



Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación  
Centro de Computación Distribuida y Paralela

**Desarrollo de una interfaz gráfica en R para la ejecución  
de métodos de minería de datos sobre  
plataformas Hadoop para Big Data.**

Trabajo Especial de Grado presentado ante la ilustre  
Universidad Central de Venezuela por el  
Br. Pascual Madrid

Tutor:  
Prof. Jesús Lares  
Prof. José Sosa

Caracas, Octubre de 2015

## ACTA

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por el Bachiller Pascual Madrid C.I.: 20.604.893, con el título: “Desarrollo de una interfaz gráfica en R para la ejecución de métodos de minería de datos sobre plataformas Hadoop para Big Data”, a los fines de cumplir con el requisito legal para 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 27 de octubre de 2015, a las 09:00am, para que su autor lo defendiera en forma pública, en la Sala III Postgrado de la Escuela de Computación, Facultad de Ciencias de la Universidad Central de Venezuela, lo cual realizó mediante una exposición oral de su contenido, y luego respondió a las preguntas que le fueron 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 aprobarlo con la nota de \_\_\_\_\_ puntos.

En fe de lo cual se levanta la presente acta, en Caracas el 27 de octubre de 2015.

---

Prof. José Sosa  
Tutor

---

Prof. Jesús Lares  
Tutor

---

Prof. Héctor Navarro  
Jurado

---

Prof. Fernando Crema  
Jurado

# Resumen

El análisis de grandes volúmenes de datos representa un gran reto para los científicos de datos, ya sea desde un punto de vista intelectual y uno de recursos. No es sencillo realizar análisis en plataformas de Big Data debido a que los scripts deben seguir un paradigma de programación llamado MapReduce el cual resulta todo un reto hasta para las personas con mucha experiencia en la programación sin contar lo costoso que es implementar toda una infraestructura que de soporte a la cantidad masiva de datos.

La intención de este trabajo de grado es la realización de una aplicación que provea una interfaz gráfica para la ejecución de métodos de minería de datos sobre una plataforma Hadoop de una manera remota sin tener que implementar métodos MapReduce ni tener que preparar una infraestructura Hadoop, sólo utilizar una ya preparada previamente.

La aplicación fue programada utilizando el lenguaje de programación estadístico R utilizando una gran gama de paquetes para el desarrollo de la interfaz y de los cálculos.

La comunicación con la plataforma Hadoop se hace mediante el protocolo SSH (Secure Shell) permitiendo un tráfico de información de manera segura en todo momento.

Se realizaron pruebas sencillas que englobaron todas las funcionalidades de la aplicación.

Este trabajo dejó como fruto final una interfaz gráfica programada en R capaz de ejecutar métodos de minería de datos de manera local y remota sobre un clúster Hadoop y también la posibilidad de ejecutar funciones Map y Reduce en un clúster Hadoop utilizando la funcionalidad llamada Hadoop Streaming.

# Agradecimientos

Agradezco principalmente a Dios por permitirme hoy estar cumpliendo un sueño y por no dejarme sólo en ningún momento.

A mi madre por todo el apoyo humano tan particular que me ha dado en toda mi vida.

A mis hermanas Lucila Madrid, Carmen Cañas y a mí cuñado Xavier Capelo por todo el apoyo que me han dado durante toda la carrera.

A mis tíos Jaime Madrid, Hernán Valera y Antonio Valera, por sus orientaciones y todas las experiencias que han compartido conmigo toda mi vida.

A Laura González por brindarme todo su apoyo y cariño desde siempre.

A Vicente Ordoñez, Juan Morantes y Gustavo Morantes por todo el apoyo que me han brindado y por compartir conmigo sus conocimientos y experiencias en el ámbito académico y laboral.

A Emily Corro, Daniel Hernández, Rafael Machado, Álvaro Marciales, Eduardo Sánchez, Carlos Pereira y Claudio Torrez, por la amistad y hermandad que me han brindado por años. Son pilares y símbolos de motivación en muchos sentidos de mi vida.

A todas aquellas personas que de una u otra manera me ayudaron durante el desarrollo de este Trabajo Especial de Grado.

A mis tutores Jesús Lares y José Sosa por el apoyo y confianza que han tenido en mi trabajo y por haberme guiado en el desarrollo de este Trabajo Especial de Grado. Igualmente al profesor Fernando Crema y al profesor Héctor Navarro por haberme ayudado a corregir los errores del TEG y haber sido un ejemplo de excelencia.

*Este trabajo especial de Grado está dedicado a Pascual Madrid Manzanilla, Pascual Madrid Araujo, Carmen Verónica Blanco y Pablo Padrón*

# Tabla de contenidos

ACTA .....	ii
Resumen .....	iii
Agradecimientos .....	iv
Índice de figuras.....	x
Índice de tablas.....	xiii
Introducción .....	xiv
1. El Problema .....	16
1.1. Planteamiento del Problema .....	16
1.2. Justificación.....	17
1.2.1. ¿Por qué es un problema? .....	17
1.2.2. ¿Para quién es un problema?.....	17
1.2.3. ¿Desde cuándo es un problema?.....	17
1.3. Objetivos de la investigación .....	18
1.3.1. Objetivo General.....	18
1.3.2. Objetivos Específicos.....	18
1.4. Antecedentes .....	18
1.5. Alcance.....	20
2. Marco Conceptual.....	22
2.1. Ciencia de Datos .....	22
2.1.1. Minería de Datos .....	22
2.1.1.1. Clustering .....	23
2.1.1.1.1. K-medias.....	23
2.1.1.1.2. Clustering Jerárquico.....	25
2.1.1.1.3. Análisis de Componentes Principales.....	26
2.1.1.2. Clasificación .....	27
2.1.1.2.1. K vecinos más cercanos.....	27
2.1.1.2.2. Análisis discriminante .....	29
2.1.1.2.3. Maquinas vectoriales de soporte .....	29

## Tabla de Contenidos

2.1.1.2.4. Árbol de Decisión.....	32
2.1.1.2.5. Bosques Aleatorios.....	35
2.1.1.2.6. Redes Bayesianas.....	37
2.1.1.2.7. Redes Neuronales.....	38
2.1.1.2.8. Regresión Logística.....	39
2.1.2. Grandes Volúmenes de Datos.....	39
2.1.2.1. Las 5 V's de Grandes Volúmenes de Datos.....	39
2.2. Sistema Operativo.....	41
2.2.1. Tipos de Sistema Operativo.....	41
2.2.2. Linux.....	43
2.2.2.1. GTK+.....	43
2.2.2.2. Secure Shell.....	43
2.3. Lenguajes de Programación.....	44
2.3.1. R.....	44
2.3.1.1. gWidgets2.....	44
2.3.1.2. RHadoop.....	49
2.3.1.2.1. rmr2.....	50
2.3.1.2.2. rhdfs.....	50
2.3.2. Java.....	50
2.3.3. Python.....	51
2.3.3.1. SciPy.....	51
2.4. Apache Hadoop.....	51
2.4.1. Common.....	53
2.4.2. Hadoop Distributed File System (HDFS).....	53
2.4.2.1. Conceptos.....	55
2.4.3. MapReduce.....	57
2.4.3.1. Hadoop Streaming.....	60
2.4.4. YARN.....	62
2.4.5. Herramientas del Ecosistema Hadoop.....	65
2.4.5.1. Hive.....	65

## Tabla de Contenidos

2.4.5.2. Pig.....	66
2.4.5.3. HCatalog .....	68
2.4.5.4. Apache Spark.....	69
2.4.5.5. Apache Flume .....	70
2.4.5.6. Hue.....	71
2.4.5.7. Apache ZooKeeper .....	72
2.4.5.8. Shark.....	75
2.4.5.9. Apache Sqoop.....	76
2.4.6. Distribuciones Hadoop.....	77
2.4.6.1. MapR.....	77
2.4.6.2. Cloudera.....	77
2.4.6.3. Hortonworks .....	78
2.4.6.4 Comparación entre Hortonworks y Cloudera .....	78
2.4.6.4.1. Similitudes .....	78
2.4.6.4.2. Diferencias.....	79
2.4.6.5. Comparación general .....	80
3. Método de Desarrollo .....	82
3.1. Manifiesto Ágil .....	82
3.1.1. Principios del Manifiesto Ágil .....	83
3.2. Métodos Ágiles.....	83
3.2.1. Metodología Ad Hoc orientada a prototipos.....	84
4. Desarrollo de la Solución.....	87
4.1. Clúster .....	87
4.1.1. Instalación de R.....	87
4.1.1.1. Instalación de rmr.....	87
4.1.1.2. Instalación de rhdfs .....	87
4.2. Ambiente de Desarrollo .....	88
4.2.1. Instalación de GTK+ .....	88
4.2.2. Instalación de R.....	88
4.2.2.1. Instalación de R-Studio .....	88

## Tabla de Contenidos

4.2.2.2. Instalación y actualización de paquetes .....	88
4.2.3. Instalación de Openssh y sshpass .....	90
4.3. Aplicación .....	90
4.3.1. Prototipo 0 .....	91
4.3.1.1. Objetivos .....	91
4.3.1.2. Resultados .....	91
4.3.1.3. Conclusiones.....	95
4.3.2. Prototipo 1 .....	96
4.3.2.1. Objetivos .....	96
4.3.2.2. Resultados .....	96
4.3.2.3. Conclusiones.....	103
4.3.3. Prototipo 2 .....	104
4.3.3.1. Objetivos .....	104
4.3.3.2. Resultados .....	104
4.3.3.3. Conclusiones.....	104
4.3.4. Prototipo 3 .....	105
4.3.4.1. Objetivos .....	105
4.3.4.2. Resultados .....	105
4.3.4.3. Conclusiones.....	113
4.3.5. Prototipo 4 .....	113
4.3.5.1. Objetivos .....	114
4.3.5.2. Resultados .....	114
4.3.5.3. Conclusiones.....	115
4.3.6. Prototipo 5 .....	115
4.3.6.1. Objetivos .....	116
4.3.6.2. Resultados .....	116
4.3.6.3. Conclusiones.....	118
4.3.7. Prototipo 6 .....	119
4.3.7.1. Objetivos .....	119
4.3.7.2. Resultados .....	119



## Tabla de Contenidos

4.3.7.3. Conclusiones.....	128
5. Conclusiones y Resultados.....	129
5.1. Resultados .....	129
5.2. Conclusiones.....	141
5.3. Recomendaciones.....	142
5.4. Trabajos Futuros .....	142
Anexos.....	143
Guía de instalación de R, rmr y rhdfs .....	143
Guía de instalación de gtk+ .....	145
Guía de instalación de R .....	146
Guía de instalación de R-Studio.....	147
Guía de instalación Openssh y sshpass .....	148
Algoritmo de K-medias en MapReduce en R .....	149
Algoritmo de Regresión Logística en MapReduce en R.....	151
Función Map del algoritmo Wordcount en R .....	152
Función Reduce del algoritmo Wordcount en R .....	153
Bibliografía.....	154

# Índice de figuras

Figura 1: Interfaz Gráfica de Weka ( <a href="http://cs.calstatela.edu/wiki/index.php/Courses/CS_491ab/Winter_2009/Lalanthanth_Sathkumara">http://cs.calstatela.edu/wiki/index.php/Courses/CS_491ab/Winter_2009/Lalanthanth_Sathkumara</a> ) .....	19
Figura 2: Interfaz gráfica Rcommander ( <a href="http://uce.uniovi.es/CURSOICE/Informese2.html">http://uce.uniovi.es/CURSOICE/Informese2.html</a> ) .....	19
Figura 3: Interfaz gráfica de Rattle .....	20
Figura 4: Ejemplo agrupación K-medias .....	25
Figura 5: Ejemplo Agrupamiento jerárquico ascendente .....	26
Figura 6: Ejemplo K vecinos más cercanos .....	28
Figura 7: Ejemplo Maquinas vectoriales de Soporte .....	30
Figura 8: Maquina vectorial de soporte de margen máximo .....	31
Figura 9: Maquina vectorial de soporte de margen blando .....	31
Figura 10: Ejemplo Árbol de Decisión .....	32
Figura 11: Árbol Podado .....	35
Figura 12: Ejemplo métodos de consenso .....	35
Figura 13: Ejemplo Bosques Aleatorios .....	36
Figura 14: Ejemplo de ventana creada con el paquete gWidgets2 en R .....	45
Figura 15: Arquitectura HDFS .....	55
Figura 16: Ejemplo de MapReduce .....	57
Figura 17: Arquitectura YARN de Hortonworks .....	62
Figura 18: Proceso MapReduce YARN .....	64
Figura 19: Ciclo de la metodología utilizada .....	85
Figura 20: Ejemplo de widget utilizando R con el paquete gWidgets .....	91
Figura 21: Sección K-medias del Prototipo 0 .....	92
Figura 22: Sección de Análisis de Componentes Principales del Prototipo 092	
Figura 23: Sección Clúster Jerárquico del Prototipo 0 .....	93
Figura 24: Resultado del método K medias del prototipo 0 .....	93
Figura 25: Resultado Análisis de componentes Principales del Prototipo 0 .	94

## Índice de Figuras

Figura 26: Resultados gráficos de Análisis de componentes principales del Prototipo 0 .....	94
Figura 27: Resultado Agrupamiento Jerárquico.....	95
Figura 28: Vista principal del prototipo 1 .....	97
Figura 29: Sección de carga de datos del prototipo 1 .....	98
Figura 30: Sección de análisis exploratorio de datos del prototipo 1 .....	99
Figura 31: Sección para el agrupamiento de datos del prototipo 1 .....	100
Figura 32: Sección de Big Data del prototipo 1 .....	101
Figura 33: Resultados gráficos de la sección de exploración de datos del prototipo 1 .....	102
Figura 34: Resultados gráficos del método K medias del prototipo 1 .....	102
Figura 35: Resultados gráficos del método agrupamiento jerárquico del prototipo 1 .....	103
Figura 36: Resultado K medias MapReduce del prototipo 2.....	104
Figura 37: Sección modelo del prototipo 3.....	106
Figura 38: Sección para la creación de conexiones del prototipo 3 .....	107
Figura 39: Resultado de carga previa al almacenamiento de conexiones en el prototipo 3.....	108
Figura 40: Datos de una conexión del prototipo 3.....	109
Figura 41: Sección dónde se muestran todas las conexiones disponibles del prototipo 3.....	110
Figura 42: Métodos disponibles para las conexiones de tipo archivo csv del prototipo 3.....	111
Figura 43: Métodos disponibles para las conexiones de tipo archivo HDFS del prototipo 3 .....	112
Figura 44: Resultado de clasificación por el método de agrupamiento jerárquico .....	113
Figura 45: Sección crear nueva conexión del prototipo 4 .....	115
Figura 46: Resultado del método K medias del prototipo 5 .....	116
Figura 47: Resultado del método Bosques Aleatorios del prototipo 5.....	117
Figura 48: Gráfico del codo de jambu .....	118
Figura 49: Jerárquia del proyecto .....	120

## Índice de Figuras

Figura 50: Vista principal del prototipo 6.....	121
Figura 51: Funcionalidad de Subir función MAP del prototipo 6 .....	122
Figura 52: Funcionalidad para crear conexión de tipo Archivo HDFS.....	123
Figura 53: Sección para ejecutar funciones MAP y REDUCE sobre un clúster Hadoop .....	124
Figura 54: Sección que muestra las conexiones almacenadas del prototipo 6 .....	125
Figura 55: Ejemplo de tooltip en el prototipo 6.....	126
Figura 56: Ejecución del método K medias MapReduce .....	127
Figura 57: Ejecución del método Regresión Logística MapReduce.....	128
Figura 58: Estadísticas de la Conexión 1 .....	129
Figura 59: Estadísticas de la Conexión 2.....	130
Figura 60: Resultado Análisis de Componentes Principales.....	131
Figura 61: Resultado Análisis de Componentes Principales (Individuos) ...	132
Figura 62: Resultados Análisis de Componentes Principales (Variables)...	133
Figura 63: Resultados del método K medias .....	134
Figura 64: Diagrama de datos del método K medias .....	135
Figura 65: Clasificación después de aplicar el método K medias .....	136
Figura 66: Resultado Agrupamiento Jerárquico.....	137
Figura 67: Clasificación del método Agrupamiento Jerárquico .....	138
Figura 68: Clasificación después de ejecutar el método Agrupamiento Jerárquico .....	139
Figura 69: Centros de grupos generados tras la ejecución del método K medias MapReduce sobre un clúster Hadoop .....	139
Figura 70: Resultado del método Bosques Aleatorios utilizando la conexión 2 .....	140
Figura 71: Resultado del método Regresión Logística utilizando la conexión 3 .....	140

# Índice de tablas

Tabla 1: Parametros de Hadoop Streaming.....	62
Tabla 2: Comparación de las principales distribuciones de Hadoop.....	81
Tabla 3: Primeros siete resultados del Hadoop Streaming.....	141

# Introducción

Actualmente vivimos en un mundo globalizado regido por la Internet y un sin fin de tecnologías basadas de igual manera en la Internet, sabemos que estamos en una época en la que cada día se genera información, la cual muchas veces no nos enteramos que de verdad se haya generado. La mayoría de esta información está no estructurada lo que implica que es indescifrable para el ojo común, debido a que oculta patrones que no se ven a simple vista, al descubrir estos patrones las empresas tendrán una gran ventaja en conocimiento sobre sus principales competidores.

Todo lo planteado ha dado un gran impulso a una gran gama de oportunidades basadas en el conocimiento que se puede obtener a partir de todos esos datos generados diariamente.

A raíz de esto podrían surgir una serie de interrogantes validas que describimos a continuación:

- ¿Cómo hago para acceder de forma eficaz y eficiente a todos estos datos?
- ¿Existe alguna estructura o esquema de almacenamiento que permita mantener en el tiempo estos datos?
- Una vez obtenido los datos ¿Cómo los proceso eficientemente para obtener información relevante que permita optimizar el proceso de toma de decisiones dentro de la organización?

La respuesta a todas estas interrogantes la tiene una tecnología emergente llamada Big Data o Grandes Volúmenes de Datos, la cual se definirá en el contenido de la investigación.

El trabajo de investigación se divide en cinco capítulos. En el primer capítulo se describe el problema a solucionar. Se plantea el problema, su justificación, los objetivos y el alcance de la misma.

En el segundo capítulo se definen las bases teóricas de la investigación. Se define lo que es ciencia de datos, minería de datos, grandes volúmenes de datos la herramienta Apache Hadoop, el lenguaje de programación estadístico R, entre otras definiciones. En el tercer capítulo se define la metodología de desarrollo utilizada para la aplicación que da respuesta al problema de investigación.

## Introducción

En el cuarto capítulo se describe en detalle el proceso de desarrollo de la aplicación y finalmente en el quinto capítulo se describen los resultados y las conclusiones obtenidas.

# 1. El Problema

## 1.1. Planteamiento del Problema

En la Escuela de Computación de la Universidad Central de Venezuela ha surgido la iniciativa de incursionar en el mundo de la ciencia de datos realizando investigaciones y proyectos relacionados con el análisis de datos complejos y/o masivos. Para realizar estos proyectos se necesita un clúster de computadores, específicamente un clúster basado en Apache Hadoop, tener acceso al mismo y escribir métodos para realizar los análisis sobre el clúster.

No siempre se tiene acceso a un clúster de manera directa pero para dar los primeros pasos las 3 grandes compañías especializadas en el procesamiento de grandes volúmenes de datos (Cloudera, Hortonworks y MapR) ofrecen instalaciones de pruebas llamadas "SandBox". Pero estas instalaciones no tienen la capacidad de hacer análisis de envergadura debido a que se encuentran en la maquina local.

Sin embargo se pueden hacer análisis de envergadura sobre un clúster remoto si se encuentra en algún centro dedicado siempre y cuando se tenga acceso al mismo. La mejor manera para conectarse a un clúster remoto es utilizando el protocolo SSH (Secure Shell), pero no todos los interesados en utilizar el clúster conocen los pasos para realizar una conexión.

Para realizar un análisis en un clúster remoto se deben seguir los siguientes pasos:

- 1) Generar los scripts y los datos de entrada de manera local
- 2) Enviar los archivos generados al clúster
- 3) Establecer una conexión dedicada
- 4) Ejecutar los scripts
- 5) Visualizar el resultado.



Esos scripts deben seguir el paradigma de programación MapReduce lo cual resulta muy complejo para cualquier analista debido a que deben cambiar cualquier solución válida a este paradigma para que pueda funcionar en el clúster.

Por todo lo anterior sería de gran utilidad la creación de una aplicación que brinde una interfaz gráfica que provea la funcionalidad de conectarse a un clúster Apache Hadoop y realizar análisis de datos complejos abstrayendo a los usuarios finales de los pasos de conexión remota y del desarrollo de complejos scripts MapReduce utilizando métodos previamente validados por otros analistas con mayores conocimientos en el paradigma MapReduce.

### **1.2. Justificación**

#### **1.2.1. ¿Por qué es un problema?**

Es un problema porque resulta muy engorroso realizar los pasos para la conexión remota a un clúster Apache Hadoop y la realización de scripts MapReduce válidos.

#### **1.2.2. ¿Para quién es un problema?**

Es un problema para cualquier persona que desee incursionar en el mundo de la ciencia de datos y no posea los conocimientos necesarios para la realización de los pasos previos para una conexión remota ni la pericia para desarrollar scripts en el paradigma MapReduce.

#### **1.2.3. ¿Desde cuándo es un problema?**

A grandes rasgos es un problema desde que se desea realizar un análisis de datos teniendo acceso a un cluster pero sin saber cómo acceder al mismo ni como realizar scripts MapReduce.

### 1.3. Objetivos de la investigación

#### 1.3.1. Objetivo General

Desarrollar una aplicación capaz de ejecutar algoritmos de minería de datos en un cluster Apache Hadoop para hacer análisis de datos accediendo al mismo de manera remota y mostrando los resultados de manera local.

#### 1.3.2. Objetivos Específicos

- a) Realizar las instalaciones necesarias en el ambiente de desarrollo local y en el cluster de prueba.
- b) Programar la aplicación.
- c) Definir casos de estudio para las pruebas integradas de la aplicación.
- d) Visualizar los resultados obtenidos.

### 1.4. Antecedentes

No conocemos antecedentes de una aplicación capaz de ejecutar algoritmos MapReduce de manera remota en un clúster Apache Hadoop pero si conocemos antecedentes de aplicaciones que ejecutan modelos de minería de datos, las cuales sirvieron de inspiración para el desarrollo de esta investigación.

- a) **Weka:** Weka es una colección de algoritmos de aprendizaje automático para tareas de minería de datos [1]. Contiene también una colección de herramientas de visualización unidas a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. Las últimas versiones de Weka fueron implementadas en Java [2].

## Capítulo 1. El Problema

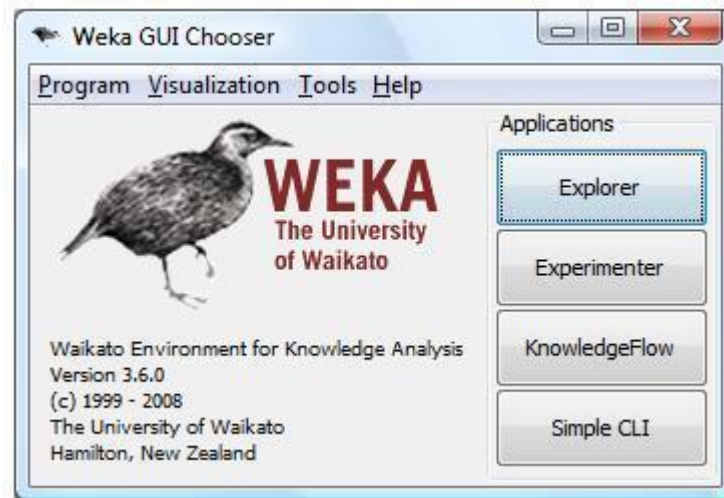


Figura 1: Interfaz Gráfica de Weka ([http://cs.calstatela.edu/wiki/index.php/Courses/CS\\_491ab/Winter\\_2009/Lalantha\\_Sathkumara](http://cs.calstatela.edu/wiki/index.php/Courses/CS_491ab/Winter_2009/Lalantha_Sathkumara))

- b) **Rcommander**: Rcommander es una interfaz gráfica para el uso de las funcionalidades que proporciona el lenguaje de programación R. R proporciona un sistema potente para el análisis de datos y cuando se utiliza con RCommander proporciona una interfaz gráfica que es fácil e intuitiva de usar [3].

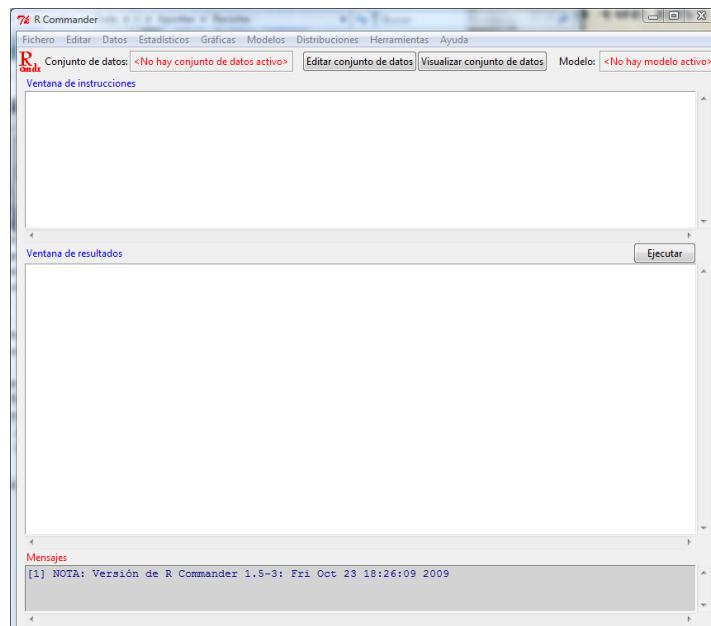


Figura 2: Interfaz gráfica Rcommander (<http://uce.uniovi.es/CURSOICE/Informese2.html>)

## Capítulo 1. El Problema

- c) **Rattle:** Rattle es una interfaz gráfica para la minería de datos utilizando el lenguaje de programación R. Rattle proporciona funcionalidades de minería de datos exponiendo considerablemente el poder R a través de una interfaz gráfica de usuario. Rattle se puede utilizar para el análisis estadístico, o la generación de modelos [4].

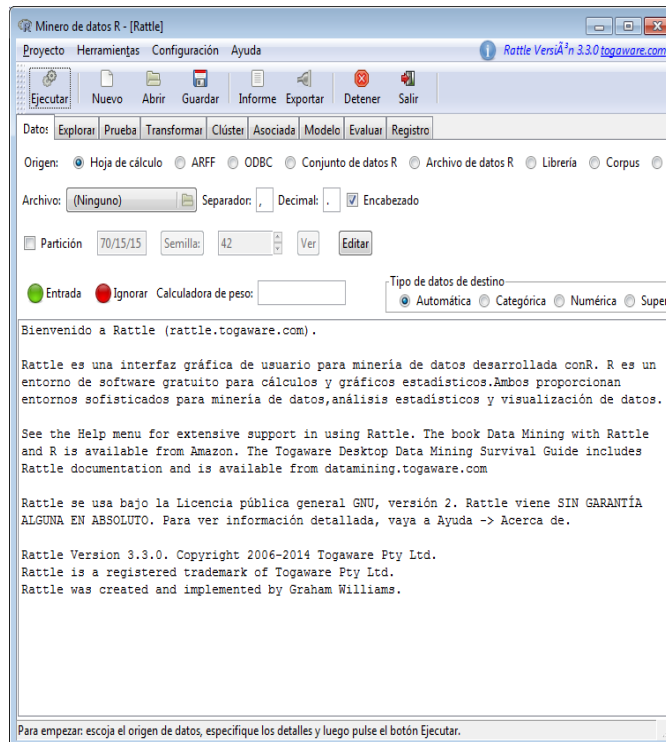


Figura 3: Interfaz gráfica de Rattle

Rattle es la aplicación que más ha inspirado el desarrollo de esta investigación debido a que se programó utilizando las mismas herramientas de desarrollo.

### 1.5. Alcance

Esta aplicación va dirigida a todas las personas que estén interesadas en incursionar en el mundo de la ciencia de datos. La aplicación cuenta con

## Capítulo 1. El Problema

una interfaz sencilla que contempla a grandes rasgos las siguientes funcionalidades:

- a) Crear conexiones de datos
- b) Verificar las conexiones creadas
- c) Realizar análisis exploratorio a las conexiones creadas
- d) Realizar clustering de datos a las conexiones creadas de manera local y en un cluster Apache Hadoop
- e) Aplicación de modelos de minería de datos a las conexiones creadas de manera local y en un cluster Apache Hadoop
- f) Ejecución de funciones map y reduce en un cluster Apache Hadoop

## **2. Marco Conceptual**

### **2.1. Ciencia de Datos**

La Ciencia de Datos (Data Science) es el estudio de la extracción de conocimiento generalizable a partir de datos. El término “Ciencia” implica conocimiento ganado a través de un estudio sistemático. En una definición la Ciencia de Datos, es una empresa sistemática que construye y organiza conocimiento en forma de explicaciones comprobables y predicciones [5]. La Ciencia de Datos solamente es útil cuando se utilizan los datos para responder una pregunta [6].

#### **2.1.1. Minería de Datos**

Es el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Es decir, la tarea fundamental de la minería de datos es encontrar modelos inteligibles a partir de los datos. Para que este proceso sea efectivo debería ser automático o semi-automático (asistido) y el uso de los patrones descubiertos debería ayudar a tomar decisiones más seguras que reporten, por tanto, algún beneficio a la organización [7].

Por lo tanto, dos son los retos de la minería de datos: por un lado, trabajar con grandes volúmenes de datos, procedentes mayoritariamente de sistemas de información, con los problemas que ello conlleva (ruido, datos ausentes, intratabilidad, volatilidad de los datos...), y por el otro usar técnicas adecuadas para analizar los mismos y extraer conocimiento novedoso y útil [7].

Otro concepto: La Minería de Datos es un tópico que involucra el aprendizaje en un sentido práctico. Se interesa en técnicas para encontrar y describir patrones estructurales en los datos, como una herramienta para ayudar a explicar los datos y hacer predicciones a partir de los mismos. Aunque la salida de un proceso de minería de datos no siempre es una predicción, también puede ser una descripción actual de una estructura la cual puede ser utilizada para clasificar ejemplos desconocidos [8].

La minería de datos tiene como objetivo analizar los datos para extraer conocimiento. Este conocimiento puede ser en forma de relaciones, patrones o reglas inferidos de los datos y (previamente) desconocidos, o bien en forma de una descripción más concisa (es decir, un resumen de los mismos). Estas relaciones o resúmenes constituyen el modelo de los datos analizados. Existen muchas formas de representar los modelos y cada una de ellas determina el tipo de técnica que puede usarse para inferirlos.

En la práctica, los modelos pueden ser de dos tipos: predictivos o de clasificación y descriptivos o de Clustering [7].

### **2.1.1.1. Clustering**

Los modelos descriptivos o de Clustering identifican patrones que explican o resumen los datos, es decir, sirven para explorar las propiedades de los datos examinados, no para predecir nuevos datos. Por ejemplo una agencia de viaje desea identificar grupos de personas con unos mismos gustos, con el objeto de organizar diferentes ofertas para cada grupo y poder así remitirles esta información; para ello analiza los viajes que han realizado sus clientes e infiere un modelo descriptivo que caracteriza estos grupos [7].

#### **2.1.1.1.1. K-medias**

El algoritmo de K medias (del inglés Kmeans) se trata de un método de agrupamiento por vecindad en el que se parte de un número determinado de prototipos y de un conjunto de ejemplos a agrupar, sin etiquetar. Es el método más popular de los métodos de agrupamiento denominados “por partición”, en contraposición de los métodos jerárquicos. La idea del K medias es situar a los prototipos o centros en el espacio, de forma que los datos pertenecientes al mismo prototipo tengan características similares [7].

Las regiones se definen minimizando la suma de las distancias cuadráticas entre cada vector de entrada y el centro de su correspondiente clase, representado por el prototipo correspondiente. Cuando se inicia el algoritmo, se debe seleccionar arbitrariamente una partición inicial de forma que cada clase disponga de, al menos, un ejemplo. Como la totalidad de los datos están disponibles, los centros de cada partición se calculan como la media de los ejemplos pertenecientes a esa clase. A medida que el algoritmo se va ejecutando, algunos ejemplos cambian de una clase a otra debiendo

## Capítulo 2. Marco Conceptual

recalcularse los centros a cada paso, o sea, desplazar convenientemente los prototipos. [7]

El procedimiento es el siguiente:

- Se calcula para cada ejemplo  $x_k$ , el prototipo más próximo  $A_g$  y se incluye en la lista de ejemplos de dicho prototipo
- Después de haber introducido todos los ejemplos, cada prototipo  $A_k$  tendrá un conjunto de ejemplos a los que representa
- Se desplaza el prototipo hacia el centro de masas de su conjunto de ejemplos
- Se repite el procedimiento hasta que ya no se desplacen los prototipos.

Mediante este algoritmo el espacio de ejemplos de entrada se divide en  $k$  clases o regiones, y el prototipo de cada clase estará en el centro de la misma. Dichos centros se determinan con el objetivo de minimizar las distancias cuadráticas euclídeas entre los patrones de entrada y el centro más cercano. [7]



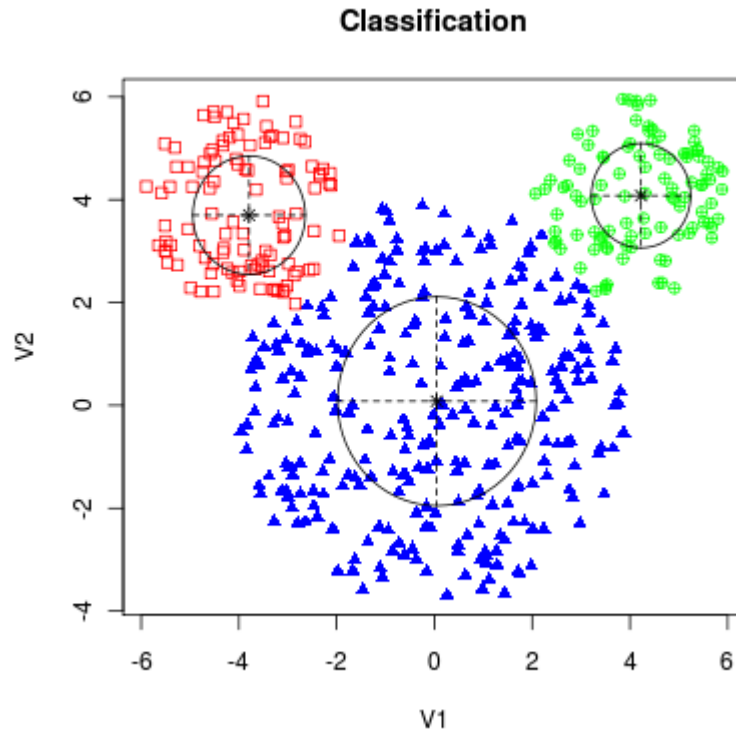


Figura 4: Ejemplo agrupación K-medias

#### 2.1.1.1.2. Clustering Jerárquico

En minería de datos, el agrupamiento jerárquico es un método de análisis de grupos el cual busca construir una jerarquía de grupos. Estrategias para agrupamiento jerárquico generalmente caen en dos tipos [10]:

- Aglomerativas: Este es un agrupamiento ascendente. Cada observación comienza en su propio grupo, y los pares de grupos son mezclados mientras uno sube en la jerarquía.
- Divisivas: Este es un agrupamiento descendente. Todas las observaciones comienzan en un grupo, y se realizan divisiones mientras uno baja en la jerarquía.

En general, las mezclas y divisiones son determinadas de forma golosa. Los resultados del agrupamiento jerárquico son usualmente presentados en un dendograma. [10]

En orden de decidir cuales grupos deberían ser combinados (para aglomerativo), o cuando un grupo debería der dividido (para divisivo), una medida de disimilitud entre conjuntos de observaciones es requerida. En la mayoría de los métodos de agrupamiento jerárquico, esto es logrado mediante el uso de una métrica apropiada (una medida de distancia entre pares de observaciones), y un criterio de enlace el cual especifica la disimilitud de conjuntos como una función de las distancias dos a dos entre observaciones en los conjuntos. [10]

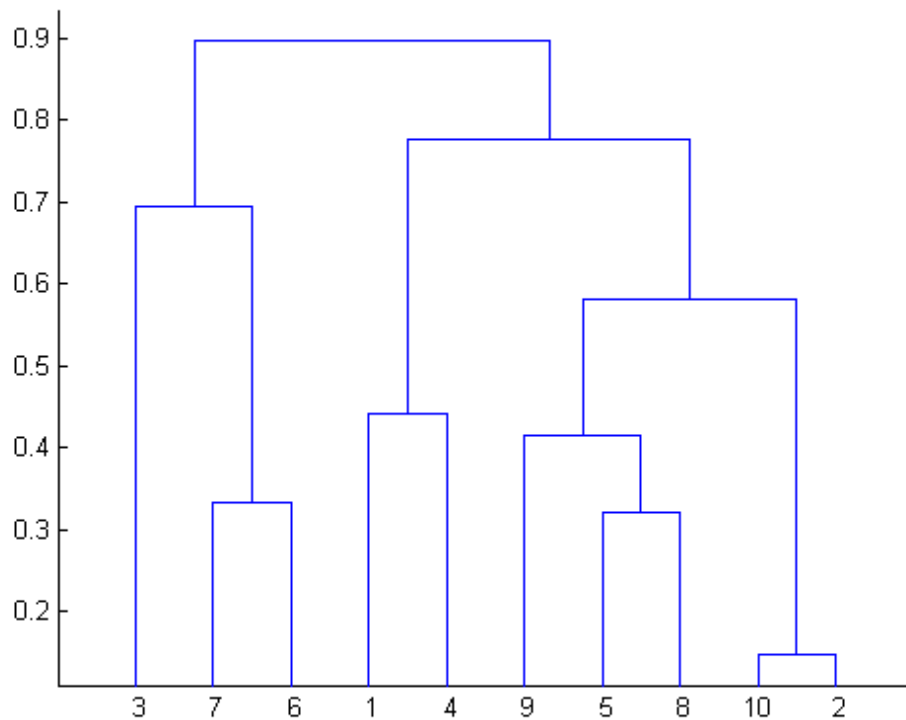


Figura 5: Ejemplo Agrupamiento jerárquico ascendente

### 2.1.1.1.3. Análisis de Componentes Principales

En estadística, el análisis de componentes principales (en español ACP, en inglés, PCA) es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos. Intuitivamente la técnica sirve para hallar las causas de la variabilidad de un conjunto de datos y ordenarlos por importancia.

## Capítulo 2. Marco Conceptual

Técnicamente, el ACP busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. El ACP se emplea sobre todo en el análisis exploratorio de datos y para construir modelos predictivos. El ACP comporta el cálculo de la descomposición en autovalores de la matriz de covarianza, normalmente tras centrar los datos en la medida de cada atributo [9].

El ACP construye una transformación lineal que escoge un nuevo sistema de coordenadas para el conjunto original de datos en el cual la varianza de mayor tamaño del conjunto de datos es capturada en el primer eje (llamado el Primer Componente Principal), la segunda varianza más grande es el segundo eje, y así sucesivamente. Para construir esta transformación lineal debe construirse primero la matriz de covarianza o matriz de coeficientes de correlación. Debido a la simetría de esta matriz existe una base completa de vectores propios de la misma. La transformación que lleva de las antiguas coordenadas a las coordenadas de la nueva base es precisamente la transformación lineal necesaria para reducir la dimensionalidad de datos.

### **2.1.1.2. Clasificación**

Los modelos predictivos o de Clasificación pretenden estimar valores futuros o desconocidos de variables de interés, que se denominan como variables objetivo o dependientes, usando otras variables o campos de la base de datos (o cualquier otra fuente), a las que se refieren como variables independientes o predictivas. Por ejemplo, un modelo predictivo sería aquel que permite estimar la demanda de un nuevo producto en función del gasto en publicidad [7].

#### **2.1.1.2.1. K vecinos más cercanos**

Es un método de clasificación supervisada que sirve para estimar la función de densidad  $F(x/C_j)$  de las predictoras  $x$  por cada clase  $C_j$ . [23]

En este método se determina para cada región del espacio la probabilidad de que un elemento que esté situado en ella pertenezca a cada una de las clases existentes. En este caso no hay reglas prefijadas, sino que la clasificación se irá haciendo para cada caso nuevo en particular. Cuando

## Capítulo 2. Marco Conceptual

un caso nuevo aparece, se genera un círculo con centro en dicho punto y un radio prefijado como parámetro del sistema en el cual se encuentran los  $k$  ejemplos más cercanos al nuevo elemento. Se etiqueta al nuevo caso como perteneciente a la clase más numerosa dentro del círculo.[7]

En la Figura 6 se aprecia cómo en un primer caso con  $k$  igual a cuatro, dentro del círculo hay tres ejemplos de la clase A y uno de la clase B, luego el nuevo dato se etiqueta como perteneciente a la clase A.

En la Figura también se aprecia que el  $k$  elegido influye en la clasificación. Si se aumenta o disminuye el  $k$ , la predicción realizada puede variar. Por lo tanto, el  $k$  será un parámetro crítico a tener en cuenta. También las predicciones pueden variar si se varía la función de distancia.

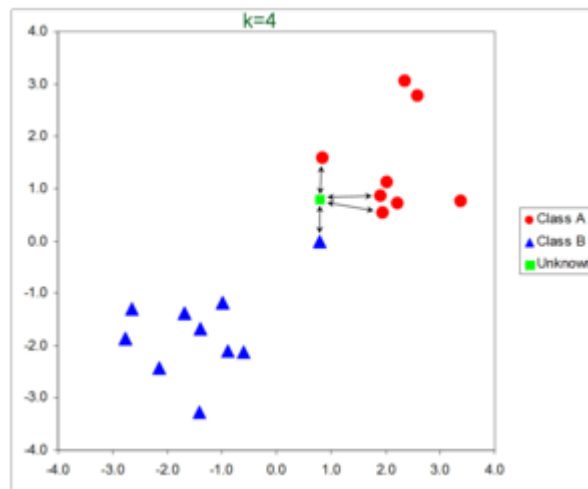


Figura 6: Ejemplo K vecinos más cercanos

El método espera hasta la aparición de un nuevo dato a clasificar para la utilización del conjunto de ejemplos. Cuando el dato está disponible se recurre a los ejemplos para realizar la clasificación, se crea una regla local al dato que acaba de llegar, se realiza la clasificación, y se abandona dicha regla. Si ahora hubiera que clasificar otro dato más, los cálculos realizados para la clasificación del dato anterior serían inservibles y habría que realizar de nuevo todo el proceso. Este es un típico ejemplo de aprendizaje retardado. En este caso hay que almacenar los ejemplos de entrenamiento siempre, ya que son utilizados una y otra vez. [7]

### **2.1.1.2.2. Análisis discriminante**

El objetivo del análisis discriminante es encontrar reglas de asignación de individuos a una de las clases de una clasificación preestablecida. El término análisis discriminante es el nombre que se utiliza tradicionalmente en estadística para englobar las técnicas de clasificación supervisada.

Para resolver este problema se dispone de una muestra de  $n$  individuos, de los cuales sabemos su clase de pertenencia y en los que tenemos medidas un conjunto de  $p$  variables que permiten diferenciar a las clases. [7]

Geoméricamente, las  $p$  variables permiten situar a los individuos en un espacio  $R^p$  euclídeo. Por otro lado, se supone que estas variables permiten diferenciar los individuos según su clase de pertenencia, es decir, se supone que cada clase viene definida por una distribución de probabilidad distinta de las restantes clases. Naturalmente, las variables  $x_j$  no tienen por qué ser las originales sino que puede ser transformaciones que optimicen la separación entre las clases [7]

### **2.1.1.2.3. Maquinas vectoriales de soporte**

Las máquinas de vectores de soporte pertenecen a la familia de los clasificadores lineales puesto que inducen separadores lineales o hiperplanos en espacios de características de muy alta dimensionalidad (introducidos por funciones de núcleo o kernel) con un sesgo inductivo muy particular (maximización del margen). [7]

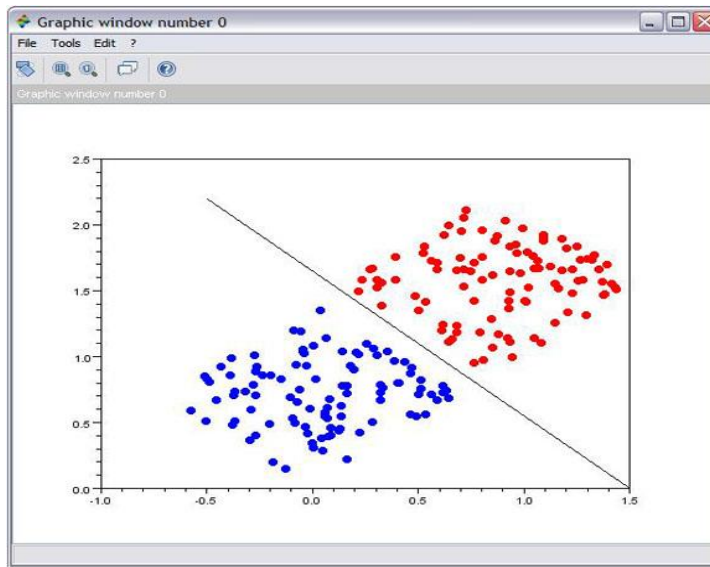


Figura 7: Ejemplo Maquinas vectoriales de Soporte

La idea básica de las máquinas de vectores de soporte es:

Dado un conjunto de puntos, en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría se desconoce) pertenece a una categoría o a la otra.

Como en la mayoría de los métodos de clasificación supervisada, los datos de entrada (los puntos) son vistos como un vector  $p$ -dimensional.

Las SVM buscan un hiperplano que separe de forma óptima a los puntos de una clase de la otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

En ese concepto de “separación óptima” es donde reside la característica fundamental de las SVM: este tipo de algoritmos busca el hiperplano que tenga la máxima distancia (margen) con los puntos que están más cerca de él mismo. Por eso también a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

Al vector formado por los puntos más cercanos al hiperplano se le llama vector de soporte. [43]

## Capítulo 2. Marco Conceptual

La SVM lineal con margen máximo es el modelo más sencillo e intuitivo de SVM, aunque también el que tiene condiciones de aplicabilidad más restringidas, puesto que parte de la hipótesis de que el conjunto de datos es linealmente separable en el espacio de entrada. [7]

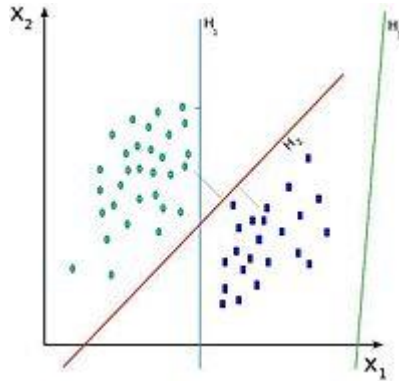


Figura 8: Máquina vectorial de soporte de margen máximo

Desgraciadamente, no siempre es posible encontrar una transformación de los datos que permita separarlos linealmente, y si se logra, el resultado del modelo no puede ser generalizado para otros datos.

Con el fin de permitir cierta flexibilidad, los SVM manejan un parámetro  $C$  que controla la compensación entre errores de entrenamiento y los márgenes rígidos, creando así un margen blando que permita algunos errores en la clasificación a la vez que los penaliza. [43]

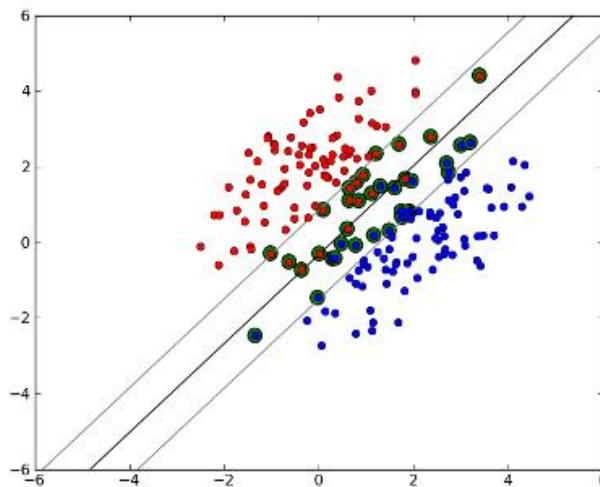


Figura 9: Máquina vectorial de soporte de margen blando

#### 2.1.1.2.4. Árbol de Decisión

De todos los métodos de aprendizaje, los sistemas de aprendizaje basados en árboles de decisión son quizás el método más fácil de utilizar y de entender. Un árbol de de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Los árboles de decisión se utilizan desde hace siglos, y son especialmente apropiados para expresar procedimientos médicos, legales, comerciales, estratégicos, matemáticos, lógicos, etc. [7]

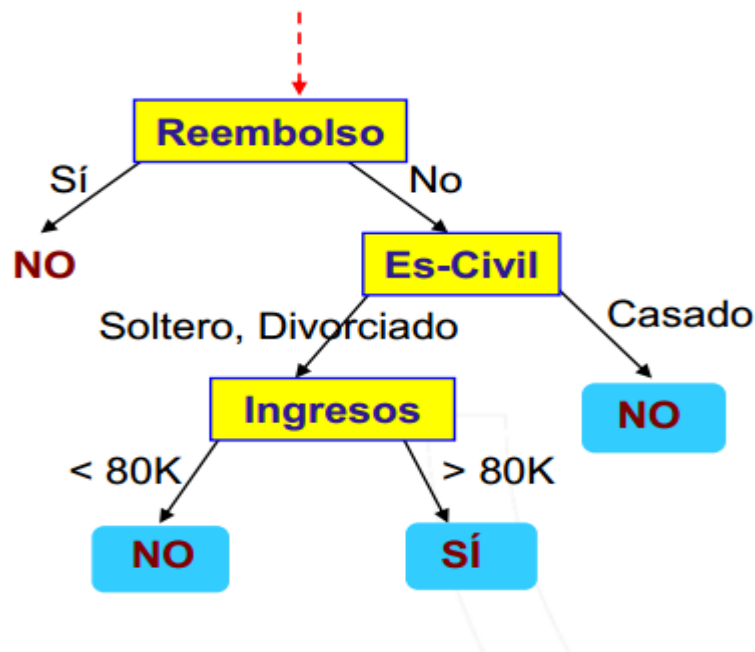


Figura 10: Ejemplo Árbol de Decisión

Una de las grandes ventajas de los árboles de decisión es que en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Esto permite analizar una situación y, siguiendo el árbol de decisión apropiadamente, llegar a una sola acción o decisión a tomar. [7]

La tarea de aprendizaje para la cual los árboles de decisión se adecuan mejor es la clasificación. De hecho, clasificar es determinar entre varias clases a qué clase pertenece un objeto; la estructura de condición y ramificación de un árbol de decisión es idónea para este problema. La



## Capítulo 2. Marco Conceptual

característica más importante del problema de la clasificación es que se asume que las clases son disjuntas.

Debido al hecho de que la clasificación trata con clases o etiquetas disjuntas, un árbol de decisión conducirá un ejemplo hasta una y sólo una hoja, asignando, por tanto, una única clase al ejemplo. Para ello las particiones existentes en el árbol deben ser también disjuntas. Es decir, cada instancia cumple o no cumple una condición.

Esta propiedad dio lugar al esquema básico de los primeros algoritmos de aprendizaje de árboles de decisión; el espacio de instancias se iba partiendo de arriba abajo, utilizando cada vez una partición, es decir, un conjunto de condiciones excluyentes y exhaustivas.

Otra característica importante de los primeros algoritmos de aprendizaje de árboles de decisión es que una vez elegida la partición dicha partición no se podía cambiar, aunque más tarde se pensara que había sido una mala elección. Por tanto, uno de los aspectos más importantes en los sistemas de aprendizaje de árboles de decisión es el denominado criterio de partición, ya que una mala elección de la partición (especialmente en las partes superiores del árbol) generará un peor árbol.

Las particiones son, como se ha dicho, un conjunto de condiciones exhaustivas y excluyentes. Lógicamente, cuantos más tipos de condiciones se permitan, más posibilidades se tendrán de encontrar los patrones que hay detrás de los datos. Cuantas más particiones se permitan más expresivos podrán ser los árboles de decisión generados y probablemente, más precisos. No obstante, cuantas más particiones elijamos, la complejidad del algoritmo será mayor. [7]

Incluso con sólo dos tipos de particiones sencillas, el número de particiones posibles en cada caso puede dispararse (si existen  $n$  atributos y  $m$  valores posibles para cada atributo, el número de particiones posibles es de  $n$  por  $m$ ). Como se ha dicho anteriormente, los algoritmos clásicos de aprendizaje de decisión son voraces, en el sentido de que una vez elegida la partición se continúa hacia abajo la construcción del árbol y no vuelve a plantearse las particiones ya construidas. Estos dos aspectos tienen como consecuencia que se busque un criterio que permita realizar una buena elección de la partición que parece más prometedora y que esto se haga sin demasiado esfuerzo computacional. Esto obliga a que calcular la optimalidad de cada partición no sea muy costoso. [7]

## Capítulo 2. Marco Conceptual

La mayoría de criterios se basan por tanto en obtener medidas derivadas de las frecuencias relativas de las clases en cada uno de los hijos de la partición respecto a las frecuencias relativas de las clases en el padre. Por ejemplo, si en un nodo tenemos cincuenta por ciento de ejemplos de clase a y un cincuenta por ciento de ejemplos de clase b, una partición que dé como resultado dos nodos  $n_1$  y  $n_2$ , donde todos los ejemplos de  $n_1$  sean de la clase a y todos los ejemplos de  $n_2$  sean de la clase b, será una buena partición, porque los nodos resultantes son más puros que el padre. Por el contrario, si ambos nodos  $n_1$  y  $n_2$  siguen teniendo proporciones cercanas al cincuenta por ciento no habremos discriminado nada y no avanzaremos hacia un árbol que nos clasifique correctamente la evidencia. [7]

Basándose en la idea de buscar particiones que discriminen o que consigan nodos más puros, se han presentado en las últimas dos décadas numerosos criterios de partición, tales como el criterio del error esperado, el criterio de Gini, los criterios de Gain, entre otros. [7]

Los algoritmos de aprendizaje de árboles de decisión y conjuntos de reglas mencionados previamente obtienen un modelo que es completo y consistente con respecto a la evidencia. Es decir, el modelo cubre todos los ejemplos vistos y los cubre todos de manera correcta. Esto puede parecer óptimo a primera vista, pero se vuelve demasiado ingenuo en la realidad. En primer lugar, ajustarse demasiado a la evidencia suele tener como consecuencia que el modelo se comporte mal para nuevos ejemplos, ya que, en la mayoría de los casos, el modelo es solamente una aproximación del concepto objetivo del aprendizaje. Por tanto, intentar aproximar demasiado hace que el modelo sea demasiado específico, poco general y, por tanto, malo con otros datos no vistos. En segundo lugar, esto es especialmente patente cuando la evidencia puede contener ruido (errores en los atributos o incluso en las clases), ya que el modelo intentará ajustarse a los errores y esto perjudicará el comportamiento global del modelo aprendido. Esto es lo que se conoce como sobreajuste (overfitting). [7]

La manera más frecuente de limitar este problema es modificar los algoritmos de aprendizaje de tal manera que obtengan modelos más generales. En el contexto de los árboles de decisión y conjuntos de reglas, generalizar significa eliminar condiciones de las ramas del árbol o de algunas reglas. En el caso de los árboles de decisión dicho procedimiento se puede ver gráficamente como un proceso de poda, como se ilustra en la Figura 11.

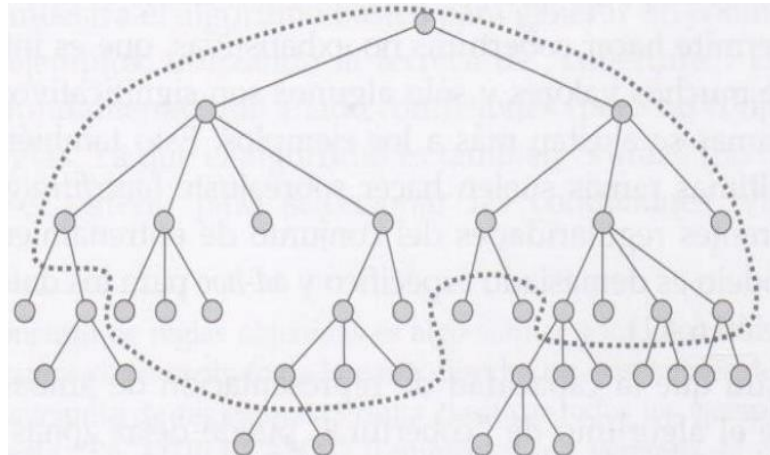


Figura 11: Árbol Podado

### 2.1.1.2.5. Bosques Aleatorios

Antes de hablar de los Bosques Aleatorios es conveniente mencionar la definición de los métodos de consenso o Bagging. La idea fundamental de estos métodos es tomar  $m$  muestras aleatorias con reemplazo de los datos originales y luego aplicar a cada una de ellas un método predictivo para luego con algún criterio establecer un consenso de todos los resultados. El consenso podría ser un promedio, un promedio ponderado basado en cuál método obtuvo los mejores resultados o el que obtenga la mayor cantidad de votos.[44]

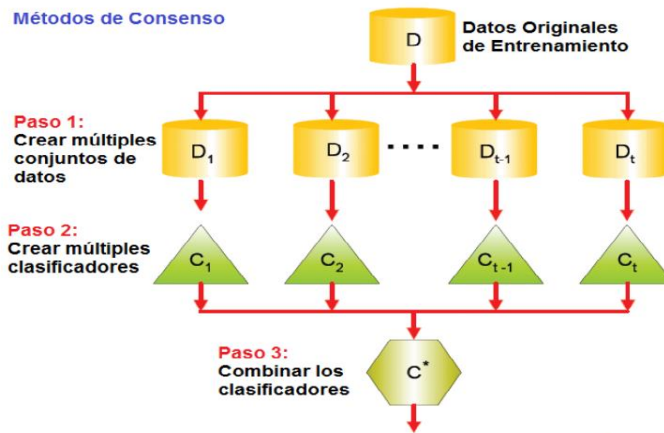


Figura 12: Ejemplo métodos de consenso

Bosques Aleatorios o Random forest es una combinación de árboles de decisión tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.

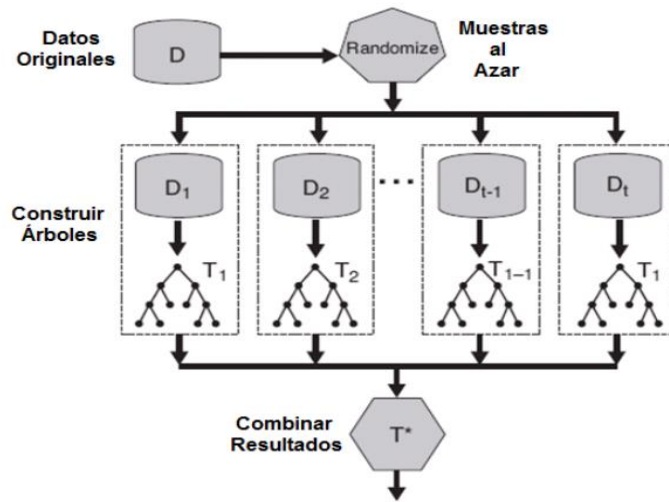


Figura 13: Ejemplo Bosques Aleatorios

Es una modificación sustancial de los métodos de consenso que construye una larga correlación de árboles no correlacionados y luego los promedia. [45]

Las ventajas de los Bosques aleatorios son [45]:

- Es uno de los algoritmos de aprendizaje más certeros que hay disponible. Para un conjunto de datos lo suficientemente grande produce un clasificador muy certero
- Corre eficientemente en bases de datos grandes
- Puede manejar cientos de variables de entrada sin excluir ninguna
- Da estimados de qué variables son importantes en la clasificación
- Tiene un método eficaz para estimar datos perdidos y mantener la exactitud cuando una gran proporción de los datos está perdida
- Computa los prototipos que dan información sobre la relación entre las variables y la clasificación
- Computa las proximidades entre los pares de casos que pueden usarse en los grupos, localizando valores atípicos, o (ascendiendo) dando vistas interesantes de los datos
- Ofrece un método experimental para detectar las interacciones de las variables

Las desventajas son [45]:

- Se ha observado que los Bosques aleatorios sobreajusta en ciertos grupos de datos con tareas de clasificación ruidosas
- A diferencia de los árboles de decisión, la clasificación hecha por Bosques aleatorios es difícil de interpretar por el hombre
- Para los datos que incluyen variables categóricas con diferente número de niveles, el random forests se parcializa a favor de esos atributos con más niveles. Por consiguiente, la posición que marca la variable no es fiable para este tipo de datos

Si los datos contienen grupos de atributos correlacionado con similar relevancia para el rendimiento, entonces los grupos más pequeños están favorecidos sobre los grupos más grandes

### 2.1.1.2.6. Redes Bayesianas

Antes de hablar de Naïve Bayes se debe hablar del Teorema de Bayes. En teoría de la probabilidad, el teorema de Bayes es la regla básica para realizar inferencias. Así, el teorema de Bayes nos permite actualizar la creencia que tenemos en un suceso o conjunto de sucesos a la luz de nuevos datos u observaciones. Es decir, nos permite pasar de la probabilidad a priori  $P(\text{suceso})$  a la probabilidad a posteriori  $P(\text{suceso}|\text{observaciones})$ . La probabilidad a priori puede verse como la probabilidad inicial, la que fijamos sin saber nada más. La probabilidad a posteriori es la que obtendríamos tras conocer cierta información, por tanto, puede verse como un refinamiento de nuestro conocimiento.[7]

Teniendo en cuenta estos conceptos, el teorema de Bayes viene representado por la siguiente expresión:

$$P(h|O) = \frac{P(O|h) \cdot P(h)}{P(O)}$$

Donde, se puede observar, lo que aparecen son la probabilidad a priori de la hipótesis ( $h$ ) y de las observaciones ( $O$ ) y las probabilidades condicionadas  $P(h|O)$  y  $P(O|h)$ .

Centrándose en el problema de la clasificación, con una variable clase ( $C$ ) y un conjunto de variables predictoras o atributos  $\{A_1, \dots, A_n\}$ , el teorema de Bayes tendría la siguiente forma:

$$P(C|A_1 \dots A_n) = \frac{P(A_1 \dots A_n|C) \cdot P(C)}{P(A_1 \dots A_n)}$$

Evidentemente, si C tiene k posibles valores  $\{c_1, \dots, c_k\}$ , lo que interesa es identificar el más plausible y devolverlo como resultado de la clasificación. En el marco bayesiano, la hipótesis más plausible no es otra que aquella que tiene máxima probabilidad a posteriori dados los atributos, y es conocida como la hipótesis máxima a posteriori.

Por tanto, el teorema de Bayes facilita un método sencillo y con una semántica clara para resolver esta tarea. Sin embargo, este método tiene un problema, y es su altísima complejidad computacional, debido a que se necesita trabajar con distribuciones de probabilidad que involucran muchas variables, haciéndolas en la mayoría de los casos inmanejables.[7]

#### 2.1.1.2.7. Redes Neuronales

Las redes neuronales artificiales son un método de aprendizaje cuya finalidad inicial era la de emular los procesadores biológicos de información. Las RNA parten de la presunción de que la capacidad humana de procesar información se debe a la naturaleza biológica del cerebro. Por tanto, para imitar esta característica se debe estudiar y basarse en el uso de soportes artificiales semejantes a los existentes en el cerebro.[7]

Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones [46]:

1. Una función de propagación (también conocida como función de excitación), que por lo general consiste en el sumatorio de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, denomina inhibitoria.
2. Una función de activación, que modifica a la anterior. Puede no existir, siendo en este caso la salida la misma función de propagación.
3. Una función de transferencia, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas

son la función sigmoidea (para obtener valores en el intervalo  $[0, 1]$ ) y la tangente hiperbólica (para obtener valores en el intervalo  $[-1, 1]$ ).

### **2.1.1.2.8. Regresión Logística**

Antes de hablar de lo que es el modelo de regresión logística se debe saber qué es un modelo de regresión. Un modelo de regresión es cuando la variable de respuesta y las variables explicativas son todas ellas cuantitativas [7]. Si sólo se dispone de una variable explicativa hablamos de regresión simple, mientras que si se dispone de varias variables explicativas se trata de una regresión múltiple.

La regresión logística es un tipo especial de regresión que se utiliza para predecir el resultado de una variable categórica en función de las variables explicativas [11].

### **2.1.2. Grandes Volúmenes de Datos**

Una parte fundamental de la Ciencia de Datos es Grandes Volúmenes de datos o en inglés Big Data el cual es un término global para cualquier colección de conjuntos de datos tan grandes y complejos que complican el procesamiento de los mismos inhabilitando el uso de herramientas de gestión de datos y las aplicaciones tradicionales de tratamiento de datos.[24]

#### **2.1.2.1. Las 5 V's de Grandes Volúmenes de Datos**

Se sabe que los datos se están convirtiendo en la base de cara a obtener ventajas competitivas con el afán de desmarcarnos del resto de compañías. Sin embargo, el gran inconveniente del Big Data y del tratamiento de datos es la ausencia de un diccionario, guía o 'librillo' que dicte la praxis para exprimir todo el potencial que posee. No obstante, en el año 2001, el experto analista de datos Doug Laney definió los tres vectores de los volúmenes de datos. Desde entonces, numerosos autores han aparecido con otras definiciones y descripciones que enriquecen la teoría de Laney.[37]

## Capítulo 2. Marco Conceptual

Finalmente, la comunidad tecnológica que rodea al Big Data y el Business Intelligence se ha puesto de acuerdo y ha establecido cinco directrices que describen.

- I. **V de Volumen:** El primer aspecto que se nos viene a la cabeza cuando pensamos en el Big Data es un torrente de datos desestructurados que guardan un inmenso potencial en sí mismos. Dicho esto, no es de extrañar que las empresas ya sepan con los volúmenes que tienen ante sí. Y es que el panorama tecnológico referente ha sufrido variaciones considerables. Lo que antes se consideraba grande, ahora ya no lo es tanto, sino basta con echar un vistazo al Gigabyte, que al parecer ya se ha convertido en la unidad “básica” de almacenamiento, frente a los Petabytes que engloba el Big Data. [37]
- II. **V de Velocidad:** Para un gran volumen de datos que no sufre variaciones muy a menudo, el análisis lleva horas e incluso días. No obstante, en el ámbito del Big Data el montaje de información crece por Terabytes, de ahí que el tiempo de procesamiento de la información sea un factor fundamental para que dicho tratamiento aporte ventajas que marquen la diferencia. [37]
- III. **V de Variedad:** De sobra se sabe que el Big Data no versa en la mayoría de las ocasiones sobre datos estructurados y que no siempre es sencillo incorporar grandes volúmenes a una base de datos relacional. Infinidad de tipos de datos se aglutinan dispuestos a ser tratados y es por ello que frente a esa variedad aumenta el grado de complejidad tanto en el almacenamiento como en su análisis. [37]
- IV. **V de Veracidad:** Con un alto volumen de información que crece a tal velocidad y es de tamaño variedad, en ocasiones es inevitable dudar del grado de veracidad que éstos poseen. Para ello, se incide en ejercer una limpieza en los datos para así asegurar el mayor aprovechamiento de los mismos. No obstante, supone un gran esfuerzo que a grosso modo no reflejará variaciones esenciales de los resultados finales



relativos al tratamiento de la información. Por lo tanto, dependiendo de la aplicación que se les dé, su veracidad y su verificación puede ser imprescindible o simplemente un acto secundario sin llegar a ser vital. [37]

- V. **V de Valor:** Sin duda el aspecto más relevante del Big Data. Es muy costoso poner en práctica las infraestructuras informáticas para almacenar estos volúmenes de datos, y por ende, las empresas van a necesitar gran cantidad de dinero para rentabilizar su gasto. Si no se consigue extraer todo el valor de ellos, no habrá lugar para almacenar ni administrar. [37]

### 2.2. Sistema Operativo

Un sistema operativo (SO) es el software de sistema que gestiona los recursos de hardware y software del ordenador y proporciona servicios comunes para los programas del mismo. El sistema operativo es un componente esencial del software del sistema en un sistema informático. Los programas de aplicación por lo general requieren un sistema operativo para funcionar [12].

Para las funciones de hardware como las entrada y salida y la asignación de memoria, el sistema operativo actúa como intermediario entre los programas y el hardware del ordenador, aunque el código de la aplicación se suele ejecutarse directamente por el hardware y con frecuencia hace que el sistema llame a una función del Sistema Operativo o ser interrumpido por ella. Los sistemas operativos se encuentran en muchos dispositivos que contienen un ordenador desde teléfonos celulares y consolas de videojuegos a los servidores web y supercomputadoras [12].

#### 2.2.1. Tipos de Sistema Operativo

- i. **Single- and Multi-Tasking:** Un sistema Single-Tasking sólo puede ejecutar un programa a la vez, mientras que un sistema operativo Multi-Tasking permite que más de un programa que se ejecuta de manera concurrente. Esto se logra por un tiempo

compartido, dividiendo el tiempo de procesador disponible entre múltiples procesos que son cada uno interrumpidos repetidamente por un subsistema de programación de tareas del sistema operativo [12].

- ii. **Single- and Muti-User:** Los sistemas operativos de un solo usuario no tienen instalaciones para distinguir a los usuarios, pero puede permitir que varios programas se ejecuten en paralelo. Un sistema operativo multiusuario extiende el concepto básico de la multitarea con instalaciones que identifican los procesos y recursos, tales como el disco espacio, que pertenece a varios usuarios, y el sistema permite que varios usuarios interactúen con el sistema al mismo tiempo [12].
- iii. **Distribuido:** Un sistema operativo distribuido gestiona un grupo de equipos distintos y los hace parecer como un solo equipo. El desarrollo de ordenadores conectados en red que podrían estar vinculados y se comunican entre sí dio lugar a la computación distribuida. [12]
- iv. **Templated:** El término se refiere a la creación de una única imagen de máquina virtual como un sistema operativo invitado. La técnica se utiliza tanto en la virtualización y la gestión de la computación en la nube, y es común en grandes almacenes de un servidor. [12]
- v. **Embebidos:** Los sistemas operativos embebidos están diseñados para ser utilizados en sistemas informáticos integrados. Están diseñados para operar en pequeñas máquinas como PDAs con menos autonomía. Ellos son capaces de operar con un número limitado de recursos. [12]
- vi. **Real-time:** Un sistema operativo de tiempo real es un sistema operativo que garantiza procesar los eventos o datos dentro de una corta cantidad de tiempo. Un sistema operativo de tiempo real puede ser de una o varias tareas a la vez. [12]

### **2.2.2. Linux**

Linux es un sistema operativo de software libre, compatible Unix. El sistema lo forman el núcleo del sistema (kernel) más un gran número de programas / bibliotecas que hacen posible su utilización. Muchos de estos programas y bibliotecas han sido posibles gracias al proyecto GNU, por esto mismo, muchos llaman a Linux, GNU/Linux, para resaltar que el sistema lo forman tanto el núcleo como gran parte del software producido por el proyecto GNU.

Linux se distribuye bajo la GNU General Public License por lo tanto, el código fuente tiene que estar siempre accesible y cualquier modificación ó trabajo derivado tiene que tener esta licencia. [13]

#### **2.2.2.1. GTK+**

GTK+, o GIMP toolkit, es un kit de herramientas multiplataforma para crear interfaces gráficas de usuario. Ofreciendo un conjunto completo de los widgets. GTK+ está escrito en C, pero ha sido diseñado desde cero para apoyar una amplia gama de lenguajes, no sólo C/C++. El uso de GTK+ de lenguajes como Perl y Python proporciona un método eficaz para el desarrollo rápido de aplicaciones. GTK+ es software libre y parte del proyecto GNU. Sin embargo, los términos de licencia para GTK+, la licencia GNU LGPL, permiten que sea utilizado por todos los desarrolladores, incluyendo aquellos que desarrollan software propietario, sin ningún tipo de derechos de licencia o regalías. [14]

#### **2.2.2.2. Secure Shell**

Secure Shell (SSH) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos ejecutando un Servidor X (en sistemas Unix y Windows). Además de la conexión a otros dispositivos, SSH nos permite copiar datos de forma segura (tanto archivos sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves

al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH. [15]

### **2.3. Lenguajes de Programación**

#### **2.3.1. R**

El proyecto R es a la vez un lenguaje especializado y un conjunto de herramientas de módulos dirigido a cualquier persona trabajando con las estadísticas. Abarca todo, desde la carga de los datos a la ejecución de análisis sofisticados en él y luego ya sea de exportación o la visualización de los resultados. La consola interactiva hace que sea fácil de experimentar con sus datos, ya que se puede probar un montón de diferentes enfoques muy rápidamente. La desventaja más grande desde una perspectiva de proceso de datos es que está diseñado para trabajar con conjuntos de datos que se ajustan a la memoria de una sola máquina. Es posible utilizarlo dentro de Hadoop como otro lenguaje para streaming, pero una gran cantidad de las más potentes funciones requieren acceso al conjunto completo de datos para ser eficaz. R hace una gran plataforma de creación de prototipos para el diseño de soluciones que necesitan para funcionar con cantidades masivas de datos, sin embargo, o para dar sentido a los resultados de menor escala del procesamiento. [25]

R se compone de una serie de paquetes programados por una comunidad activa de desarrolladores. A continuación se describen los paquetes más importantes utilizados en el marco del desarrollo de esta investigación.

##### **2.3.1.1. gWidgets2**

El paquete gWidgets2 proporciona una interfaz de programación para la fabricación de interfaces gráficas de usuario dentro de R. El paquete es una reescritura del paquete gWidgets. El paquete se basa en uno de los varios paquetes toolkit subyacentes que dan acceso a las bibliotecas de gráficos. Estos incluyen RGtk2, tcltk, qtbase, y una colección de widgets del navegador proporcionado por ExtJS. [16]

El paquete proporciona constructores para desarrollar controles, widgets con los que un usuario interactúa, contenedores, objetos GUI

utilizados para organizar los controles dentro de una ventana y diálogos simples. Estos objetos son manipulados a través de diversos métodos. El paquete proporciona algunas funcionalidades genéricas y, en lo posible, aprovecha los métodos existentes para R. [16]

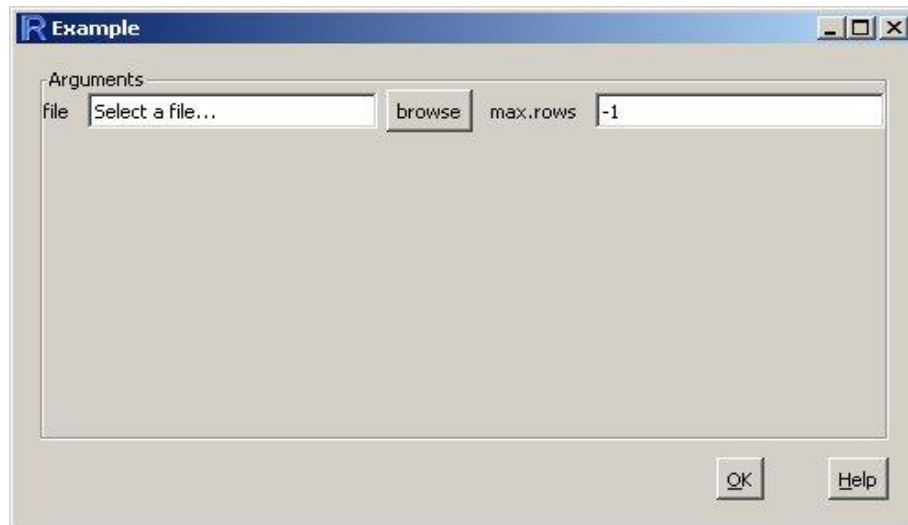


Figura 14: Ejemplo de ventana creada con el paquete gWidgets2 en R

Los constructores se pueden dividir en las siguientes categorías:

i) Constructores de control:

1. `gbutton`: Provee un botón básico para iniciar una acción
2. `gcalendar`: Provee una entrada de texto con formato de fecha
3. `gcheckbox`: Provee un checkbox junto con su etiqueta para permitir a los usuario realizar selecciones
4. `gcheckboxgroup`: Igual que el `gcheckbox` pero permitiendo la selección de cero o más objetos
5. `gcombobox`: Provee una lista desplegable con una lista de opciones para una libre selección
6. `gdf`: Provee un widget para la edición de un dataframe
7. `gedit`: Provee un entrada de texto
8. `ggraphics`: Provee la función de mostrar gráficos embebidos en un widget

## Capítulo 2. Marco Conceptual

9. `gimage`: Provee la función para que un widget soporte imágenes
10. `glabel`: Provee etiquetas
11. `gmenu`: Provee menús en la parte superior de la ventana
12. `gradio`: Proporciona un medio para seleccionar uno de los muchos objetos
13. `gseparator`: Proporciona una línea visual para separar partes de una ventana
14. `gslider`: Proporciona un medio para seleccionar un valor de uno de continuo de valores
15. `gspinbutton`: Proporciona los medios para seleccionar un valor de una secuencia de valores
16. `gstatusbar`: Proporciona un widget para mostrar los mensajes de estado en una ventana de nivel superior
17. `gtable`: Proporciona un widget para mostrar datos tabulares para la selección
18. `gtext`: Proporciona un widget de edición de texto de varias líneas
19. `gtimer`: Proporciona un temporizador
20. `gtoolbar`: Proporciona barras de herramientas para ventanas de nivel superior
21. `gtree`: Proporciona una pantalla para los datos jerárquicos
22. `gvarbrowser`: Proporciona un widget que muestra una instantánea del espacio actual de trabajo
23. `gaction`: Proporciona un medio para encapsular las acciones para su uso con barras de menú, barras de herramientas y botones.
24. `gexpandgroup`: Proporciona un contenedor con una opción de revelar u ocultar sus hijos
25. `gframe`: Proporciona un contenedor cuadro enmarcado
26. `gggroup`: Proporciona un contenedor caja horizontal o vertical para el embalaje en componentes hijos
27. `glayout`: Proporciona un contenedor para organizar los datos por filas y columnas
28. `gnotebook`: Proporciona un contenedor portátil
29. `gpanedgroup`: Proporciona un contenedor dividido con divisor ajustable
30. `gstackwidget`: Proporciona un recipiente como un bloc de notas, pero sin etiquetas pestaña

31. gwindow: Proporciona una ventana de nivel superior

### ii) Constructores de dialogo:

1. gmessage: Produce un diálogo sencillo para mostrar un mensaje
2. gconfirm: Produce un cuadro de diálogo para un usuario para confirmar una acción
3. ginput: Proporciona un cuadro de diálogo para recoger la entrada del usuario
4. gbasicdialog: Proporciona un medio para producir cuadros de diálogo modales generales
5. galert: Proporciona un diálogo de mensaje transitoria corta
6. gfile: Proporciona un cuadro de diálogo para seleccionar un nombre de archivo o directorio

### iii) Métodos

1. svalue: Esto se utiliza para recuperar o establecer la propiedad principal asociado con un widget
2. enabled: Un widget está activado si es sensible a la entrada del usuario. Los widgets no activados normalmente se presentan en un estado en gris.
3. visible: La idea genérica de un widget visible es uno que se dibuja. Sin embargo, varias clases anulan esto como parte del widget es visible o no visible.
4. focus: Un widget con focus recibe cualquier entrada de teclado.
5. editable: Un widget es editable si puede recibir la entrada de teclado.
6. font: La fuente para un objeto se especifica a través de este método que utiliza una convención se ilustra en la página de ayuda.
7. size: El tamaño de un widget se recupera o se da a través de estos métodos
8. tooltip: Una tooltip ofrece información contextual cuando un ratón pasa sobre un objeto
9. undo, redo: Algunos widgets apoyan las funciones deshacer y rehacer
10. isExtant: Un método para comprobar si todavía existe una parte de un widget.

- 11.tag: Un método utilizado para establecer los atributos de un objeto que se almacenan en un entorno de modo que se pasan por referencia, no por copia. Esto permite a los controladores de eventos manipular atributos de un objeto fuera del ámbito del callback.
- 12.getToolkitWidget: Devuelve el conjunto de herramientas del objeto que subyacen empaquetados en un objeto gWidgets2
- 13.add: Método utilizado para añadir un componente secundario de un contenedor primario
- 14.delete: Método utilizado para eliminar un componente de su contenedor
- 15.dispose: Método utilizado para eliminar un componente
- 16.dim: Se utiliza para volver fila y tamaño de la columna de información según corresponda.
- 17.names: Se utiliza para definir los nombres asociados a un objeto. Estos pueden ser nombres de columna en el widget table, o nombres de ficha en el contenedor portátil.
- 18.dimnames: Se utiliza para definir nombres de fila y columna, según corresponda.
- 19.update: Llamada para actualizar el estado de un widget.

### iv) Manejadores de eventos

1. addHandlerChanged: Asigna un handler y lo ejecuta cuando el objeto cambia
2. addHandlerClicked: Asigna un handler y lo ejecuta cuando se le hace click al objeto
3. addHandlerDoubleClick: Asigna un handler y lo ejecuta cuando se le hace doble click al objeto
4. addHandlerRightclick: Asigna un handler y lo ejecuta cuando se le hace click derecho al objeto
5. addHandlerColumnClicked: Asigna un handler y lo ejecuta cuando se le hace click a una columna del objeto
6. addHandlerColumnDoubleClicked: Asigna un handler y lo ejecuta cuando se le hace doble click a una columna del objeto
7. addHandlerColumnRightClicked: Asigna un handler y lo ejecuta cuando se le hace click derecho a una columna del objeto



8. `addHandlerSelect`: Asigna un handler y lo ejecuta cuando se selecciona el objeto
9. `addHandlerFocus`: Asigna un handler y lo ejecuta cuando se el objeto recibe un focus
10. `addHandlerBlur`: Asigna un handler y lo ejecuta cuando el objeto recibe un blur
11. `addHandlerDestroy`: Ejecuta un handler cuando el objeto es destruido
12. `addHandlerUnrealize`: Para un `gwindow` esta función es llamada antes de la destrucción del objeto y puede prevenir que eso suceda
13. `addHandlerExpose`: Asigna un handler y lo ejecuta cuando el objeto es expuesto
14. `addHandlerKeystroke`: Asigna un handler y lo ejecuta cuando un evento sobre el identificador ocurre
15. `addHandlerMouseMotion`: Asigna un handler y lo ejecuta cuando el ratón pasa sobre el objeto
16. `addHandler`: Método base para asignar un handler
17. `addHandlerIdle`: Método para asignar un handler por un determinado tiempo
18. `addPopupMenu`: Agrega un menú popup
19. `add3rdmousePopupMenu`: Agrega un menú popup para el ratón derecho
20. `addDropSource`: Especificar un widget como una ruta para activar el drag and drop
21. `addDropTarget`: Especificar un widget como una destino para activar el drag and drop
22. `addDragMotion`: Ejecuta un handler cuando un evento drag ocurre en el objeto
23. `blockHandlers`, `blockHandler`: Bloquea el handler del objeto
24. `unblockHandlers`, `unblockHandler`: Desbloquea el handler del objeto
25. `removeHandler`: Remueva el handler del objeto

### 2.3.1.2. RHadoop

RHadoop es una colección de cinco paquetes del lenguaje R que permiten a los usuarios manipular y analizar datos con Hadoop [48].

### 2.3.1.2.1. rmr2

Corre en la parte superior de Hadoop, este paquete permite definir y ejecutar trabajos de MapReduce, incluida la especificación del mapper y el reducer como funciones de investigación, y para mover datos entre R y Hadoop de una manera casi transparente. El objetivo es hacer que la escritura de los trabajos map y reduce sea muy similar y tan fácil como escribir un lapply y tapply. Las características adicionales proporcionan composición de trabajo fácil, gestión de resultado intermedio transparente, soporte para diferentes formatos de datos y mucho más. [17]

Especificando la siguiente configuración se puede utilizar el paquete sin estar en una plataforma Hadoop

```
rmr.options(backend="local")
```

### 2.3.1.2.2. rhdfs

El paquete rhdfs suministra las funciones para interactuar con un sistema de archivos distribuido Hadoop desde dentro R. Hay funciones para la gestión del sistema de archivos, así como funciones para leer, escribir, abrir, y cerrar archivos. [18]

Sus principales funciones son: hdfs.copy, hdfs.move, hdfs.rename, hdfs.put, hdfs.get, hdfs.file, hdfs.write, hdfs.close, hdfs.flush, hdfs.read, hdfs.seek, hdfs.tell, hdfs.defaults, hdfs.ls, hdfslist.files, hdfs.delete, hdfs.rm, hdfs.del, hdfs.dircreate, hdfs.mkdir, hdfs.chmod, hdfs.chown, hdfs.file.info, hdfs.exists, hdfs.init, hdfs.line.reader, hdfs.read.text.file.

## 2.3.2. Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. [32]

### 2.3.3. Python

Python es un lenguaje de programación orientado a objetos claro y potente, comparable a Perl, Ruby, Scheme o Java. [35]

Python tiene un gran soporte para aplicaciones de Ciencias de Datos, especialmente con librerías como NumPy/SciPy, Pandas, Scikit-learn, IPython para un análisis exploratorio y Matplotlib para visualizaciones.[40]

Python para el análisis de de Big Data se enfoca en la manipulación, procesamiento y limpieza de los datos. Se apoya en algunas librerías como PyDooop y SciPy. [41]

#### 2.3.3.1. SciPy

Scipy es un software de código abierto para matemáticas, ciencias e ingeniería. Incluye módulos para estadísticas, optimización, integración, álgebra lineal, transformaciones de Fourier, procesamiento de imágenes, y más [49]. A continuación se listan algunos de sus módulos [50]:

1. NumPy: Provee una manipulación rápida y conveniente de arreglos n-dimensionales
2. SciPy library: Librería fundamental para la computación científica
3. Matplotlib: Maneja gráficos comprensibles de dos dimensiones
4. IPython: Consola interactiva
5. SymPy: Provee simbología matemática
6. Pandas: Provee rutinas para la estructuración y análisis de los datos

## 2.4. Apache Hadoop

Antes de hablar de Apache Hadoop primero se debe mencionar que es Apache. Apache Software Foundation (ASF) es una organización no lucrativa creada para dar soporte a los proyectos de software bajo la denominación Apache. Apache Software Foundation es una comunidad descentralizada de desarrolladores que trabajan cada uno en sus propios proyectos de código abierto. [39]

## Capítulo 2. Marco Conceptual

Los proyectos Apache se caracterizan por un modelo de desarrollo basado en el consenso y la colaboración y en una licencia de software abierta y pragmática. Cada proyecto es gestionado por un grupo autoseleccionado de expertos técnicos que son participantes activos en dicho proyecto. [39]

Entre los objetivos de la ASF se encuentra el proporcionar protección legal a los voluntarios que trabajan en proyectos Apache, y al propio nombre Apache de ser empleado por otras organizaciones. El proyecto Apache es el origen de la licencia Apache y de todas las licencias que siguen un esquema similar. [39]

Apache Hadoop es un framework de software que soporta aplicaciones distribuidas bajo una licencia libre. Fue creado por Doug Cutting. Tiene sus orígenes en Apache Nutch, el cual es un motor de búsqueda en la web de código abierto. [24]

Nutch se inició en 2002. Sin embargo, se dieron cuenta de que su arquitectura no escalaría a los miles de millones de páginas en la Web. La ayuda estaba a la mano con la publicación de un artículo en 2003 que describe la arquitectura del sistema de archivos distribuido de Google, llamado GFS, que estaba siendo utilizado en producción en Google. GFS, o algo parecido, podría resolver las necesidades de almacenamiento para los archivos muy grandes que se generan como parte del proceso de indexación y rastreo web. En particular, GFS liberaría tiempo que se gasta en tareas administrativas, tales como la gestión de nodos de almacenamiento. En 2004, se empezó a escribir una implementación de código abierto, el Sistema de archivos distribuidos Nutch (NDFS). [24]

En 2004, Google publicó el documento que presentó MapReduce para el mundo. Temprano en 2005, los desarrolladores de Nutch tenían una implementación de MapReduce trabajando en Nutch, y a mediados de ese año. Todos los algoritmos principales Nutch habían sido adecuados para ejecutarse utilizando MapReduce y NDFS. [24]

NDFS y la implementación de MapReduce en Nutch eran aplicables más allá del ámbito de búsqueda, y en febrero de 2006 los desarrolladores se mudaron de Nutch para formar un subproyecto independiente de Lucene llamado Hadoop. Casi al mismo tiempo, Doug Cutting se unió a Yahoo!, la que proporcionó un equipo dedicado y los recursos para convertir a Hadoop en un sistema que corrió a escala web. Este se demostró en febrero de 2008,

## Capítulo 2. Marco Conceptual

cuando Yahoo! anunció que su búsqueda de índices en producción fue generada por 10000 núcleos de un clúster Hadoop. [24]

En enero de 2008, Hadoop se hizo su propio proyecto de nivel superior en Apache, lo que confirma su éxito y su diversa comunidad activa. En ese momento, Hadoop estaba siendo utilizado por muchas otras empresas, además de Yahoo!, como Last.fm, Facebook, y el New York Times. [24]

En abril de 2008, Hadoop rompió un récord mundial al convertirse en el sistema más rápido para ordenar una terabyte de datos. Ejecutándose en un clúster de 910 nodos, Hadoop ordenó un terabyte en 209 segundos, superando al ganador de 297 segundos del año anterior. [24]

Desde entonces, Hadoop ha visto una rápida adopción de las empresas dominantes. El papel de Hadoop como una plataforma de almacenamiento y análisis de propósito general para grandes volúmenes de datos ha sido reconocido por la industria, y este hecho se refleja en el número de productos que utilizan o incorporan Hadoop de alguna manera. Hay distribuciones de Hadoop de las grandes empresas establecida, incluyendo EMC, IBM, Microsoft y Oracle, así como de empresas especialistas como Cloudera Hadoop, Hortonworks y MapR. [24]

El nombre Hadoop proviene del nombre que le dio su hijo a un elefante de juguete. [24]

Permite a las aplicaciones trabajar con miles de nodos y petabytes de datos. Hadoop se inspiró en los documentos de Google para MapReduce y Google File System (GFS). [24]

### **2.4.1. Common**

Son un conjunto de librerías que soportan varios subproyectos de Hadoop. [27]

### **2.4.2. Hadoop Distributed File System (HDFS)**

Hadoop viene con un sistema de archivos distribuidos llamados HDFS, siglas de Hadoop Distributed Filesystem o en español como Sistema de Archivos Distribuidos de Hadoop. [24]

## Capítulo 2. Marco Conceptual

El Sistema de archivos distribuido Hadoop (HDFS) está diseñado para soportar aplicaciones como trabajos MapReduce que leen y escriben grandes cantidades de datos en lotes, en lugar de más accesos aleatorios a un montón de archivos pequeños. Sólo se puede escribir en un archivo una vez en el tiempo de la creación, para que sea más fácil de manejar problemas de coherencia cuando los datos se alojan en un clúster de máquinas, por lo que las copias del archivo en caché se pueden leer en cualquiera de las máquinas que tienen uno, sin tener que comprobar si el contenido ha cambiado. El software cliente almacena los datos escritos en un archivo local temporal, hasta que haya suficiente para llenar un bloque completo HDFS. Todos los archivos son almacenados en estos bloques, con un tamaño predeterminado de 64 MB. Una vez que suficientes datos se hayan guardado, o la operación de escritura este cerrada, los datos locales se envían a través de la red y son escritos a varios servidores del clúster, para asegurar que no se pierde si hay una falla de hardware.

Para simplificar la arquitectura, HDFS utiliza un único nodo maestro o namenode para tener un rastreo sobre que archivos son almacenados y en dónde. Esto quiere decir que hay un único punto de fallo y el rendimiento potencial a un cuello de botella. La intervención manual necesaria para un fallo de namenode puede ser un dolor de cabeza para el mantenimiento del sistema. [25]

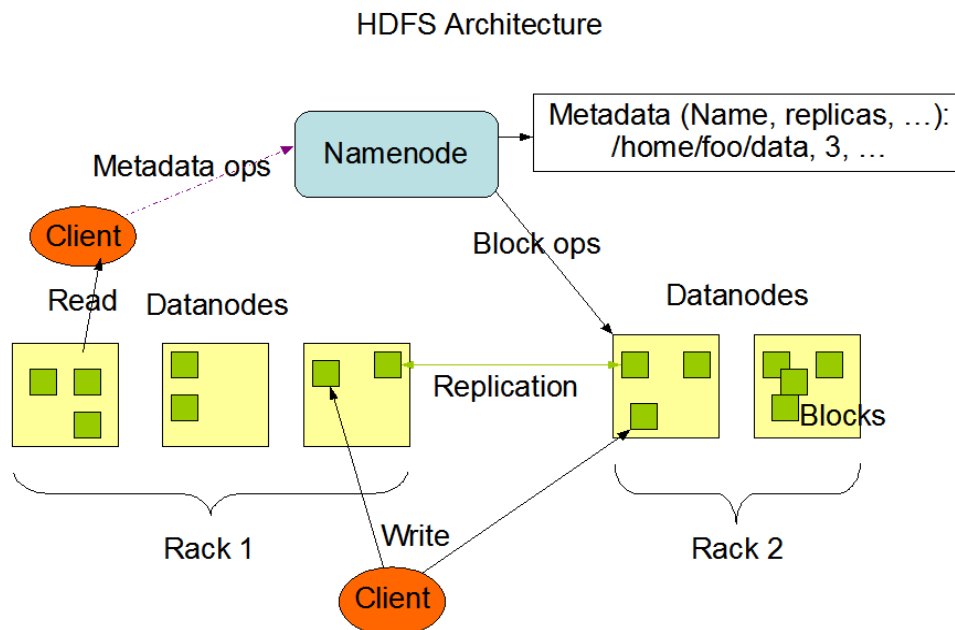


Figura 15: Arquitectura HDFS

### 2.4.2.1. Conceptos

**Bloques:** HDFS tiene el concepto de un bloque, que a diferencia de otros sistemas de archivos la unidad es mucho más grande. De 64 MB son los bloques por defecto. Al igual que en un sistema de archivos de un solo disco, los archivos en HDFS se rompen en chunks del tamaño de bloque, que se almacenan como unidades independientes. A diferencia de un sistema de archivos de un solo disco, un archivo en HDFS que es más pequeño que un solo bloque no ocupa el valor de un bloque completo lo cual hace que valga la pena el almacenamiento subyacente.

Los bloques HDFS son más grandes en comparación con los bloques de disco, y la razón es para minimizar el costo de búsqueda. Al hacer un bloque lo suficientemente grande, el tiempo para transferir los datos desde el disco puede ser significativamente más largo que el tiempo de búsqueda al principio del bloque. Así, el tiempo para transferir un archivo grande hecho de múltiples bloques opera a la velocidad de transferencia de disco. [24]

**Namenodes y datanodes:** Un clúster HDFS tiene dos tipos de nodos que operan en un patrón maestro esclavo: un namenode (el maestro) y un número de datanodes (esclavos). El namenode gestiona el espacio de nombres del sistema de archivos. Él mantiene el árbol de archivos y los metadatos de todos los archivos y directorios en el árbol. Esta información se

## Capítulo 2. Marco Conceptual

almacena persistentemente en el disco local en la forma de dos archivos: la imagen de espacio de nombres y el registro de ediciones. El namenode también sabe de los datanodes que se encuentran todos los bloques de un archivo dado; sin embargo, no conoce la ubicación de manera persistente, ya que esta información se reconstruye a partir los datanodes cuando el sistema se inicia.

Un cliente tiene acceso al sistema de archivos en nombre del usuario mediante la comunicación con el namenode y datanodes. El cliente presenta una interfaz de sistema de archivos similar a una Portable Operating System Interface (POSIX), por lo que el código de usuario no necesita saber sobre el funcionamiento del namenode y del datanode.

Los datanodes son los caballos de batalla del sistema de archivos. Ellos almacenan y recuperan bloques cuando se les dice que lo haga (los clientes o el namenode), y que informen a el namenode periódicamente con las listas de bloques que están almacenando.

Sin el namenode, el sistema de ficheros no se puede utilizar. De hecho, si la máquina está en marcha y el namenode fue borrado, todos los archivos en el sistema de archivos se perderían ya que no habría manera de saber cómo reconstruir los archivos de los bloques en los datanodes. Por esta razón, es importante hacer que el namenode sea resistente a fallas, y Hadoop proporciona dos mecanismos para ello.

La primera manera es hacer una copia de seguridad de los archivos que componen el estado persistente de los metadatos del sistema de archivos. Hadoop se puede configurar de modo que el namenode escribe su estado persistente a múltiples sistemas de ficheros. Estas escrituras son sincrónicas y atómicas. La opción de configuración habitual es escribir en el disco local, así como un mando a distancia de montaje NFS.

También es posible ejecutar un namenode secundario o sustituto, que a pesar de su nombre no actúa como un namenode. Su función principal es la de integrar periódicamente la imagen de espacio de nombres con el registro de ediciones para prevenir que el registro de ediciones se haga demasiado grande. El namenode secundario generalmente se ejecuta en una máquina física independiente, ya que requiere un montón de CPU y tanta memoria como el namenode principal para realizar la combinación. Mantiene una copia de la imagen del espacio de nombres fusionada, que se puede utilizar en el caso de fallo del namenode. Sin embargo, el estado del



namenode secundario queda por detrás de la primaria, por lo que en caso de fallo total del principal, la pérdida de datos es casi seguro. El curso normal de la acción, en este caso es copiar los archivos de metadatos del namenode que están en NFS al secundario y ejecutarlo como el nuevo principal. [24]

### 2.4.3. MapReduce

Hadoop es el sistema público más conocido para el funcionamiento de los algoritmos de MapReduce, pero muchas bases de datos modernas, tales como MongoDB, también apoyan este patrón. Es muy eficaz incluso en un sistema bastante tradicional, ya que si se puede escribir su consulta en una forma MapReduce, será capaz de ejecutar de manera eficiente en tantas máquinas como se tenga disponible. [7]

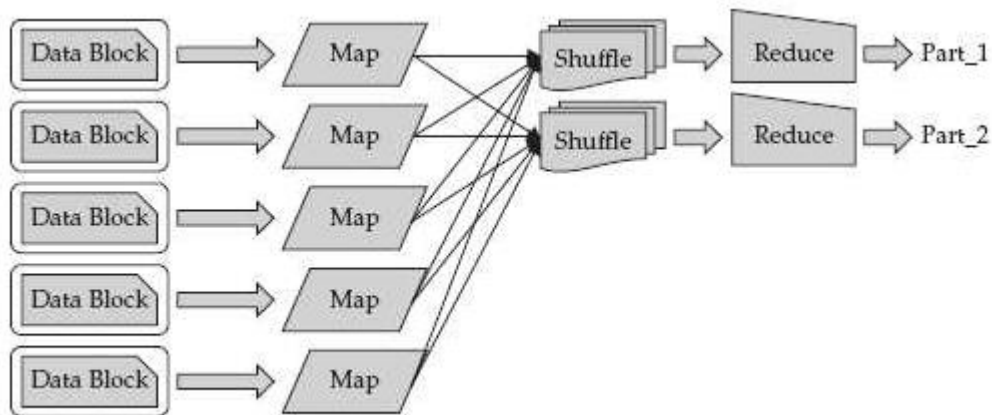


Figura 16: Ejemplo de MapReduce

Los trabajos Hadoop MapReduce se dividen en un conjunto de tareas map y tareas reduce las cuales se ejecutan de una manera distribuida en un clúster de ordenadores. Cada tarea trabaja en un pequeño subconjunto de los datos que ha sido asignado de manera que la carga se distribuye a través del clúster. Las tareas map generalmente cargan, analizan, transforman, y filtran los datos. Cada tarea reduce es responsable de la manipulación de un subconjunto de la salida de la tarea map. Los datos intermedios son copiados de las tareas map por las tareas reducir con el fin de agrupar y agregar los datos. Es increíble como una amplia gama de problemas se puede resolver con un paradigma tan directo, de agregaciones numéricas simples a complejas operaciones de combinación y productos cartesianos.

## Capítulo 2. Marco Conceptual

La entrada a un trabajo MapReduce es un conjunto de archivos en el almacén de datos que se transmiten a lo largo del Sistema de Archivos Distribuidos de Hadoop o “Hadoop Distributed File System” (HDFS). En Hadoop, estos archivos se dividen con una entrada formato, que define cómo separar un archivo en divisiones de entrada. Una partición de la entrada es una vista orientada a byte de una parte del archivo a ser cargado por una tarea map.

Cada tarea map en Hadoop se divide en las siguientes fases: record reader, mapper, combiner, y partitioner. La salida de las tareas map, son llamadas claves intermedias y valores, las cuales se envían a las tareas reducir. Las tareas reducir se dividen en las siguientes fases: shuffle, sort, reducir, y output format. Los nodos en los que las tareas map se ejecutan de manera óptima son los nodos en los que los datos se apoyan. De esta manera, los datos normalmente no se tienen que mover por la red y se puede calcular en la máquina local.

**Record reader:** El record reader traduce una partición de entrada generado por el formato de entrada en los registros. El propósito del lector de registro es para analizar los datos en los registros, pero no analiza el registro en sí. Se pasa los datos al mapper en la forma de un par clave/valor. Por lo general, la clave en este contexto es la información de posición y el valor es el fragmento o chunk de datos que compone un registro. [26]

**Map:** En el mapper, el usuario proporciona un código que se ejecuta en cada par clave/valor del record reader para producir cero o más pares clave/valor, llamados pares intermedios. La decisión de cuál es la clave y el valor que aquí no es arbitraria y es muy importante lo que está logrando el trabajo MapReduce. La clave está dada de manera que los datos se agrupan en base a ella y el valor es la información pertinente para el análisis en el reducir. [26]

**Combiner:** El combiner, un reducer opcional localizado, pueden agrupar los datos en la fase map. El combiner toma las claves intermedias desde el mapper y aplica un método proporcionado por el usuario para agregar valores en el pequeño ámbito de un mapper. Por ejemplo, debido a que el recuento de una agregación es la suma de los cargos de cada parte, se puede producir un recuento intermedio y luego sumar esos recuentos intermedios para el resultado final. En muchas situaciones, esto reduce significativamente la cantidad de datos que tienen que moverse más en red.

## Capítulo 2. Marco Conceptual

Envío (hola mundo, 3) requiere menos bytes que envía (helloworld, 1) tres veces a lo largo de la red. Muchos de los nuevos desarrolladores de Hadoop ignoran la fase combiner, pero a menudo proporcionan mejoras de rendimiento extremas con ningún inconveniente. Se puede señalar que los patrones se benefician del uso de combiner, y hay cuáles en los que no se puede utilizar un combiner. Un combiner no está garantizado para ejecutarse, por lo que no puede ser un parte del algoritmo general. [26]

**Partitioner:** El partitioner toma los pares clave/valor intermedios desde el mapper (o combiner si se utiliza) y los divide en fragmentos o shards, uno fragmento por reducer. De forma predeterminada, la herramienta partitioner interroga al objeto por su código hash, que es típicamente un md5sum. Entonces, el partitioner realiza una operación de módulo por el número de reducer: **key.hashCode ()% (número de reducers)**. Esto distribuye aleatoriamente el espacio de claves de manera uniforme sobre los reducers, pero aún asegura que las claves con el mismo valor en diferentes mappers terminen en el mismo reducer. El comportamiento por defecto de la herramienta partitioner se puede personalizar, y estará en algunos patrones más avanzados, tales como el sorting. Sin embargo, el cambio de la herramienta partitioner rara vez es necesario. Los datos particionados se escriben en el sistema de archivos local de cada tarea map y espera a ser pulled por su respectivo reducer. [26]

**Shuffle and sort:** La tarea reduce comienza con la etapa shuffle and sort. Este paso toma los archivos de salida escrito por todos los partitioners y lo descarga en el equipo local en el que el reducer se está ejecutando. Estas piezas individuales de datos se ordenan por llave en una lista de datos más grande. El propósito de este tipo es agrupar las llaves equivalentes juntas para que sus valores puedan iterarse más fácilmente en la tarea reduce. Esta fase no es adaptable y el framework maneja todo automáticamente. El único control que tiene un desarrollador es cómo se clasifican y agrupan las claves especificando una costumbre **Comparator object**. [26]

**Reduce:** El reducer toma los datos agrupados como entrada y ejecuta una función reduce una vez por agrupación clave. La función se le pasa la clave y un iterador sobre todos los valores asociados a la clave pasada. Una amplia gama de procesamientos pueden ocurrir en esta función. Los datos pueden ser agregados, filtrados, y combinados en un número amplio de maneras. Una vez que la función reduce se haga, se envían cero o más clave/valor par a la etapa final, el output format o formato de salida. Al igual

que la función map, la función reduce cambiará de un trabajo a otro, ya que es una pieza central de la lógica en la solución. [26]

**Output format:** El formato de salida o output format traduce el par clave/valor final de la función reduce y lo escribe en un archivo de un registro escritor. Por defecto, se separará la clave y el valor en una ficha de registros separados con un carácter de nueva línea. Esto puede ser típicamente personalizado para ofrecer formatos de salida más entendibles, pero al final, los datos se escriben en el HDFS, independientemente del formato.[26]

### 2.4.3.1. Hadoop Streaming

Hadoop streaming es un componente que viene con las distribuciones de Hadoop. La utilidad le permite crear y ejecutar trabajos map / reduce con cualquier ejecutable o script. [19] Por ejemplo:

```
$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/hadoop-streaming.jar \  
-input myInputDirs \  
-output myOutputDir \  
-mapper /bin/cat \  
-reducer /bin/wc
```

En el ejemplo anterior, tanto el map como el reduce son ejecutables que leen la entrada por entrada estándar (línea por línea) y emiten la salida a como salida estándar. El componente creará un trabajo Map/Reduce, enviará la tarea a un clúster apropiado y monitoreará el progreso del trabajo hasta que se complete.

Parametro	Requerido	Descripción
-input	Requerido	Localización de la entrada del mapper
-output	Requerido	Localización de la salida del reducer
-mapper	Requerido	Ejecutable Mapper
-reducer	Requerido	Ejecutable Reducer

## Capítulo 2. Marco Conceptual

-file	Opcional	Ruta donde se encuentra el ejecutable mapper o reducer
-inputformat	Opcional	Clase que especifica el tipo de entrada. Si no se especifica, TextInputFormat es usado por defecto
-outputformat	Opcional	Clase que especifica el tipo de entrada. Si no se especifica, TextOutputFormat es usado por defecto
-partitioner	Opcional	Clase que determina que clave reduce es enviada
-combiner	Opcional	Ejecutable Combiner para la salida map
-cmdenv	Opcional	Pasa variables de entorno para los comandos de streaming
-inputreader	Opcional	Por compatibilidad con versiones anteriores: especifica una clase record reader (en lugar de una clase de formato de entrada)
-verbose	Opcional	Mostrar log de salida
-lazyOutput	Opcional	Crea una salida perezosa

-numReduceTasks	Opcional	Especifica el número de reducers
-mapdebug	Opcional	Script a ser llamado cuando una tarea map falle

Tabla 1: Parametros de Hadoop Streaming

### 2.4.4. YARN

Como parte de Hadoop 2.0, YARN toma las capacidades de gestión de los recursos que estaban en MapReduce y las empaqueta para que puedan ser utilizados por los nuevos motores. Esto también simplifica MapReduce para hacer lo que mejor hace, procesar los datos. Con YARN, ahora puede ejecutar varias aplicaciones en Hadoop, todos compartiendo una gestión común de los recursos. Muchas organizaciones ya están construyendo aplicaciones sobre YARN con el fin de traerlos a Hadoop. [28]

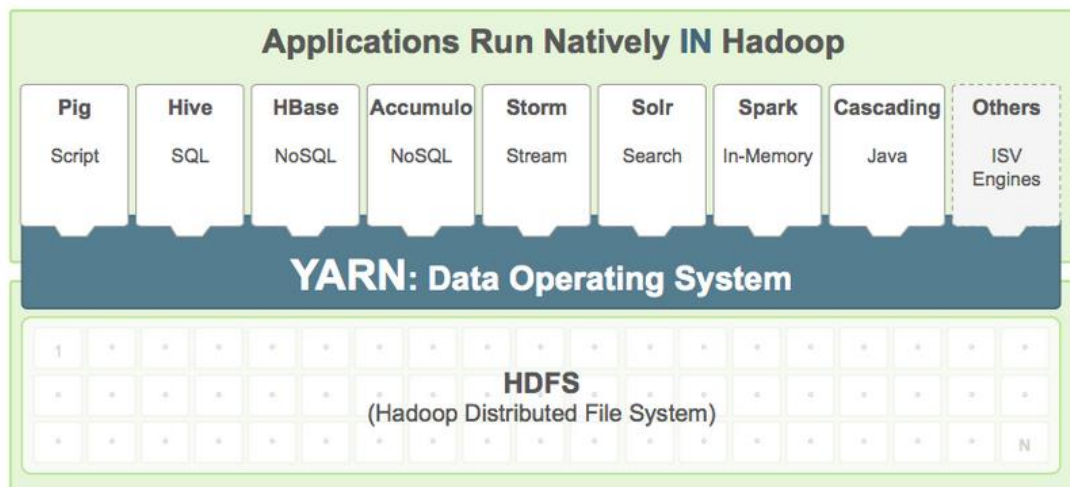


Figura 17: Arquitectura YARN de Hortonworks

YARN combina un administrador de recursos central que reconcilia la forma en que las aplicaciones utilizan los recursos del sistema de Hadoop con agentes gestores de nodos que controlan las operaciones de procesamiento de los nodos individuales del clúster. La separación de HDFS

## Capítulo 2. Marco Conceptual

y MapReduce con YARN hace que el ambiente Hadoop sea más adecuado para las aplicaciones operativas que no pueden esperar que terminen trabajos por lotes. [29]

YARN aumenta el poder de un clúster de cálculo Hadoop de las siguientes maneras:

- a. Escalabilidad: El poder de procesamiento en los centros de datos continúa creciendo rápidamente. Debido a que YARN ResourceManager se centra exclusivamente en la programación, se puede administrar esos grandes grupos con mucha más facilidad. [28]
- b. Compatibilidad con MapReduce: Aplicaciones MapReduce y usuarios existentes pueden ejecutar en la parte superior de YARN y sin interrupción a sus procesos existentes. [28]
- c. Utilización clúster mejorada: El ResourceManager es un programador puro que optimiza la utilización del clúster de acuerdo con criterios tales como garantías de capacidad, equidad y SLAs. [28]
- d. Soporte para cargas de trabajo distintas de MapReduce: Modelos de programación adicionales, tales como procesamiento gráfico y modelado iterativo ahora son posibles para el procesamiento de datos. [28]
- e. Agilidad: MapReduce se convierte en una biblioteca de espacio de usuario, que puede evolucionar de forma independiente de la capa de administrador de recursos subyacente y de una manera mucho más ágil. [28]

### ¿Cómo trabaja YARN?

La idea fundamental es la de dividir las dos principales funcionalidades del JobTracker/TaskTracker en varias entidades. La idea es tener un Manejador de Recursos global ResourceManager (RM), un ApplicationMaster (AM) por aplicación

## Capítulo 2. Marco Conceptual

El ResourceManager y el esclavo por nodo, el NodeManager (NM), forman el marco de datos de cálculo. El ResourceManager es la última autoridad que arbitra los recursos entre todas las aplicaciones en el sistema.

El ApplicationMaster es, en efecto, un framework específico de biblioteca y se encarga de la negociación de los recursos del ResourceManager y trabajar con los NodeManager para ejecutar y supervisar las tareas que lo componen.[30]

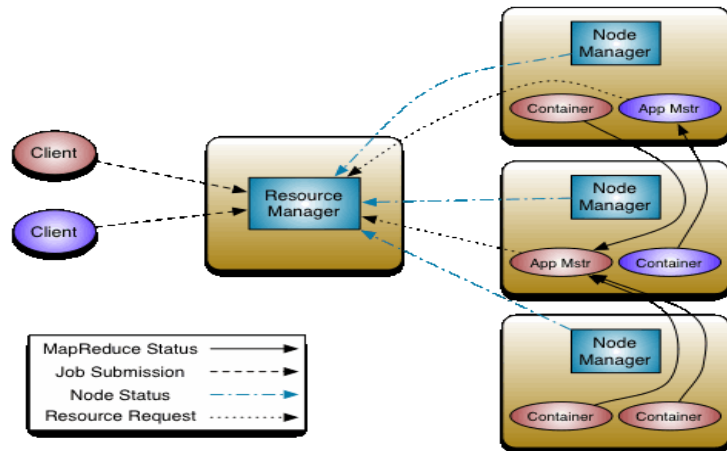


Figura 18: Proceso MapReduce YARN

El ResourceManager tiene un planificador, que es responsable de la asignación de recursos para las diversas aplicaciones en ejecución, de acuerdo con las limitaciones como las capacidades de colas, los plazos de usuario, etc. El programador realiza su función de programación basado en las necesidades de recursos de las aplicaciones. El NodeManager es responsable del lanzamiento de los contenedores de las aplicaciones, el seguimiento de su uso de los recursos (CPU, memoria, disco, red) e informar del mismo al ResourceManager. Cada ApplicationMaster tiene la responsabilidad de negociar contenedores de recursos adecuados desde el planificador, el seguimiento de su estado, y el seguimiento de su progreso. Desde la perspectiva del sistema, el ApplicationMaster se ejecuta como un recipiente normal.



### 2.4.5. Herramientas del Ecosistema Hadoop

#### 2.4.5.1. Hive

Apache Hive es el estándar de facto para las consultas SQL que tienen más de petabytes de datos en Hadoop. Es un motor integral y compatible que ofrece la gama más amplia de la semántica de SQL para Hadoop, proporcionando un potente conjunto de herramientas para los analistas y desarrolladores para acceder a los datos de Hadoop.[31]

Con Hive, se pueden programar trabajos Hadoop utilizando SQL. Es una excelente interfaz para cualquier persona que viene del mundo de bases de datos relacionales, aunque los detalles de la implementación subyacente no están completamente ocultos. El usuario todavía tiene que preocuparse acerca de algunas diferencias en cosas como la forma más óptima para especificar los joins para un mejor rendimiento y algunas características del lenguaje que faltan. Hive ofrece la posibilidad de conectar en código personalizado para situaciones que no encajan en SQL, así como una gran cantidad de herramientas para el manejo de entrada y salida. Para usarlo, debe configurar las tablas estructuradas que describen su entrada y salida, los problemas en comandos de carga para ingerir archivos y, a continuación, escribir sus consultas como lo haría en cualquier otra base de datos relacional. No se debe tener en cuenta, sin embargo, que debido al enfoque de Hadoop de procesamientos en gran escala, la latencia puede significar que incluso los trabajos más simples toman minutos para completarse, así que no es un sustituto de una base de datos transaccional en tiempo real.[7]

Hive fue creado para hacer posible que los analistas con fuertes habilidades de SQL (pero con escaso conocimientos de programación Java) puedan ejecutar consultas sobre los enormes volúmenes de datos que Facebook ha almacenado en HDFS. Hoy, Hive es un proyecto Apache exitoso utilizado por muchas organizaciones. [24]

¿Qué hace Hive?

Hadoop fue construido para organizar y almacenar grandes cantidades de datos. Un cluster Hadoop es un depósito de datos heterogéneos, provenientes de múltiples fuentes y en diferentes formatos. Hive permite al usuario explorar y estructura que los datos, analizarlos, y luego convertirlo en el conocimiento del negocio. [31]

### ¿Cómo funciona Hive?

Las tablas en Hive son similares a las tablas en una base de datos relacional, y las unidades de datos están organizadas en una taxonomía más grande que muchas unidades granulares. Las bases de datos se componen de tablas, que se componen de particiones. Los datos se pueden acceder a través de un lenguaje de consulta simple, llamado HiveQL, que es similar a SQL. Hive apoya sobrescribir o anexar datos, pero no las actualizaciones y eliminaciones. [31]

Dentro de una base de datos particular, los datos de las tablas son serializados y cada tabla tiene un directorio en el sistema de archivos distribuido Hadoop correspondiente (HDFS). Cada tabla puede estar subdividida en particiones que determinan cómo los datos se distribuyen dentro de los subdirectorios del directorio de la tabla. Los datos dentro de particiones se pueden desglosar en cubos. [31]

Estas son algunas de las características ventajosas de Hive:

- a) Familiar: Cientos de usuarios únicos pueden consultar simultáneamente los datos utilizando un lenguaje familiar para los usuarios de SQL. [31]
- b) Velocidad: Los tiempos de respuesta son normalmente mucho más rápido que otros tipos de consultas en el mismo tipo de conjuntos de datos enormes. [31]
- c) Escalable y extensible: Como la variedad de datos y el volumen crece, más máquinas de las materias primas se pueden añadir a la agrupación, sin una reducción correspondiente en el rendimiento. [31]
- d) Informativo: Controladores JDBC y ODBC familiares permiten muchas aplicaciones para extraer datos de Hive para la presentación de informes sin fisuras. Hive permite a los usuarios leer los datos en formatos arbitrarios, usando SerDes y formatos de entrada/salida. [31]

#### **2.4.5.2. Pig**

El proyecto Apache Pig es un lenguaje procedimental para el procesamiento de datos diseñado para Hadoop. En contraste con el enfoque de Hive, con Pig se especifican una serie de pasos a realizar en los datos. Está más cerca de un lenguaje de programación todos los días, pero con un

## Capítulo 2. Marco Conceptual

conjunto especializado de funciones que ayudan con problemas de procesamiento de datos comunes.[25]

Pig eleva el nivel de abstracción para el procesamiento de grandes conjuntos de datos. MapReduce permite, como el programador especifica una función de map seguida por una función de reduce, pero trabajando fuera de cómo encontrar la manera de adaptar su procesamiento de datos a este patrón, que a menudo requiere múltiples etapas de MapReduce, por ende la adaptación puede ser un desafío. Con Pig, las estructuras de datos son mucho más ricas, suelen ser múltiples valores y anidados, y el conjunto de transformaciones que se le pueden aplicar los datos son mucho más poderosas. Ellas incluyen joins.[32]

Pig se compone de dos piezas:

- a) El lenguaje para expresar flujos de datos llamado, Pig Latin.
- b) El entorno de ejecución para ejecutar los programas de Pig Latin. Actualmente hay dos entornos de ejecución: Una ejecución local en una única JVM y una ejecución distribuida en un clúster Hadoop. [32]

¿Qué hace Pig?

Pig fue diseñado para llevar a cabo una larga serie de operaciones de datos, por lo que es ideal para tres categorías de empleos Big Data:

Extracción, transformación y carga (ETL) de pipelines de datos, la investigación sobre los datos en bruto, y el procesamiento de datos iterativo. [33]

Cualquiera que sea el caso de uso, Pig será:

- a) Extensible. Los usuarios Pig pueden crear funciones personalizadas para satisfacer sus necesidades de procesamiento particulares.
- b) Fácil de programar. Las tareas complejas que implican transformaciones de datos relacionados entre sí se pueden simplificar y codificar como secuencias de flujo de datos. Los programas Pig realizan tareas enormes, pero son fáciles de escribir y mantener.
- c) Auto-optimización. El sistema optimiza automáticamente la ejecución de trabajos Pig, por lo que el usuario puede centrarse en la semántica. [33]

### ¿Cómo funciona Pig?

Pig se ejecuta en Hadoop y hace uso del sistema de archivos distribuido Hadoop (HDFS) y MapReduce. El idioma de la plataforma se llama Pig Latin, que abstrae del lenguaje Java MapReduce en una forma similar a SQL. Pig Latin es un lenguaje de flujo mientras que SQL es un lenguaje declarativo. SQL es ideal para hacer una pregunta de sus datos, mientras Pig Latin le permite escribir un flujo de datos que describe cómo se transforman los datos. Los scripts Pig Latin pueden ser gráficos (en lugar de requerir una sola salida) es posible construir flujos de datos compleja que involucren múltiples entradas, transformaciones y salidas. Los usuarios pueden ampliar Pig Latin escribiendo sus propias funciones, utilizando Java, Python, Ruby, u otros lenguajes de script. [33]

El usuario puede ejecutar Pig en dos modos:

1. Modo Local. Con acceso a una sola máquina, todos los archivos se instalan y se ejecutan utilizando un sistema anfitrión y el archivo local. [19]
2. Modo MapReduce. Este es el modo por defecto, lo que requiere el acceso a un cluster Hadoop. [33]

El usuario puede ejecutar Pig en cualquiera de los modos mediante el comando "java" o el comando "pig".

### **2.4.5.3. HCatalog**

Apache HCatalog es una capa que gestiona el almacenamiento de datos en tablas que permite a los usuarios utilizando diferentes herramientas de procesamiento leer y escribir datos de manera más fácil en la red. La abstracción que provee HCatalog en base a tablas presenta a los usuarios una visión relacional de los datos en el sistema de archivos distribuidos de Hadoop (HDFS) y además asegura a los usuarios que no deben preocuparse acerca de dónde o en qué formato se almacenan los datos. Hcatalog muestra datos provenientes desde fuentes con varios formatos en una vista tabular. También proporciona medios para que los sistemas externos puedan acceder a los datos de las tablas.[34]

### ¿Qué hace HCatalog?

## Capítulo 2. Marco Conceptual

Apache HCatalog ofrece los siguientes beneficios a los administradores de la red:

- Libera al usuario de tener que saber dónde se almacenan los datos, con la abstracción en base a tablas [34]
- Activa las notificaciones de la disponibilidad de datos [34]
- Proporciona visibilidad para herramientas de limpieza y almacenamiento de datos [34]

¿Cómo trabaja HCatalog?

HCatalog apoya la lectura y la escritura de archivos en cualquier formato para el que Hive SerDe funcione. Por defecto HCatalog soporta los siguientes formatos: RCFile, CSV, JSON y SequenceFile. Para utilizar un formato personalizado se debe proporcionar el formato de entrada, el de salida y el SerDe. [34]

HCatalog se construye en la parte superior de Hive e incorpora componentes DDL de Hive. HCatalog proporciona interfaces de lectura y escritura interfaces para Pig y MapReduce y usa la interfaz de línea de comandos de Hive para la emisión de comando para la definición y exploración de datos. También presenta una interfaz REST para permitir a las herramientas externas el acceso a las operaciones DDL (Data Definition Language) de Hive, como "create table" y "describe table". [34]

HCatalog presenta una vista relacional de datos. Los datos se almacenan en tablas y estas tablas se pueden colocar en las bases de datos. Las tablas también se pueden dividir en una o más claves. Para un valor dado de una clave (o conjunto de claves) habrá una partición que contiene todas las filas con ese valor (o conjunto de valores). [34]

### **2.4.5.4. Apache Spark**

A medida que la relación de la memoria a la potencia de procesamiento evoluciona rápidamente, muchos dentro de la comunidad

## Capítulo 2. Marco Conceptual

Hadoop están gravitando hacia Apache Spark para procesamiento rápido de datos en memoria.[35]

Apache Spark es un framework de código abierto para análisis de datos en clústeres, desarrollado originalmente en el AMPLab en UC Berkeley. [36]

Spark encaja en la comunidad de código abierto Hadoop, sobre la parte superior del sistema de archivos distribuido Hadoop (HDFS). Sin embargo, Spark no está ligado al paradigma MapReduce de dos etapas, y promete un rendimiento de hasta 100 veces más rápido que Hadoop MapReduce para ciertas aplicaciones.[36]

Spark proporciona primitivas para la computación de clústeres en memoria que permite a los programas de usuario cargar datos en un la memoria de clústeres y consultarla varias veces, por lo que es muy adecuado para los algoritmos de aprendizaje automático. [36]

Apache Spark permite a los científicos de datos implementar de manera efectiva y simple algoritmos iterativos para análisis avanzados como la agrupación y clasificación de conjuntos de datos. Actualmente es un proyecto Apache nivel superior y se está convirtiendo en una alternativa atractiva para ejecutar algunas cargas discretas de trabajo de ciencia de datos. [35]

### **2.4.5.5. Apache Flume**

Apache Flume es un servicio disponible, distribuido y confiable para la recolección, agregación y el movimiento de manera eficiente de grandes cantidades de datos en el sistema de archivos distribuidos de Hadoop (HDFS). Cuenta con una arquitectura simple y flexible basada en la transmisión de flujos de datos; es robusto y tolerante a fallos con mecanismos de confiabilidad sintonizables para la conmutación por error y recuperación. [38]

¿Qué hace Flume?

Flume permite a los usuarios de Hadoop sacar mayor provecho a los datos de registro con mayor valor. En concreto, Flume permite a los usuarios hacer:

## Capítulo 2. Marco Conceptual

1. Secuencias de datos a partir de múltiples fuentes en Hadoop para el análisis. [38]
2. Recoger registros Web de alto volumen en tiempo real. [38]
3. Aislarse de picos transitorios cuando la tasa de datos entrantes excede la tasa a la que los datos se pueden escribir en el destino. [38]
4. Garantía en la entrega de datos. [38]
5. Escala horizontal para manejar el volumen de datos adicional. [38]

### ¿Cómo funciona Flume?

La arquitectura de alto nivel de Flume se centra en la entrega de una base de código optimizada que es fácil de usar y fácil de extender. El equipo del proyecto ha diseñado Flume con los siguientes componentes:

- a) Event - una unidad singular de los datos que se transporta por el canal de flujo (por lo general una sola entrada de registro). [38]
- b) Source - la entidad a través de la cual los datos de entrada en Flume. Source o bien sondea activamente para datos o pasivamente espera que los datos sean entregados a ellos. [38]
- c) Sink - la entidad que entrega los datos al destino. [38]
- d) Channel - el conducto entre el Source y el Sink. Source ingieren Events en el Channel y los Sinks drenan el Channel. [38]
- e) Agent - cualquier máquina virtual física Java que ejecute Flume. Se trata de una colección de Source, Sink y Channel. [38]
- f) Client - produce y transmite el Event a Source que opera dentro del Agent. [38]

Un flujo de Flume se inicia desde el Client. El Client transmite el Event a una Source que opera dentro del Agent. La Source de recibir este Event lo entrega a uno o más Channels. Estos Channels son drenados por uno o más Sinks que operan dentro del mismo Agent. [38]

### **2.4.5.6. Hue**

Hue es una interfaz Web para analizar datos con Apache Hadoop. Es compatible con un explorador de archivos y trabajos, Hive, Pig, Impala,

## Capítulo 2. Marco Conceptual

Spark, editores oozie, Solr Search cuadros de mando, HBase, Sqoop2, y más.[39]

Hue cuenta con:

- Explorador de archivos para acceder a HDFS
- Editor Hive para desarrollar y ejecutar consultas Hive
- Search App para consultar, explorar, visualizar datos y cuadros de mando con Solr
- Impala App para ejecutar consultas SQL interactivas
- Spark Editor y Dashboard
- Editor Pig para la presentación de scripts Pig
- Editor Oozie y Dashboard para la presentación y seguimiento de flujos de trabajo, de coordinadores y de los paquetes
- HBase Browser para visualizar, consultar y modificar tablas HBase
- MetaStore Browser para acceder a los metadatos de Hive y HCatalog
- Navegador de empleo para acceder a puestos de trabajo de MapReduce (MR1/MR2-YARN)
- Diseñador de trabajos para la creación de trabajos MapReduce/Streaming/Java
- Un editor y dashboard para Sqoop 2
- Un navegador y editor ZooKeeper
- Un editor de consultas DB para MySql, Postgres, SQLite y Oracle

### 2.4.5.7. Apache ZooKeeper

Escribir aplicaciones distribuidas es difícil. Es complicado principalmente debido a las fallas parciales. Cuando es enviado un mensaje a través de la red entre dos nodos y la red falla, el remitente no sabe si el receptor recibió el mensaje. La única manera de que el emisor pueda saber si se recibió el mensaje o no es reconectarse con el receptor y preguntarle. Este es el fracaso parcial, cuando no se sabe si una operación ha fallado. [40]

Apache ZooKeeper es un servidor de código abierto que coordina de forma fiable procesos distribuidos. [40]

Apache ZooKeeper ofrece servicios operativos para un clúster Hadoop. ZooKeeper ofrece un servicio de configuración distribuida, un



## Capítulo 2. Marco Conceptual

servicio de sincronización y un registro de denominación para los sistemas distribuidos. Las aplicaciones distribuidas utilizan Zookeeper para almacenar y mediar cambios a la información de configuración importante. [40]

ZooKeeper no puede hacer que las fallas parciales se vayan, ya que son intrínsecos al sistema distribuido. Ciertamente tampoco esconde los fracasos parciales. Pero lo que hace ZooKeeper es proporcionar un conjunto de herramientas para construir aplicaciones distribuidas que puedan manejar de forma segura los fracasos parciales. [40]

ZooKeeper también tiene las siguientes características:

ZooKeeper es simple. ZooKeeper es, en su esencia, un sistema de archivos que expone algunas simples operaciones, y algunas abstracciones adicionales, tales como ordenaciones y notificaciones. [40]

ZooKeeper es expresivo. Las primitivas ZooKeeper son un rico conjunto de bloques de construcción que se puede utilizar para construir una gran clase de estructuras de datos y protocolos de coordinación. Los ejemplos incluyen: colas distribuidas, cerraduras distribuidas y elección de un líder entre un grupo de compañeros. [40]

ZooKeeper es altamente disponible. ZooKeeper se ejecuta en una colección de máquinas y está diseñado para ser altamente disponible, por lo que las aplicaciones pueden depender de él. [40]

ZooKeeper facilita las interacciones débilmente acopladas. Las interacciones de ZooKeeper apoyan a los participantes que no necesitan saber acerca de otros. Por ejemplo, ZooKeeper se puede utilizar como un mecanismo de encuentro de manera que los procesos que no saben de la existencia de otros (o detalles de la red) puedan descubrir e interactuar con los demás. Las partes de coordinación pueden incluso no ser contemporáneas, ya que un proceso puede dejar un mensaje en ZooKeeper el cual puese ser leído por otro después de que el primero se haya apagado. [40]

ZooKeeper es una biblioteca. ZooKeeper proporciona un repositorio compartido de implementaciones y recetas de patrones de coordinación comunes, de código abierto. Los programadores individuales se esparcen la carga de la escritura protocolos comunes entre ellos mismos (que a menudo son difíciles de conseguir de manera correcta). Con el tiempo, la comunidad

## Capítulo 2. Marco Conceptual

puede aumentar y mejorar las bibliotecas, que es para beneficio de todos. [40]

ZooKeeper es de gran rendimiento también. En Yahoo!, donde fue creado, el rendimiento para un clúster ZooKeeper se ha evaluado en más de 10.000 operaciones por segundo para cargas de trabajo de escritura dominante generados por cientos de clientes. Para cargas de trabajo donde la lectura domina, que es la norma, el rendimiento es varias veces alto. [40]

¿Qué hace ZooKeeper?

ZooKeeper proporciona una interfaz muy simple y servicios. ZooKeeper trae los siguientes beneficios clave:

- a) Rápido. ZooKeeper es especialmente rápido con cargas de trabajo donde la lectura de los datos es más común que la escritura. La proporción ideal de lectura/escritura es de aproximadamente 10: 1. [40]
- b) Fiable. ZooKeeper se replica a través de una serie de hosts (llamado un conjunto) y los servidores son conscientes unos de otros. Mientras una masa crítica de servidores está disponible, el servicio ZooKeeper también estará disponible. No hay ningún punto único de fallo. [40]
- c) Sencillo. ZooKeeper mantiene un espacio de nombres jerárquico estándar, similar a los archivos y directorios.
- d) Ordenado. El servicio mantiene un registro de todas las transacciones, que se pueden utilizar para abstracciones de nivel superior, como primitivas de sincronización.

¿Cómo trabaja ZooKeeper?

ZooKeeper permite procesos distribuidos para coordinar entre sí a través de un espacio de nombres jerárquico compartido de registros de datos, conocidos como znodes. Cada znode se identifica por un camino, con elementos de ruta separados por una barra ("/"). Aparte de la raíz, cada znode tiene un padre y un znode no puede ser eliminado si tiene hijos. [40]

Esto es muy similar a un sistema de archivos normal, pero ZooKeeper ofrece una fiabilidad superior a través de servicios redundantes. Un servicio

se replica a través de una serie de máquinas y cada uno mantiene una imagen en memoria de los árboles de los datos y de transacciones. Los clientes se conectan a un único servidor ZooKeeper y mantienen una conexión TCP a través del cual envían peticiones y reciben respuestas. [40]

Esta arquitectura permite a ZooKeeper proporcionar un alto rendimiento y disponibilidad con baja latencia, pero el tamaño de la base de datos que puede gestionar ZooKeeper está limitado por la memoria. [40]

### **2.4.5.8. Shark**

Shark es un sistema de almacenamiento de datos a gran escala para Spark diseñado para ser compatible con Apache Hive. Puede ejecutar consultas Hive QL hasta 100 veces más rápido que Hive sin ninguna modificación a los datos o consultas existentes. Shark soporta el lenguaje de consulta Hive, MetaStore, formatos de serialización, y funciones definidas por el usuario, proporcionando una integración perfecta con los despliegues Hive existentes y una opción familiar, más potente para los nuevos. [41]

Shark se construye en la parte superior de Spark, un motor de ejecución de datos en paralelo que es rápido y de alta disponibilidad. Incluso si los datos están en el disco, Shark puede ser notablemente más rápido que Hive debido al motor de ejecución rápida. Evita la tarea lanzamiento de alta sobrecarga de Hadoop MapReduce y no requiere la materialización de datos intermedios entre las etapas en el disco. Gracias a este motor rápido, Shark puede responder a las consultas en menos de un segundo de latencia. [41]

Las consultas analíticas por lo general se centran en un subgrupo en particular o en una ventana de tiempo, por ejemplo, los registros de HTTP desde el mes anterior, tocando solamente las tablas de dimensiones y una pequeña porción de la tabla de hechos. Estas consultas tienen una fuerte localidad temporal, y en muchos casos, es plausible para encajar el conjunto de trabajo en la memoria de un clúster.

Shark permite a los usuarios explotar esta localidad temporal mediante el almacenamiento de su conjunto de trabajo de datos a través de la memoria de un grupo, o en términos de base de datos, para crear en memoria vistas materializadas.[41]

### 2.4.5.9. Apache Sqoop

Apache Sqoop es una herramienta diseñada para transferir datos de manera eficiente a granel entre Hadoop y almacenes de datos estructurados como bases de datos relacionales. Sqoop importa datos de almacenes estructurados externos dentro de los sistemas HDFS o cualquier medio de almacenamiento que se utilice, como Hive y HBase. Sqoop también se puede utilizar para extraer datos de Hadoop y exportarlo a almacenes estructurados externos tales como bases de datos relacionales y almacenes de datos empresariales. Sqoop trabaja con bases de datos relacionales, tales como: Teradata, Netezza, Oracle, MySQL, Postgres, y HSQLDB. [42]

¿Qué hace Sqoop?

Está diseñado para transferir datos de manera eficiente a granel entre Hadoop y almacenes estructurados como bases de datos relacionales, Apache Sqoop:

- Permite la importación de datos de almacenes externos y almacenes de datos empresariales en Hadoop [42]
- Paraleliza la transferencia de datos para un rendimiento rápido y la utilización óptima del sistema [42]
- Copia datos rápidamente de sistemas externos a Hadoop [42]
- Hace un análisis de datos más eficiente [42]
- Mitiga cargas excesivas a sistemas externos. [42]

¿Cómo trabaja Sqoop?

Sqoop proporciona un mecanismo conector enchufable para una óptima conectividad a sistemas externos. La API de extensión de Sqoop proporciona un marco conveniente para la construcción de nuevos conectores que pueden ser ignorados en las instalaciones de Sqoop para proporcionar conectividad a varios sistemas. Sqoop sí viene incluido con varios conectores que pueden ser utilizados para los sistemas de base de datos y almacenes de datos populares. [42]

### **2.4.6. Distribuciones Hadoop**

La arquitectura flexible y modular de hadoop permite añadir nuevas funcionalidades para la realización de diversas tareas de Big Data. Un número de vendedores han aprovechado el framework de composición abierta de Hadoop ajustando sus códigos para cambiar o mejorar sus funcionalidades. En el proceso han sido capaces de solucionar algunos de los inconvenientes inherentes de Hadoop. En lo que se refiere a las distribuciones de Hadoop, las tres empresas que realmente se destacan en la terminación son: Cloudera, MapR y Hortonworks. [20]

Cloudera ha estado presente por mucho más tiempo, desde la creación de Hadoop. Hortonworks vino después. Mientras Cloudera y Hortonworks son 100 por ciento de código abierto, la mayoría de las versiones de MapR vienen con módulos propietarios. [20]

#### **2.4.6.1. MapR**

MapR reemplaza el componente HDFS y en su lugar utiliza su propio sistema de archivo propietario, llamado MapRFS. MapRFS ayuda a incorporar características de nivel empresarial en Hadoop, lo que permite una gestión más eficiente de los datos, fiabilidad y lo más importante, la facilidad de uso. [20]

A través de una alianza con Canonical, el creador del sistema operativo Ubuntu, MapR está ofreciendo Hadoop como un componente predeterminado del sistema operativo Ubuntu. Bajo los términos de la alianza, la edición M3 de MapR para Hadoop se integrará en el sistema operativo Ubuntu. [20]

#### **2.4.6.2. Cloudera**

Cloudera Inc. fue fundada por los genios de Big Data de Facebook, Google, Oracle y Yahoo en 2008. Fue la primera empresa en desarrollar y distribuir software basado en Apache Hadoop y todavía tiene la mayor base de usuarios con mayor número de clientes. Aunque el núcleo de la distribución está basado en Hadoop, también proporciona una Suite de

gestión propietaria llamada Management Suite Cloudera para automatizar el proceso de instalación y proporcionar otros servicios para mejorar la comodidad de los usuarios, que incluyen la reducción de tiempo de implementación, mostrando recuento nodos en tiempo real, etc. [20]

### **2.4.6.3. Hortonworks**

Hortonworks, fundada en 2011, se ha convertido rápidamente en uno de los principales proveedores de Hadoop. La distribución proporciona la plataforma de código abierto basado en Hadoop para analizar, almacenar y gestionar grandes volúmenes de datos. Hortonworks es el único proveedor comercial para distribuir código abierto completamente Hadoop sin software propietario adicional. La distribución 2.0 de Hadoop Data Platform (HDP2.0) de Hortonworks se puede descargar directamente desde su página web de forma gratuita y es fácil de instalar. Los ingenieros de Hortonworks están detrás de la mayoría de las innovaciones recientes de Hadoop incluido YARN, que es mejor que MapReduce en el sentido de que permitirá inclusión de más frameworks para el procesamiento de datos. [20]

### **2.4.6.4 Comparación entre Hortonworks y Cloudera**

#### **2.4.6.4.1. Similitudes**

- Ofrecen distribuciones de Hadoop listas para la empresa. Las distribuciones han resistido la prueba del tiempo, así como los consumidores, garantizando la seguridad y la estabilidad. Además, proporcionan servicios de capacitación y familiarización pagos a los recién llegados que pisan el camino de Big Data.
- Han establecido comunidades que participan de forma activa y ayudan con los problemas que enfrentan.
- Ambas distribuciones tienen una arquitectura maestro-esclavo.
- Soportan y dan apoyo a MapReduce así como a YARN.

### 2.4.6.4.2. Diferencias

- Cloudera ha anunciado que su objetivo a largo plazo es convertirse en un "centro de datos empresariales", disminuyendo así la necesidad de almacenamiento de datos. Hortonworks, por el contrario, sigue siendo firmemente un proveedor de Hadoop, y se ha asociado con la compañía de almacenamiento de datos Teradata.
- Mientras Cloudera CDH se puede ejecutar en un servidor de Windows, HDP está disponible como un componente nativo en un servidor de Windows. Un cluster Hadoop basado en Windows se puede implementar en Windows Azure a través del Servicio HDInsight.
- Cloudera tiene un software de gestión propietario llamado Cloudera Manager, una interfaz de manejo de consulta SQL llamada Impala y Cloudera Search que es un acceso fácil y en tiempo real de los productos. Hortonworks no tiene software propietario, utiliza Ambari para la gestión y Stinger para el manejo de consultas y Apache Solr para búsquedas de datos.
- Cloudera tiene uso comercial, mientras Hortonworks tiene licencia de código abierto.
- Cloudera también permite el uso de sus proyectos de código abierto de manera gratuita, pero el paquete no incluye la suite de gestión de Cloudera Manager o cualquier otro software propietario.
- Cloudera tiene una prueba gratuita de 60 días, Hortonworks es totalmente gratuito.

**2.4.6.5. Comparación general**

	Hortonworks	Cloudera	MapR
<b>Rendimiento</b>			
Ingreso de datos	Por lotes	Por lotes	Por lotes y escritura de tipo streaming
Arquitectura de los metadatos	Centralizada	Centralizada	Distribuida
Rendimiento de HBase	Picos en latencia	Picos en latencia	Consistente de baja latencia
Aplicaciones NoSQL	Principalmente aplicaciones con datos por lotes	Principalmente aplicaciones con datos por lotes	Aplicaciones con datos por lotes y de tiempo real
<b>Dependencia</b>			
Alta disponibilidad	Recuperación de fallas simples	Recuperación de fallas simples	Auto recuperación a través de múltiples fallas
Replicación	Datos	Datos	Datos y meta-datos
Recuperación tras catástrofe	No	Planificación de copia de archivos (BDR)	Mirroring
<b>Manejabilidad</b>			
Herramientas de manejo	Ambari	Cloudera Manager	MapR Control System
Integración con API's REST	Sí	Sí	Sí
<b>Acceso a los Datos</b>			
Acceso al	HDFS, sólo	HDFS, sólo	HDFS, lectura y



## Capítulo 2. Marco Conceptual

Sistema de Archivos	lectura NFS	lectura NFS	escritura NFS
Entrada y Salida de Archivos	Sólo Append	Sólo Append	Lectura y escritura
Autenticación	Kerberos	Kerberos	Kerberos y Nativa

Tabla 2: Comparación de las principales distribuciones de Hadoop

## **3. Método de Desarrollo**

Para el desarrollo de software de manera eficiente se emplean métodos de desarrollo acorde a los requerimientos que amerite el software. Dentro del conjunto de métodos de desarrollo de Software, existen unos para el desarrollo ágil, estos tienen como objetivo fundamental minimizar las actividades que no se considera relevantes, aumentar la productividad del equipo de desarrollo y elevar la adaptabilidad del resultado.

En este capítulo se describen algunos tópicos requeridos para el desarrollo del Trabajo Especial de Grado. Entre estos tópicos se encuentran el Manifiesto Ágil y métodos ágiles Ad Hoc.

### **3.1. Manifiesto Ágil**

El 17 de febrero de 2001 diecisiete críticos de los modelos de mejora del desarrollo de software basados en procesos, convocados por Kent Beck, quien había publicado un par de años antes Extreme Programming Explained, libro en el que exponía una nueva metodología denominada Extreme Programming, se reunieron en Snowbird, Utah para tratar sobre técnicas y procesos para desarrollar software. En la reunión se acuñó el término “Métodos Ágiles” para definir a los métodos que estaban surgiendo como alternativa a las metodologías formales (CMMI, SPICE) a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo.

Los integrantes de la reunión resumieron los principios sobre los que se basan los métodos alternativos en cuatro postulados, lo que ha quedado denominado como Manifiesto Ágil. [21]

### 3.1.1. Principios del Manifiesto Ágil

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

### 3.2. Métodos Ágiles

El desarrollo ágil de software refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto

organizados y multidisciplinarios. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, sino que la meta es tener una "demo" (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (bullpen en inglés). La oficina debe incluir revisores, escritores de documentación y ayuda, diseñadores de iteración y directores de proyecto. Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica. [22]

### **3.2.1. Metodología Ad Hoc orientada a prototipos**

Antes de hablar de una metodología Ad hoc se debe explicar que significa ad hoc.

Ad hoc es una locución latina que significa literalmente "para esto". [23]

Generalmente se refiere a una solución específicamente elaborada para un problema o fin preciso y, por tanto, no generalizable ni utilizable para otros propósitos.

Se usa pues para referirse a algo que es adecuado sólo para un determinado fin o en una determinada situación.

Cuando hablamos de una metodología de desarrollo de software Ad hoc se refiere a una metodología particular para un determinado problema tomando en cuenta aspectos como el número de desarrolladores, el objetivo a cumplir, entre otros.

### Capítulo 3. Método de Desarrollo

Para el desarrollo particular de esta aplicación se adecuo de manera tal que no se necesitaría ningún artefacto entregable y se manejaría en base a prototipos funcionales o no cumpliendo los distintos objetivos plasmados para esa entrega. Dependiendo de los resultados obtenidos por cada prototipo se generarán nuevos objetivos o se eliminaran anteriores para el siguiente prototipo.

Se manejan dos roles para este desarrollo.

El rol de líder, el cual se encarga de liderizar el desarrollo. Sus actividades principales son las de definir los objetivos, verificar los resultados de cada prototipo y estimar el tiempo de desarrollo de cada prototipo.

El segundo rol es el de desarrollador, el cual se encarga de desarrollar valga la redundancia. Su actividad principal es la de completar los objetivos plasmados por el líder.

En la entrega de cada prototipo ambos roles pueden discutir sobre los objetivos siguientes o los anteriores, tomando en cuenta la opinión de cada ente.

En la siguiente figura se ejemplifica el ciclo de desarrollo de cada prototipo.

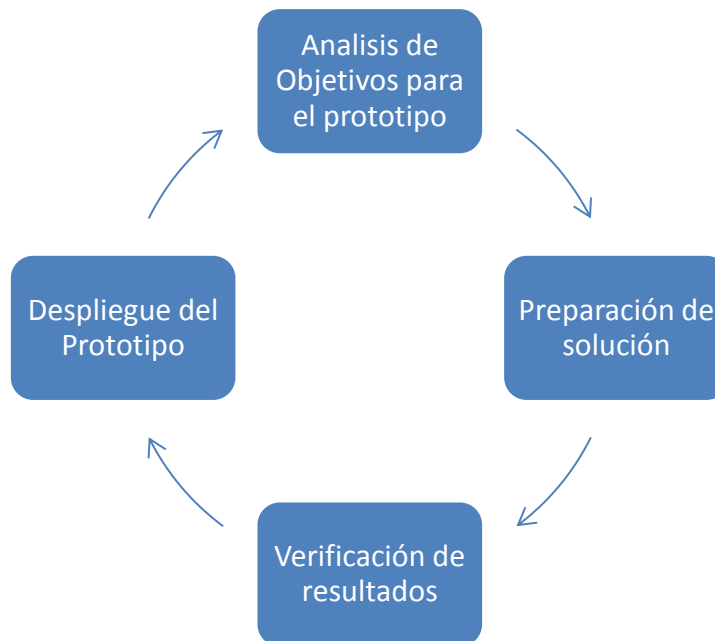


Figura 19: Ciclo de la metodología utilizada

### Capítulo 3. Método de Desarrollo

En la fase de análisis los dos roles involucrados discuten sobre los objetivos a cumplir para el prototipo.

En la siguiente fase el desarrollador ejecuta los objetivos planteados en la fase anterior, luego se verifican los resultados y se hace el despliegue del mismo. Las conclusiones obtenidas sirven para la fase de análisis del siguiente prototipo.

## **4. Desarrollo de la Solución**

Para el desarrollo de la aplicación son necesarios dos ambientes, el primero es el clúster basado en Hadoop dónde se ejecutaran los algoritmos MapReduce y el segundo es el ambiente de desarrollo como tal en el cual se ejecutará la aplicación. En ambos es necesaria la instalación de R debido a que en ese lenguaje se desarrolló la aplicación.

En estos dos ambientes se tienen que configurar con las librerías, herramientas y paquetes necesarios para el funcionamiento correcto de la aplicación. A continuación se listan en detalle las configuraciones necesarias que se realizaron para el desarrollo de la aplicación.

### **4.1. Clúster**

En lo que se refiere al clúster basado en Hadoop sólo se necesita la instalación de R en cada nodo.

#### **4.1.1. Instalación de R**

Para la instalación de R en cada nodo se siguen los siguientes pasos (ver anexo 1). Junto con R son necesarios dos paquetes junto con sus dependencias los cuales se listan a continuación:

##### **4.1.1.1. Instalación de rmr**

Para la instalación del paquete rmr en cada nodo primero se deben instalar sus dependencias y luego seguir los pasos que se reflejan en el anexo 1.

##### **4.1.1.2. Instalación de rhdfs**

Para la instalación del paquete rhdfs en cada nodo primero se deben instalar sus dependencias y luego seguir los pasos que se reflejan en el anexo 1.

## 4.2. Ambiente de Desarrollo

En lo que se refiere al ambiente de desarrollo de la aplicación si son necesarias más instalaciones en comparación con el clúster. A continuación se listan y se describen las mismas.

### 4.2.1. Instalación de GTK+

Para poder crear widgets es necesaria la herramienta GTK+. Para ver el proceso de instalación por favor véase el anexo 2. Cabe resaltar que esta especificación solamente aplica para sistemas operativos Linux, en sistemas operativos Windows no es necesaria debido a que se instala automáticamente al instalar el paquete gWidgets2 en R.

### 4.2.2. Instalación de R

La instalación de R es necesaria en el ambiente local en el anexo 3 se especifican los pasos a seguir para instalar el mismo.

#### 4.2.2.1. Instalación de R-Studio

R-Studio es la interfaz gráfica para el desarrollo de proyectos en R. Su instalación se especifica en el anexo 4.

#### 4.2.2.2. Instalación y actualización de paquetes

A diferencia del clúster que sólo necesita dos paquetes, en el ambiente de desarrollo se necesitan muchos más. A continuación se listan los paquetes necesarios instalados, la forma en la que se instaló y la función de cada uno:

- gWidgets2: Se instala ejecutando el siguiente comando en una consola de R: **install.packages('gWidgets2', dependencies = TRUE)**. Véase el capítulo dos para más detalles del paquete.
- RGtk2: Es un paquete en el lenguaje R para el desarrollo de interfaces gráficas utilizando GTK. Se instala ejecutando el



siguiente comando en una consola de R:  
**install.packages('RGtk2', dependencies = TRUE).**

- **gWidgets2RGtk2**: Es el puerto de conexión entre el API gWidgets2 hacia RGtk2. Se instala ejecutando el siguiente comando en una consola de R:  
**install.packages('gWidgets2RGtk2', dependencies = TRUE).**
- **FactoMineR**: Es un paquete que contiene métodos para el análisis exploratorio de datos, tales como los métodos de componentes principales y de clustering. Se instala ejecutando el siguiente comando en una consola de R:  
**install.packages('FactoMineR', dependencies = TRUE).**
- **kknn**: Paquete que contiene el método k vecinos más cercanos. Se instala ejecutando el siguiente comando en una consola de R:  
**install.packages('kknn', dependencies = TRUE).**
- **e1071**: Paquete que contiene funciones para el análisis de datos tales como: transformada de Fourier, fuzzy clustering, máquinas de vectores soporte, cálculo de camino más corto, bagged clustering, clasificador naive Bayes, entre otros. Se instala ejecutando el siguiente comando en una consola de R:  
**install.packages('e1071', dependencies = TRUE).**
- **MASS**: Contiene funciones y conjuntos de datos para su posterior análisis. Se instala ejecutando el siguiente comando en una consola de R:  
**install.packages('MASS', dependencies = TRUE).**
- **class**: Contiene varias funciones para la clasificación. Se instala ejecutando el siguiente comando en una consola de R:  
**install.packages('class', dependencies = TRUE).**
- **rpart**: Provee particionamiento recursivo para árboles de clasificación, regresión y supervivencia. Se instala ejecutando el

siguiente comando en una consola de R: **install.packages('rpart', dependencies = TRUE).**

- **rpart.plot**: Paquete para gráficar los modelos generados con **rpart**. Se instala ejecutando el siguiente comando en una consola de R: **install.packages('rpart.plot', dependencies = TRUE).**
- **randomForest**: Provee bosques aleatorios para la clasificación y regresión. Se instala ejecutando el siguiente comando en una consola de R: **install.packages('randomForest', dependencies = TRUE).**
- **nnet**: Software para redes neuronales con una sola capa oculta, y para los modelos multinomiales. Se instala ejecutando el siguiente comando en una consola de R: **install.packages('nnet', dependencies = TRUE).**

### 4.2.3. Instalación de Openssh y sshpass

Para la comunicación entre el ambiente de desarrollo y el clúster es necesario utilizar el protocolo Secure Shell (SSH) el cual se obtiene instalando la aplicación Openssh y sshpass como se ve en el anexo 5. Esta instalación sólo es válida para los sistemas operativos basados en Linux debido a que estos programas son netos de estos sistemas.

## 4.3. Aplicación

Después de tener ambos ambientes completamente configurados se empezó con el desarrollo de la aplicación utilizando la metodología previamente definida. Como se dijo anteriormente esta metodología consistía de la realización de prototipos funcionales o no utilizando el lenguaje de programación R.

A continuación se listan todos los prototipos desarrollados con sus objetivos, resultados y conclusiones:

### 4.3.1. Prototipo 0

Este prototipo se basó en la exploración de las herramientas que provee R y sus paquetes para la realización de widgets, por esto previo a trazar objetivos se realizó una exploración a fondo del paquete gWidgets.

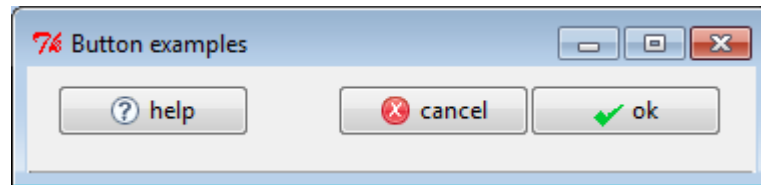


Figura 20: Ejemplo de widget utilizando R con el paquete gWidgets

Luego de realizar esta exploración se procedió a la primera reunión con el fin de establecer los objetivos para el primer prototipo.

#### 4.3.1.1. Objetivos

1. Crear una interfaz gráfica utilizando R y el paquete gWidgets
2. Ejecutar el método K medias sobre un conjunto de datos provenientes de un archivo con valores separados por coma.
3. Ejecutar el método Análisis de componentes principales sobre un conjunto de datos provenientes de un archivo con valores separados por coma.
4. Ejecutar el método de clúster jerárquico sobre un conjunto de datos provenientes de un archivo con valores separados por coma.

#### 4.3.1.2. Resultados

Para el primer objetivo se logro crear una interfaz sencilla con una barra de herramientas, un toolbar y tres secciones, una para cada método. En cada sección se requieren los parámetros necesarios para cada método Véanse las Figuras 21, 22 y 23.

## Capítulo 4. Desarrollo de la Solución

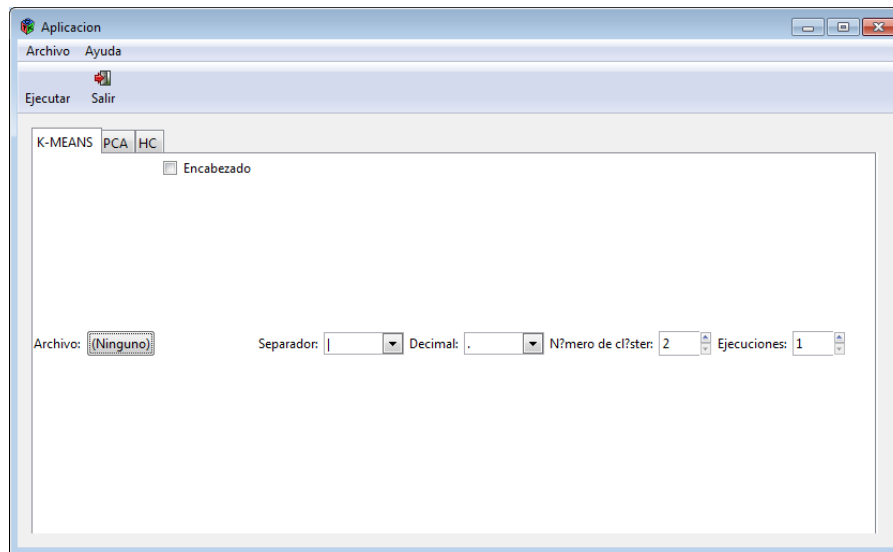


Figura 21: Sección K-medias del Prototipo 0

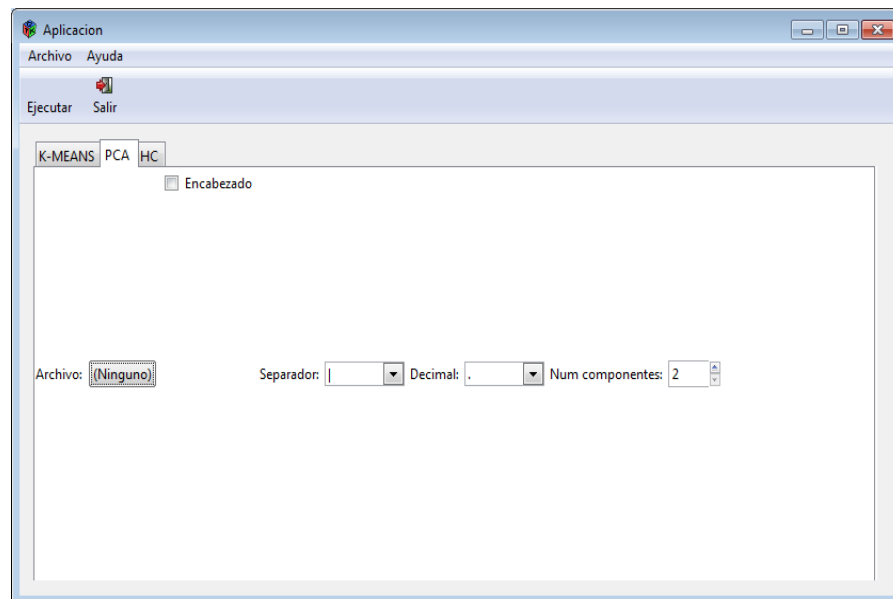


Figura 22: Sección de Análisis de Componentes Principales del Prototipo 0

## Capítulo 4. Desarrollo de la Solución

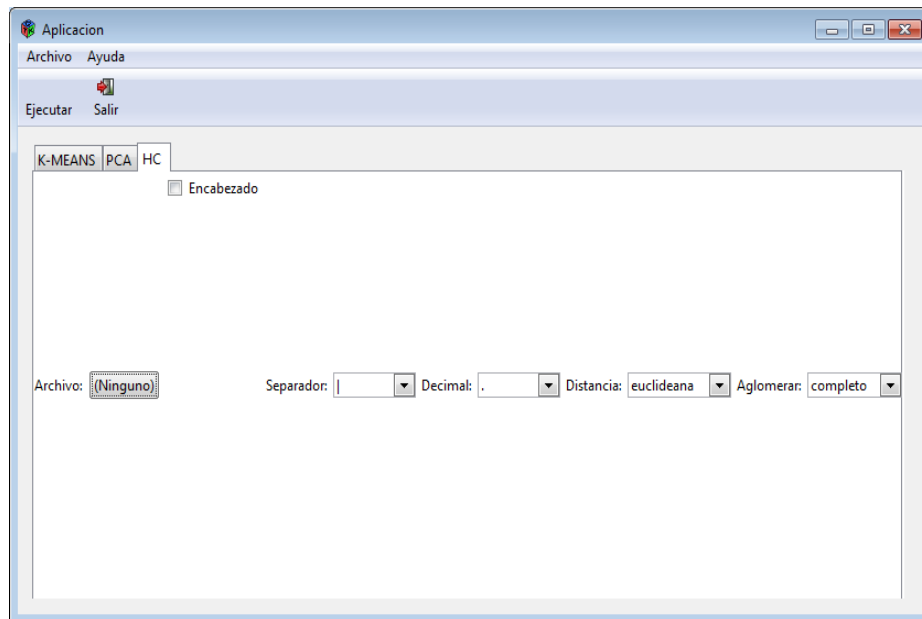


Figura 23: Sección Clúster Jerárquico del Prototipo 0

Para el segundo objetivo se utilizaron funciones de R para realizar el análisis. El resultado de la ejecución del método se muestra por una consola de R como se ve en la Figura 24.

```
Console H:/Tesis/SmartR/ ↵
K-means clustering with 3 clusters of sizes 2, 4, 4

Cluster means:
  Matematicas Ciencias Espanol Historia EducFisica
1    5.500    6.250    6.500    6.250    8.850
2    7.700    9.475    7.625    7.750    6.750
3    6.525    6.525    8.475    8.875    7.375

Clustering vector:
  Lucia Pedro  Ines  Luis Andres  Ana Carlos  Jose  Sonia  Maria
    3     2     2     1     3     2     3     2     1     3

within cluster sum of squares by cluster:
[1] 1.7950 2.5150 3.1575
(between_SS / total_SS = 86.9 %)

Available components:
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> |
```

Figura 24: Resultado del método K medias del prototipo 0

## Capítulo 4. Desarrollo de la Solución

Para el tercer objetivo al igual que en el objetivo anterior se imprime el resultado del modelo por una consola de R, véase la figura 25, pero también se muestra a través de otro widget el círculo de correlaciones y las distancias entre cada individuo, véase la figura 26.

```
Console H:/Tesis/SmartR/
**Results for the Principal Component Analysis (PCA)**
The analysis was performed on 10 individuals, described by 5 variables
*The results are available in the following objects:

  name          description
1 "$eig"        "eigenvalues"
2 "$var"        "results for the variables"
3 "$var$coord" "coord. for the variables"
4 "$var$cor"    "correlations variables - dimensions"
5 "$var$cos2"  "cos2 for the variables"
6 "$var$contrib" "contributions of the variables"
7 "$ind"        "results for the individuals"
8 "$ind$coord" "coord. for the individuals"
9 "$ind$cos2"  "cos2 for the individuals"
10 "$ind$contrib" "contributions of the individuals"
11 "$call"      "summary statistics"
12 "$call$centre" "mean of the variables"
13 "$call$secart.type" "standard error of the variables"
14 "$call$row.w" "weights for the individuals"
15 "$call$col.w" "weights for the variables"
>
```

Figura 25: Resultado Análisis de componentes Principales del Prototipo 0

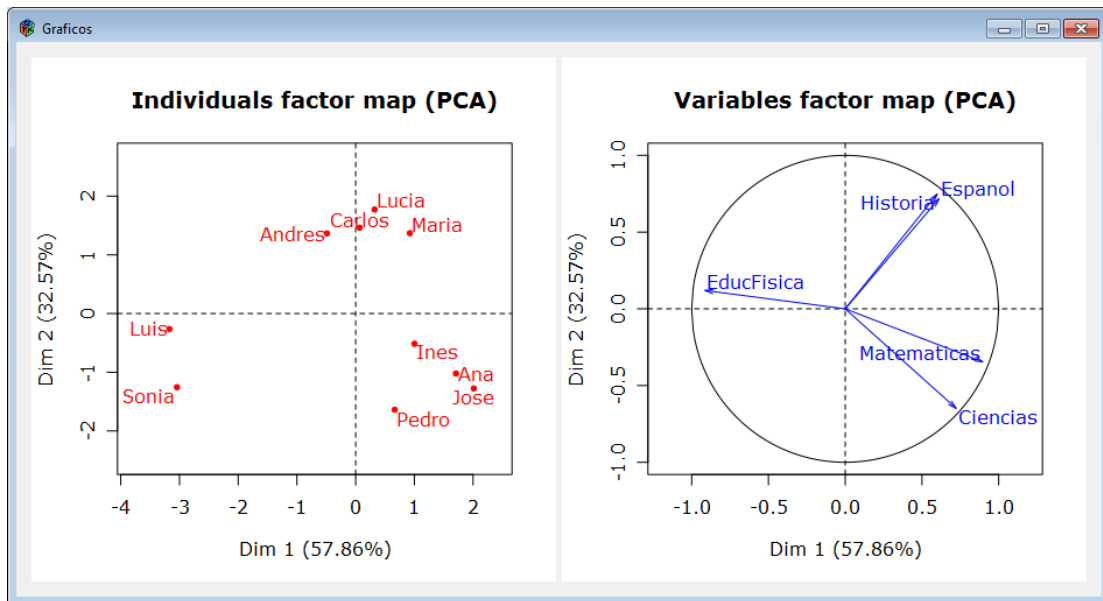


Figura 26: Resultados gráficos de Análisis de componentes principales del Prototipo 0

Para el cuarto y último objetivo solamente se imprime el dendrograma resultante en un nuevo widget, véase la figura 27.

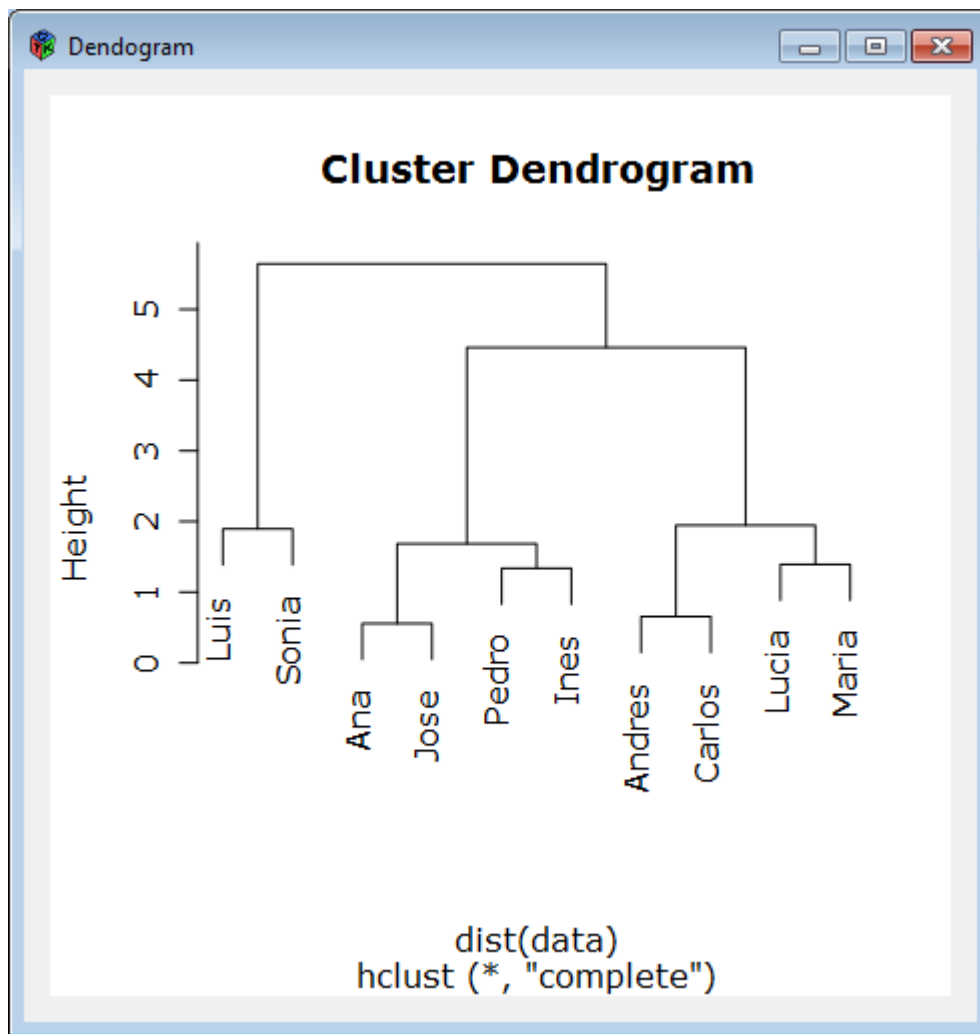


Figura 27: Resultado Agrupamiento Jerárquico

### 4.3.1.3. Conclusiones

Luego de entregar este prototipo se llegaron a las siguientes conclusiones:

- Se deben mejorar las interfaces de usuario debido a que existieron muchos descontroles en los elementos.
- Los resultados se deben ver en algún widget generado por la aplicación.
- El agrupamiento jerárquico no agrupa, sólo muestra el dendograma.
- Sería mejor tener una sección para la carga de datos solamente.

## Capítulo 4. Desarrollo de la Solución

- No se hace referencia a la teoría de grandes volúmenes de datos.

### **4.3.2. Prototipo 1**

Tomando como premisa los resultados y las conclusiones del prototipo 0 se procedió a establecer los objetivos para a siguiente entrega bajo el nombre de Prototipo 1.

#### **4.3.2.1. Objetivos**

1. Mejorar la interfaz de usuario colocando iconos a los botones acorde a sus funcionalidades
2. Crear sección única para la carga de datos de entrada
3. Crear sección única para el análisis exploratorio de datos
4. Crear sección única para el agrupamiento de datos
5. Crear sección única para los algoritmos de Big Data
6. Mostrar resultados de los métodos de manera gráfica
7. Ejecutar método K medias bajo el paradigma MapReduce
8. Establecer funcionalidad para determinar el K óptimo para el método K medias

#### **4.3.2.2. Resultados**

El primer objetivo se cumplió mejorando el marco de la aplicación, véase la figura 28.



## Capítulo 4. Desarrollo de la Solución

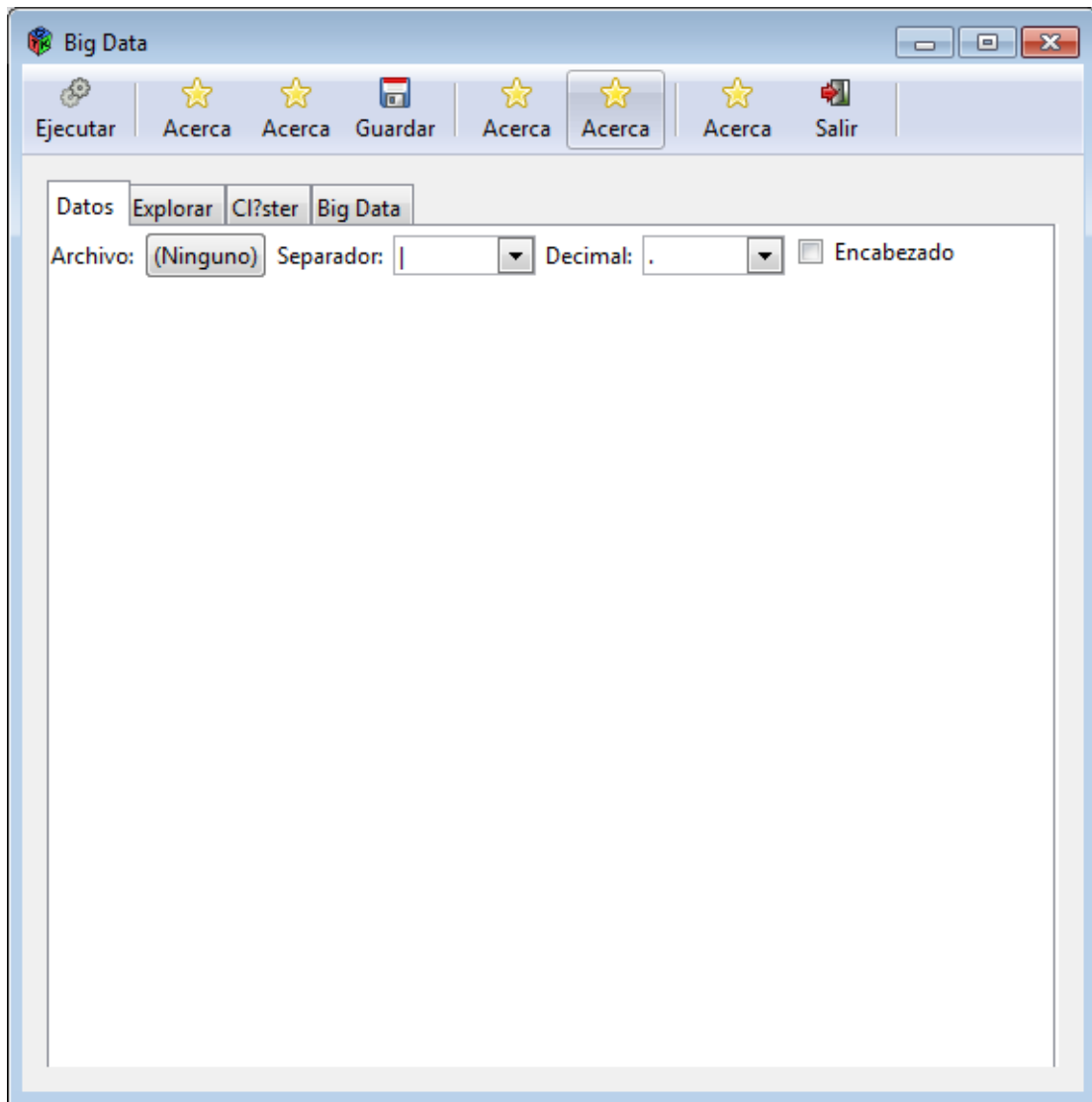


Figura 28: Vista principal del prototipo 1

Los objetivos dos, tres, cuatro y cinco se ven representados en las siguientes imágenes:

## Capítulo 4. Desarrollo de la Solución

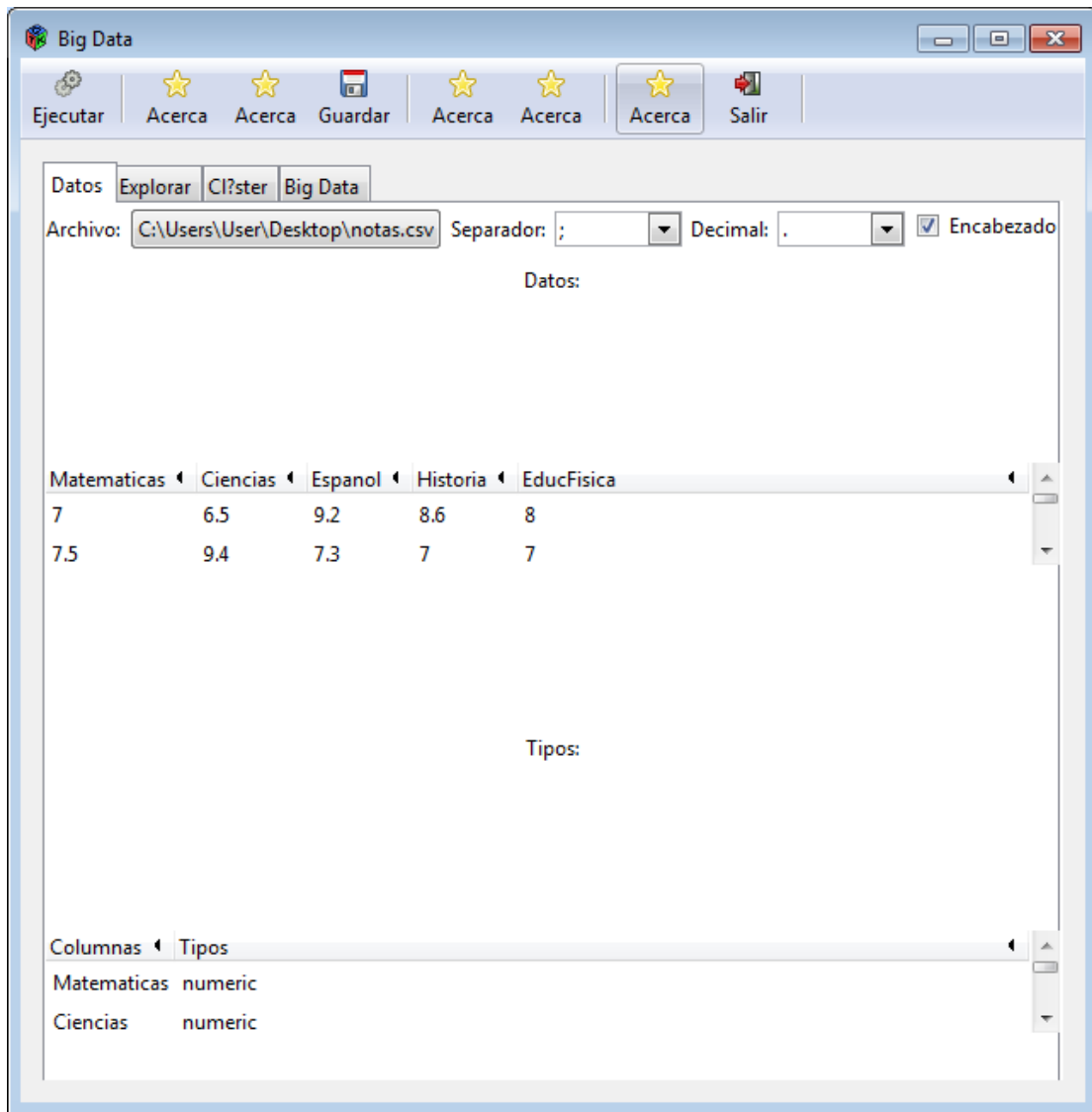


Figura 29: Sección de carga de datos del prototipo 1

Capítulo 4. Desarrollo de la Solución

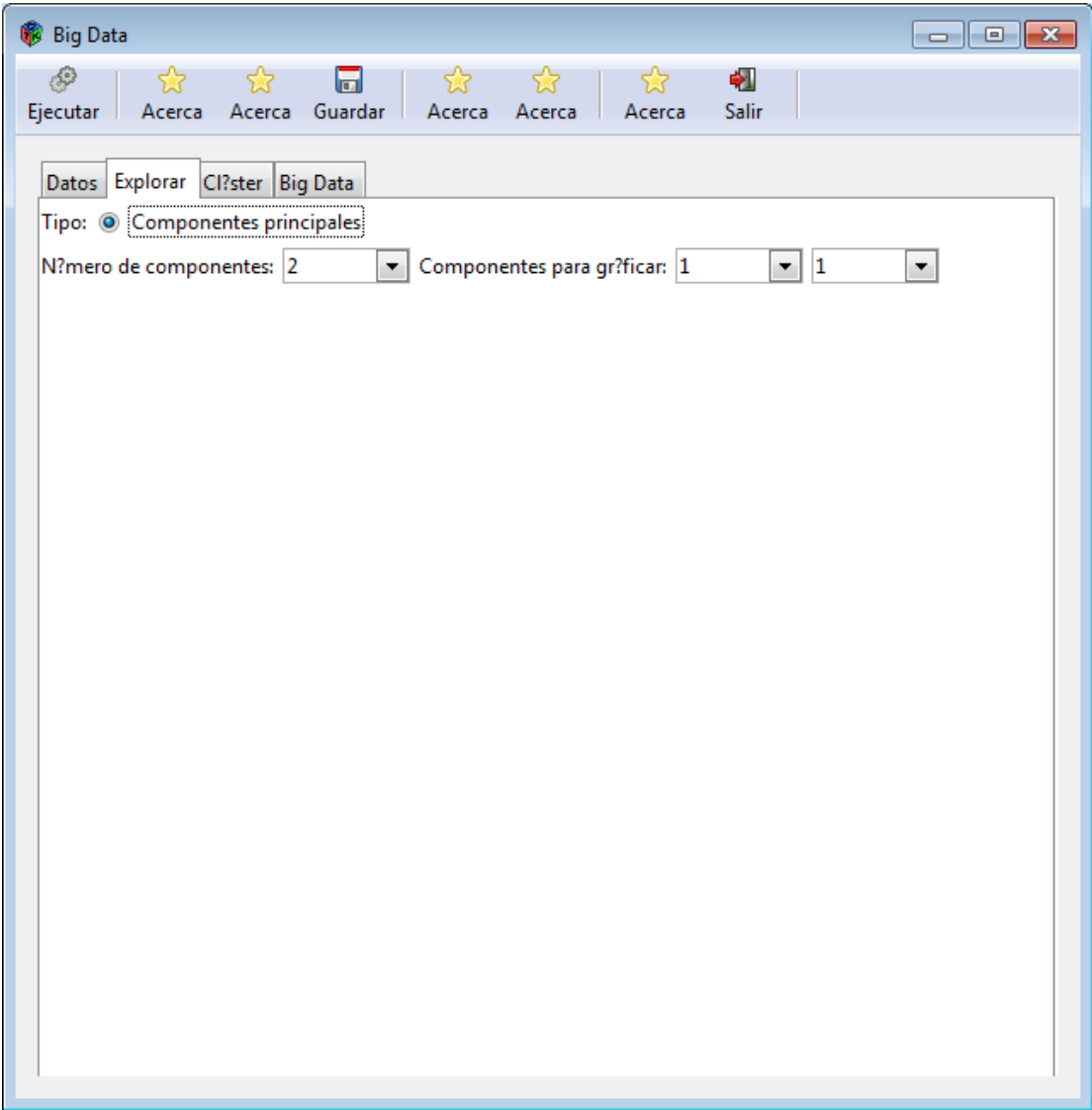


Figura 30: Sección de análisis exploratorio de datos del prototipo 1

## Capítulo 4. Desarrollo de la Solución

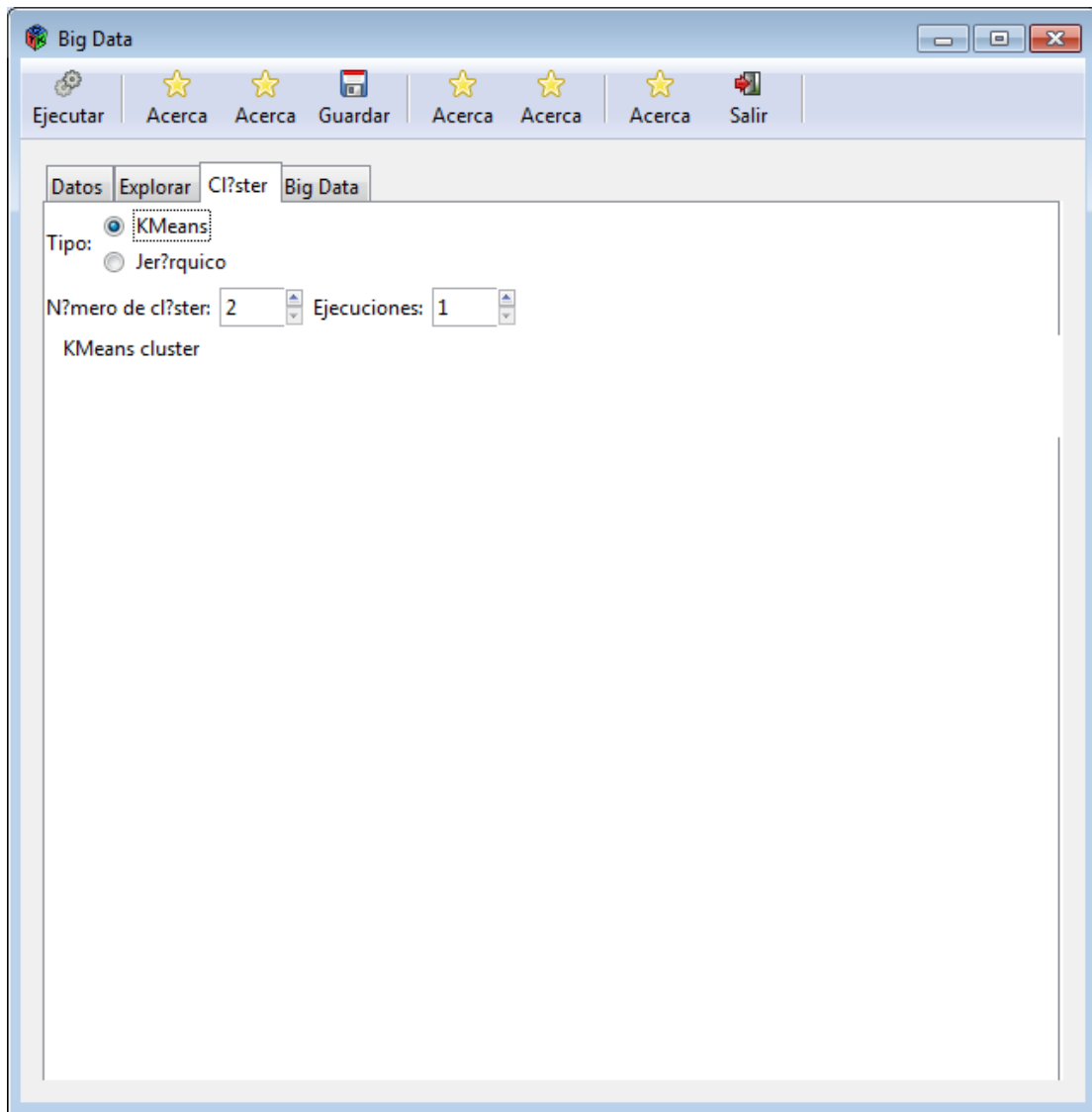


Figura 31: Sección para el agrupamiento de datos del prototipo 1

## Capítulo 4. Desarrollo de la Solución

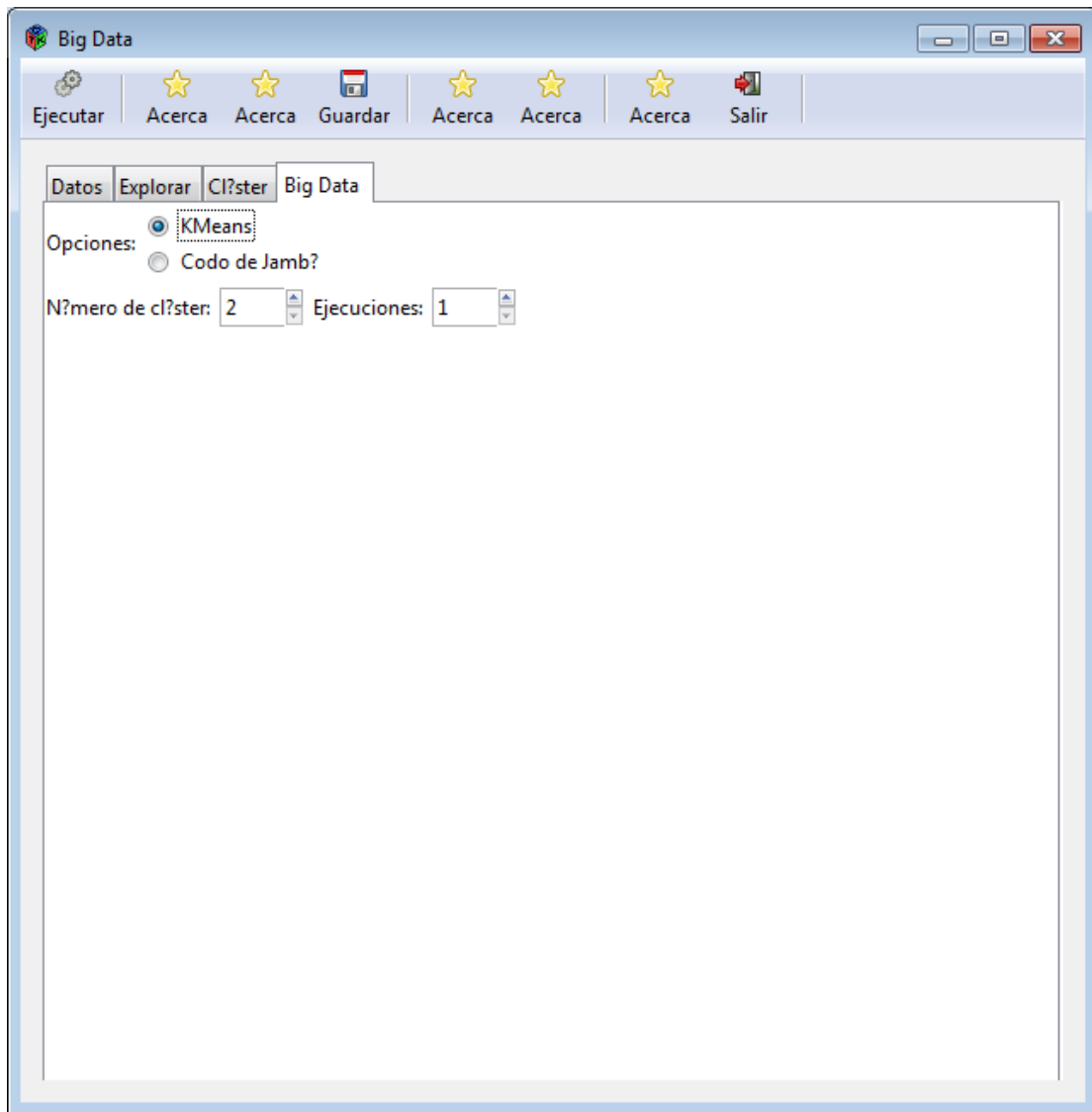


Figura 32: Sección de Big Data del prototipo 1

El objetivo seis se cumplió para todas las secciones como se muestra en las siguientes imágenes:

## Capítulo 4. Desarrollo de la Solución

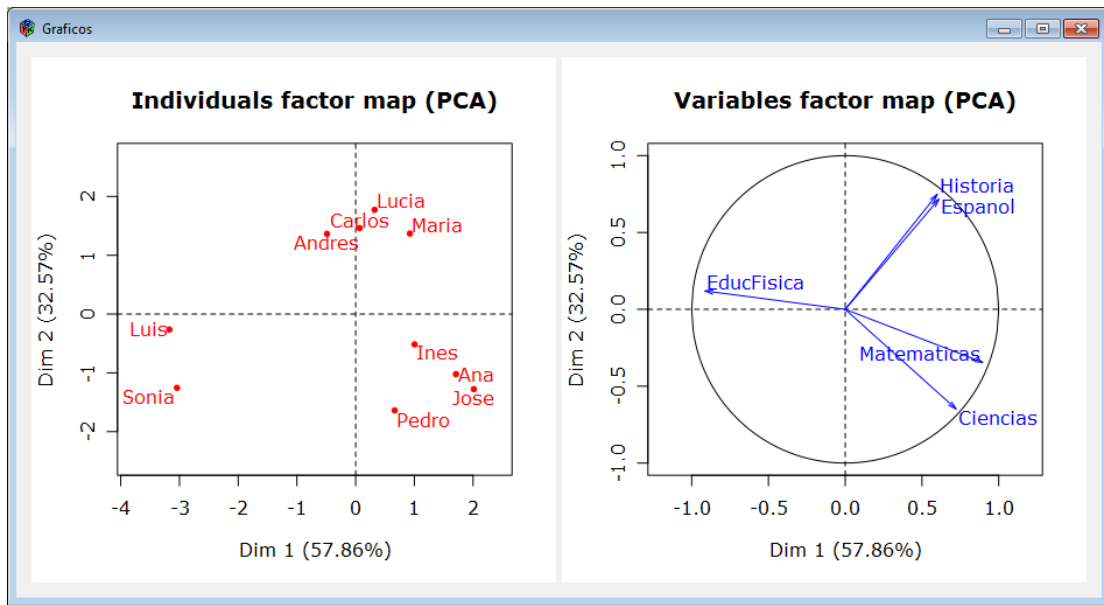


Figura 33: Resultados gráficos de la sección de exploración de datos del prototipo 1

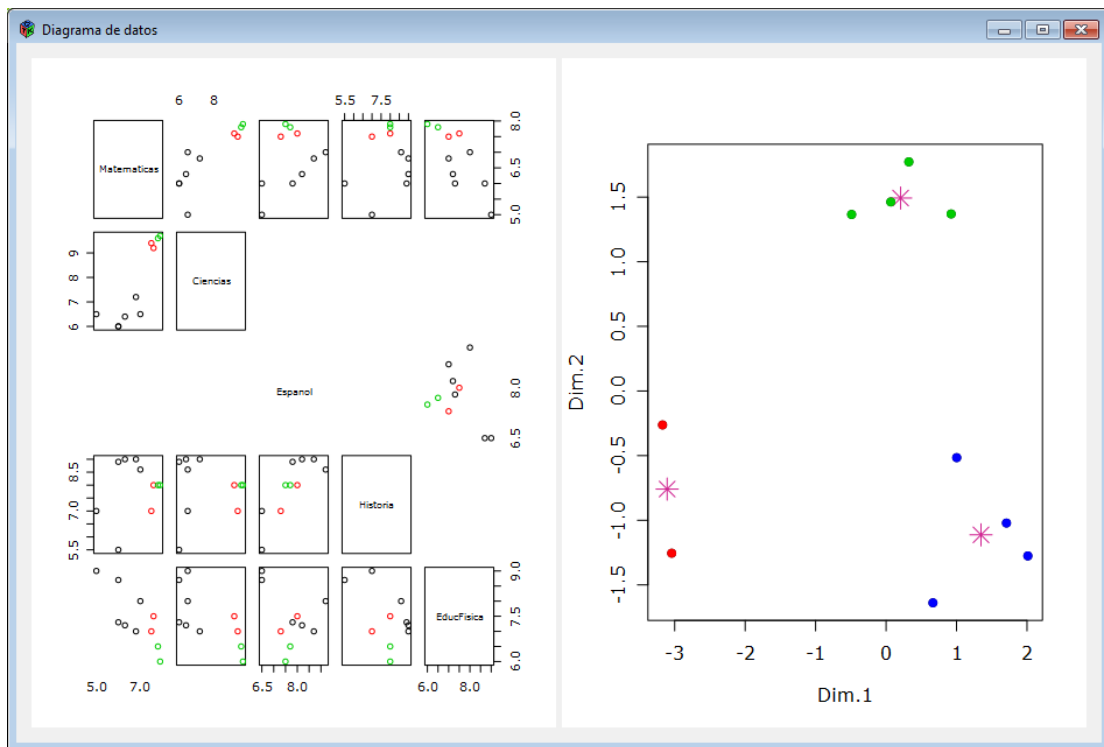


Figura 34: Resultados gráficos del método K medias del prototipo 1

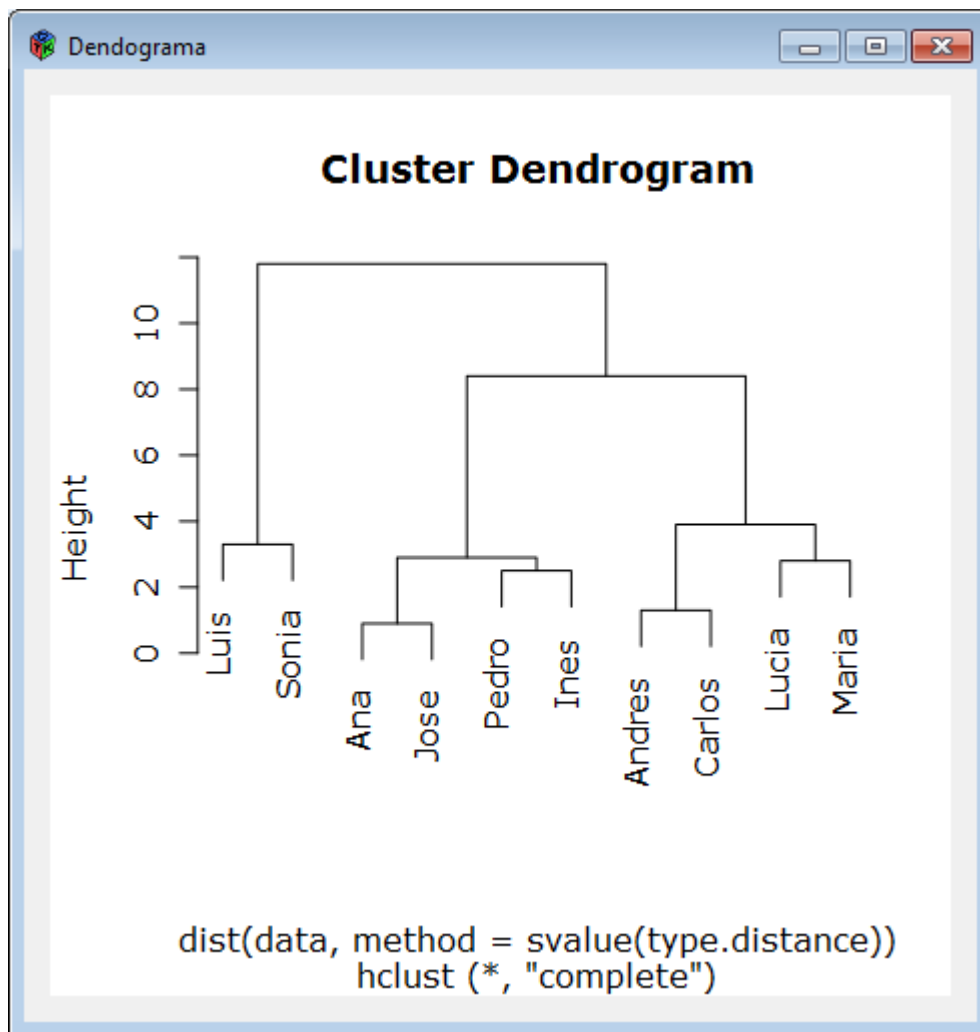


Figura 35: Resultados gráficos del método agrupamiento jerárquico del prototipo 1

Los objetivos siete y ocho no pudieron ser cumplidos por limitaciones de tiempo.

#### 4.3.2.3. Conclusiones

Luego de la entrega de este prototipo se llegó a estas conclusiones:

- Es necesaria la ejecución del algoritmo K medias MapReduce
- Se debe mostrar gráfico con la agrupación del método de agrupación jerárquico

### 4.3.3. Prototipo 2

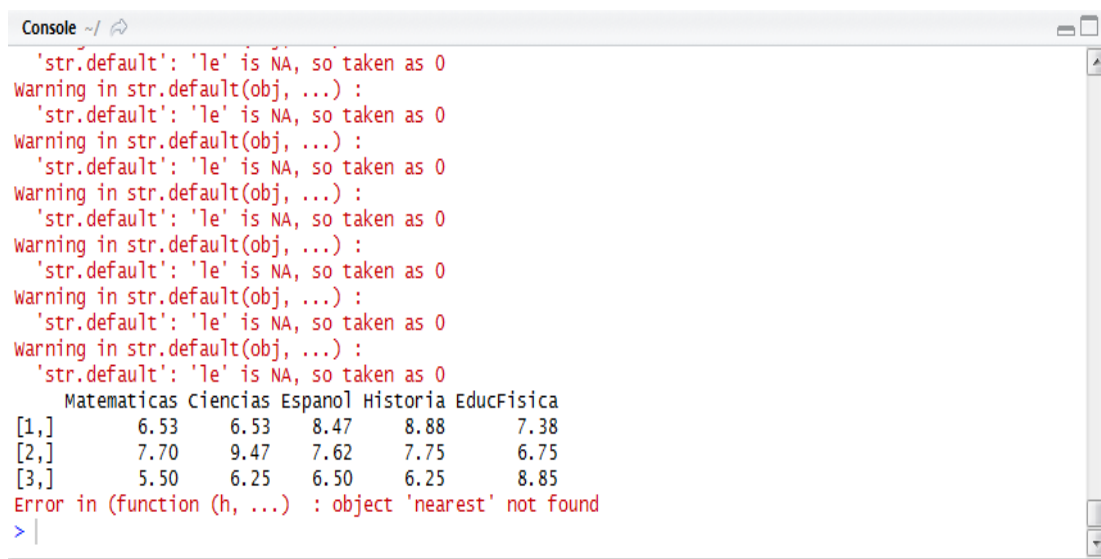
El prototipo 2 no tuvo muchos cambios con respecto al prototipo 1. A continuación se listan los objetivos

#### 4.3.3.1. Objetivos

1. Ejecutar el método K medias MapReduce a un conjunto de datos en el ambiente local.

#### 4.3.3.2. Resultados

El único objetivo que se trazó se cumplió perfectamente cómo se muestra en la siguiente figura que muestra los resultados obtenidos.



```
Console ~/ |
'usr.default': 'le' is NA, so taken as 0
warning in str.default(obj, ...) :
'usr.default': 'le' is NA, so taken as 0
warning in str.default(obj, ...) :
'usr.default': 'le' is NA, so taken as 0
warning in str.default(obj, ...) :
'usr.default': 'le' is NA, so taken as 0
warning in str.default(obj, ...) :
'usr.default': 'le' is NA, so taken as 0
warning in str.default(obj, ...) :
'usr.default': 'le' is NA, so taken as 0
warning in str.default(obj, ...) :
'usr.default': 'le' is NA, so taken as 0
Matematicas Ciencias Espanol Historia EducFisica
[1,]      6.53      6.53      8.47      8.88      7.38
[2,]      7.70      9.47      7.62      7.75      6.75
[3,]      5.50      6.25      6.50      6.25      8.85
Error in (function (h, ...) : object 'nearest' not found
> |
```

Figura 36: Resultado K medias MapReduce del prototipo 2

El método K medias MapReduce utilizado se encuentra en el anexo 6.

#### 4.3.3.3. Conclusiones

Las conclusiones más importantes tras el despliegue del prototipo 2 son:



## Capítulo 4. Desarrollo de la Solución

- El prototipo tiene muy pocos métodos disponibles
- Resulta engorroso tener que cambiar entre conjuntos de datos
- No existe una visualización de datos previos a la carga

### 4.3.4. Prototipo 3

Tras las conclusiones obtenidas en el prototipo dos se empezó a utilizar el paquete gWidgets2 en vez de gWidgets y también se procedió a la generación de los nuevos objetivos para el siguiente prototipo.

#### 4.3.4.1. Objetivos

Los objetivos principales fueron:

1. Incluir métodos de clasificación en una nueva sección
2. Capacidad de crear conexiones de datos y acceder a ellas siempre
3. Crear los siguientes tipos de conexiones de datos: Archivos de tipo csv, archivos de tipo HDFS y muestra de archivos de tipo HDFS
4. Ver la estadística de los archivos de tipo csv antes de ser almacenados
5. Ver los datos antes de ser almacenados
6. Decidir que variable será la variable destino para los métodos de clasificación
7. Mostrar la lista de todas las conexiones creadas
8. Poder editar la lista de todas las conexiones creadas
9. Mostrar los métodos disponibles dependiendo que tipo de conexión se haya escogido
10. Poder mostrar la clasificación del método agrupamiento jerárquico

#### 4.3.4.2. Resultados

El primer objetivo se ve reflejado en la siguiente figura en la cual se listan todos los métodos disponibles para la conexión seleccionada.

## Capítulo 4. Desarrollo de la Solución

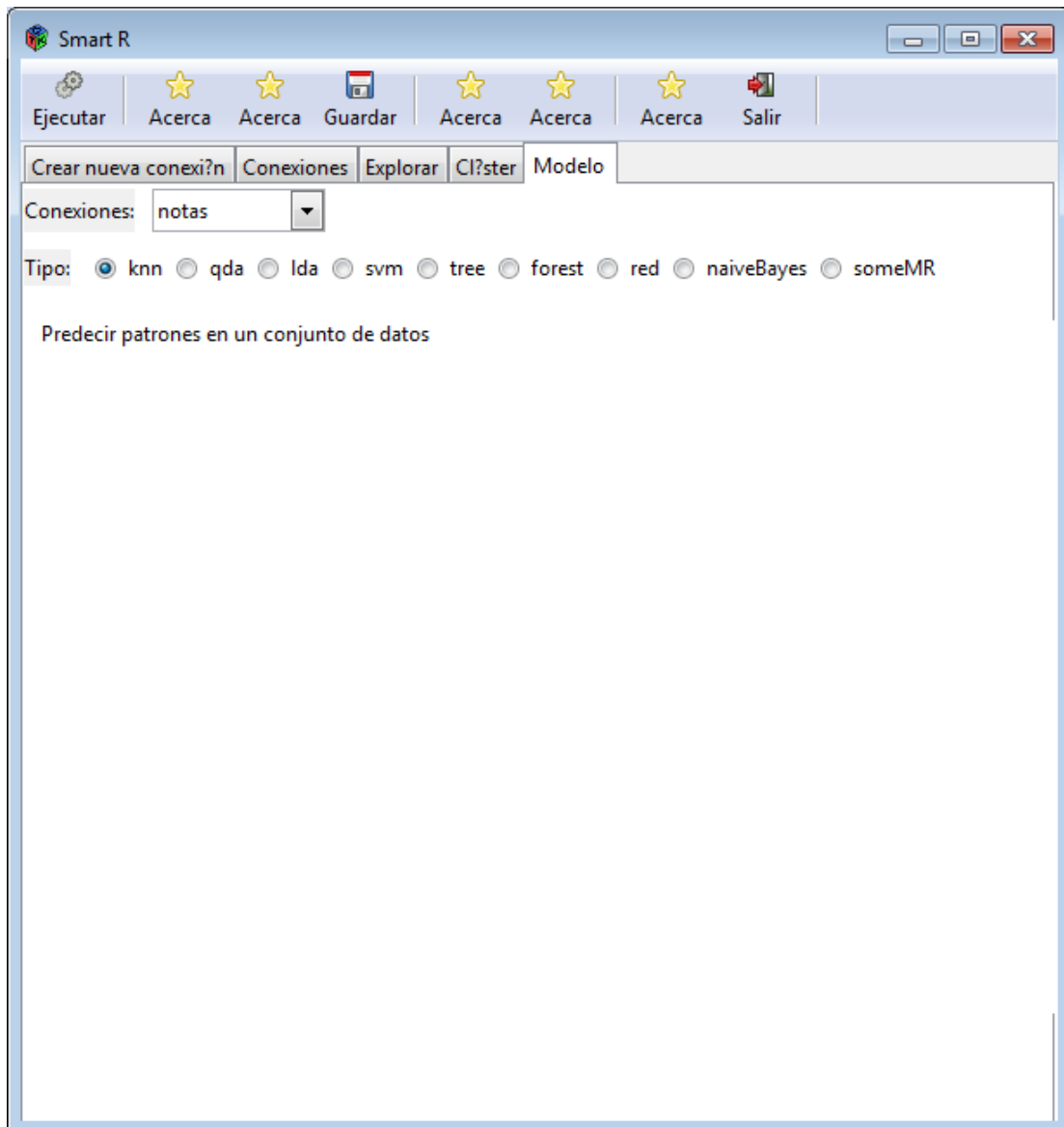


Figura 37: Sección modelo del prototipo 3

El objetivo número dos se logró pero no de manera completa, se crean las conexiones y se pueden acceder a ellas siempre y cuando no se haya reiniciado la aplicación, es decir, si la aplicación se cierra con algunas conexiones las mismas no se podrán acceder cuando se inicie otra vez, se deben volver a crear.

El objetivo número tres se logró aunque los archivos de tipo HDFS y muestra HDFS son iguales a los archivos csv, esto debido sólo para ilustrar

## Capítulo 4. Desarrollo de la Solución

que la selección de métodos disponibles dependiendo del tipo de conexión funciona.

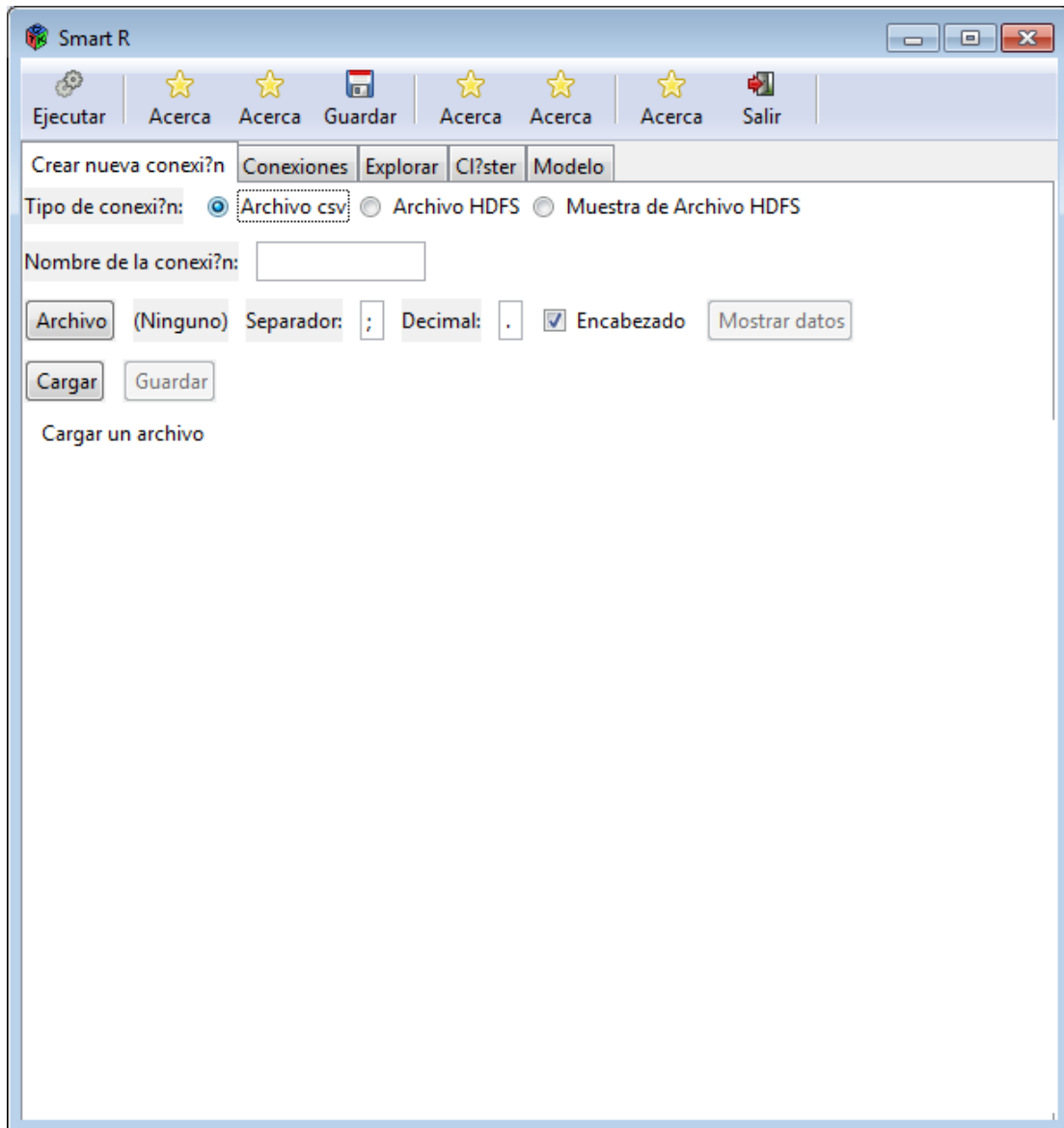


Figura 38: Sección para la creación de conexiones del prototipo 3

Los objetivos cuatro, cinco y seis se ven en las siguientes figuras:

## Capítulo 4. Desarrollo de la Solución

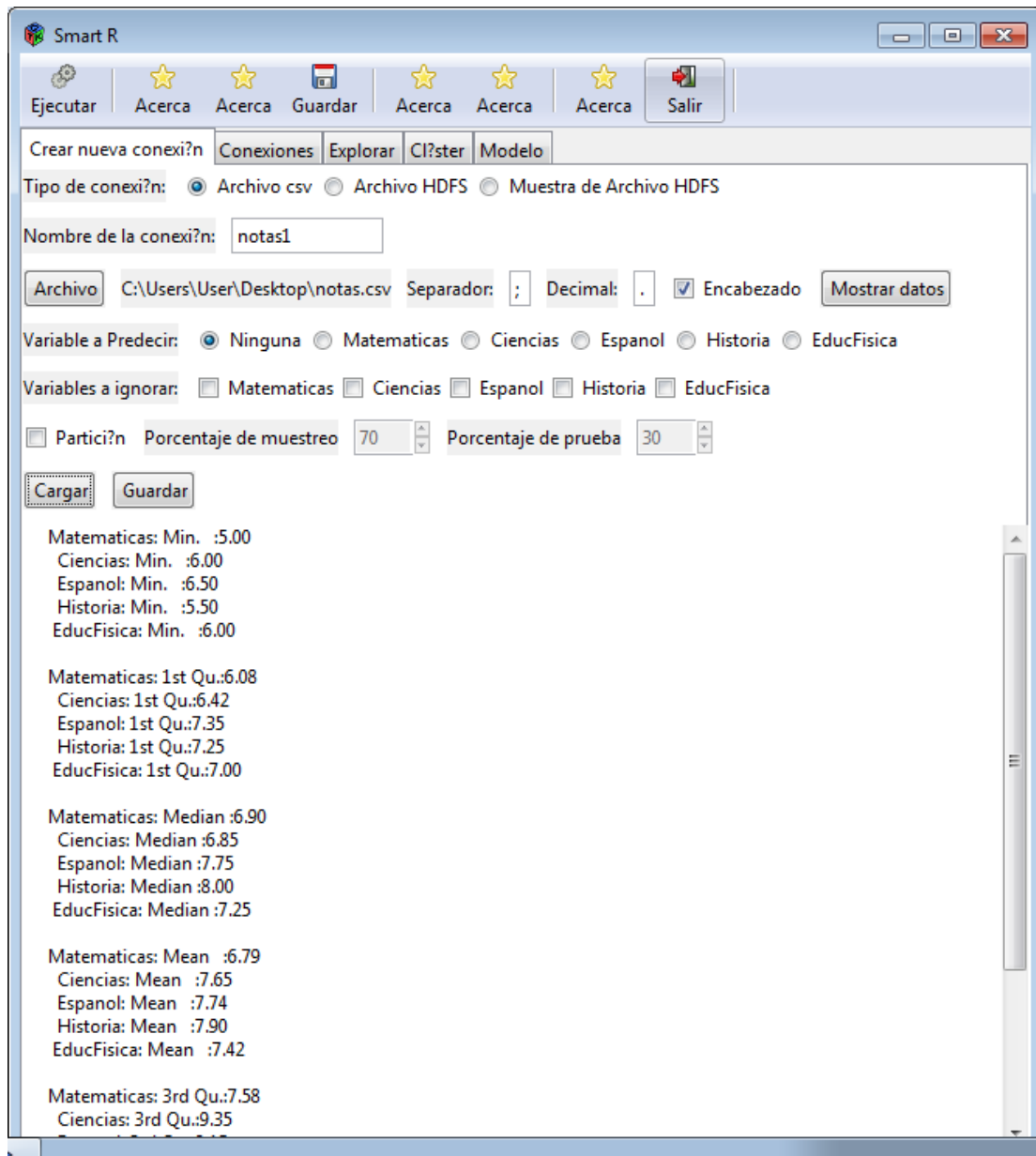
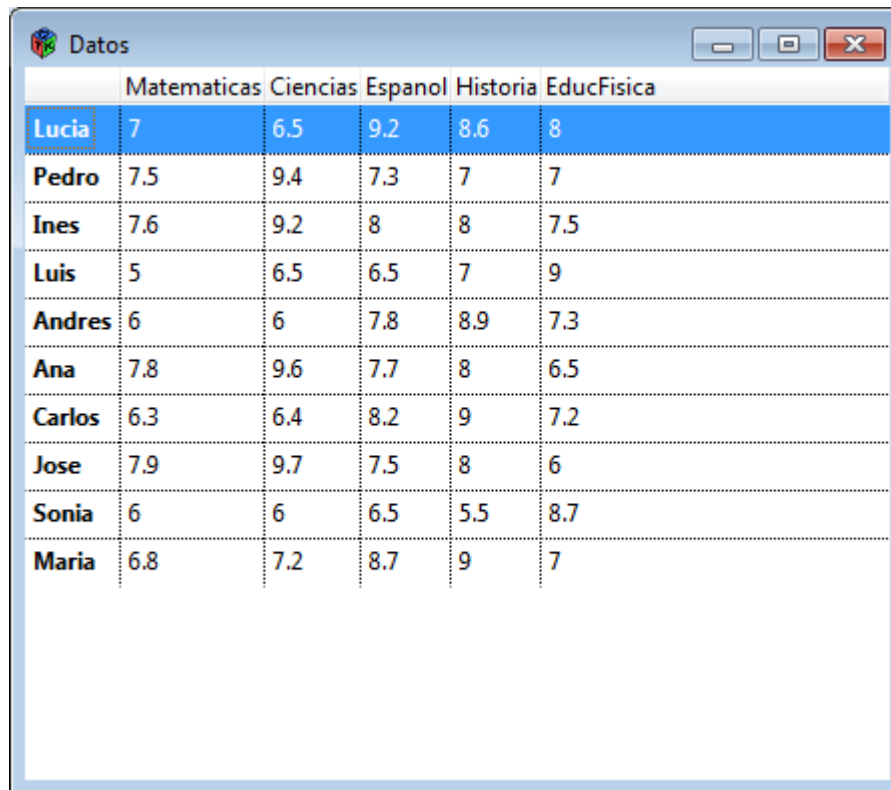


Figura 39: Resultado de carga previa al almacenamiento de conexiones en el prototipo 3

## Capítulo 4. Desarrollo de la Solución



The image shows a window titled 'Datos' with a table of student scores. The table has six columns: 'Matematicas', 'Ciencias', 'Espanol', 'Historia', and 'EducFisica'. The rows list the names of the students: Lucia, Pedro, Ines, Luis, Andres, Ana, Carlos, Jose, Sonia, and Maria. The scores are as follows:

	Matematicas	Ciencias	Espanol	Historia	EducFisica
Lucia	7	6.5	9.2	8.6	8
Pedro	7.5	9.4	7.3	7	7
Ines	7.6	9.2	8	8	7.5
Luis	5	6.5	6.5	7	9
Andres	6	6	7.8	8.9	7.3
Ana	7.8	9.6	7.7	8	6.5
Carlos	6.3	6.4	8.2	9	7.2
Jose	7.9	9.7	7.5	8	6
Sonia	6	6	6.5	5.5	8.7
Maria	6.8	7.2	8.7	9	7

Figura 40: Datos de una conexión del prototipo 3

El resultado de la realización del objetivo siete se muestra a continuación:

## Capítulo 4. Desarrollo de la Solución

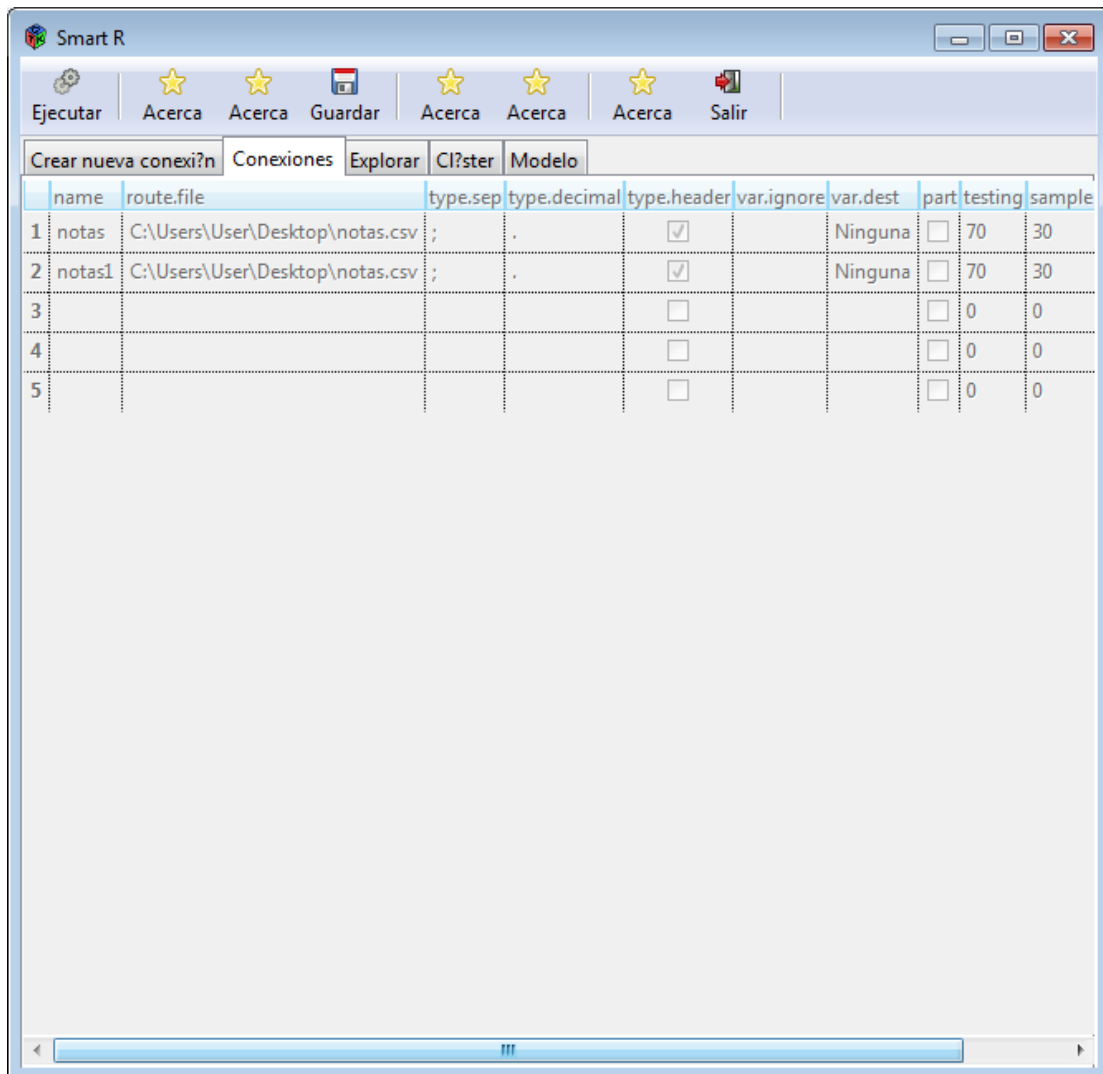


Figura 41: Sección dónde se muestran todas las conexiones disponibles del prototipo 3

El objetivo ocho no se cumplió y quedo rezagado para la siguiente entrega, mientras el objetivo nueve sí como se muestra a continuación:

## Capítulo 4. Desarrollo de la Solución

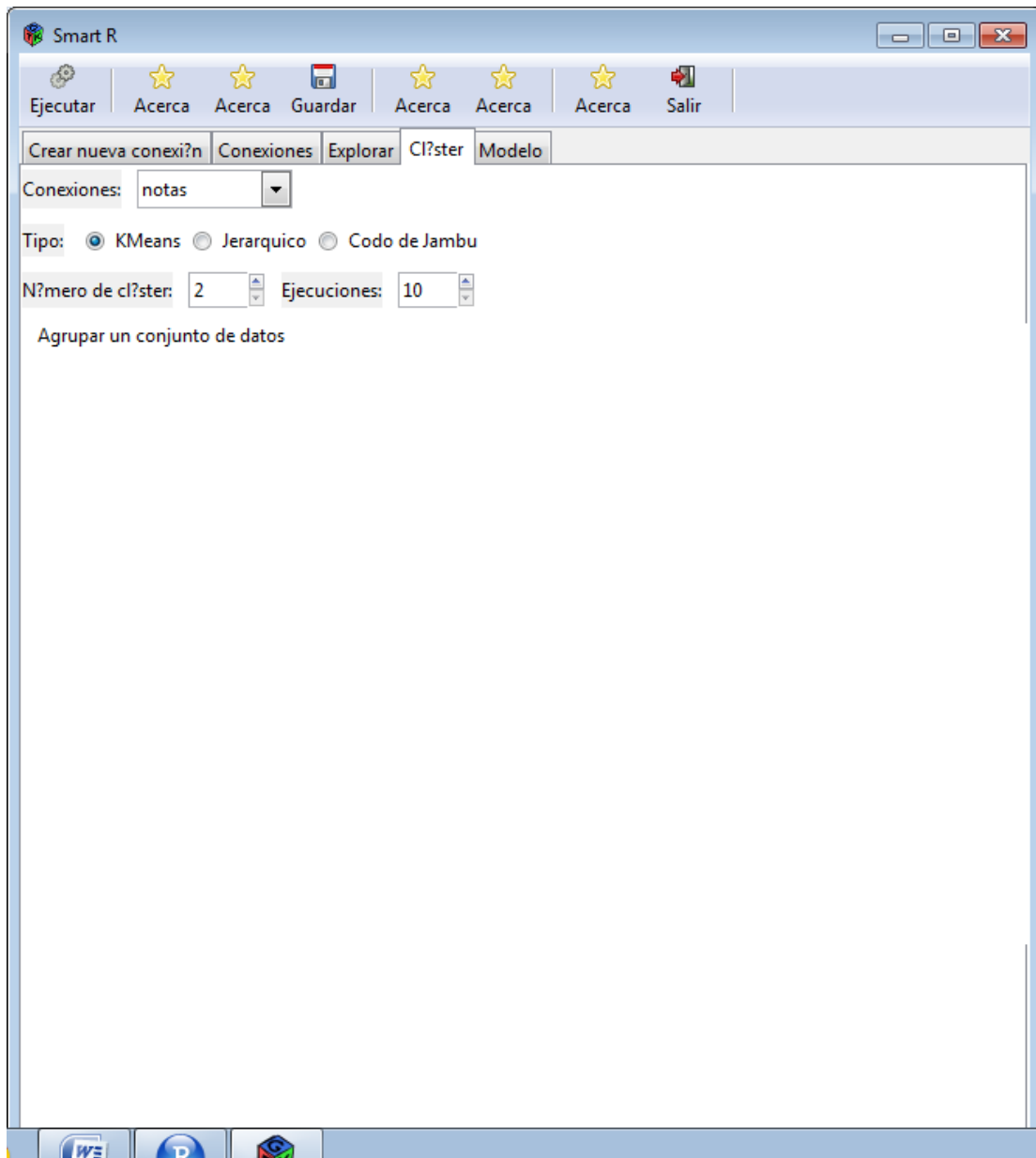


Figura 42: Métodos disponibles para las conexiones de tipo archivo csv del prototipo 3

## Capítulo 4. Desarrollo de la Solución

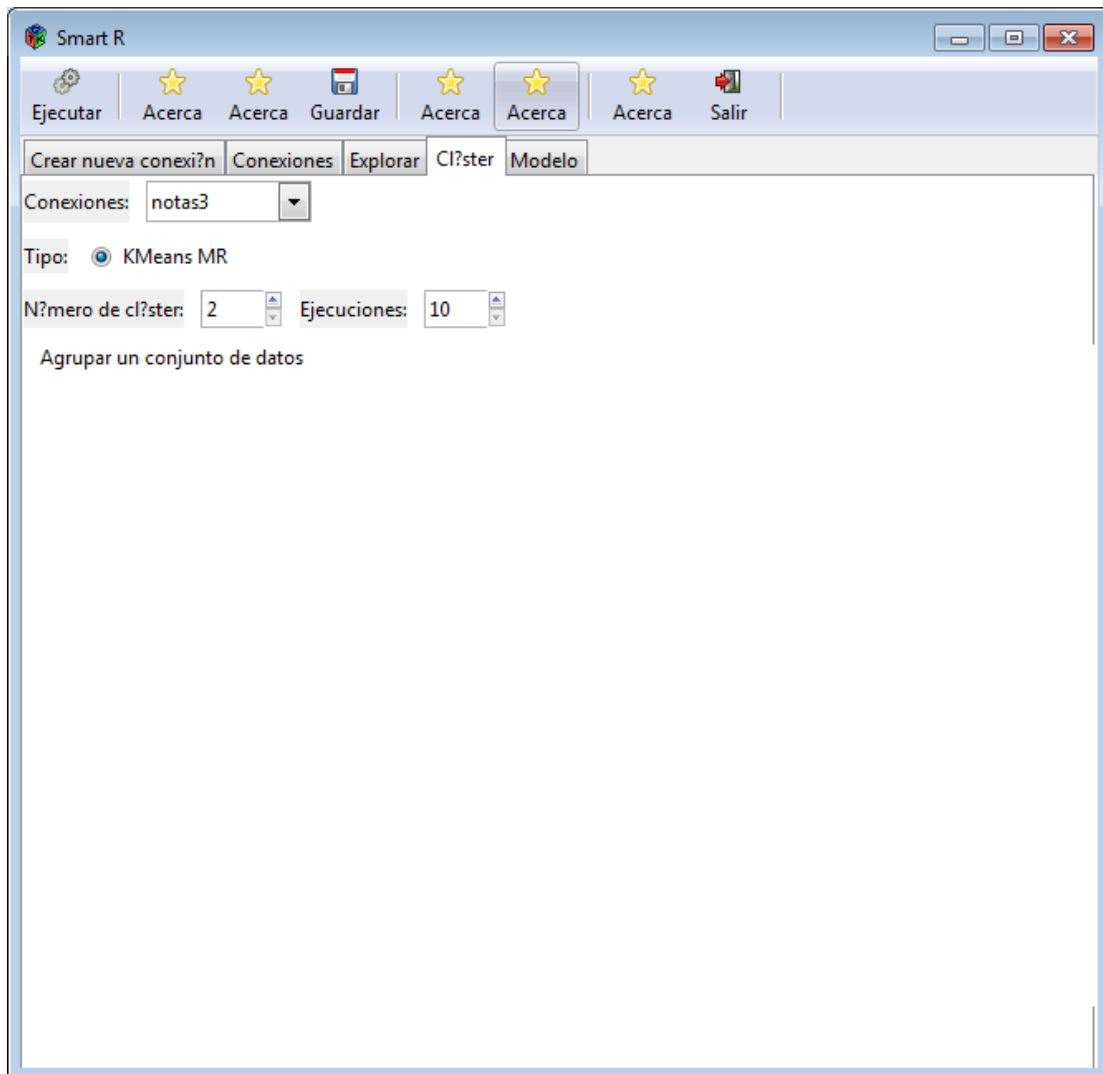


Figura 43: Métodos disponibles para las conexiones de tipo archivo HDFS del prototipo 3

El objetivo diez se cumplió a cabalidad y se ve representado en la siguiente figura:



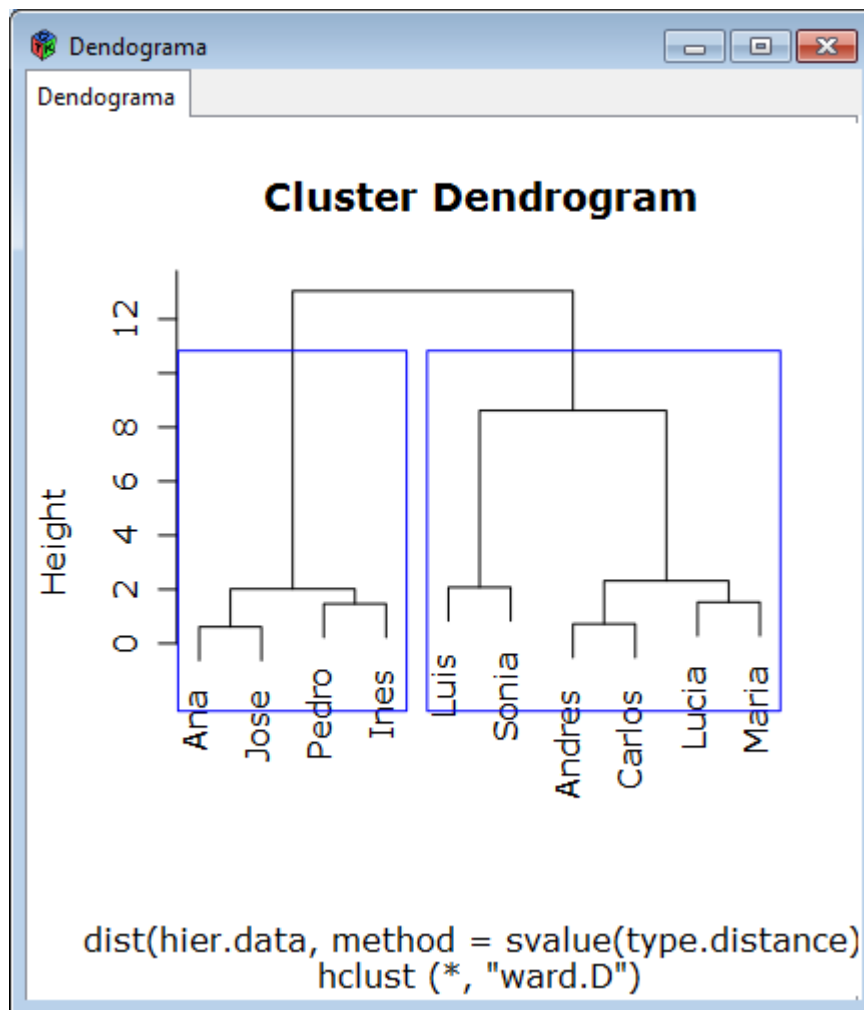


Figura 44: Resultado de clasificación por el método de agrupamiento jerárquico

#### 4.3.4.3. Conclusiones

Las principales conclusiones son que para cada método sigue sin imprimir el resultado en la aplicación, sólo lo imprime en una consola de R. Además de esta también se concluyó lo siguiente:

- La clasificación y los modelos generados sólo se tienen de manera visual, sería ideal poder descargarlos.

#### 4.3.5. Prototipo 4

## Capítulo 4. Desarrollo de la Solución

El prototipo cuatro no tuvo muchas diferencias con respecto al tres. Sólo se incluyó un nuevo tipo de conexión llamada datos de R el cual provee algunos conjuntos de datos para la utilización de la aplicación.

### **4.3.5.1. Objetivos**

El único objetivo fue el de crear el nuevo tipo de conexión.

### **4.3.5.2. Resultados**

El único objetivo fue cumplido como se muestra en la siguiente imagen:

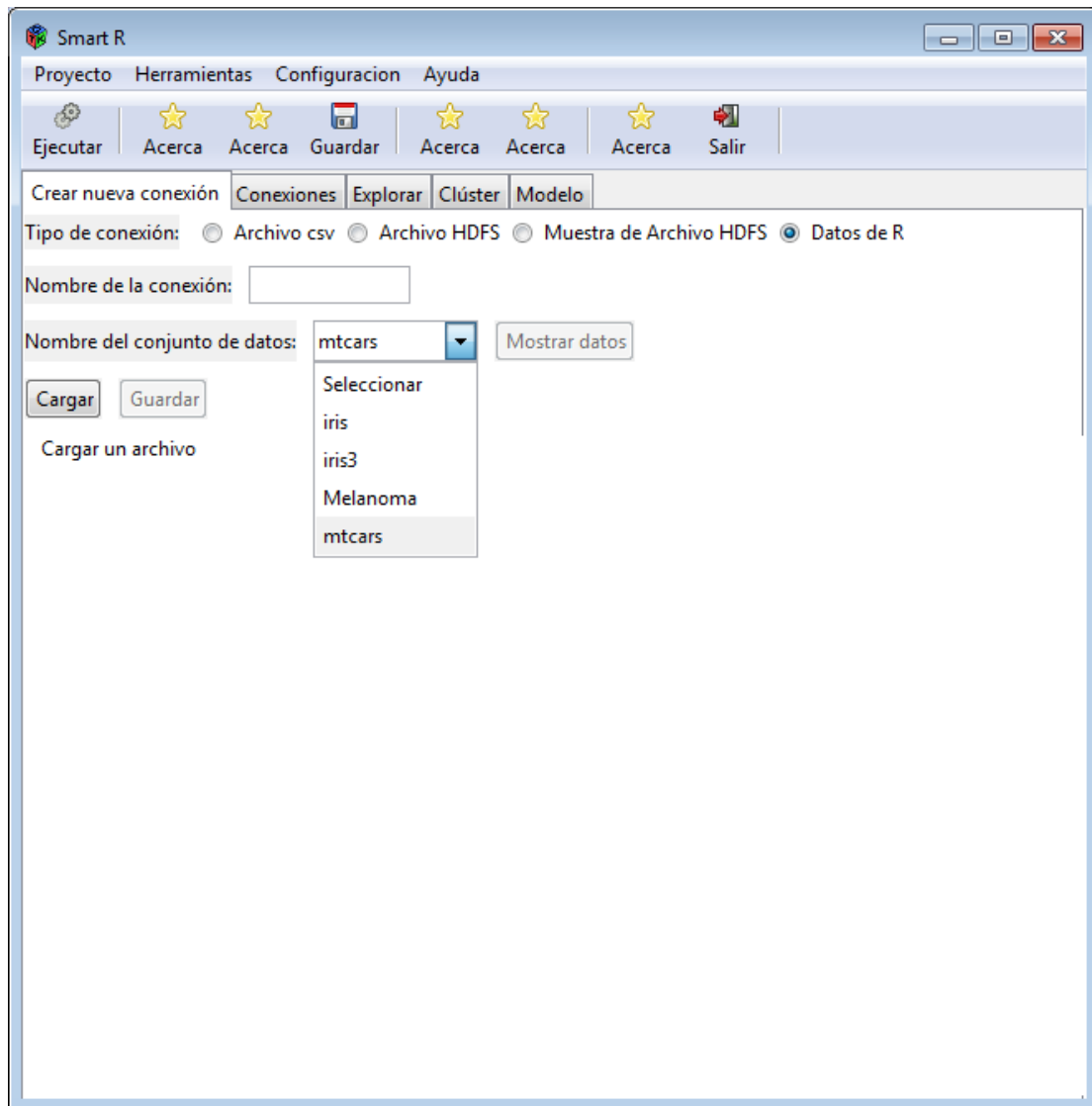


Figura 45: Sección crear nueva conexión del prototipo 4

### 4.3.5.3. Conclusiones

Las conclusiones de este prototipo son las de terminar los objetivos que ha quedado pendiente.

### 4.3.6. Prototipo 5

En este prototipo se cumplieron varios de los objetivos anteriores que no se lograron los cuales se listan a continuación:

## Capítulo 4. Desarrollo de la Solución

### 4.3.6.1. Objetivos

1. Imprimir resultado de las clasificaciones y de los modelos en la aplicación
2. Mostrar funcionalidad que grafique el número óptimo de grupos para el método K medias
3. Posibilidad de descargar los resultados obtenidos de cualquier método ya sea la clasificación o el modelo predictivo generado.

### 4.3.6.2. Resultados

El cumplimiento del objetivo uno se muestra en las siguientes figuras:

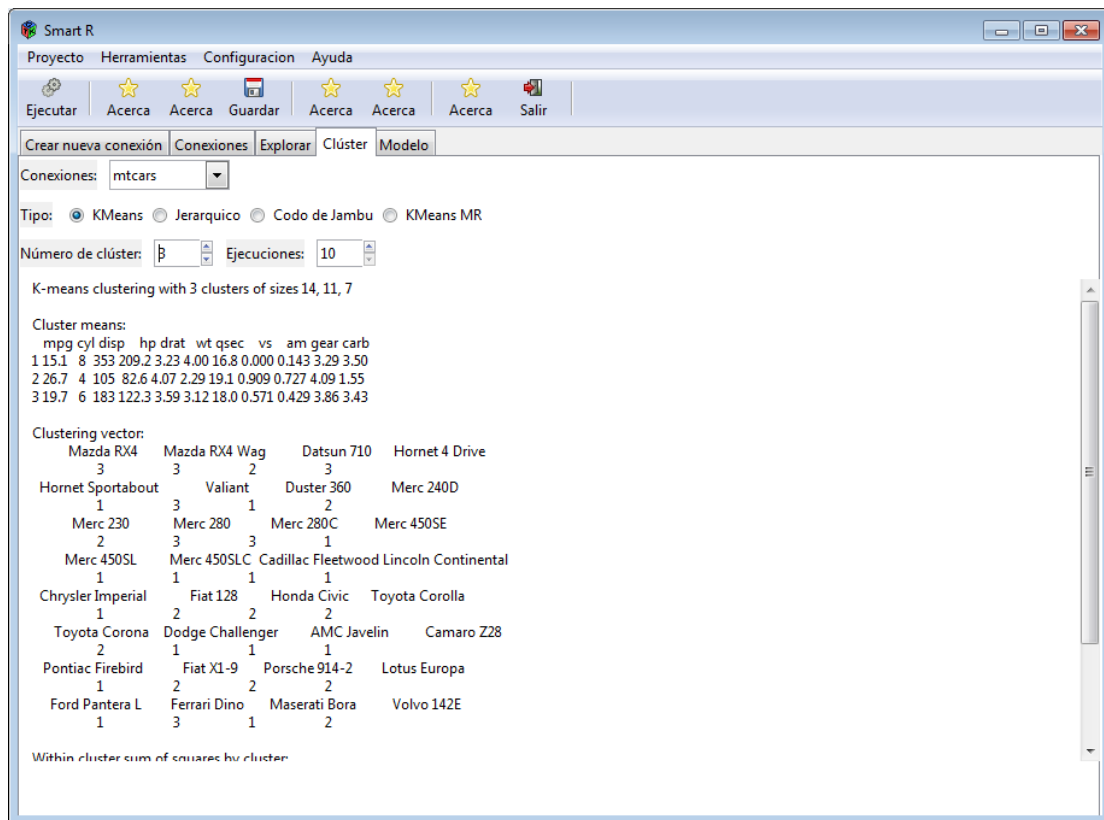


Figura 46: Resultado del método K medias del prototipo 5

## Capítulo 4. Desarrollo de la Solución

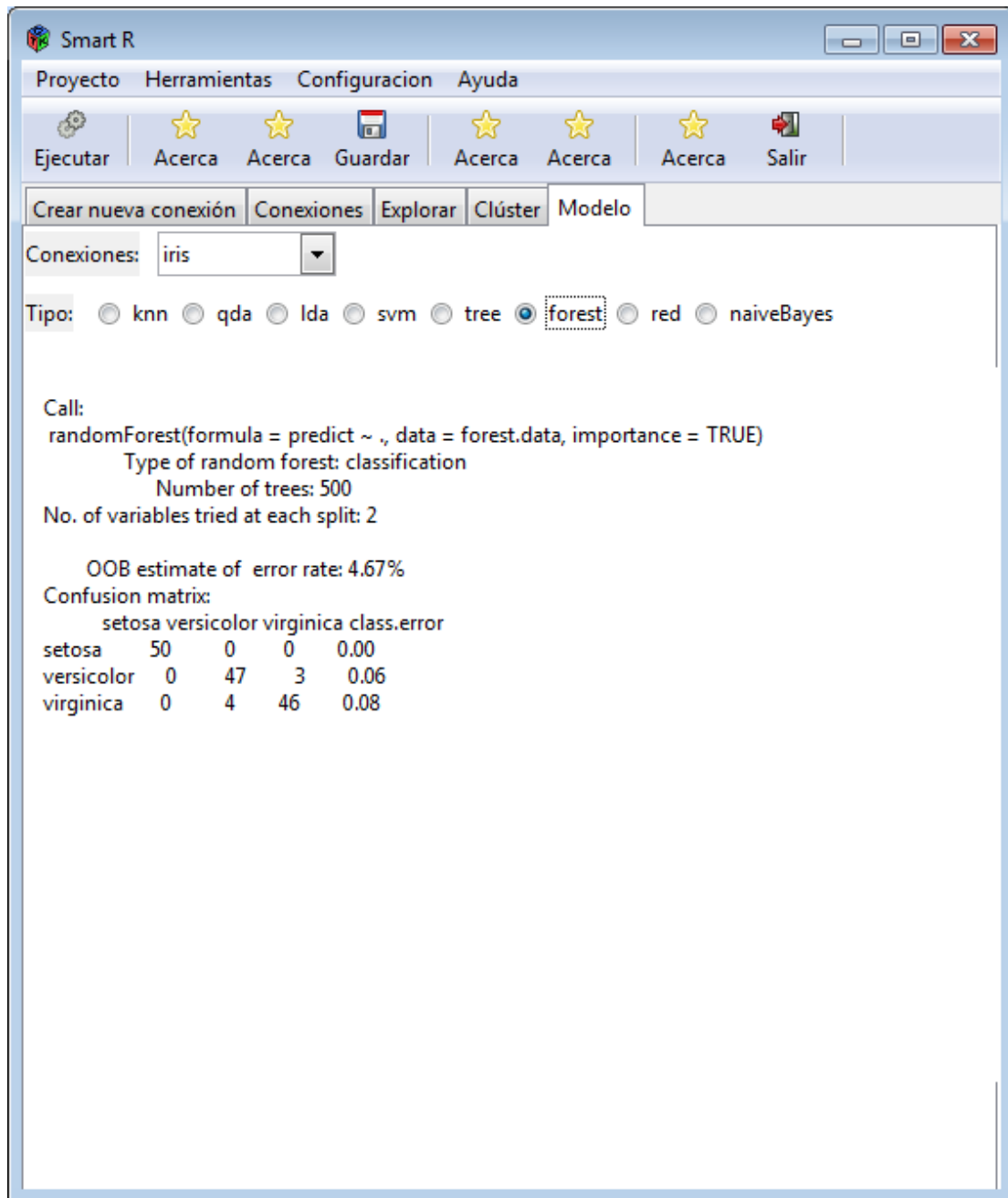


Figura 47: Resultado del método Bosques Aleatorios del prototipo 5

El gráfico que muestra el número de grupos óptimos en cumplimiento con el objetivo dos se muestra a continuación:

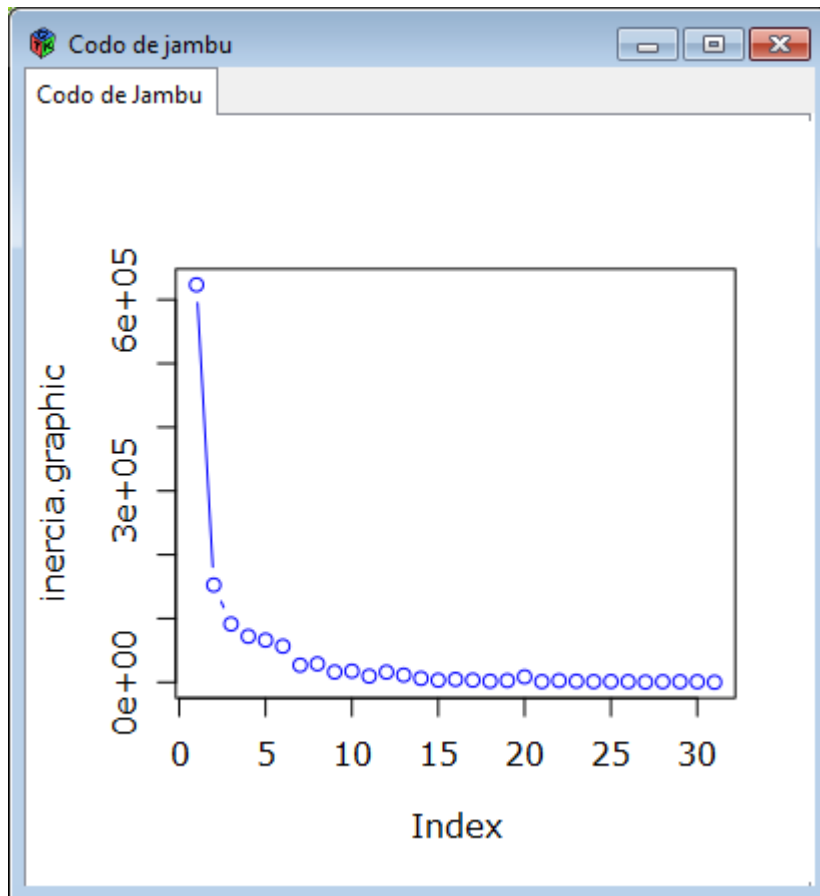


Figura 48: Gráfico del codo de jambu

Por último en cumplimiento con el objetivo tres se cumple que por los métodos de clasificación se descargue un archivo csv con la clasificación obtenida y por los métodos predictivos el modelo generado para poder utilizarlo en otro conjunto de datos desde R.

#### 4.3.6.3. Conclusiones

Las conclusiones principales fueron:

- Es necesario para la investigación poder utilizar un clúster Hadoop por eso se debe programar el tipo de conexiones de archivos HDFS para ejecutar algoritmos en un clúster.

## Capítulo 4. Desarrollo de la Solución

- Es indispensable poder tener conexiones constantes escribiéndolas en un archivo o en algún lugar de almacenamiento.
- Tener sólo un algoritmo MapReduce es muy poco

### 4.3.7. Prototipo 6

Todos los prototipos anteriores solamente eran un archivo, aunque según las buenas prácticas de programación en R dicen que es mejor tener todo en un solo archivo a la hora de codificar resultaba muy engorroso por ende se decidió cambiar la estructura del proyecto en base a una jerarquía de archivos y otros cambios que se listan a continuación:

#### 4.3.7.1. Objetivos

1. Cambiar la estructura de archivos de la aplicación de un archivo a varios módulos
2. Remover botones de más
3. Crear un nuevo tipos de conexión, para Subir funciones MAP y funciones REDUCE
4. Guardar tipo de conexión en la cual se guarden las credenciales de acceso para conectarse a un clúster Hadoop
5. Crear sección para la ejecución de las funciones MAP y REDUCE sobre un clúster Hadoop utilizando la funcionalidad del Hadoop Streaming
6. Mejorar sección donde se muestran las conexiones con la posibilidad de ver el detalle y eliminarlas
7. Colocar tooltips a algunas secciones de la aplicación
8. Ejecutar el algoritmo K medias MapReduce sobre un clúster Hadoop
9. Colocar el algoritmo regresión logística MapReduce para ejecutarlo en un clúster Hadoop

#### 4.3.7.2. Resultados

Como se mencionó de manera previo el proyecto de la aplicación pasó de ser un solo archivo a la siguiente estructura de archivos:

## Capítulo 4. Desarrollo de la Solución

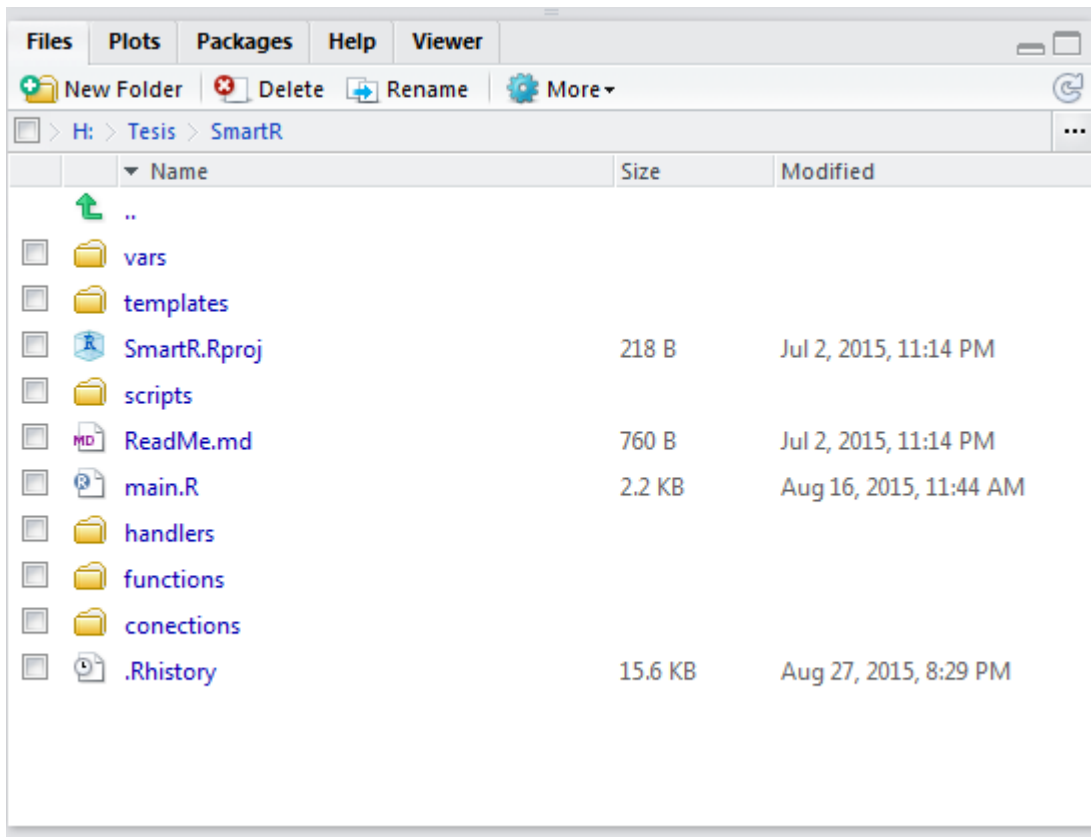


Figura 49: Jerarquía del proyecto

- vars: Almacena un script que contiene la definición de todas las variables a utilizar
- templates: Contiene los scripts necesarios para el diseño de las secciones de la aplicación
- scripts: Almacena de manera dinámica los scripts que serán enviados para la ejecución en el clúster
- handlers: Contiene los scripts que realizan la lógica de negocio
- functions: Contiene las funciones MAP y REDUCE que serán ejecutadas en el clúster
- conections: Contiene el archivo que funciona como almacenamiento de las conexiones de datos.

La aplicación se inicia ejecutando solamente el archivo main.R desde una consola de R.

Los objetivos dos, tres, cuatro, cinco, seis y siete se cumplieron a cabalidad como se muestra en las figuras 50, 51, 52, 53, 54 y 55.



## Capítulo 4. Desarrollo de la Solución

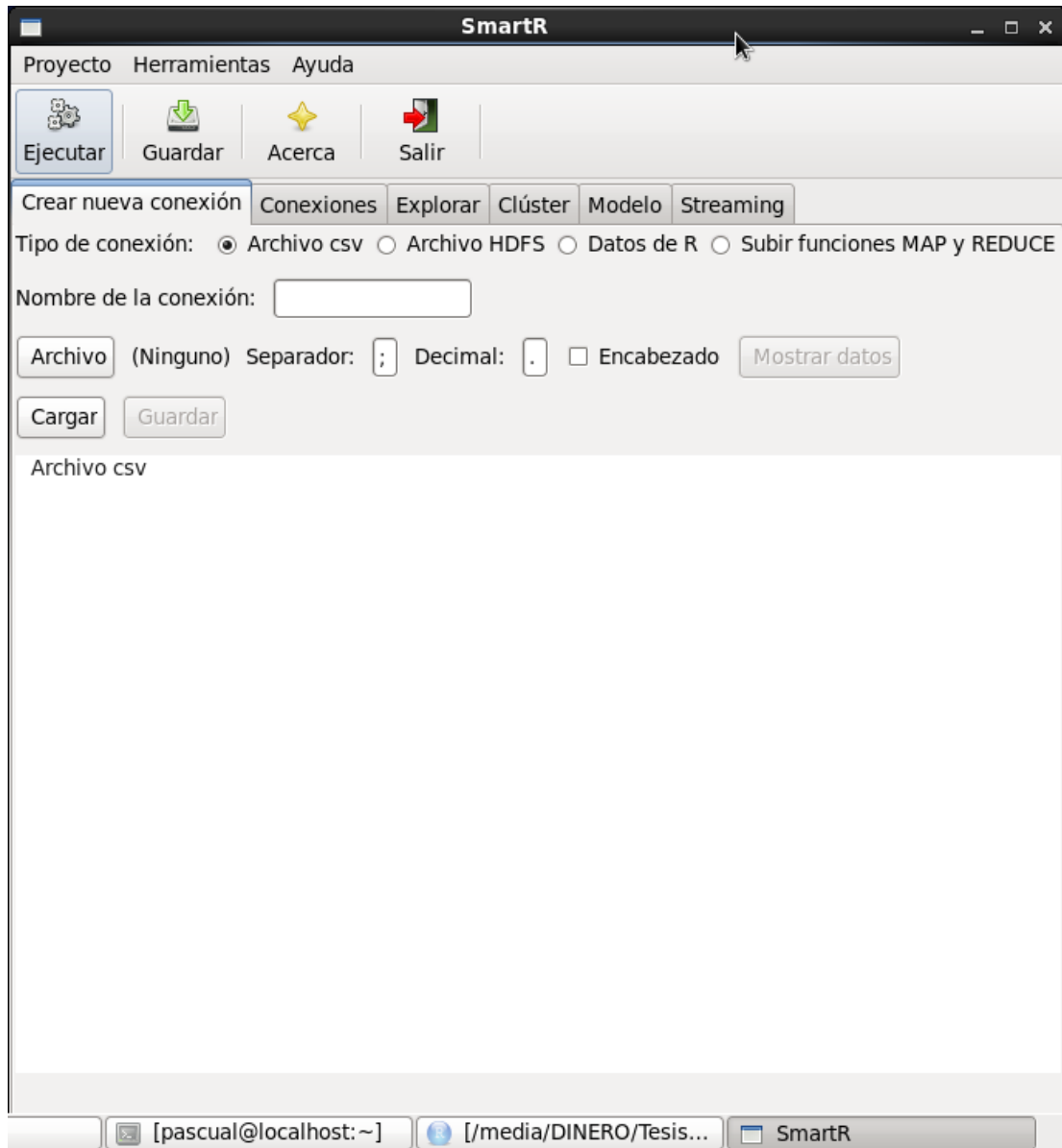


Figura 50: Vista principal del prototipo 6

## Capítulo 4. Desarrollo de la Solución

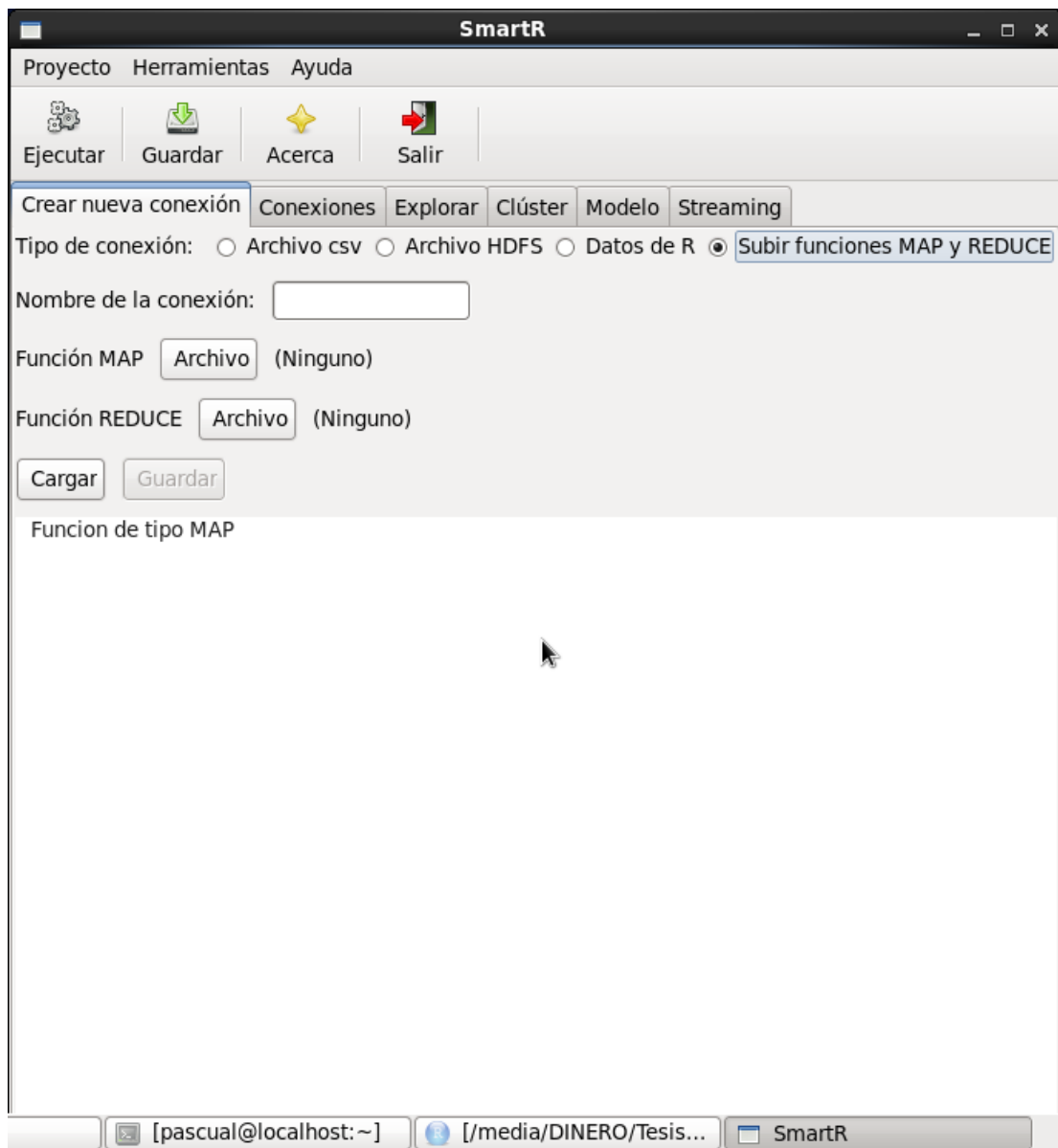


Figura 51: Funcionalidad de Subir funciones MAP y REDUCE del prototipo 6

## Capítulo 4. Desarrollo de la Solución

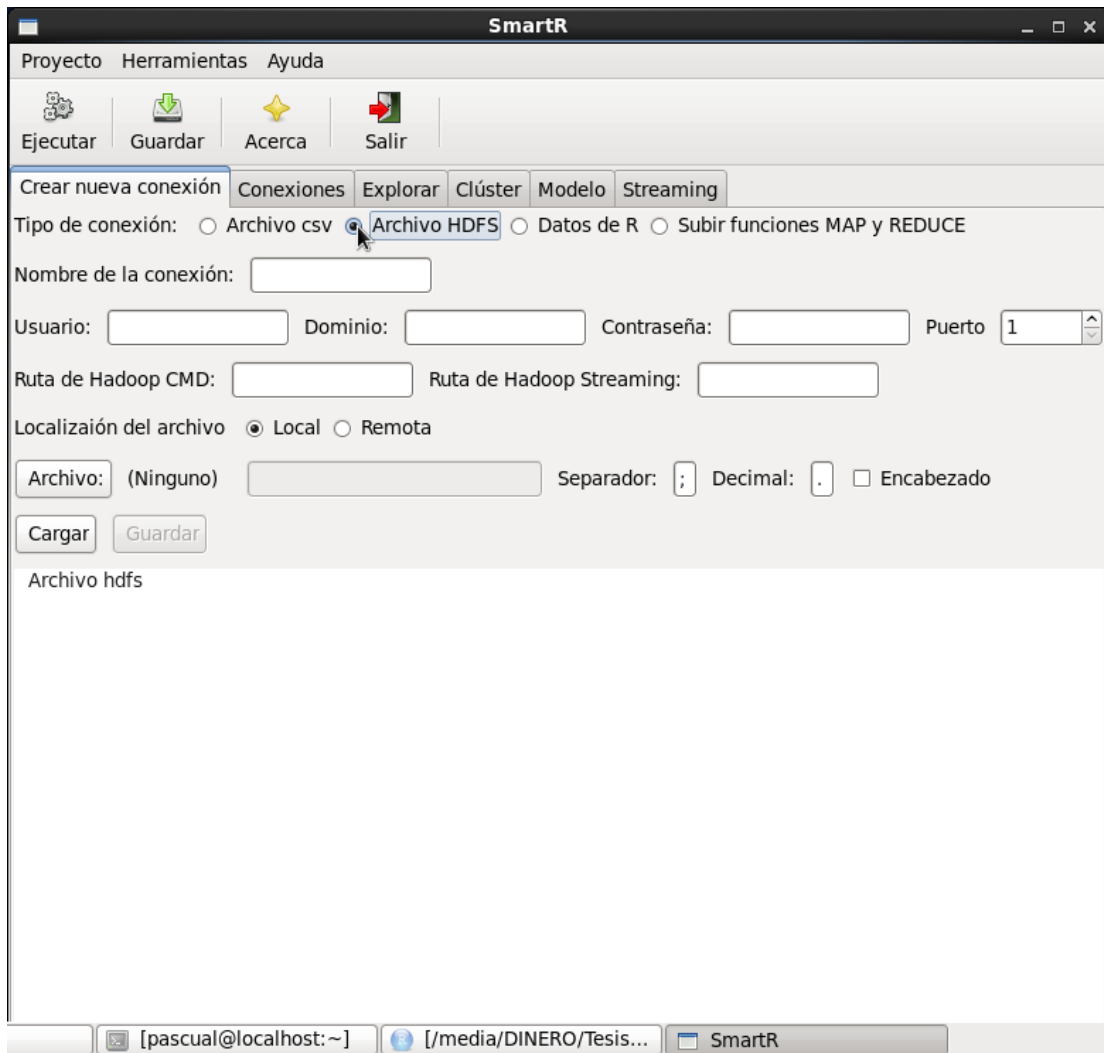


Figura 52: Funcionalidad para crear conexión de tipo Archivo HDFS

## Capítulo 4. Desarrollo de la Solución

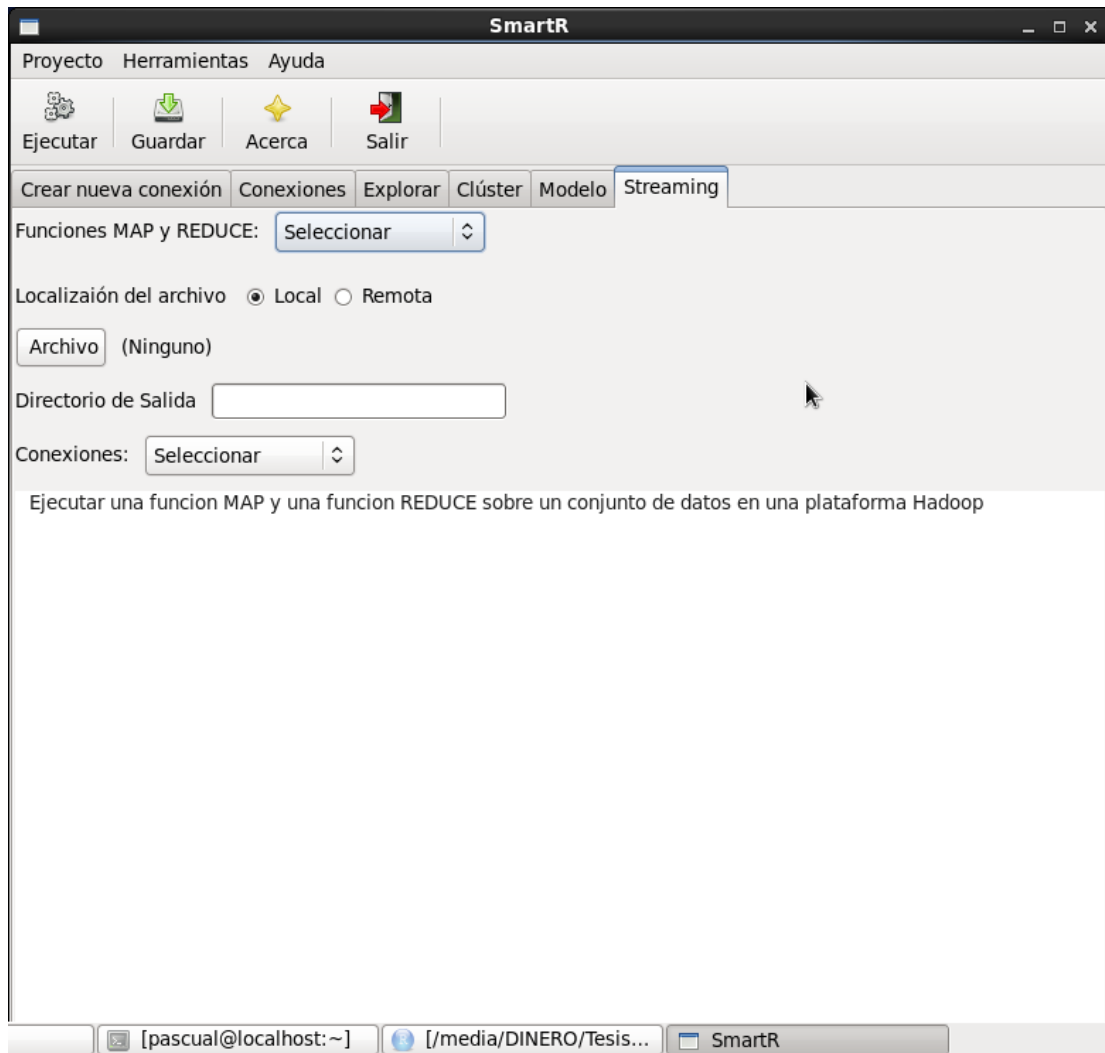


Figura 53: Sección para ejecutar funciones MAP y REDUCE sobre un clúster Hadoop

## Capítulo 4. Desarrollo de la Solución

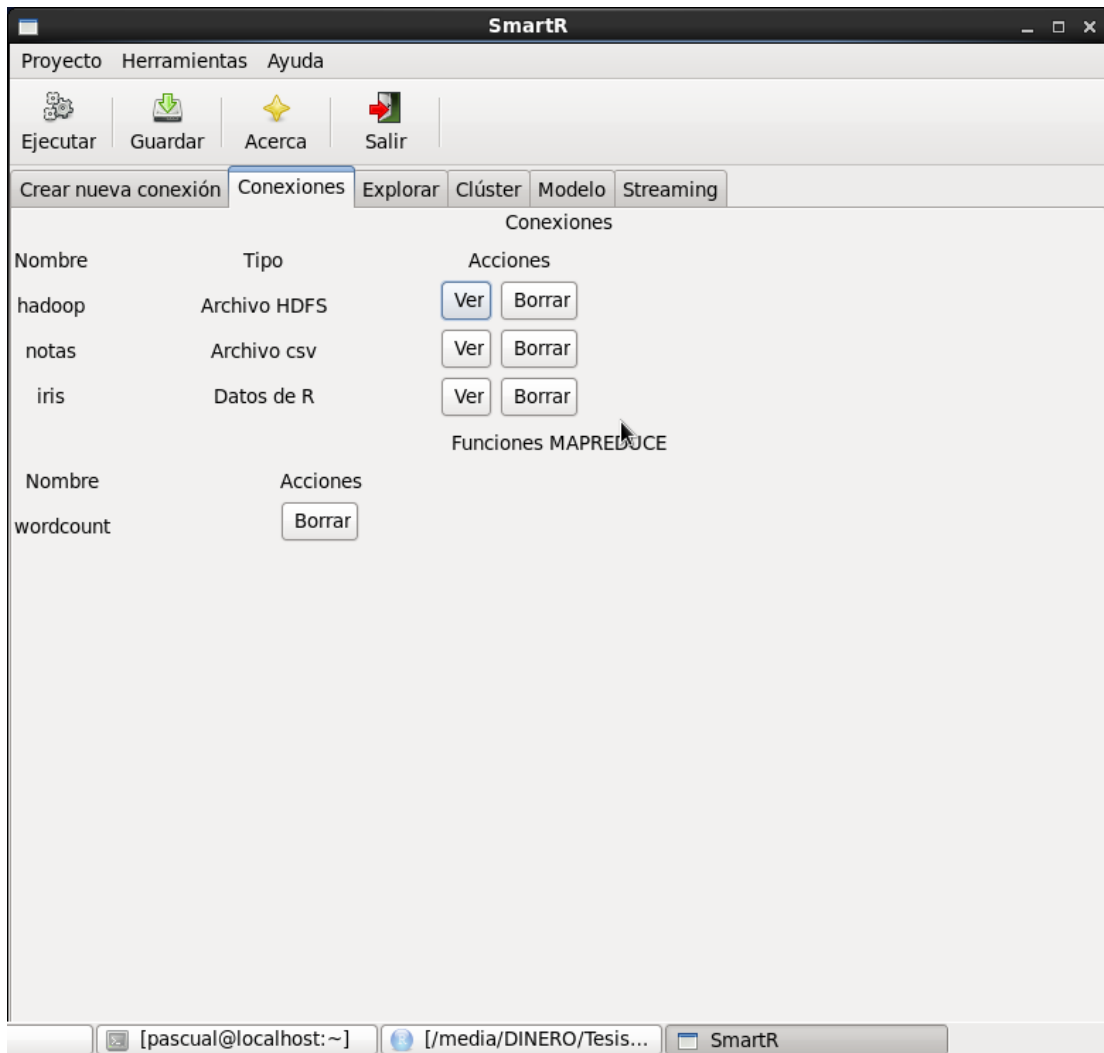


Figura 54: Sección que muestra las conexiones almacenadas del prototipo 6

## Capítulo 4. Desarrollo de la Solución

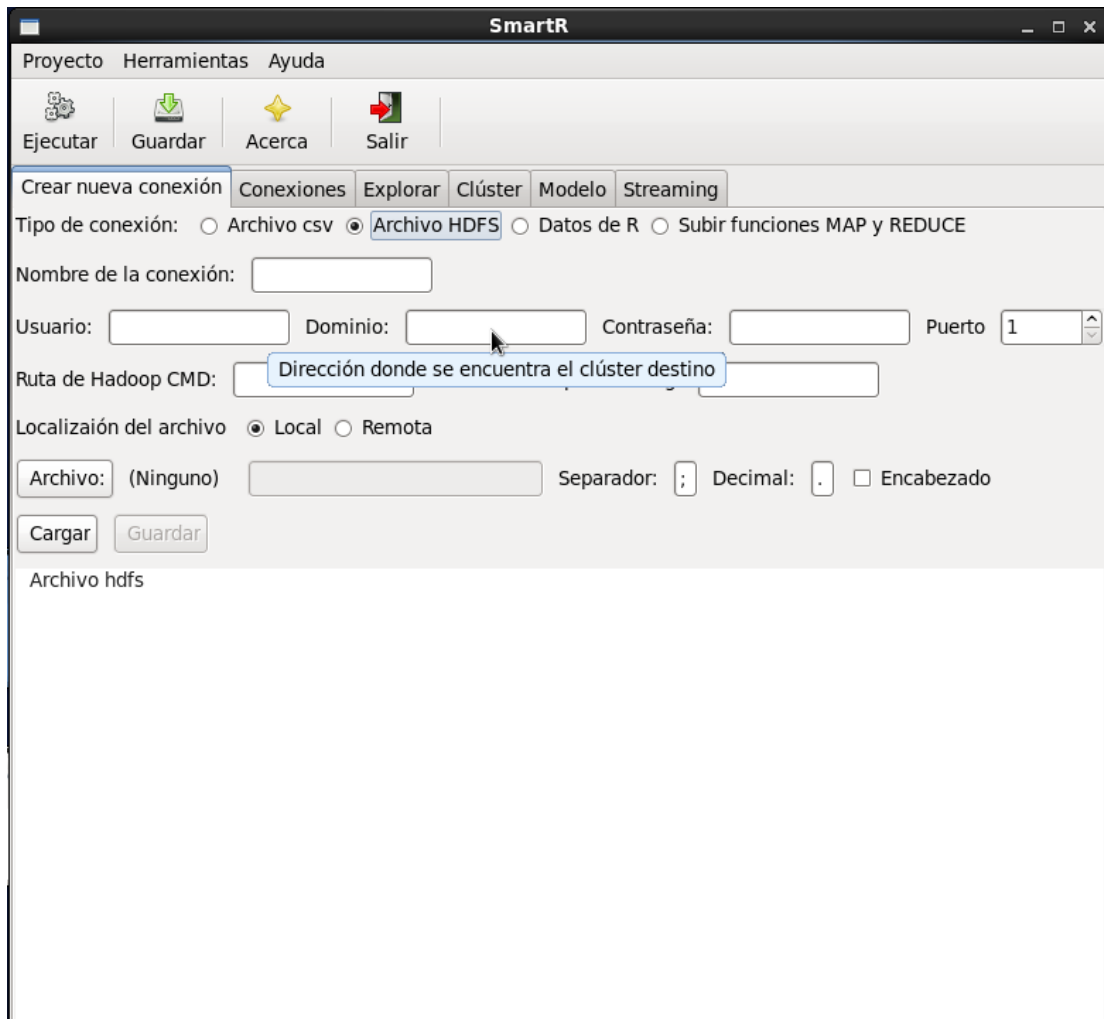


Figura 55: Ejemplo de tooltip en el prototipo 6

El resultado del objetivo ocho se muestra a continuación:

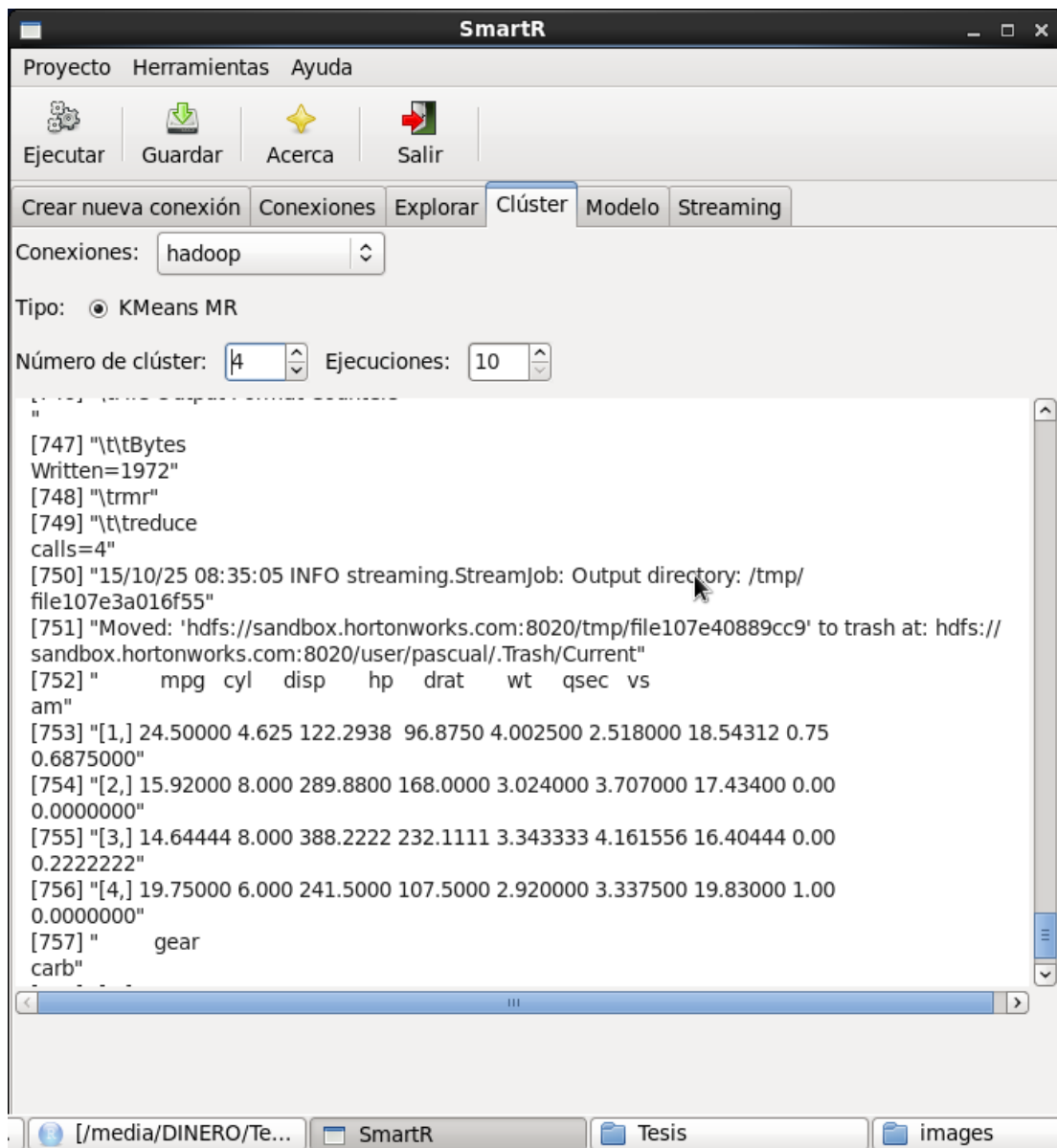


Figura 56: Ejecución del método K medias MapReduce

Se incorporó el algoritmo Regresión Logística bajo el marco MapReduce como se muestra en la siguiente imagen. El método se encuentra en el anexo siete.

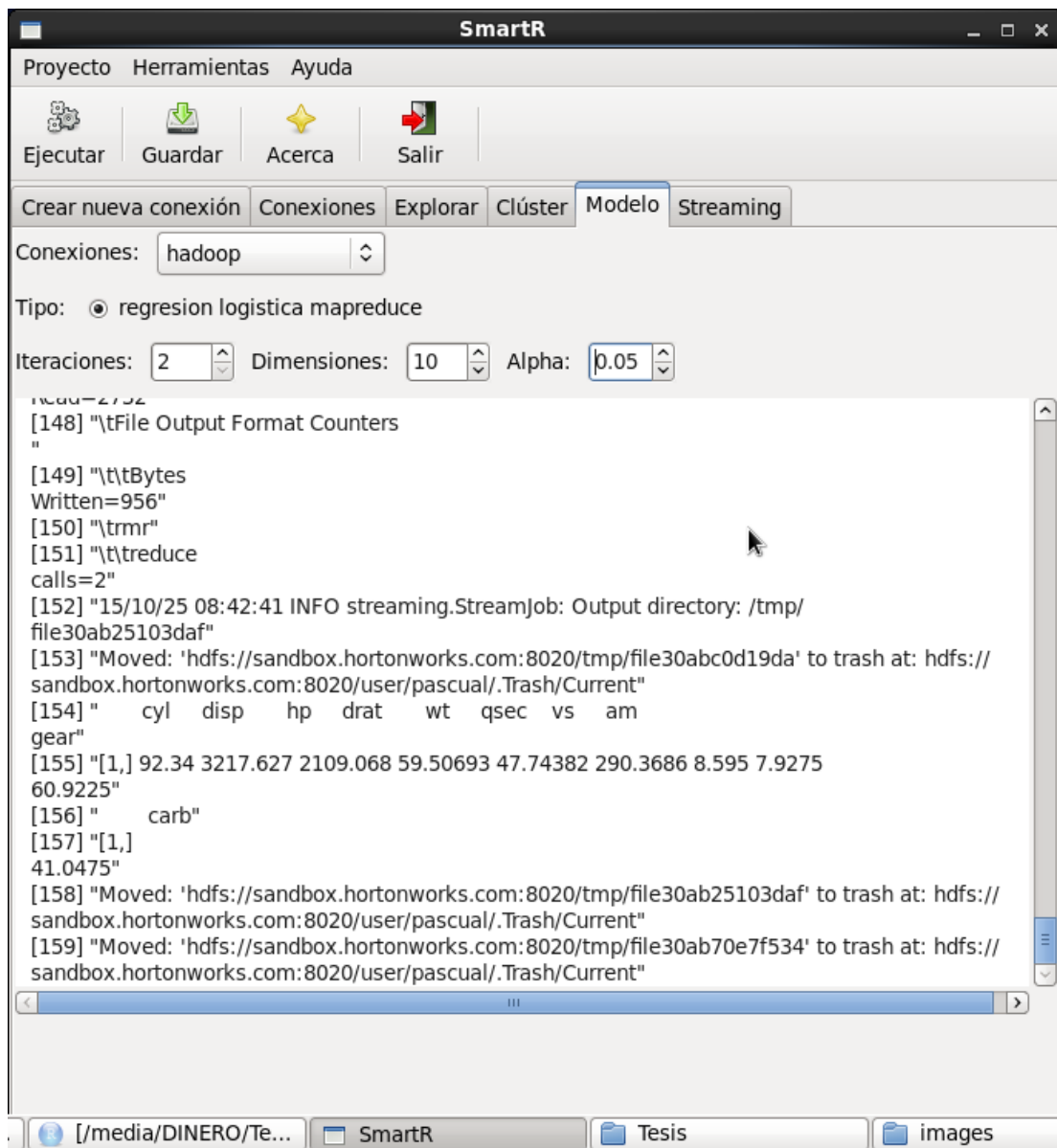


Figura 57: Ejecución del método Regresión Logística MapReduce

### 4.3.7.3. Conclusiones

Las conclusiones para este prototipo se especificaran el próximo capítulo luego de verificar los resultados.



# 5. Conclusiones y Resultados

## 5.1. Resultados

Se realizaron las siguientes pruebas con el fin de obtener los resultados pertinentes para así concluir la investigación.

Para realizar las pruebas se definieron las siguientes conexiones:

- 1) Conexión 1:
  - a. Nombre: mtcars
  - b. Tipo: Archivo csv
  - c. Variable a predecir: Ninguna
  - d. Variables a ignorar: Ninguna
  - e. Estadísticas por Variable:

mpg	cyl	disp	hp	drat	wt
Min. :10.40	Min. :4.000	Min. : 71.1	Min. : 52.0	Min. :2.760	Min. :1.513
1st Qu.:15.43	1st Qu.:4.000	1st Qu.:120.8	1st Qu.: 96.5	1st Qu.:3.080	1st Qu.:2.581
Median :19.20	Median :6.000	Median :196.3	Median :123.0	Median :3.695	Median :3.325
Mean :20.09	Mean :6.188	Mean :230.7	Mean :146.7	Mean :3.597	Mean :3.217
3rd Qu.:22.80	3rd Qu.:8.000	3rd Qu.:326.0	3rd Qu.:180.0	3rd Qu.:3.920	3rd Qu.:3.610
Max. :33.90	Max. :8.000	Max. :472.0	Max. :335.0	Max. :4.930	Max. :5.424
qsec	vs	am	gear	carb	
Min. :14.50	Min. :0.0000	Min. :0.0000	Min. :3.000	Min. :1.000	
1st Qu.:16.89	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.:3.000	1st Qu.:2.000	
Median :17.71	Median :0.0000	Median :0.0000	Median :4.000	Median :2.000	
Mean :17.85	Mean :0.4375	Mean :0.4062	Mean :3.688	Mean :2.812	
3rd Qu.:18.90	3rd Qu.:1.0000	3rd Qu.:1.0000	3rd Qu.:4.000	3rd Qu.:4.000	
Max. :22.90	Max. :1.0000	Max. :1.0000	Max. :5.000	Max. :8.000	

Figura 58: Estadísticas de la Conexión 1

- 2) Conexión 2:
  - a. Nombre: iris
  - b. Tipo: Datos de R
  - c. Variable a predecir: Species
  - d. Variables a ignorar: Ninguna
  - e. Porcentaje de muestreo: 70
  - f. Porcentaje de prueba: 30
  - g. Estadísticas por Variable:

## Capítulo 5. Conclusiones y Resultados

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100	setosa :50
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300	versicolor:50
Median :5.800	Median :3.000	Median :4.350	Median :1.300	virginica :50
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199	
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800	
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	

Figura 59: Estadísticas de la Conexión 2

### 3) Conexión 3:

- a. Nombre: hadoop
- b. Tipo: Archivo HDFS
- c. Usuario: pascual
- d. Contraseña: hadoop
- e. Dominio: 192.168.0.107
- f. Puerto: 2222
- g. Hadoop\_cmd: /usr/lib/hadoop/bin/hadoop
- h. Hadoop\_streaming: /usr/lib/hadoop-mapreduce/hadoop-streaming-2.4.0.2.1.1.0-385.jar
- i. Archivo: El mismo que se utilizó en la conexión 1 ya almacenado en HDFS

### 4) Conexión 4:

- a. Nombre: wordcount
- b. Función MAP: La función que se utilizó se puede ver en el anexo 8
- c. Función REDUCE: La función que se utilizó se puede ver en el anexo 9

Luego de tener definidas las conexiones se procedió a probar las funcionalidades más relevantes de la aplicación.

### I. Componentes principales:

Utilizando como entrada la Conexión 1, generando cinco componentes y graficando las componentes uno y dos se tuvieron los siguientes resultados:

## Capítulo 5. Conclusiones y Resultados

Eigenvalues												
	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5	Dim.6	Dim.7	Dim.8	Dim.9	Dim.10	Dim.11	
Variance	6.608	2.650	0.627	0.270	0.223	0.212	0.135	0.123	0.077	0.052	0.022	
% of var.	60.076	24.095	5.702	2.451	2.031	1.924	1.230	1.117	0.700	0.473	0.200	
Cumulative % of var.	60.076	84.172	89.873	92.324	94.356	96.279	97.509	98.626	99.327	99.800	100.000	
Individuals (the 10 first)												
	Dist	Dim.1	ctr	cos2	Dim.2	ctr	cos2	Dim.3	ctr	cos2		
Mazda RX4	2.234	-0.657	0.204	0.087	1.735	3.551	0.604	-0.601	1.801	0.072		
Mazda RX4 Wag	2.081	-0.629	0.187	0.091	1.550	2.833	0.555	-0.382	0.728	0.034		
Datsun 710	2.987	-2.779	3.653	0.866	-0.146	0.025	0.002	-0.241	0.290	0.007		
Hornet 4 Drive	2.521	-0.312	0.046	0.015	-2.363	6.584	0.879	-0.136	0.092	0.003		
Hornet Sportabout	2.456	1.974	1.844	0.646	-0.754	0.671	0.094	-1.134	6.412	0.213		
Valiant	3.014	-0.056	0.001	0.000	-2.786	9.151	0.855	0.164	0.134	0.003		
Duster 360	3.187	3.003	4.264	0.888	0.335	0.132	0.011	-0.363	0.656	0.013		
Merc 240D	2.841	-2.055	1.998	0.523	-1.465	2.531	0.266	0.944	4.439	0.110		
Merc 230	3.733	-2.287	2.474	0.375	-1.984	4.639	0.282	1.797	16.094	0.232		
Merc 280	1.907	-0.526	0.131	0.076	-0.162	0.031	0.007	1.493	11.103	0.613		
Variables (the 10 first)												
	Dim.1	ctr	cos2	Dim.2	ctr	cos2	Dim.3	ctr	cos2			
mpg	-0.932	13.143	0.869	0.026	0.026	0.001	-0.179	5.096	0.032			
cyl	0.961	13.981	0.924	0.071	0.191	0.005	-0.139	3.073	0.019			
disp	0.946	13.556	0.896	-0.080	0.243	0.006	-0.049	0.378	0.002			
hp	0.848	10.894	0.720	0.405	6.189	0.164	0.111	1.960	0.012			
drat	-0.756	8.653	0.572	0.447	7.546	0.200	0.128	2.598	0.016			
wt	0.890	11.979	0.792	-0.233	2.046	0.054	0.271	11.684	0.073			
qsec	-0.515	4.018	0.266	-0.754	21.472	0.569	0.319	16.255	0.102			
vs	-0.788	9.395	0.621	-0.377	5.366	0.142	0.340	18.388	0.115			
am	-0.604	5.520	0.365	0.699	18.440	0.489	-0.163	4.234	0.027			
gear	-0.532	4.281	0.283	0.753	21.377	0.567	0.229	8.397	0.053			

Figura 60: Resultado Análisis de Componentes Principales

## Capítulo 5. Conclusiones y Resultados

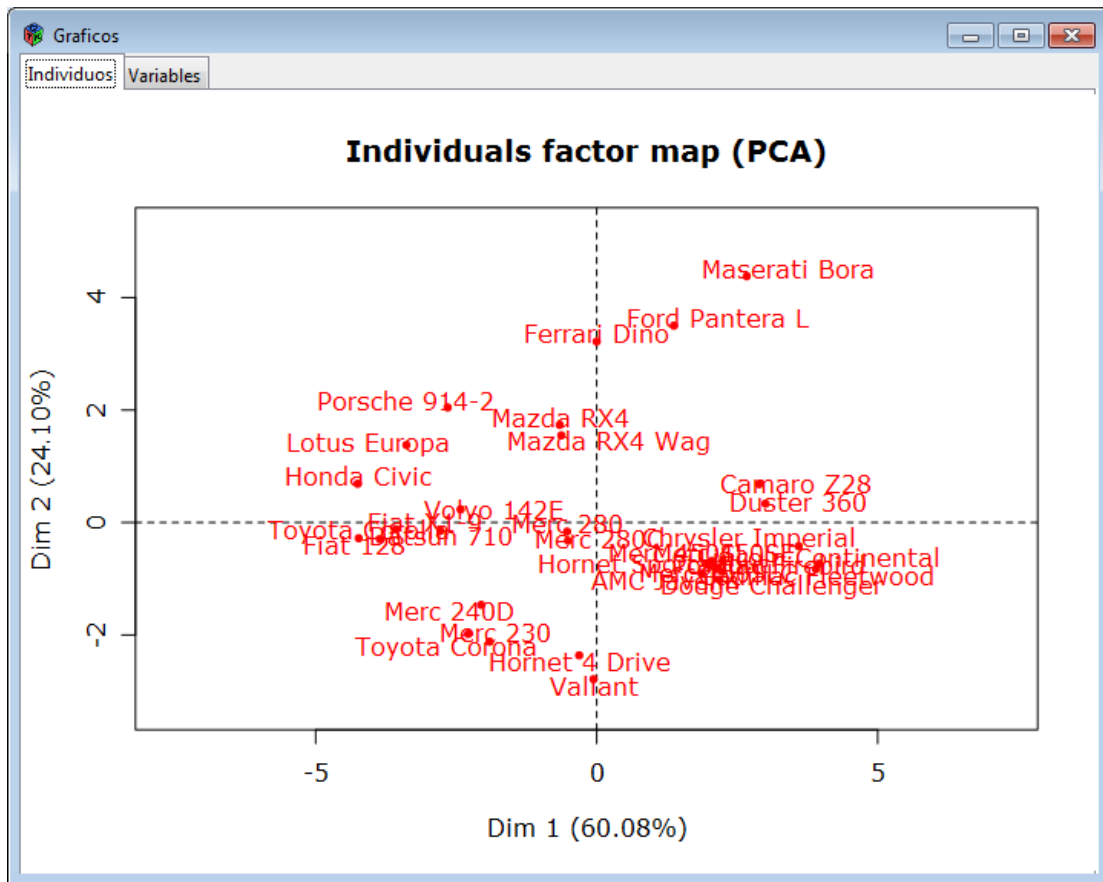


Figura 61: Resultado Análisis de Componentes Principales (Individuos)

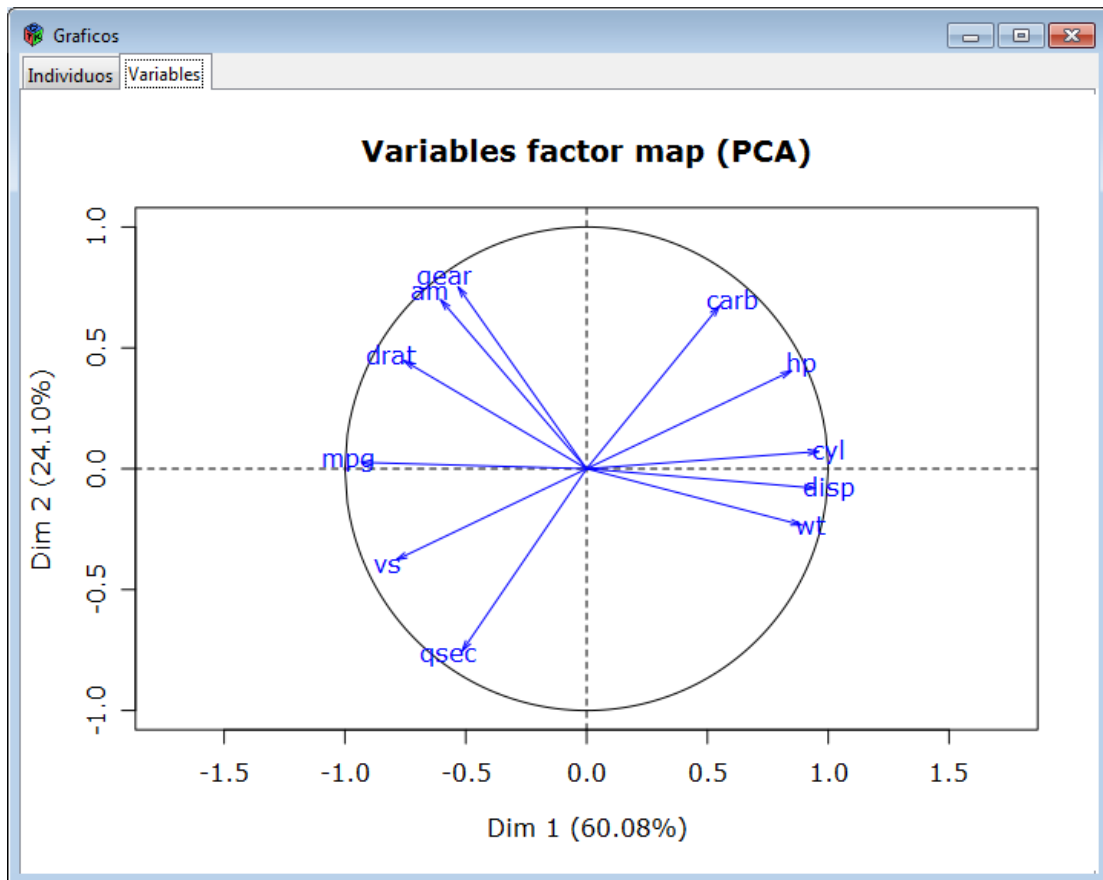


Figura 62: Resultados Análisis de Componentes Principales (Variables)

## II. K medias

Utilizando la Conexión 1 con un número de grupos igual a cuatro y un número de iteraciones igual a diez se obtuvieron los siguientes resultados en el método de K medias disponible para las conexiones locales:

## Capítulo 5. Conclusiones y Resultados

```

K-means clustering with 4 clusters of sizes 10, 7, 6, 9

Cluster means:
      mpg      cyl      disp      hp      drat      wt      qsec      vs      am      gear      carb
1 27.05000  4.000000 101.5700  81.4000  4.086000  2.199300 18.76100  0.9000000  0.8000000  4.100000  1.500000
2 19.94286  5.714286 166.5714 120.1429  3.705714  3.107857 18.47143  0.5714286  0.4285714  4.000000  3.571429
3 16.83333  7.666667 284.5667 158.3333  3.033333  3.625000 17.76833  0.1666667  0.0000000  3.000000  2.333333
4 14.64444  8.000000 388.2222 232.1111  3.343333  4.161556 16.40444  0.0000000  0.2222222  3.444444  4.000000

Clustering vector:
      Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive      Hornet Sportabout
      2              2              1              3              4
      Valiant      Duster 360      Merc 240D      Merc 230      Merc 280
      2              4              1              2              2
      Merc 280C      Merc 450SE      Merc 450SL      Merc 450SLC      Cadillac Fleetwood
      2              3              3              3              4
Lincoln Continental  Chrysler Imperial      Fiat 128      Honda Civic      Toyota Corolla
      4              4              1              1              1
      Toyota Corona      Dodge Challenger      AMC Javelin      Camaro Z28      Pontiac Firebird
      1              3              3              4              4
      Fiat X1-9      Porsche 914-2      Lotus Europa      Ford Pantera L      Ferrari Dino
      1              1              1              4              2
      Maserati Bora      Volvo 142E
      4              1

Within cluster sum of squares by cluster:
[1] 10247.471  8808.032  6355.581 46659.317
(between_SS / total_SS =  88.4 %)

```

Figura 63: Resultados del método K medias

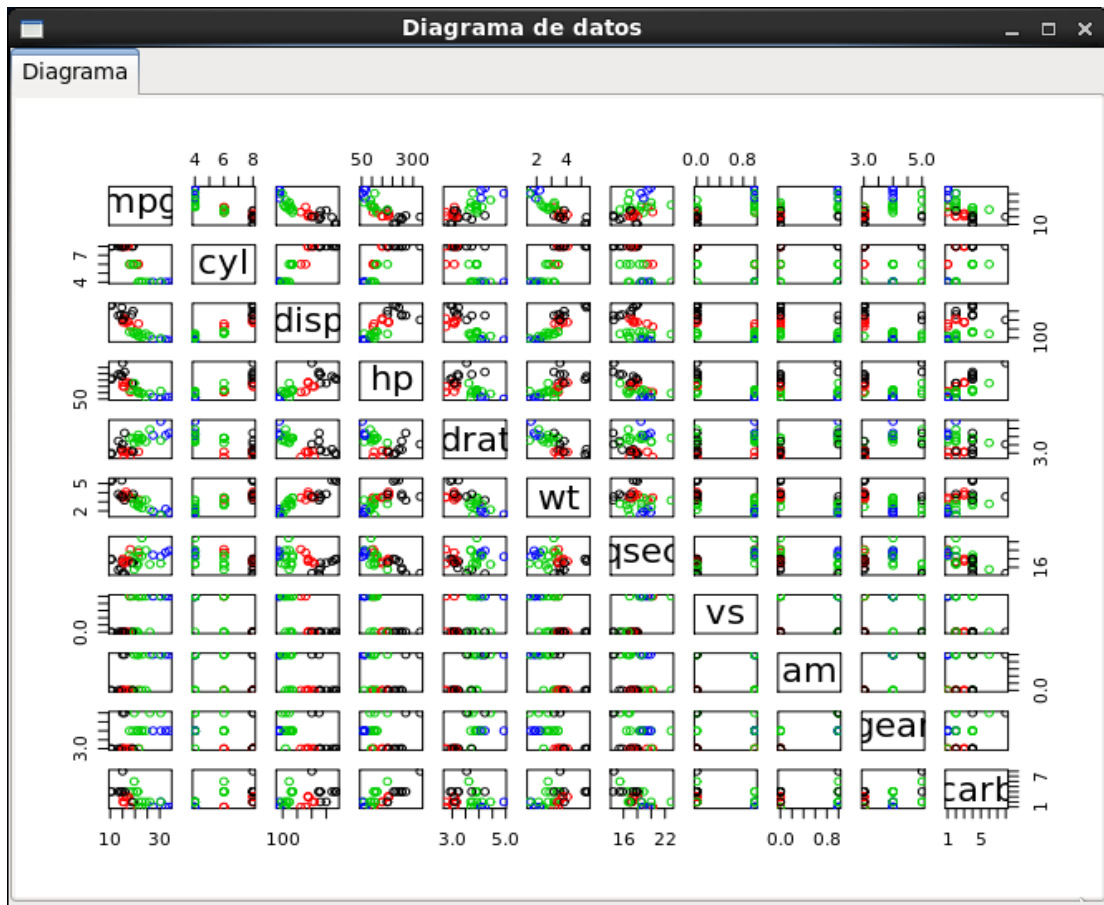


Figura 64: Diagrama de datos del método K medias

## Capítulo 5. Conclusiones y Resultados

```
"", "mpg", "cyl", "disp", "hp", "drat", "wt", "qsec", "vs", "am", "gear", "carb", "clasification"
"Mazda RX4", 21, 6, 160, 110, 3.9, 2.62, 16.46, 0, 1, 4, 4, 2
"Mazda RX4 Wag", 21, 6, 160, 110, 3.9, 2.875, 17.02, 0, 1, 4, 4, 2
"Datsun 710", 22.8, 4, 108, 93, 3.85, 2.32, 18.61, 1, 1, 4, 1, 1
"Hornet 4 Drive", 21.4, 6, 258, 110, 3.08, 3.215, 19.44, 1, 0, 3, 1, 3
"Hornet Sportabout", 18.7, 8, 360, 175, 3.15, 3.44, 17.02, 0, 0, 3, 2, 4
"Valiant", 18.1, 6, 225, 105, 2.76, 3.46, 20.22, 1, 0, 3, 1, 2
"Duster 360", 14.3, 8, 360, 245, 3.21, 3.57, 15.84, 0, 0, 3, 4, 4
"Merc 240D", 24.4, 4, 146.7, 62, 3.69, 3.19, 20, 1, 0, 4, 2, 1
"Merc 230", 22.8, 4, 140.8, 95, 3.92, 3.15, 22.9, 1, 0, 4, 2, 2
"Merc 280", 19.2, 6, 167.6, 123, 3.92, 3.44, 18.3, 1, 0, 4, 4, 2
"Merc 280C", 17.8, 6, 167.6, 123, 3.92, 3.44, 18.9, 1, 0, 4, 4, 2
"Merc 450SE", 16.4, 8, 275.8, 180, 3.07, 4.07, 17.4, 0, 0, 3, 3, 3
"Merc 450SL", 17.3, 8, 275.8, 180, 3.07, 3.73, 17.6, 0, 0, 3, 3, 3
"Merc 450SLC", 15.2, 8, 275.8, 180, 3.07, 3.78, 18, 0, 0, 3, 3, 3
"Cadillac Fleetwood", 10.4, 8, 472, 205, 2.93, 5.25, 17.98, 0, 0, 3, 4, 4
"Lincoln Continental", 10.4, 8, 460, 215, 3, 5.424, 17.82, 0, 0, 3, 4, 4
"Chrysler Imperial", 14.7, 8, 440, 230, 3.23, 5.345, 17.42, 0, 0, 3, 4, 4
"Fiat 128", 32.4, 4, 78.7, 66, 4.08, 2.2, 19.47, 1, 1, 4, 1, 1
"Honda Civic", 30.4, 4, 75.7, 52, 4.93, 1.615, 18.52, 1, 1, 4, 2, 1
"Toyota Corolla", 33.9, 4, 71.1, 65, 4.22, 1.835, 19.9, 1, 1, 4, 1, 1
"Toyota Corona", 21.5, 4, 120.1, 97, 3.7, 2.465, 20.01, 1, 0, 3, 1, 1
"Dodge Challenger", 15.5, 8, 318, 150, 2.76, 3.52, 16.87, 0, 0, 3, 2, 3
"AMC Javelin", 15.2, 8, 304, 150, 3.15, 3.435, 17.3, 0, 0, 3, 2, 3
"Camaro Z28", 13.3, 8, 350, 245, 3.73, 3.84, 15.41, 0, 0, 3, 4, 4
"Pontiac Firebird", 19.2, 8, 400, 175, 3.08, 3.845, 17.05, 0, 0, 3, 2, 4
"Fiat X1-9", 27.3, 4, 79, 66, 4.08, 1.935, 18.9, 1, 1, 4, 1, 1
"Porsche 914-2", 26, 4, 120.3, 91, 4.43, 2.14, 16.7, 0, 1, 5, 2, 1
```

Figura 65: Clasificación después de aplicar el método K medias

### III. Agrupamiento Jerárquico

Utilizando la Conexión 1 con un parámetro de distancia igual a Euclidean y un parámetro de aglomeración igual a Ward.D se obtuvieron los siguientes resultados en el método de Agrupamiento Jerárquico:



## Capítulo 5. Conclusiones y Resultados

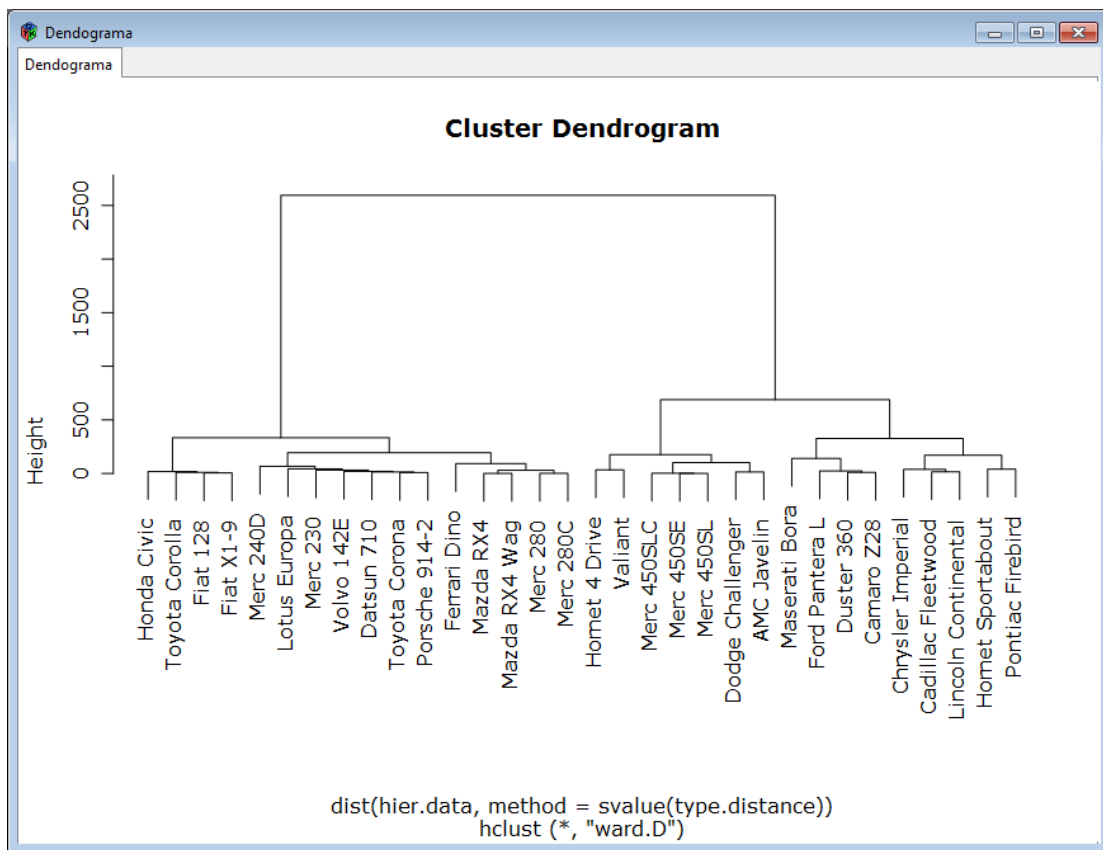


Figura 66: Resultado Agrupamiento Jerárquico

Tomando un número de grupos igual a cuatro se obtuvieron los siguientes resultados:

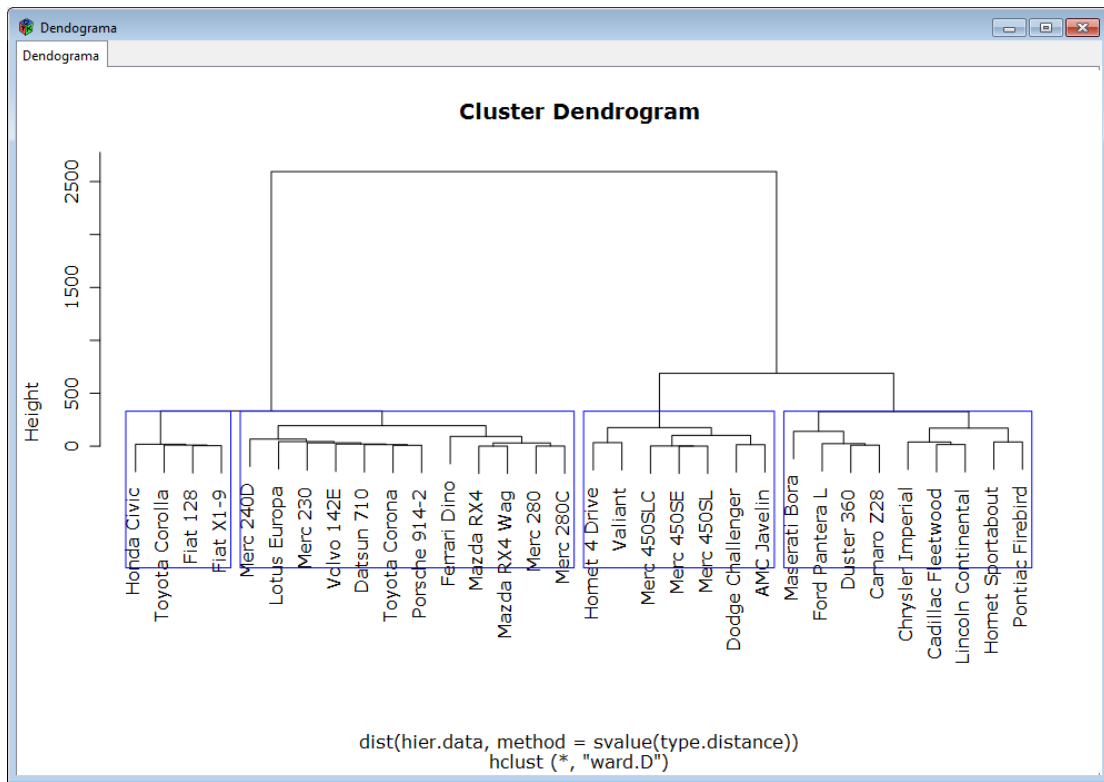


Figura 67: Clasificación del método Agrupamiento Jerárquico

## Capítulo 5. Conclusiones y Resultados

```
"", "mpg", "cyl", "disp", "hp", "drat", "wt", "qsec", "vs", "am", "gear", "carb", "clasificacion"
"Mazda RX4", 21, 6, 160, 110, 3.9, 2.62, 16.46, 0, 1, 4, 4, 1
"Mazda RX4 Wag", 21, 6, 160, 110, 3.9, 2.875, 17.02, 0, 1, 4, 4, 1
"Datsun 710", 22.8, 4, 108, 93, 3.85, 2.32, 18.61, 1, 1, 4, 1, 1
"Hornet 4 Drive", 21.4, 6, 258, 110, 3.08, 3.215, 19.44, 1, 0, 3, 1, 2
"Hornet Sportabout", 18.7, 8, 360, 175, 3.15, 3.44, 17.02, 0, 0, 3, 2, 3
"Valiant", 18.1, 6, 225, 105, 2.76, 3.46, 20.22, 1, 0, 3, 1, 2
"Duster 360", 14.3, 8, 360, 245, 3.21, 3.57, 15.84, 0, 0, 3, 4, 3
"Merc 240D", 24.4, 4, 146.7, 62, 3.69, 3.19, 20, 1, 0, 4, 2, 1
"Merc 230", 22.8, 4, 140.8, 95, 3.92, 3.15, 22.9, 1, 0, 4, 2, 1
"Merc 280", 19.2, 6, 167.6, 123, 3.92, 3.44, 18.3, 1, 0, 4, 4, 1
"Merc 280C", 17.8, 6, 167.6, 123, 3.92, 3.44, 18.9, 1, 0, 4, 4, 1
"Merc 450SE", 16.4, 8, 275.8, 180, 3.07, 4.07, 17.4, 0, 0, 3, 3, 2
"Merc 450SL", 17.3, 8, 275.8, 180, 3.07, 3.73, 17.6, 0, 0, 3, 3, 2
"Merc 450SLC", 15.2, 8, 275.8, 180, 3.07, 3.78, 18, 0, 0, 3, 3, 2
"Cadillac Fleetwood", 10.4, 8, 472, 205, 2.93, 5.25, 17.98, 0, 0, 3, 4, 3
"Lincoln Continental", 10.4, 8, 460, 215, 3.5, 5.424, 17.82, 0, 0, 3, 4, 3
"Chrysler Imperial", 14.7, 8, 440, 230, 3.23, 5.345, 17.42, 0, 0, 3, 4, 3
"Fiat 128", 32.4, 4, 78.7, 66, 4.08, 2.2, 19.47, 1, 1, 4, 1, 4
"Honda Civic", 30.4, 4, 75.7, 52, 4.93, 1.615, 18.52, 1, 1, 4, 2, 4
"Toyota Corolla", 33.9, 4, 71.1, 65, 4.22, 1.835, 19.9, 1, 1, 4, 1, 4
"Toyota Corona", 21.5, 4, 120.1, 97, 3.7, 2.465, 20.01, 1, 0, 3, 1, 1
"Dodge Challenger", 15.5, 8, 318, 150, 2.76, 3.52, 16.87, 0, 0, 3, 2, 2
"AMC Javelin", 15.2, 8, 304, 150, 3.15, 3.435, 17.3, 0, 0, 3, 2, 2
"Camaro Z28", 13.3, 8, 350, 245, 3.73, 3.84, 15.41, 0, 0, 3, 4, 3
"Pontiac Firebird", 19.2, 8, 400, 175, 3.08, 3.845, 17.05, 0, 0, 3, 2, 3
"Fiat X1-9", 27.3, 4, 79, 66, 4.08, 1.935, 18.9, 1, 1, 4, 1, 4
"Porsche 914-2", 26, 4, 120.3, 91, 4.43, 2.14, 16.7, 0, 1, 5, 2, 1
"Lotus Europa", 30.4, 4, 95.1, 113, 3.77, 1.513, 16.9, 1, 1, 5, 2, 1
"Ford Pantera L", 15.8, 8, 351, 264, 4.22, 3.17, 14.5, 0, 1, 5, 4, 3
```

Figura 68: Clasificación después de ejecutar el método Agrupamiento Jerárquico

### IV. K medias MapReduce

Utilizando la Conexión 3 con un número de grupos igual a cuatro y un número de iteraciones igual a diez se obtuvieron los siguientes resultados en el método de K medias disponible para las conexiones que corren sobre un clúster Hadoop:

```
mpg    cyl    disp    hp    drat    wt    qsec    vs    am    gear    carb
[1,]  24.500  4.625 122.2938  96.875  4.00250  2.5180 18.54312  0.75  0.6875  4.125  2.4375
[2,]  17.550  7.000 276.2500 128.750  2.93750  3.4075 18.45750  0.50  0.0000  3.000  1.5000
[3,]  14.600  8.000 399.1250 219.250  3.31875  4.2355 16.63000  0.00  0.1250  3.250  3.5000
[4,]  15.975  8.000 282.1000 218.750  3.18750  3.7875 16.90000  0.00  0.2500  3.500  4.2500
```

Figura 69: Centros de grupos generados tras la ejecución del método K medias MapReduce sobre un clúster Hadoop

Cabe resaltar que el método tardó alrededor de sesenta minutos en dar el resultado.

## V. Bosques Aleatorios

Utilizando la Conexión 2 como entrada al método de Bosques Aleatorios se obtuvieron los siguientes resultados:

```

Type of random forest: classification
                Number of trees: 500
No. of variables tried at each split: 2

      OOB estimate of error rate: 4.67%
Confusion matrix:
      setosa versicolor virginica class.error
setosa      50         0         0         0.00
versicolor  0         47         3         0.06
virginica   0         4         46         0.08
    
```

Figura 70: Resultado del método Bosques Aleatorios utilizando la conexión 2

## VI. Regresión Logística MapReduce

Utilizando la Conexión 3 con un número de iteraciones igual a dos, un número de dimensiones igual a diez y un parámetro alpha igual a 0.05 se obtuvieron los siguientes resultados en el método de regresión logística disponible para las conexiones que corren sobre un clúster Hadoop:

```

cyl   disp   hp   drat   wt   qsec   vs   am   gear   carb
[1,]  92.34 3217.627 2109.068 59.50693 47.74382 290.3686 8.595 7.9275 60.9225 41.0475
    
```

Figura 71: Resultado del método Regresión Logística utilizando la conexión 3

## VII. Streaming

Utilizando la conexión 3 junto con la conexión 4 como funciones map y reduce respectivamente y como parámetro de entrada un archivo de texto se obtuvo el siguiente resultado:

Palabra	Repeticiones
jAh	1

¡Feliz	1
¡Oh!	2
¿Eres	1
¿Porqué	2
a	23
abandoné.	1

Tabla 3: Primeros siete resultados del Hadoop Streaming

Sólo se muestran los primeros siete resultados, esto debido a que la salida arrojó quinientas palabras.

### 5.2. Conclusiones

Se logró desarrollar todos los objetivos planeados para el último prototipo junto con los objetivos específicos de la investigación. Se instaló R en el clúster junto con los paquetes rmr2 y rhdfs. En el ambiente local también se instaló R junto con R Studio, GTK+, SSH y todos los paquetes necesarios que se definieron de manera previa.

La aplicación se programó utilizando una metodología propia basada en la entrega de prototipos con ningún otro artefacto entregable más que el prototipo como tal, esto debido a que sólo existió un desarrollador por lo cual sería muy engorroso generar artefactos de sincronización cuando no había con quien sincronizarse.

Se definieron siete casos de estudios para así probar las funcionalidades principales de la aplicación.

Los resultados fueron lo esperado aunque hubo ciertos problemas de rendimiento esto debido a que la aplicación es muy propensa a la funcionalidad del clúster, es decir si el clúster es de bajo rendimiento la aplicación se comportará de igual manera.

El desarrollo de esta aplicación fue todo un reto personal debido a que la programación en R es orientada a scripts que resuelvan un solo problema no al desarrollo de aplicaciones, pero pese a esto se obtuvo una aplicación

## Capítulo 5. Conclusiones y Resultados

potente que puede hacer análisis de datos de manera local y remota incentivando así al desarrollo de nuevas investigaciones.

Este trabajo de investigación contribuye principalmente con la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, ya que esta aplicación fue creada inicialmente para ser utilizada por los docentes y alumnos de la escuela. Más allá de la UCV, esta aplicación puede ser útil para todo aquel que desee explorar en el análisis de grandes volúmenes de datos también queda abierta para cualquier contribución futura ya sea un nuevo trabajo de grado o simplemente alguna mejora.

### 5.3. Recomendaciones

Tras las conclusiones obtenidas las recomendaciones basadas en esas conclusiones son las siguientes:

- Incentivar más las ciencias de datos con la apertura de más materias en la Facultad de Ciencias
- La programación de interfaces en R son algo complicadas, por ende sería muy útil utilizar un lenguaje distinto para la realización de las interfaces y los cálculos en R
- Investigar más métodos de análisis de datos basados en el marco MapReduce

### 5.4. Trabajos Futuros

El principal trabajo de investigación que puede desencadenar desde el presente es la mejora de las interfaces utilizando tecnologías como html, css y javascript junto con un lenguaje del lado del servidor que pueda conectarse con R como java o php.

También se podrían incluir más métodos y/o módulos a este trabajo de investigación.

# Anexos

## Guía de instalación de R, rmr y rhdfs

Pasos para distribuciones de Linux basadas en Red Hat.

### 1. Actualizar los repositorios epel

Se actualizan estos repositorios para futuras instalaciones de paquetes. Se utiliza el siguiente comando:

```
sudo rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

### 2. Instalar R

Se instala R con el siguiente comando:

```
sudo yum -y install git wget R
```

### 3. Instalar Rstudio Server

Se descarga he instala con los siguientes comandos:

```
wget http://download2.rstudio.org/rstudio-server-0.97.332-x86_64.rpm
```

```
sudo yum install --nogpgcheck rstudio-server-0.97.332-x86_64.rpm
```

### 4. Instalar rmr2

Primero se descargan las dependencias con el siguiente comando

```
install.packages( c('RJSONIO', 'itertools', 'digest', 'Rcpp', 'functional', 'plyr', 'stringr'), repos='http://cran.revolutionanalytics.com')
```

```
install.packages( c('reshape2'),  
repos='http://cran.revolutionanalytics.com')
```

## Anexos

Se descarga e instala con los siguientes comandos:

```
wget --no-check-certificate  
https://github.com/RevolutionAnalytics/rmr2/releases/download/3.3.1/r  
mr2_3.3.1.tar.gz
```

```
sudo R CMD INSTALL rmr2_3.3.1.tar.gz
```

### 5. Instalar rhdfs

Primero se deben resolver las dependencias que se listan a continuación:

- a. `export JAVA_HOME=/usr/lib/java/jdk-version`
- b. `export PATH=$PATH:$JAVA_HOME/bin`
- c. `export HADOOP_CMD=/usr/lib/hadoop/bin/hadoop`
- d. En una consola de R instalar el paquete 'rJava'

Luego ejecutar los siguientes comandos

```
wget --no-check-certificate  
https://github.com/RevolutionAnalytics/rhdfs/blob/master/build/rhdfs_1.  
0.8.tar.gz?raw=true
```

```
sudo R CMD INSTALL rhdfs_1.0.8.tar.gz?raw=true
```



## **Guía de instalación de gtk+**

Se deben ejecutar las dos siguientes líneas en distribuciones de Linux basadas en Red Hat:

1. `yum groupinstall "Development Tools"`
2. `yum install gtk+-devel gtk2-devel`

## **Guía de instalación de R**

En distribuciones de Linux basadas en Red Hat:

1. `sudo yum -y install git wget R`

## Guía de instalación de R-Studio

En distribuciones de Linux basadas en Red Hat:

1. `wget http://download2.rstudio.org/rstudio-server-0.97.332-x86_64.rpm`
2. `sudo yum install --nogpgcheck rstudio-server-0.97.332-x86_64.rpm`

## **Guía de instalación Openssh y sshpass**

En distribuciones de Linux basadas en Red Hat:

1. `yum -y install openssh-server openssh-clients`
2. `yum install sshpass`

## Algoritmo de K-medias en MapReduce en R

```
Sys.setenv(HADOOP_CMD="/usr/lib/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-mapreduce/hadoop-streaming-2.4.0.2.1.1.0-385.jar")
```

```
library(rhdfs)
hdfs.init()
library(rmr2)
rmr.options(backend="hadoop")

## @knitr kmeans-signature
kmeans.mr =
  function(
    P,
    num.clusters,
    num.iter,
    combine,
    in.memory.combine) {
  ## @knitr kmeans-dist.fun
  dist.fun =
    function(C, P) {
      apply(
        C,
        1,
        function(x)
          colSums((t(P) - x)^2))}
  ## @knitr kmeans.map
  kmeans.map =
    function(., P) {
      nearest = {
        if(is.null(C))
          sample(
            1:num.clusters,
            nrow(P),
            replace = TRUE)
        else {
          D = dist.fun(C, P)
          nearest = max.col(-D)}
      if(!(combine || in.memory.combine))
        keyval(nearest, P)
      else
        keyval(nearest, cbind(1, P))}
  ## @knitr kmeans.reduce
  kmeans.reduce = {
    if (!(combine || in.memory.combine) )
      function(., P)
        t(as.matrix(apply(P, 2, mean)))
    else
      function(k, P)
        keyval(
          k,
          t(as.matrix(apply(P, 2, sum))))}
  ## @knitr kmeans-main-1
```

## Anexos

```
C = NULL
for(i in 1:num.iter ) {
  C =
  values(
    from.dfs(
      mapreduce(
        P,
        map = kmeans.map,
        reduce = kmeans.reduce)))
  if(combine || in.memory.combine)
    C = C[, -1]/C[, 1]
  ## @knitr end
  #   points(C, col = i + 1, pch = 19)
  ## @knitr kmeans-main-2
  if(nrow(C) < num.clusters) {
    C =
    rbind(
      C,
      matrix(
        rnorm(
          (num.clusters -
            nrow(C)) * nrow(C)),
          ncol = nrow(C) %*% C )})
  }
  C}
testdata = data
kmeans.mr(testdata, num.clusters = 3, num.iter = 10, combine =
FALSE, in.memory.combine = FALSE)
```

Disponible en [51]

## Algoritmo de Regresión Logística en MapReduce en R

```

Sys.setenv(HADOOP_CMD="/usr/lib/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-mapreduce/hadoop-streaming-2.4.0.2.1.1.0-385.jar")

library(rhdfs)
hdfs.init()
library(rmr2)
rmr.options(backend="hadoop")

logistic.regression =
  function(input, iterations, dims, alpha){
    lr.map =
      function(., M) {
        Y = M[,1]
        X = M[,-1]
        keyval(
          1,
          Y * X *
          g(-Y * as.numeric(X %*% t(plane))))
      }

    lr.reduce =
      function(k, Z)
        keyval(k, t(as.matrix(apply(Z,2,sum))))

    plane = t(rep(0, dims))
    g = function(z) 1/(1 + exp(-z))
    for (i in 1:iterations) {
      gradient =
        values(
          from.dfs(
            mapreduce(
              input,
              map = lr.map,
              reduce = lr.reduce,
              combine = T))
          plane = plane + alpha * gradient
        }
      plane
    }

    testdata = data
    logistic.regression(testdata, 10, 10, 0.05)
  }

```

Disponible en [51]

## Función Map del algoritmo Wordcount en R

```
#!/usr/bin/env Rscript

# mapper.R - Wordcount program in R
# script for Mapper (R-Hadoop integration)

trimWhiteSpace <- function(line) gsub("(^ +)|( +$)", "", line)
splitIntoWords <- function(line) unlist(strsplit(line,
"[:,space:]+"))

## **** could wo with a single readLines or in blocks
con <- file("stdin", open = "r")
while (length(line <- readLines(con, n = 1, warn = FALSE)) > 0) {
  line <- trimWhiteSpace(line)
  words <- splitIntoWords(line)
  ## **** can be done as cat(paste(words, "\t1\n", sep=""),
  sep="")
  for (w in words)
    cat(w, "\t1\n", sep="")
}
close(con)
```



## Función Reduce del algoritmo Wordcount en R

```

#!/usr/bin/env Rscript

# reducer.R - Wordcount program in R
# script for Reducer (R-Hadoop integration)

trimWhiteSpace <- function(line) gsub("(^ +)|( +$)", "", line)

splitLine <- function(line) {
  val <- unlist(strsplit(line, "\t"))
  list(word = val[1], count = as.integer(val[2]))
}

env <- new.env(hash = TRUE)

con <- file("stdin", open = "r")
while (length(line <- readLines(con, n = 1, warn = FALSE)) > 0) {
  line <- trimWhiteSpace(line)
  split <- splitLine(line)
  word <- split$word
  count <- split$count
  if (exists(word, envir = env, inherits = FALSE)) {
    oldcount <- get(word, envir = env)
    assign(word, oldcount + count, envir = env)
  }
  else assign(word, count, envir = env)
}
close(con)

for (w in ls(env, all = TRUE))
  cat(w, "\t", get(w, envir = env), "\n", sep = "")

```

# Bibliografía

[1] *Weka 3: Data Mining Software in Java* [en línea]: documenting electronic sources on the Internet. s.f. [Fecha de consulta: 4 de septiembre 2015, 10:11 am] Disponible en  
< <http://www.cs.waikato.ac.nz/ml/weka/> >

[2] *Weka (aprendizaje automático)* [en línea]: documenting electronic sources on the Internet. s.f. [Fecha de consulta: 4 de septiembre 2015, 10:15 am] Disponible en  
< [https://es.wikipedia.org/wiki/Weka\\_\(aprendizaje\\_autom%C3%A1tico\)](https://es.wikipedia.org/wiki/Weka_(aprendizaje_autom%C3%A1tico)) >

[3] *Rcommander a graphical interface for R* [en línea]: documenting electronic sources on the Internet. 2013. [Fecha de consulta: 4 de septiembre 2015, 10:18 am] Disponible en  
< <http://www.rcommander.com/> >

[4] *Rattle GUI* [en línea]: documenting electronic sources on the Internet. 2015. [Fecha de consulta: 4 de septiembre 2015, 10:21 am] Disponible en  
< [https://en.wikipedia.org/wiki/Rattle\\_GUI](https://en.wikipedia.org/wiki/Rattle_GUI) >

[5] DHAR, V. *Data Science and Prediction* [en línea]: documenting electronic sources on the Internet. 2013 [Fecha de consulta: 5 de octubre 2014, 8:06 pm] Disponible en  
<<http://cacm.acm.org/magazines/2013/12/169933-data-science-and-prediction/fulltext/>>

[6] LEAK, J. *The keyword in "Data Science" is not Data, It is Science* [en línea]: documenting electronic sources on the Internet. 2013 [Fecha de consulta: 5 de octubre de 2014, 8:06 pm] Disponible en  
<<http://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/>>

[7] HERNANDEZ, J. *Introducción a la Minería de Datos*. Pearson Prentice Hall 2004.

[8] WITTEN, I. *Data Mining*, Third Edition. Morgan Kauffman 2011.

## Bibliografía

[9] *Análisis de componentes principales* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 17 de febrero de 2015, 8:05 pm] Disponible en

< [http://es.wikipedia.org/wiki/An%C3%A1lisis\\_de\\_componentes\\_principales](http://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_principales)>

[10] *Agrupamiento jerárquico* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 17 de febrero de 2015, 8:52 pm] Disponible en

< [http://es.wikipedia.org/wiki/Agrupamiento\\_jer%C3%A1rquico](http://es.wikipedia.org/wiki/Agrupamiento_jer%C3%A1rquico)>

[11] *Regresión Logística* [en línea]: documenting electronic sources on the Internet. 2014 [Fecha de consulta: 4 de septiembre de 2015, 10:24 am] Disponible en

< [https://es.wikipedia.org/wiki/Regresi%C3%B3n\\_log%C3%ADstica](https://es.wikipedia.org/wiki/Regresi%C3%B3n_log%C3%ADstica) >

[12] *Operating System* [en línea]: documenting electronic sources on the Internet. 2015 [Fecha de consulta: 4 de septiembre de 2015, 10:27 am] Disponible en

< [https://en.wikipedia.org/wiki/Operating\\_system](https://en.wikipedia.org/wiki/Operating_system) >

[13] MARTINEZ, R. *Sobre Linux* [en línea]: documenting electronic sources on the Internet. 2014. [Fecha de consulta: 4 de septiembre de 2015, 10: 29 am] Disponible en

< [http://www.linux-es.org/sobre\\_linux](http://www.linux-es.org/sobre_linux) >

[14] *The GTK+ Project* [en línea]: documenting electronic sources on the Internet. 2015. [Fecha de consulta: 4 de septiembre de 2015, 10:32 am] Disponible en

< <http://www.gtk.org/> >

[15] *Secure Shell* [en línea]: documenting electronic sources on the Internet. 2015. [Fecha de consulta: 4 de septiembre de 2015, 10:34 am] Disponible en

< [https://es.wikipedia.org/wiki/Secure\\_Shell](https://es.wikipedia.org/wiki/Secure_Shell) >

[16] VERZANI, J. *gWidgets2-package* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 4 de septiembre de 2015, 10:37 am] Disponible en

< <http://www.rdocumentation.org/packages/gWidgets2/functions/gWidgets2-package> >

## Bibliografía

[17] *rmr-package* [en línea]: documenting sources on the Internet. s.f [Fecha de consulta: 4 de septiembre de 2015, 10:39 am] Disponible en < <http://www.rdocumentation.org/packages/rmr2/functions/rmr-package> >

[18] *rhdfs* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 4 de septiembre de 2015, 10:41 am] Disponible en < <http://www.rdocumentation.org/packages/rhdfs/functions/rhdfs> >

[19] *Hadoop Streaming* [en línea]: documenting electronic sources on the Internet. 2013 [Fecha de consulta: 4 de septiembre de 2015, 10:43 am] Disponible en < <http://hadoop.apache.org/docs/r1.2.1/streaming.html> >

[20] *Cloudera vs Hortonworks vs MapR: Comparing Hadoop Distributions* [en línea]: documenting electronic sources on the Internet. 2014 [Fecha de consulta: 4 de septiembre de 2015, 10:47 am] Disponible en: < <http://www.experfy.com/blog/cloudera-vs-hortonworks-comparing-hadoop-distributions/> >

[21] *Manifiesto Ágil* [en línea]: documenting electronic sources on the Internet. 2015 [Fecha de consulta: 4 de septiembre de 2015, 10:50 am] Disponible en < [https://es.wikipedia.org/wiki/Manifiesto\\_%C3%A1gil](https://es.wikipedia.org/wiki/Manifiesto_%C3%A1gil) >

[22] *Desarrollo ágil de software* [en línea]: documenting electronic sources on the Internet. 2015 [Fecha de consulta: 4 de septiembre de 2015, 10:52 am] Disponible en < [https://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software) >

[23] *Knn* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 17 de febrero de 2015, 1:33 pm] Disponible en < <http://es.wikipedia.org/wiki/Knn>>

[24] WHITE, T. *Hadoop: The Definitive Guide*, Third Edition. O'Reilly Media 2012

[25] WARDEN, P. *Big Data Glossary*. O'Reilly Media 2011.

[26] MINER, D. *MapReduce Design Patterns*. O'Reilly Media 2013.

## Bibliografía

[27] *¿Qué es Big Data?* [en línea]: documenting electronic sources on the Internet. 2012 [Fecha de consulta: 5 de octubre de 2014, 8:08 pm] Disponible en

<<http://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/>>

[28] *Hadoop YARN* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:11 pm] Disponible en

<<http://hortonworks.com/hadoop/yarn/>>

[29] ROUSE, Margaret. *Apache Hadoop YARN (Yet Another Resource Negotiator)* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:12 pm] Disponible en

<<http://searchdatamanagement.techtarget.com/definition/Apache-Hadoop-YARN-Yet-Another-Resource-Negotiator>>

[30] *Apache Hadoop NextGen MapReduce (YARN)* [en línea]: documenting electronic sources on the Internet. 2014 [Fecha de consulta: 5 de octubre de 2014, 8:13 pm] Disponible en

<<http://hadoop.apache.org/docs/r2.3.0/hadoop-yarn/hadoop-yarn-site/YARN.html>>

[31] *Apache Hive* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:14 pm] Disponible en

<<http://hortonworks.com/hadoop/hive/>>

[32] *amplab/shark* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:19 pm] Disponible en

<<https://github.com/amplab/shark/wiki>>

[33] *Apache Pig* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:15 pm] Disponible en

<<http://hortonworks.com/hadoop/pig/>>

[34] *Apache HCatalog* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 16 de febrero de 2015, 9:34 pm] Disponible en <<http://hortonworks.com/hadoop/hcatalog/>>

## Bibliografía

[35] *Apache Spark* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:16 pm] Disponible en <<http://hortonworks.com/hadoop/spark/>>

[36] METZ, C. (2013). *Spark: Open Source Superstar Rewrites Future of Big Data* [en línea]: documenting electronic sources on the Internet. 2013 [Fecha de consulta: 5 de octubre de 2014, 8:17 pm] Disponible en <<http://www.wired.com/2013/06/yahoo-amazon-amplab-spark/all/>>

[37] *Las 5 V's del Big Data* [en línea]: documenting electronic sources on the Internet. 2014 [Fecha de consulta: 13 de noviembre de 2014, 10:38 pm] Disponible en <<http://www.quanticsolutions.es/big-data/las-5-v-big-data/>>

[38] *Apache Flume* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:18 pm] Disponible en <<http://hortonworks.com/hadoop/flume/>>

[39] *cloudera/hue* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:20 pm] Disponible en <<https://github.com/cloudera/hue>>

[40] *Apache ZooKeeper* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:18 pm] Disponible en <<http://hortonworks.com/hadoop/zookeeper/>>

[41] *amplab/shark* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 5 de octubre de 2014, 8:19 pm] Disponible en <<https://github.com/amplab/shark/wiki>>

[42] *Apache Sqoop* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 16 de febrero de 2015, 10:06 pm] Disponible en <<http://hortonworks.com/hadoop/hcatalog/>>

[43] *Máquinas de vectores de soporte* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 17 de febrero de 2015, 2:35 pm] Disponible en

<[http://es.wikipedia.org/wiki/M%C3%A1quinas\\_de\\_vectores\\_de\\_soporte](http://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte)>

[44] RODRIGUEZ, Oldemar. *Aprendizaje Supervisado. Redes Neuronales, Métodos de Consenso y Potenciación* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 17 de febrero de 2015, 7:15 pm] Disponible en

## Bibliografía

<[http://www.oldemarrodriguez.com/yahoo\\_site\\_admin/assets/docs/Presentaci%C3%B3n - Redes - Consenso - Potenciacion.293124147.pdf](http://www.oldemarrodriguez.com/yahoo_site_admin/assets/docs/Presentaci%C3%B3n_-_Redes_-_Consenso_-_Potenciacion.293124147.pdf)>

[45] *Random forest* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 17 de febrero de 2015, 7:31 pm] Disponible en <[http://es.wikipedia.org/wiki/Random\\_forest](http://es.wikipedia.org/wiki/Random_forest)>

[46] *Red neuronal artificial* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 17 de febrero de 2015, 7:53 pm] Disponible en <[http://es.wikipedia.org/wiki/Red\\_neuronal\\_artificial](http://es.wikipedia.org/wiki/Red_neuronal_artificial)>

[47] *Análisis de componentes principales* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 17 de febrero de 2015, 8:05 pm] Disponible en  
<[http://es.wikipedia.org/wiki/An%C3%A1lisis\\_de\\_componentes\\_principales](http://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_principales)>

[48] *RevolutionAnalytics/RHadoop* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 21 de febrero de 2015, 5:58 pm] Disponible en <<https://github.com/RevolutionAnalytics/RHadoop/wiki>>

[49] *Scipy/Scipy* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 21 de febrero de 2015, 6:15 pm] Disponible en <<https://github.com/scipy/scipy>>

[50] *Scipy.org* [en línea]: documenting electronic sources on the Internet. s.f [Fecha de consulta: 21 de febrero de 2015, 6:16 pm] Disponible en <<http://www.scipy.org>>

[51] DEVLIN, H. *Mapreduce in R* [en línea]: documenting electronic sources on the Internet. 2014 [Fecha de consulta: 4 de septiembre de 2015, 10:59 am] Disponible en  
<<https://github.com/RevolutionAnalytics/rmr2/blob/master/docs/tutorial.md>>