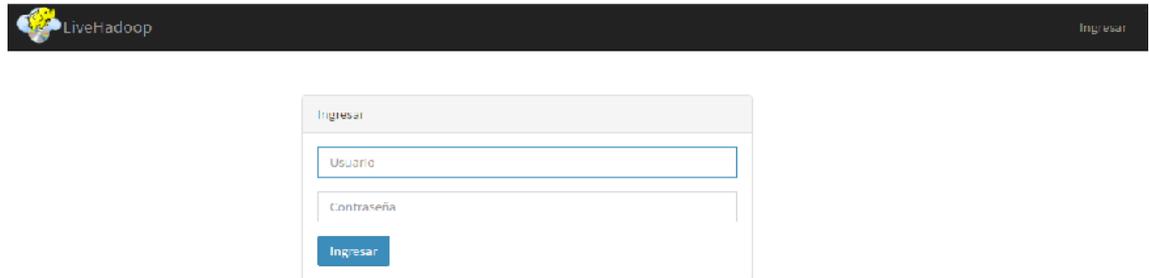


Ilustración 29: Diseño Módulo Administrar y Monitorear – Ingresar

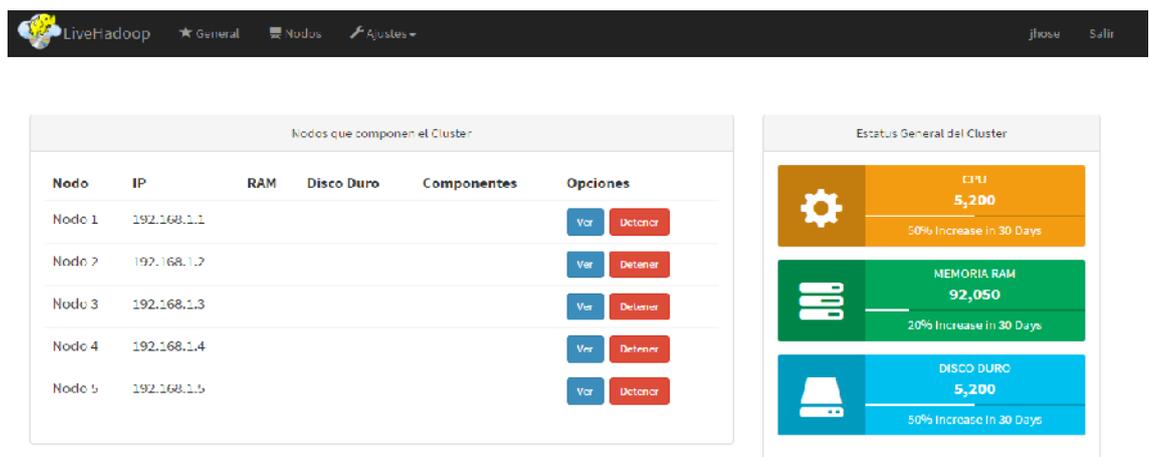


Ilustración 30: Diseño Módulo Administrar y Monitorear - General

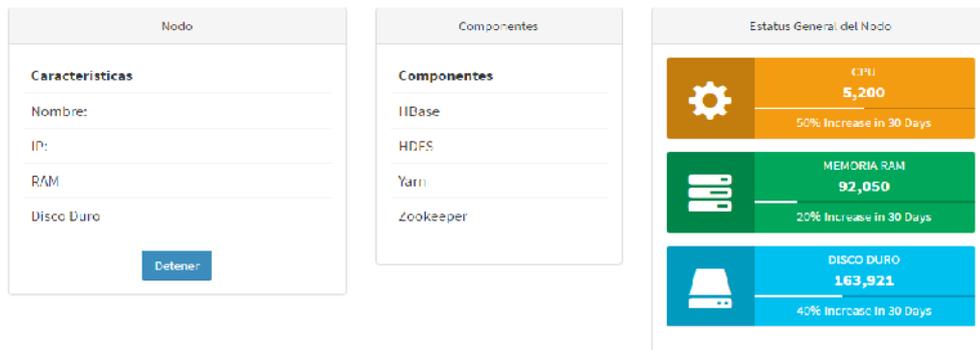


Ilustración 31: Diseño Módulo Administrar y Monitorear - Nodo

Codificación

En esta fase se crean los archivos.php para generar cada una de las funcionalidades. Cada uno de los interfaces existentes corresponde a una funcionalidad que el usuario puede realizar.

En las siguientes ilustraciones se puede apreciar una porción de código que muestra la implementación de las funciones mencionadas anteriormente.

```
<plugin name="introimage" url="viveros-iris-virtual-tourdata/graphics/introImage.png" align="center" onclick="
<action name="hideintroimage">
  if(plugin[introimage].enabled, set(plugin[introimage].enabled,false);tween(plugin[introimage].alpha, 0,0,
</action>
<action name="autohideintroimage">
  if(plugin[data].introImageHasBeenPlayed==false, set(plugin[introimage].alpha,0); tween(plugin[introimage].alpha,0,0,
  delayedcall(10, hideintroimage() );
  set(plugin[data].introImageHasBeenPlayed,true));
</action>
<!--***** Logos *****-->
<plugin name="logo4"  onhover="showtext(Website);" url="viveros-iris-virtual-tourdata/graphics/logo4.png" keep="true"
<plugin name="logo6" url="viveros-iris-virtual-tourdata/graphics/logo6.png" keep="true" align="leftbottom" x="0"
<plugin name="logo1"  onhover="showtext(Website);" url="viveros-iris-virtual-tourdata/graphics/logo1.png" keep="true"
<!--***** Lensflares *****-->
<lensflareset name="DEFAULT" url="viveros-iris-virtual-tourdata/graphics/flares.jpg"/>
<!--***** Menu *****-->
<plugin name="in" align="bottom" height="prop" keep="true" ondown="set(fov_moveforce,-1);" onhover="showtext(
<plugin name="out" align="bottom" height="prop" keep="true" ondown="set(fov_moveforce,+1);" onhover="showtext(
<plugin name="left" align="bottom" height="prop" keep="true" ondown="set(hlookat_moveforce,-1);" onhover="show
<plugin name="right" align="bottom" height="prop" keep="true" ondown="set(hlookat_moveforce,+1);" onhover="show
<plugin name="up" align="bottom" height="prop" keep="true" ondown="set(vlookat_moveforce,-1);" onhover="showte
<plugin name="down" align="bottom" height="prop" keep="true" ondown="set(vlookat_moveforce,+1);" onhover="show
<plugin name="home" align="bottom" height="prop" keep="true" onclick="lookto(get(panoview.h),get(panoview.v),g
<plugin name="openfs" align="bottom" blendmode="normal" customColor="true" height="prop" keep="true" onclick="
<plugin name="closefs" align="bottom" blendmode="normal" customColor="true" height="prop" keep="true" onclick="

<plugin name="info" align="bottom" customColor="true" height="prop" keep="true" onclick="displayHelp();" onh
<plugin name="helpScreen" align="center" customColor="true" keep="true" onclick="closeHelp();" url="viveros-ir
<action name="displayHelp">
  set(plugin[helpScreen].visible,true);
  set(plugin[info].onclick,closeHelp());
</action>
<action name="closeHelp">
  set(plugin[helpScreen].visible,false);
  set(plugin[info].onclick,displayHelp());
</action>
```

Ilustración 32: Extracto de código Módulo Administrar y Monitorear - Ingresar

```

<!DOCTYPE html>
<html lang="en-US" class="y-fp-bg y-fp-pg-grad ua-ff ua-win ua-ff3_6 bkt701">
<!-- m2 template 0 -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Yahoo!</title>
<meta http-equiv="X-UA-Compatible" content="chrome=1">
<meta name="description" content="Welcome to Yahoo!, the world's most visited home page. Quickly find what you're sea<
<meta name="keywords" content="yahoo, yahoo home page, yahoo homepage, yahoo search, yahoo mail, yahoo messenger, yah

<script type="text/javascript">
//Roundtrip
rtTop = Number(new Date());

document.documentElement.className += ' jsenabled!';
</script>
<script type="text/javascript">
(function () {
//refresh check
var d=document,c=""; "+d.cookie + ";",msc=c.indexOf("; MSC="),re=false,s,r,l="","n,n='0',b,lck='',fps=c.indexOf("FI

//reload for lite page if screen resolution < 1024
fpsVal = fps === -1 ? 'dl' : d.cookie.substring(fps+2, fps+4);
if (fpsVal === 'dl' && sw < 1024) {
fpsVal = 'ds';
re=true;
} else if (fpsVal === 'ds' && sw >=1024) {
fpsVal = 'dl';
re=true;
}
d.cookie = "FPS=" + fpsVal + "; expires=Sun, 01 Jan 2012 00:00:00 GMT; domain=www.yahoo.com"; //check cook:
if ((b=c.indexOf(' Y=v'))>=0) {
c=c.substring(b,c.indexOf(';')+1)+'&';
if ((b=c.indexOf('l='))>=0) {
l=c.substring(b+2,c.indexOf('&'));
if ((b=c.indexOf('n='))>=0)n=c.substring(b+2,c.indexOf('&'));
}
}

//if login info isn't the same...

```

Ilustración 33: Extracto de código Módulo Administrar y Monitorear – General

```

$(window).bind("load", function() {
// IMAGE RESIZE
$('#product_cat_list img').each(function() {
var maxWidth = 120;
var maxHeight = 120;
var ratio = 0;
var width = $(this).width();
var height = $(this).height();

if(width > maxWidth){
ratio = maxWidth / width;
$(this).css("width", maxWidth);
$(this).css("height", height * ratio);
height = height * ratio;
}
var width = $(this).width();
var height = $(this).height();
if(height > maxHeight){
ratio = maxHeight / height;
$(this).css("height", maxHeight);
$(this).css("width", width * ratio);
width = width * ratio;
}
});
//$("#contentpage img").show();
// IMAGE RESIZE
});

```

Ilustración 34: Extracto de código Módulo Administrar y Monitorear archivo js

Pruebas

Para comprobar el cumplimiento de los lineamientos establecidos en esta iteración, se realizan pruebas funcionales para verificar que el sistema esté accesible a los usuarios autorizados.

También se realizan pruebas funcionales para asegurar que las solicitudes de registro se estén cargando de forma correcta en la base de datos.

4.5 Iteración 4: Pruebas

Esta iteración se enfoca en la ejecución de diversos casos de prueba, que permiten determinar si el software desarrollado es parcial o totalmente satisfactorio para el cliente, por ello se realizaron pruebas de usabilidad. Dichas pruebas consistieron en la creación de un cuestionario, donde se empleó la escala de Likert donde el usuario de la aplicación muestra su nivel de acuerdo o desacuerdo en relación a un planteamiento de carácter positivo o negativo, utilizando cinco opciones de respuesta para cada pregunta: Totalmente de acuerdo, de acuerdo, ni de acuerdo ni en desacuerdo, en desacuerdo y totalmente en desacuerdo. El cuestionario se realizó de forma escrita y se realizó a 15 personas que pertenecen a la comunidad de estudiantes de la Universidad central de Venezuela

Estas pruebas contemplan un grupo de criterios basados en las Historias de Usuario permiten evidenciar el grado de satisfacción del cliente con la aplicación desarrollada.

Resultados de las Pruebas de Usabilidad

Del cuestionario planteado anteriormente se obtuvieron los resultados:

Como se puede apreciar en el ilustración 35, en la primera pregunta el 100% de los usuarios estuvo totalmente de acuerdo y consideran que la aplicación dispone de una navegación fácil de entender.

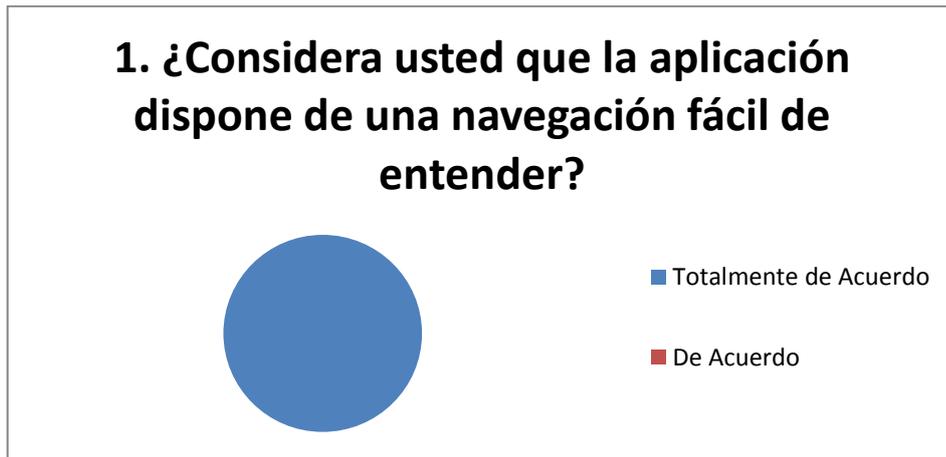


Ilustración 35: Resultados Item 1

En la segunda pregunta, como se refleja en la ilustración 36, el 40% de los usuarios opinó que estaba totalmente de acuerdo con que las funcionalidades de la aplicación son fáciles de reconocer, mientras que el 60% restante indicó que estaba de acuerdo.

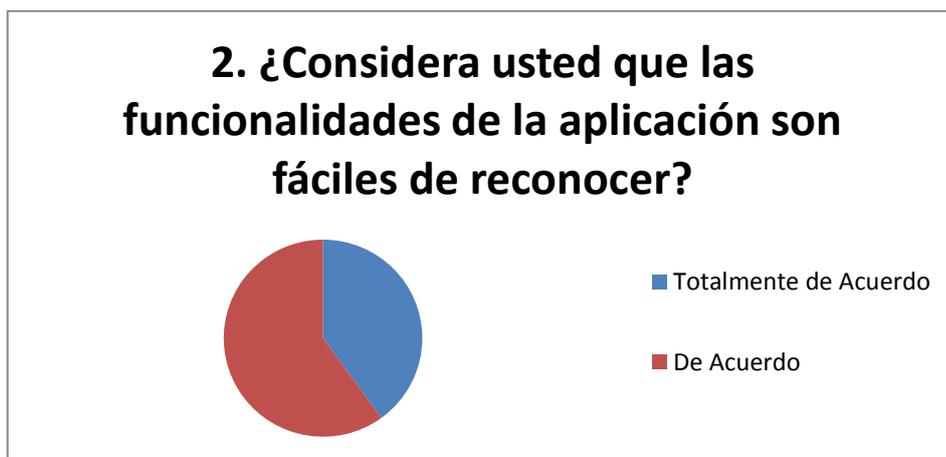


Ilustración 36 : Resultados Item 2

En la tercera pregunta, el 100% de los usuarios estuvo de acuerdo con que el diseño de la aplicación favorece a su uso. Estos resultados se ven reflejados en la figura 55.



Ilustración 37: Resultados Item 3

En la cuarta pregunta, como se refleja en la ilustración 38, el 80% de los usuarios indicó que estaba completamente de acuerdo con que la aplicación posee una interfaz simple sencilla y organizada, mientras que el 20% restante opinó que estaba de acuerdo.

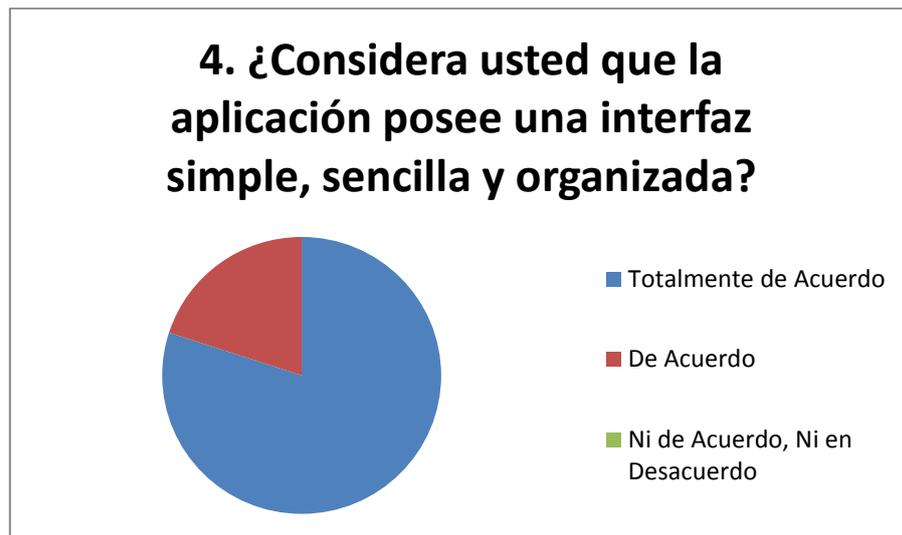


Ilustración 38: Resultados Item 4

En la quinta pregunta, como se refleja en la ilustración 39, un 20% de los usuarios indicó que estaba completamente de acuerdo con que la aplicación está

diseñada para minimizar las posibilidades de cometer errores, otro 60% indicó que estaba de acuerdo y el 20% restante señaló que no estaba ni de acuerdo ni en desacuerdo.

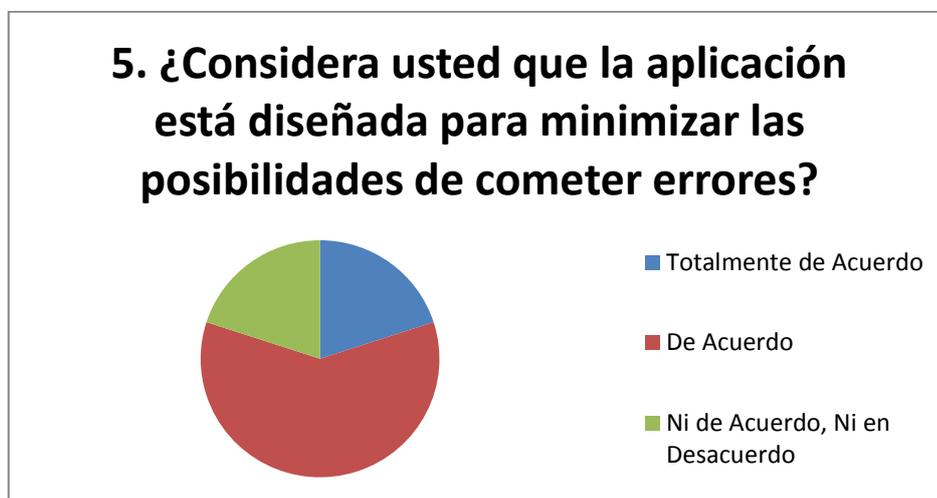


Ilustración 39: Resultados Item 5

En la sexta pregunta, el 60% de los usuarios indicó que estaba completamente de acuerdo con que el sistema usa utiliza un lenguaje consistente y apropiado, mientras que el 40% restante estuvo de acuerdo. Estos resultados se reflejan en la ilustración 40.



Ilustración 40: Resultados Item 6

En la séptima pregunta, como se refleja en la figura 59, el 40% de los usuarios estuvo completamente de acuerdo con que es un sistema dispone de

elementos de ayuda útiles, otro 40% está de acuerdo y el otro 20% indicó que no estaba de acuerdo ni de acuerdo.

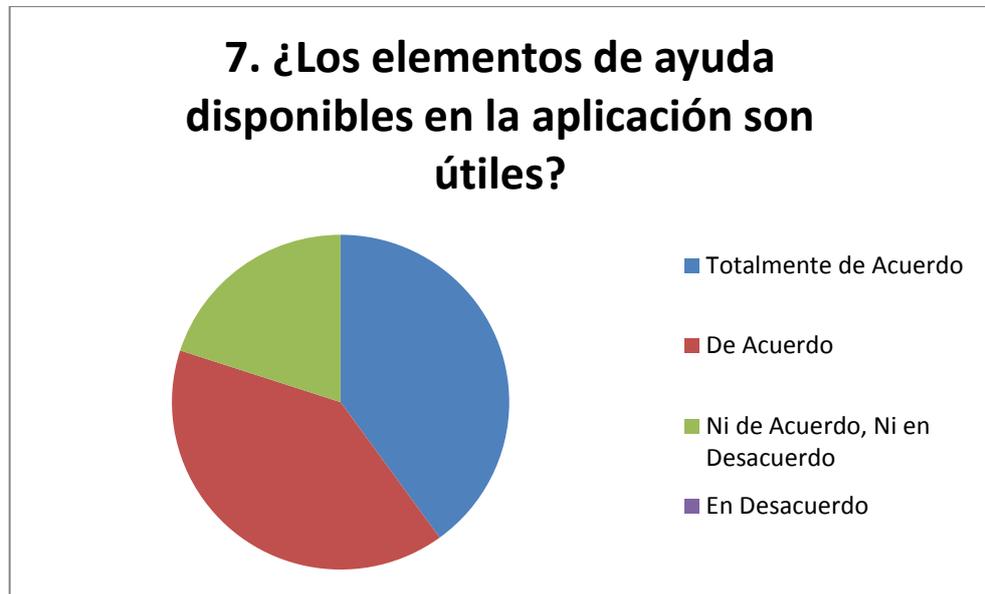


Ilustración 41: Resultados Item 7

En la octava y última pregunta, el 60% de los usuarios estuvo completamente de acuerdo con que las pruebas funcionales de la aplicación se ejecutaron de manera satisfactoria, mientras que el 40% restante indicó que estaba de acuerdo. Dichos resultados se aprecian en la ilustración 42.



Ilustración 42: Resultados Item 8

Finalmente, los resultados fueron exitosos, debido a que la aplicación funcionó correctamente en todos los aspectos.

Capítulo V: Resultados

En este capítulo se presentan los resultados de la aplicación Web desarrollada para llevar a cabo la configuración, administración y monitoreo de un clúster Live Hadoop. La aplicación va dirigida a toda la comunidad de estudiantes y docentes de la Universidad Central de Venezuela y a todo aquel interesado en aprender sobre este tema.

5.1 Descargar Sistema Operativo Live Hadoop

Este módulo nos da acceso a descargar un archivo iso o tar.gz con el sistema operativo Live Hadoop

5.1.1 Inicio

La ilustración 43 muestra la página de inicio de este módulo, la cual nos da una breve explicación de lo que es el Sistema Operativo Live Hadoop y para que podemos usarlo. Cuenta con un cabecera y el logo de Live Hadoop.



El objetivo fundamental del proyecto Live Hadoop es la creación de una distribución de Hadoop en el formato LiveCD, que permite levantar una infraestructura de pruebas para Big Data de forma AdHoc o temporal, sobre un conjunto de equipos en red, que no necesariamente estén pre-configurados para ello. Esta tecnología permite implementar un clúster Hadoop sobre un conjunto de máquinas que pudieran tener otro uso o fin, como por ejemplo una sala de formación, salas de navegación o laboratorios.

Descargar



Ilustración 43: Pagina inicio - Módulo Descargar

5.1.2 Descargas

Presenta perfiles predeterminados con distintos módulos que van a componer el sistema operativo, tal como se muestra en la ilustración 44.

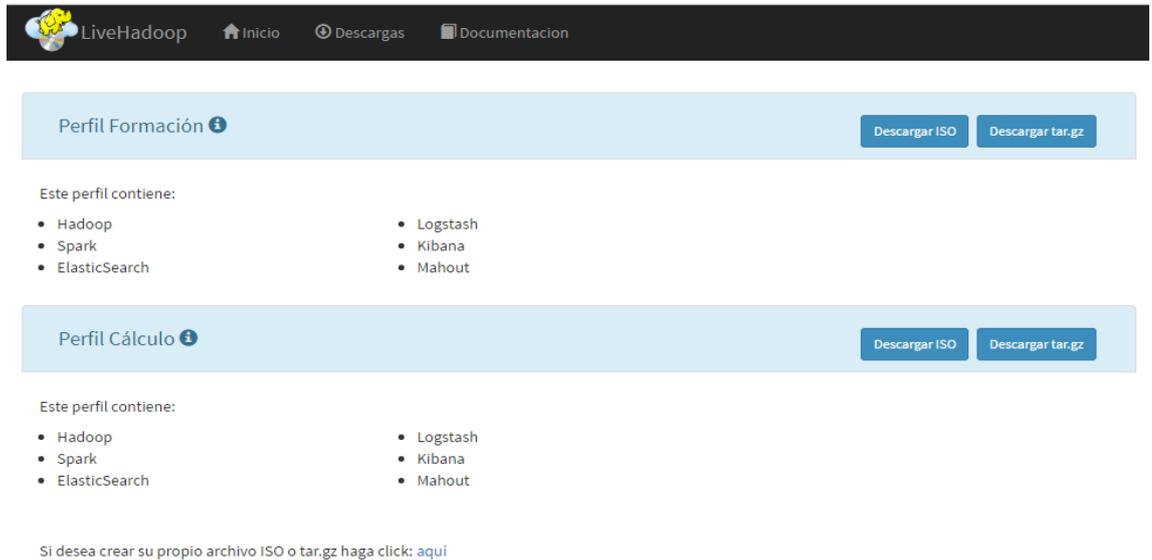


Ilustración 44: Pagina de descargas - Módulo Descargar

5.1.3 Crear

Presenta los módulos disponibles para crear el sistema operativo con los que decidamos seleccionar.

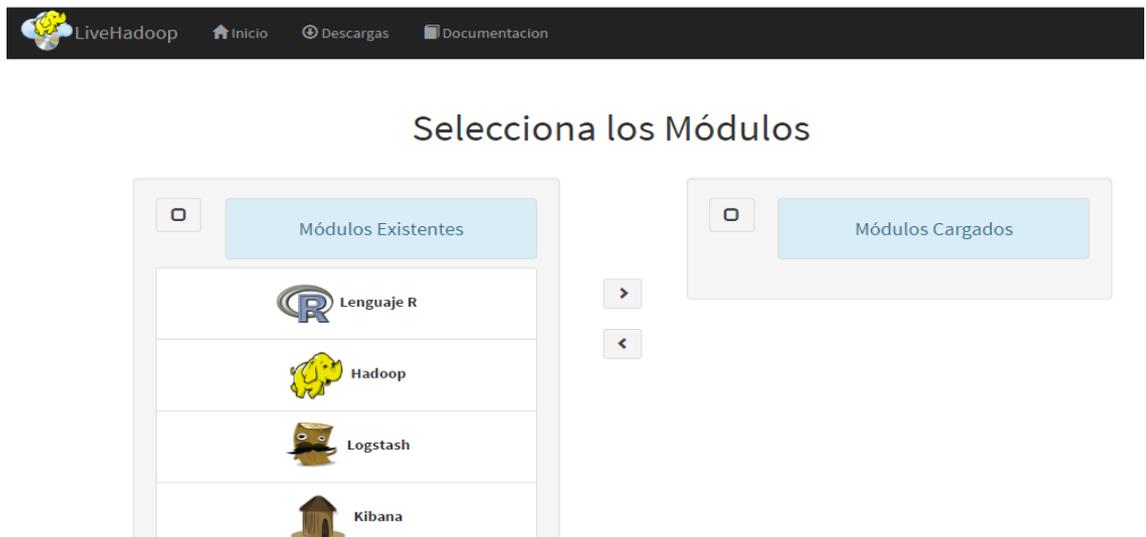


Ilustración 45: Pagina crear - Módulo Descargar

5.2 Configurar Clúster Live Hadoop

Este módulo nos permite configurar e iniciar un clúster Live Hadoop.

5.2.1 Inicio

La ilustración 46 muestra la página de inicio de este módulo, la cual nos muestra el primer paso para iniciar un clúster, aquí debemos decidir si queremos o no disponer el sistema operativo en la red.



Ilustración 46: Pagina inicio - Módulo Configurar

5.2.2 Encendido

En esta página debemos escoger si encender las máquinas de forma manual o remota.



Ilustración 47: Pagina encendido - Módulo Configurar

5.2.3 Remoto

Si se escogió encender las máquinas remotamente entonces la pagina es la mostrada en la figura 48 donde debemos ingresar las direcciones MAC de las máquinas ya sea por medio de un archivo o ingresando una por una.

Iniciar Cluster



Pasos para Iniciar un Cluster

Paso 3



Ingresar Direcciones MAC

En este tercer paso se debe ingresar cada una de las direcciones MAC de las máquinas que se desea encender remotamente, puede ser en un archivo o se puede ingresar cada una manualmente.

No se eligió archivo

Ilustración 48: Pagina remoto - Módulo Configurar

5.2.4 Manual

Si se escogió encender las máquinas manualmente entonces la página mostrará un mensaje donde recuerda que se deben tener encendidas las máquinas.

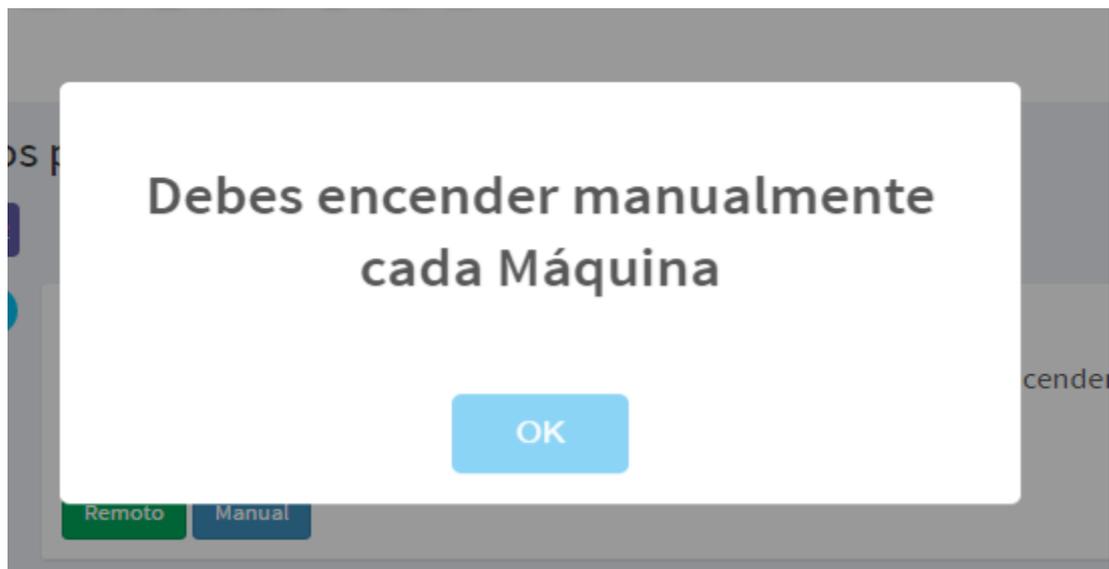


Ilustración 49: Página manual - Módulo Configurar

5.2.5 Seleccionar IP

En esta página se muestran las direcciones IP de las máquinas que están encendidas y debemos escoger las que queremos que pertenezcan al clúster.



Ilustración 50: Página seleccionar IP - Módulo Configurar

5.2.6 Fin

Como se puede observar en la ilustración 51 aquí se presenta la lista de las direcciones IP que escogimos anteriormente y al presionar el botón de “finalizar” iniciamos el clúster.



Ilustración 51: Página final - Módulo Configurar

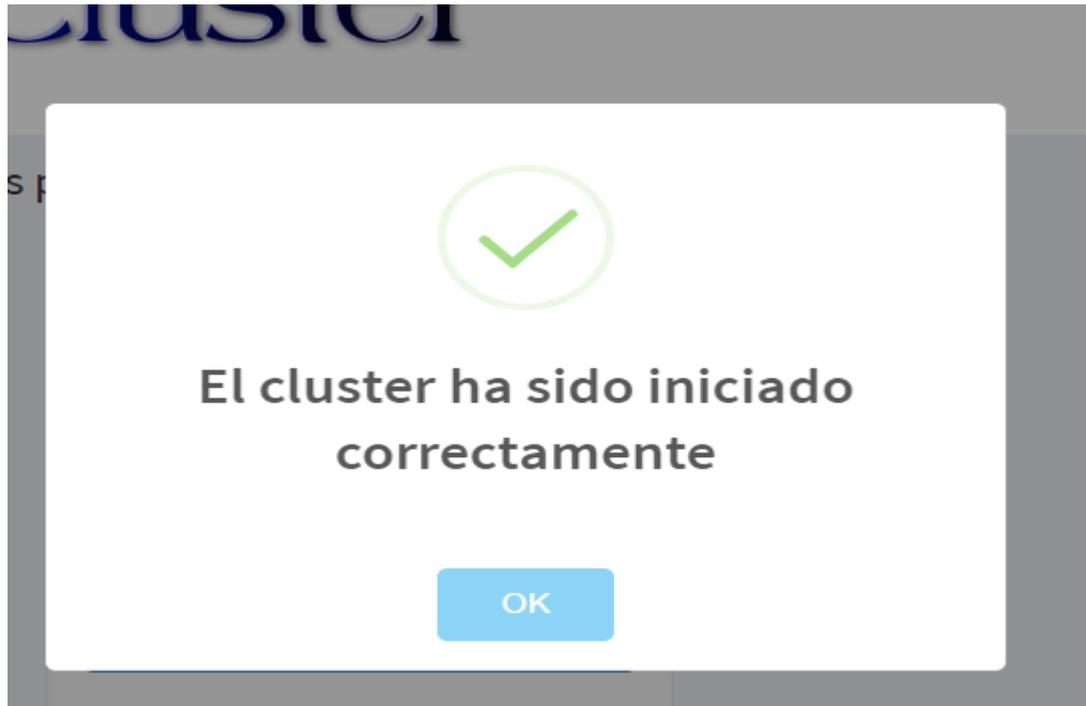


Ilustración 52: mensaje final - Módulo Configurar

5.3 Administrar y Monitorear Clúster Live Hadoop

Este módulo nos permite administrar y monitorear los aspectos generales de un clúster Live Hadoop.

5.3.1 General

Presenta el estado general del clúster, estado del CPU, memoria RAM y disco duro así como los nodos que lo componen.

Nodo	IP	RAM	Disco Duro	Componentes	Opciones
Nodo 1	192.168.1.1				Ver Detener
Nodo 2	192.168.1.2				Ver Detener
Nodo 3	192.168.1.3				Ver Detener
Nodo 4	192.168.1.4				Ver Detener
Nodo 5	192.168.1.5				Ver Detener

Ilustración 53: Página general- Módulo Administrar y Monitorear

5.3.2 Nodos

Presenta es estado del nodo, estado del CPU, memoria RAM y disco duro.

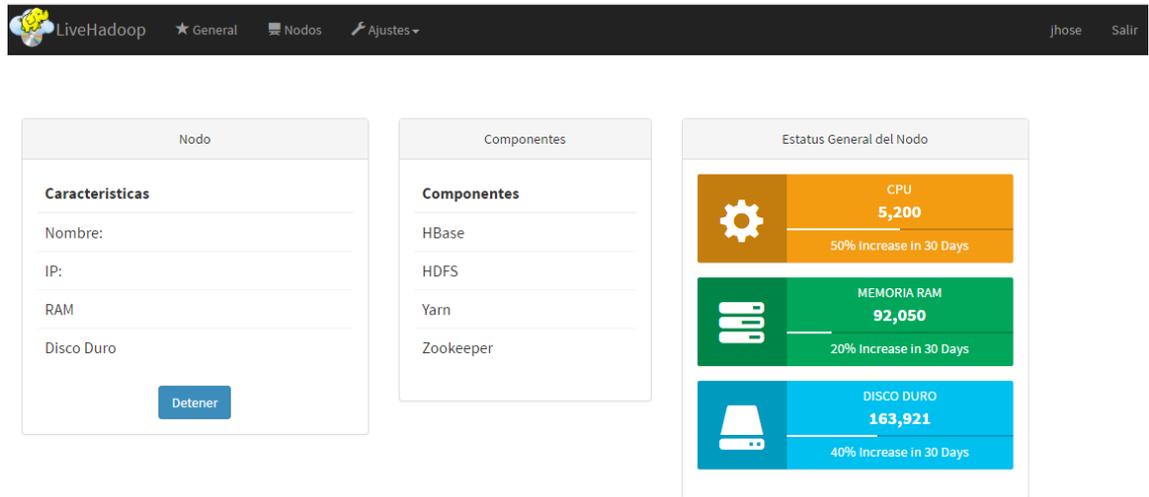


Ilustración 54: Nodos - Módulo Administrar y Monitorear

Una vez presentado los módulos de Descarga, Configuración, Administración y Monitoreo de un Clúster Live Hadoop y todas sus funcionalidades, como cierre del TEG, a continuación se presentan las conclusiones.

Conclusiones

La aplicación desarrollada provee un portal web eficiente para el proceso de configuración, administración y monitoreo de un clúster Live Hadoop. Este proceso llevado manualmente mediante scripts generaba un alto costo en tiempo además de ser un trabajo tedioso, razón por la cual se llevo un análisis de la situación actual que permitiera comprender la problemática existente para así determinar las bases teóricas y prácticas a seguir en el desarrollo de una solución ante la necesidad evidenciada. Esta solución se logro al desarrollar aplicaciones web para automatizar dichos procesos, cumpliendo así con el objetivo principal de esta investigación.

Para ello, se decidió configurar el método de desarrollo XP, principalmente, por su grado de detalle en la fase de planificación, debido a que propone como artefacto las Historias de Usuario, que permiten recopilar de forma concisa los requerimientos del usuario; hecho que facilitó la planificación inicial y por el cual se decidió agregar tablas descriptivas que indican las Historias de Usuario a considerar y el esfuerzo estimado para dar respuesta a cada una de ellas. De esta manera, se pudo dividir estratégicamente el desarrollo de la aplicación en iteraciones de acuerdo a los grupos de funcionalidades identificados en la planificación inicial. Cada iteración generó como salida un entregable o producto de software funcional desarrollado aplicando las fases de planificación, diseño, codificación y pruebas; a excepción de la última iteración debido a que en estas sólo se preparó el entorno de desarrollo y se verificó el correcto comportamiento de las aplicaciones, respectivamente.

Una vez completado el desarrollo de la aplicación, se realizaron las pruebas funcionales y de aceptación que demostraron el cumplimiento de los objetivos planteados y la satisfacción del usuario con el software entregado que, a través del uso de las tecnologías de información, fue puesto en producción, convirtiendo éste Trabajo Especial de Grado en una solución a un problema de la vida real.

Recomendaciones

Con la finalidad de mejorar los procesos para la configuración, administración y monitoreo de un clúster Live Hadoop, se propone tomar en cuenta ciertas consideraciones que de alguna u otra manera contribuirán al éxito del sistema. En tal sentido, se proponen las siguientes recomendaciones:

1. Realizar la debida promoción del sistema, para la participación total del personal involucrado en el proceso.
2. Examinar periódicamente el sistema, para evaluar las posibilidades de adaptación a nuevos requerimientos que surjan.
3. Realizar el mantenimiento preventivo, adaptativo y correctivo para garantizar la calidad y vanguardia del software desarrollado.
4. Crear y adoptar de medidas de seguridad que resguarden al clúster LiveHadoop
5. Migrar el software a un lenguaje que posea un framework ligero.

Bibliografía

1. Bracho J., Narvaez I., 2015. Diseño y desarrollo de interfaces para la configuración, administración y monitoreo del clúster Live-Hadoop (seminario).
2. (20/06/2016,3:00pm)
<http://www.pcmag.com/encyclopedia/term/39803/client-server-architecture>
3. (25/06/2016, 5:00pm)
https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web
4. Mateu, C. (2004). *Desarrollo de Aplicaciones Web*. UOC.
5. (25/06/2016, 7:00pm) <http://www.pixima.net/>
6. (28/06/2016, 4:00pm) <http://www.techopedia.com>
7. Mora, S. L. (2001). *Programación en Internet: Clientes Web*. España: Club Universitario.
8. (10/07/2016, 8:00am) <https://es.wikipedia.org/wiki/HTML>
9. (10/07/2016, 1:00pm) <http://definicion.de/html/>
10. (11/07/2016,8:00am)
https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada
11. (11/07/2016, 8:30am) http://www.htmlpoint.com/css/css_03.htm
12. (12/07/2016, 8:30am) [http://www.techopedia.com/.](http://www.techopedia.com/)
13. (15/07/2016, 10:30am) <http://librosweb.es/libro/ajax/>
14. (18/07/2016,10:00pm)<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
15. (21/07/2016, 11:00pm) <http://jquery.com/>
16. (21/07/2016, 11:30pm) <http://jqueryui.com/>
17. (29/07/2016, 7:30am) <http://www.ibrugor.com/blog/que-es-php-para-que-sirve/>
18. (29/07/2016, 11:30am) <http://www.mclibre.org/consultar/php/lecciones/>
19. (05/08/2016, 10:00am) <http://www.mysql.com/about/>
20. (07/08/2016, 1:30pm) <https://es.wikipedia.org/wiki/Git>
21. (19/08/2016, 4:00pm) <http://www.agilemanifesto.org/iso/es/principles.html>
22. (23/06/2016, 5:00pm)
<http://blogs.unellez.edu.ve/dsilva/files/2014/07/Metodologia-XP.pdf>
23. (20/06/2016, 2:00pm)
https://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema

Anexos

Cuestionario de Usabilidad

El presente cuestionario tiene como finalidad conocer su opinión con respecto a la usabilidad de la Aplicación Web teniendo en cuenta la efectividad, eficiencia y satisfacción con la que la aplicación puede ser comprendida, utilizada y ser atractiva ante el usuario. Se agradece seleccionar la puntuación que corresponda a su opinión de acuerdo a la siguiente escala:

Totalmente De acuerdo	De acuerdo	Ni de acuerdo ni Desacuerdo	En Desacuerdo	Totalmente en Desacuerdo	
5	4	3	2	1	
Pregunta	1	2	3	4	5
<i>¿Considera usted que la aplicación dispone de una navegación fácil de entender?</i>					
<i>¿Considera usted que las funcionalidades de la aplicación son fáciles de reconocer?</i>					
<i>¿El diseño de la aplicación favorece su uso?</i>					
<i>¿Considera usted que la aplicación posee una interfaz simple, sencilla y organizada?</i>					
<i>¿Considera usted que la aplicación está diseñada para minimizar las posibilidades de cometer errores?</i>					
<i>¿Considera usted que la aplicación utiliza un lenguaje consistente y apropiado?</i>					
<i>¿Los elementos de ayuda disponibles en la aplicación son útiles?</i>					
<i>En las pruebas funcionales realizadas ¿Todo se ejecuta según lo esperado?</i>					