



UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería del Software

Trabajo Fin de Grado

Big Geo Data - Análisis de datos meteorológicos  
mediante series temporales con Hadoop y Weka

Jaime Pina Cambero

Noviembre, 2017





# UNIVERSIDAD DE EXTREMADURA

## Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería del Software

### Trabajo Fin de Grado

## Big Geo Data - Análisis de datos meteorológicos mediante series temporales con Hadoop y Weka

Autor: Jaime Pina Cambero  
Fdo.:

Director: Félix Rodríguez Rodríguez  
Fdo.:

#### **Tribunal Calificador**

Presidente: José Luis González Sánchez  
Fdo.:

Secretaria: M<sup>a</sup> Luisa Durán Martín-Merás  
Fdo.:

Vocal: M<sup>a</sup> Ángeles Mariscal Araujo  
Fdo.:

CALIFICACION:  
FECHA:



## **Agradecimientos**

En primer lugar, quiero dar las gracias a mi familia, especialmente a mis padres que siempre han estado apoyándome y soportándome, sobretodo en estos últimos meses. A mi abuela, que todos los días me sacaba una sonrisa al decirme que “a ver cuando terminaba ese proyecto dichoso”. A Edurne y a mis amigos que, durante horas, me han escuchado hablar y me han dado sus mejores consejos. Por último, pero no por ello menos importante, quiero dar las gracias a mi tutor, gracias al cual he podido realizar este proyecto.

A todos vosotros. Muchas gracias.



## **Resumen**

En este trabajo fin de grado trataremos de realizar un ciclo completo de procesamiento de Big Data. Aplicaremos este ciclo al sector meteorológico, con el fin de ser capaces de extraer información útil en determinadas zonas de interés.

Los datos meteorológicos serán extraídos de las bases de datos creadas por la NASA, a partir de la información extraída con sus satélites. En concreto, nos centraremos en aquellos cuyo objetivo es obtener información acerca de los vientos sobre la superficie de los mares y océanos. Una vez obtenidos, se realizará una transformación de los mismos con el fin de convertirlos a un formato más cómodo y fácil de trabajar.

Tras la obtención y transformación de datos, se procederá a realizar el procesamiento de los mismos. Para ello, se usará una técnica de procesamiento MapReduce, debido principalmente al gran tamaño de los datos. Con esta técnica se filtrarán los datos, analizando solo los de interés según las regiones insertadas.

Por último, mediante Weka se usará un algoritmo de series temporales para analizar cada una de las regiones. Se realizará un ajuste de regresión lineal permitiendo la predicción de los datos.

# ÍNDICE

<b>RESUMEN.....</b>	<b>7</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>11</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>12</b>
<b>1. INTRODUCCIÓN .....</b>	<b>13</b>
1.1    OBJETIVOS.....	13
1.2    MOTIVACIÓN .....	14
1.3    PRESENTACIÓN DE LOS CAPÍTULOS .....	14
<b>2.    ANTECEDENTES Y TECNOLOGÍA RELACIONADA .....</b>	<b>17</b>
2.1    TRABAJO RELACIONADOS .....	17
2.2    CONOCIMIENTOS GENERALES .....	17
2.2.1 Misión SeaWinds.....	17
2.2.2 Big Data.....	21
2.2.3 Extracción, Transformación y Carga (ETL).....	22
2.2.4 Apache Hadoop.....	22
2.2.5 MapReduce .....	22
2.2.6 Data Mining .....	23
2.3    MATERIALES, ENTORNOS, HERRAMIENTAS Y LIBRERÍAS.....	23
2.3.1    Software .....	23
2.3.1.1 Sistemas operativos.....	23
2.3.1.2 Máquina Virtual con SpatialHadoop.....	24
2.3.1.3 Java .....	24
2.3.1.4 Python .....	24
2.3.1.5 Librerías para la transformación de los datos.....	25
2.3.1.6 Weka.....	25
2.3.1.7 Eclipse.....	25
2.3.1.8 WindowBuilder .....	26
2.3.1.9 FileZilla .....	26
2.3.2    Hardware.....	26
<b>3.    DESARROLLO DEL PROYECTO .....</b>	<b>27</b>
3.1 ESTUDIO DE VIABILIDAD.....	27
3.2 PREVISIÓN DEL REPARTO DEL TIEMPO .....	28
3.3 FUNCIONAMIENTO GENERAL .....	30
3.3.1 ETL.....	32
3.3.1.1 Extracción .....	32



3.3.1.2 Transformación .....	33
3.3.1.3 Carga.....	33
3.3.2 Procedimiento MapReduce .....	34
Paquete MapReduce .....	34
Clase <i>Main_MapReduce</i> .....	34
Clase <i>MapClass</i> .....	34
Clase <i>ReduceClass</i> .....	35
Clase <i>Punto</i> .....	36
Clase <i>Region</i> .....	37
Paquete UI.....	37
Clase <i>MainWindow</i> .....	38
Clase <i>Rutas</i> .....	38
Clase <i>SeleccionManual</i> .....	38
3.3.3 Weka .....	39
<b>4. RESULTADOS .....</b>	<b>43</b>
<b>5. CONCLUSIONES Y TRABAJOS FUTUROS .....</b>	<b>49</b>
5.1 CONCLUSIONES DEL PROYECTO .....	49
5.2 CONCLUSIONES PERSONALES .....	50
5.3 PRINCIPALES PROBLEMAS ENCONTRADOS.....	50
5.4 TRABAJOS FUTUROS .....	52
<b>5. ANEXOS.....</b>	<b>53</b>
ANEXO 1. MANUAL DE USUARIO .....	53
1.1 Introducción .....	53
1.2 Ejecución ETL.....	53
1.2.1 Extracción de los datos con FileZilla .....	53
1.2.2 Transformación .....	54
1.3 MapReduce .....	57
1.4 Weka .....	57
ANEXO 2. MANUAL DEL PROGRAMADOR.....	58
2.1 Introducción .....	58
2.2 Instalación de FileZilla.....	58
2.3 Instalación de Python.....	59
2.4 Instalación de Hadoop.....	61
2.4.1 Instalación en Windows.....	61
2.4.2 Instalación usando SpatialHadoop .....	67
2.4.2.1 Instalación de la máquina virtual.....	68
2.4.2.2 Ampliación de la capacidad de almacenamiento .....	69
2.5 Instalación de Weka .....	72

2.6 Instalación WindowBuilder (Eclipse) .....	73
2.7 Compartición de datos entre sistemas .....	73
<b>7. BIBLIOGRAFÍA .....</b>	<b>75</b>

## Índice de figuras

<b>Figura 1 Señal del satélite QuikSCAT atenuada por el paso a través de las nubes [7]</b> .....	18
<b>Figura 2 Imagen artística del satélite QuikSCAT [8]</b> .....	19
<b>Figura 3 Satélite RapidSCAT incrustado en la Estación Espacial Internacional [11]</b> .....	20
<b>Figura 4 Gráfico de la distribución del tiempo</b> .....	28
<b>Figura 5 Esquema general del funcionamiento del sistema</b> .....	30
<b>Figura 6 Esquema específico del funcionamiento del sistema</b> .....	31
<b>Figura 7 Repositorio FTP de la NASA</b> .....	32
<b>Figura 8 Diagrama de flujo del proceso Reduce</b> .....	36
<b>Figura 9 Interfaz gráfica del proyecto</b> .....	37
<b>Figura 10 Ejemplo de la interfaz con inserción manual</b> .....	38
<b>Figura 11 Pantalla de Weka tras insertarle un archivo</b> .....	39
<b>Figura 12 Configuración básica de Weka (paquete Forecast)</b> .....	40
<b>Figura 13 Configuración avanzada de Weka - Lag creation</b> .....	40
<b>Figura 14 Configuración avanzada de Weka – Output</b> .....	41
<b>Figura 15 Salida por consola tras realizar la ejecución</b> .....	43
<b>Figura 16 Carpeta con los archivos .arff y .txt generados tras la ejecución</b> .....	44
<b>Figura 17 Archivo .arff de las Islas Baleares de 2015</b> .....	45
<b>Figura 18 Archivo .txt generado por Weka de las Islas Baleares de 2015</b> .....	45
<b>Figura 19 Resultados gráficos sobre la velocidad máxima</b> .....	47
<b>Figura 20 Resultados gráficos sobre la velocidad media</b> .....	48
<b>Figura 21 Pantalla de FileZilla una vez ejecutada</b> .....	53
<b>Figura 22 Inicialización de la descarga de los datos</b> .....	54
<b>Figura 23 Archivos que contienen la información de un día</b> .....	55
<b>Figura 24 Menú del programa de transformación</b> .....	56
<b>Figura 25 Ejemplo de ejecución de la transformación</b> .....	56
<b>Figura 26 Configuración de la interfaz.</b> .....	57
<b>Figura 27 Descarga de la herramienta FileZilla</b> .....	58
<b>Figura 28 Descarga de la herramienta Anaconda</b> .....	59
<b>Figura 29 Comando de la instalación de numpy</b> .....	60
<b>Figura 30 Comando de la instalación de Pydhf</b> .....	60
<b>Figura 31 Comando de la instalación de netCDF4</b> .....	61
<b>Figura 32 Descarga de la herramienta Hadoop</b> .....	61

<b>Figura 33 Descarga del JDK.....</b>	<b>62</b>
<b>Figura 34 Variables de entorno añadidas .....</b>	<b>63</b>
<b>Figura 35 Modificación de la variable PATH .....</b>	<b>63</b>
<b>Figura 36 Modificación en hadoop-env.cmd .....</b>	<b>66</b>
<b>Figura 37 Ejecución de los componentes de Hadoop .....</b>	<b>66</b>
<b>Figura 38 Interfaz de Hadoop en localhost:8088 .....</b>	<b>67</b>
<b>Figura 39 Interfaz de Hadoop en localhost:50070 .....</b>	<b>67</b>
<b>Figura 40 Descarga de SpatialHadoop .....</b>	<b>68</b>
<b>Figura 41 Comando de creación de disco duro virtual .....</b>	<b>69</b>
<b>Figura 42 Comando de clonación de disco duro virtual .....</b>	<b>69</b>
<b>Figura 43 Inserción del nuevo disco duro virtual .....</b>	<b>70</b>
<b>Figura 44 Particiones con la herramienta GParted .....</b>	<b>71</b>
<b>Figura 45 Weka con el paquete TimeseriesForecasting instalado .....</b>	<b>72</b>
<b>Figura 46 Instalación de WindowBuilder .....</b>	<b>73</b>
<b>Figura 47 Configuración de las carpetas compartidas .....</b>	<b>74</b>

## Índice de tablas

<b>Tabla 1: Tareas propuestas y estimación del tiempo de realización .....</b>	<b>28</b>
<b>Tabla 2: Tareas realizadas y tiempo utilizado en las mismas.....</b>	<b>28</b>

# **1. Introducción**

Todos los días, millones de personas se sientan en sus hogares a la espera de que las noticias finalicen para visualizar el parte meteorológico de los próximos días. Otros, afines a las nuevas tecnologías, prefieren consultar sus dispositivos móviles para acceder a estas predicciones de forma rápida y cómoda. La necesidad de conocer el tiempo se ha convertido en un punto clave a la hora de organizar planes a corto o largo plazo.

Millones de bytes de datos relacionados con el tiempo son generados a diario. El estudio y análisis de estos datos es fundamental para la realización de predicciones meteorológicas, el estudio de patrones del tiempo, la comprensión del comportamiento de ciertas especies o el análisis en los cambios en la historia producidos por el clima, entre muchas otras opciones.

En este contexto se enmarca este Trabajo de Fin de Grado (TFG), cuyo objetivo es la realización de un modelo de procesamiento, mediante programación MapReduce, para llevar a cabo análisis meteorológicos de zonas exclusivas de La Tierra. A continuación, se expondrán los objetivos que se pretenden alcanzar en este proyecto.

## **1.1 Objetivos**

El objetivo principal de este proyecto es la realización de un proceso completo de análisis de datos que obtenga información útil sobre datos meteorológicos, permitiendo por lo tanto, la predicción de los mismos a través de una serie temporal. Este proceso se divide en varios subobjetivos:

- 1- Identificar la necesidad del análisis de los datos meteorológicos.
- 2- Estudiar las soluciones existentes.
- 3- Obtener conocimiento general sobre los datos y el proceso de transformación.
- 4- Realizar extracción, transformación y carga de los datos de la NASA.
- 5- Diseño e implementación de un proceso MapReduce.
- 6- Uso de Weka para el análisis de los datos.
- 7- Ajuste de Weka para mejorar la precisión de los resultados.
- 8- Implementación de una interfaz de usuario para lanzar todo el proceso.
- 9- Analizar el estado final del proyecto.
- 10- Plantear los posibles trabajos futuros.

## 1.2 Motivación

La importancia de la información siempre ha sido fundamental para la ayuda a la toma de decisiones, desde la antigüedad hasta los días actuales. El problema reside en que actualmente la mayoría de la información se encuentra en medios virtuales; ya sea la nube, bases de datos, redes sociales, apps de mensajería, etc. Es tanta la información, que resultaría imposible estudiarla y analizarla sin la ayuda de un ordenador. Aquí es donde reside la importancia del Big Data: ser capaces de tratar y analizar una gran cantidad de datos para poder obtener información útil.

En este TFG se lleva a cabo un modelo de procesamiento Big Data completo bajo el esquema de ejecución multiproceso MapReduce. Aunque en este trabajo se usen datos meteorológicos este proceso se podría aplicar sobre cualquier tipo de datos pudiendo obtener resultados e información no accesible por los métodos tradicionales, lo cual lo hace muy interesante.

## 1.3 Presentación de los capítulos

Esta documentación se compone de 7 capítulos. La información general de cada capítulo se resume a continuación:

En este **primer capítulo** se ha presentado el contexto, los objetivos y la motivación para desarrollar este TFG.

El **segundo capítulo** trata sobre los conocimientos que rodean al trabajo, ya sean conceptos teóricos, como el proceso MapReduce, o herramientas que se han utilizado ya sean software o hardware.

En el **tercer capítulo** se explica el desarrollo del proyecto. Empezando por el estudio de viabilidad y seguido de la previsión del reparto del tiempo estimada y real. Por último, se explican los pasos que se dan en el desarrollo del proyecto, mostrando las clases que se han creado y cómo interactúan entre ellas.

En el **capítulo cuarto** se muestran y explican de los resultados del proyecto.

En el **capítulo quinto** se comentan cuáles son las conclusiones del proyecto, tanto generales como personales, así mismo, también se comentan posibles trabajos futuros sobre el mismo.

El **capítulo sexto** está formado por dos anexos:

- El **primer anexo** es el manual de usuario, en él se explica detalladamente como ejecutar todas las herramientas de las que se compone el proyecto. Este Anexo también muestra una ejecución completa del proyecto.
- El **segundo anexo** es el manual del programador en el cual se explica cómo realizar las instalaciones y configuraciones de cada herramienta utilizada.

Por último, en **el capítulo sétimo** se muestran las referencias consultadas, ya sean libros, videos, páginas web o blogs.





## 2. Antecedentes y tecnología relacionada

En este capítulo se van a tratar los conocimientos necesarios para entender este proyecto. En primer lugar, se hablará sobre trabajos relacionados; tras ello se comentarán ciertas técnicas y conceptos importantes en este proyecto y, por último, se mencionarán las herramientas software y hardware que se han utilizado.

### 2.1 Trabajos relacionados

Aunque este proyecto ha sido realizado por completo por el autor del mismo, incluye algunas herramientas que se han utilizado en otros proyectos anteriores:

Para la transformación de los datos, se ha utilizado un programa escrito en Python creado por Daniel Teomiro Villa en su TFG “*Integración de datos de vientos marinos en Oracle NoSQL*”. [1]

Así mismo, para la inicialización de este proyecto también se han consultado otros trabajos parecidos como, por ejemplo, el de César Fernández Arroyo “*Técnicas Big Data con datos Geoespaciales y Hadoop*”. [2]

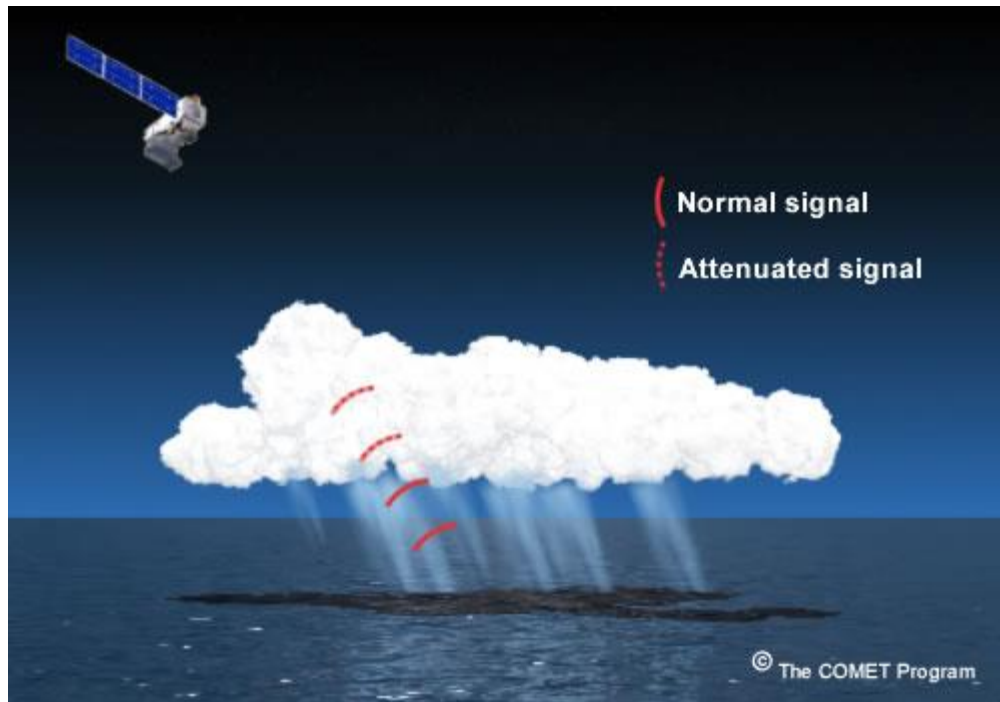
### 2.2 Conocimientos generales

En este apartado se hablará de conceptos y técnicas importantes en este TFG. Se empezará contextualizando la misión de la NASA gracias a la cual se han obtenido los datos. Tras ello, se hablará de las técnicas que subyacen debajo del desarrollo de este proyecto.

#### 2.2.1 Misión SeaWinds

La misión **SeaWinds** [3] nace con una colaboración entre la **NASA** [4] y la **NOAA** [5]. Fue creada para poder medir vientos que están presentes cerca de la superficie de los océanos independientemente de la climatología que hiciese; es decir, que pudiese captar los datos sin tener problemas con las nubes, tormentas o cualquier alteración meteorológica.

Los satélites más importantes en esta misión son QuikSCAT y RapidSCAT, pero antes se puso en órbita el **NSCAT** [6], satélite que, tras su lanzamiento en 1996, tuvo un fallo imprevisto en el panel solar en 1997 y dejó de funcionar.

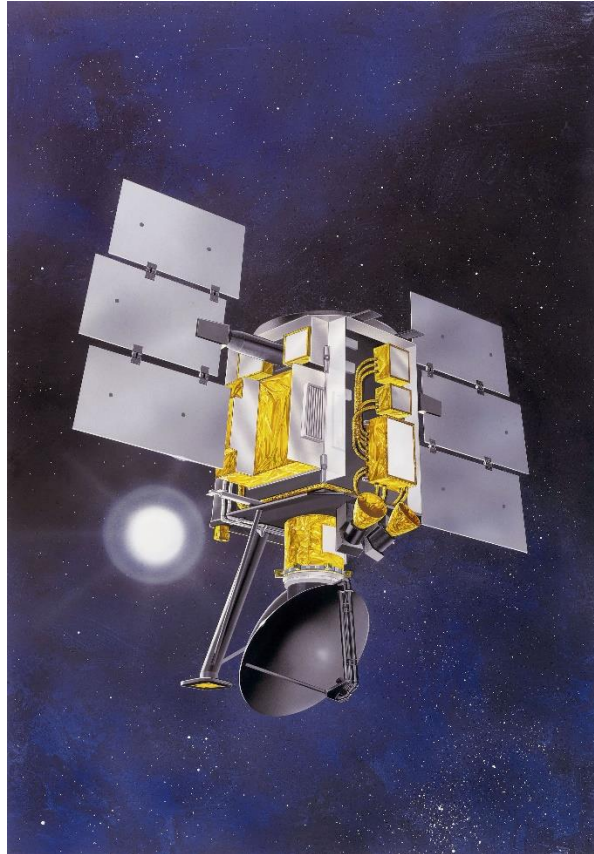


**Figura 1 Señal del satélite QuikSCAT atenuada por el paso a través de las nubes [7]**

**QuikSCAT** [8] se puso en órbita en 1999 como una “Recuperación Rápida” para llenar el vacío creado por la pérdida de datos de la NASA, tras la caída del satélite NSCAT en 1997. Este satélite orbitaba a una distancia de 800 Km de la superficie, realizando un ciclo que le permitía dar entre 14 y 16 vueltas a la Tierra cada día. La misión tenía los siguientes objetivos [8]:

- Obtener los datos de vientos cercanos a la superficie oceánica sin que se produzcan fallos en la obtención de los mismos por las condiciones climatológicas.
- Estudiar los cambios anuales y semestrales de la vegetación de la selva tropical.
- Estudiar los movimientos diarios y estacionales del hielo marino y cambios producidos en los hielos de los polos.
- Mejorar la predicción de tormentas marinas.
- Mejorar la predicción meteorológica en las zonas costeras
- Combinar datos provenientes de diferentes instrumentos científicos de otras disciplinas, para poder realizar un estudio del cambio climático.
- Determinar la respuesta de los océanos y su interacción con el aire en diferentes escalas espaciales y temporales.

El elemento principal del satélite QuikSCAT era el **dispersómetro** (scatterometer). Este dispositivo es una antena de plato giratorio que irradia pulsos de microondas a una frecuencia de 13,4 GHz a través de amplias regiones de la superficie oceánica de la Tierra, midiendo los pulsos retornados al satélite por eco.



**Figura 2** Imagen artística del satélite QuikSCAT [8]

Con estas mediciones, el satélite formaba los datos y los transmitía al centro de recepción PODAAC, gestionado por la NASA. Allí, estos datos eran procesados y almacenados en ficheros (uno por cada órbita) con formatos científicos HDF [9] y netCDF [10] para la resolución de captura más fina. La principal ventaja de estos ficheros es que son capaces de almacenar una gran cantidad de datos numéricos referidos a multitud de variables. La versión de ambos formatos científicos es la 4, HDF-4 y netCDF-4, capaces de soportar diferentes modelos de datos, en los que pueden aparecer matrices multidimensionales, imágenes y tablas. Estos datos contienen básicamente la velocidad y la dirección del viento y la ratio de precipitaciones próximas a la superficie oceánica con distintas resoluciones (una observación menos precisa cada 25 km<sup>2</sup>, o bien una con mayor precisión cada 12,5 km<sup>2</sup>), realizando barridos de superficie de una anchura aproximada de 1.800 km con unas 400.000 tomas de datos diarias, el equivalente al 90% de la superficie terrestre.



**Figura 3** Satélite RapidSCAT incrustado en la Estación Espacial Internacional [11]

Debido a la importancia que estaba teniendo esta misión, tras dejar de funcionar el satélite QuikSCAT, se le propuso a la NASA el reto de tener otro satélite en órbita que permitiese la obtención de esos datos lo antes posible. Es así como nació el ingenio **RapidSCAT** [12]. Este se basa en una reutilización del proyecto SeaWinds para obtener datos con una precisión similar a la resolución más fina de la misión QuikSCAT. Pero hay una variación importante respecto al QuikSCAT y se debe a que se reduce a la mitad la franja de mediciones, ya que este dispersómetro (scatterometer), que se terminó implantando en la Estación Espacial Internacional (ISS) el 20 de septiembre de 2014, se debe al propio recorrido orbital de la ISS, y éste cubre un espacio orbital menor que la recorrida por el proyecto original QuikSCAT. Supone que cada día se obtienen observaciones de alrededor del 71% de la superficie oceánica terrestre. La idea principal de implantarlo en la Estación Espacial Internacional radica en que la obtención de los datos meteorológicos de la superficie oceánica terrestre tuviese el mínimo número posible de anomalías y/o falta de capturas por fallos, ya que siempre hay un ingeniero que puede realizar el mantenimiento de todos los componentes del ingenio orbital.

Los datos que obtiene RapidSCAT se almacenan solamente en el formato netCDF-4, dado que este es un formato autodescriptivo, independiente de la máquina y centrado en vectores de datos, comúnmente conocidos como arrays, creado única y exclusivamente para la distribución de datos de carácter científico.

El 19 de agosto de 2016, el módulo ISS Columbus [13] experimentó una pérdida de potencia, lo que resultó en una pérdida de potencia total e irreparable para el instrumento

RapidSCAT. La desconexión de ISS-RapidSCAT comenzó en diciembre de 2016 y continuará hasta 2017. A través de la fase de desmantelamiento, el equipo ISS-RapidSCAT SDS continuará proporcionando versiones actualizadas y de mejor calidad de productos de datos científicos.

### **2.2.2 Big Data**

Big Data [14] hace referencia a toda aquella información que, por su extensión, no puede ser procesada de la forma habitual. Otra forma de definirlo es como un gran conjunto de datos tan extenso que su procesamiento y almacenamiento superarían la capacidad del software actual.

Las características principales que cumplen los datos definidos como Big Data están recogidas en las llamadas cuatro V del Big Data (Velocidad, Volumen, Veracidad y Variedad), aunque considero necesario incluir una quinta (Valor).

Las tres principales son **Volumen**, **Velocidad** y **Variedad**. Big Data será toda aquella información de gran volumen, que necesite de velocidad de procesamiento y sea de distintos tipos. Sin embargo, que los datos sean verdaderos también es muy importante para su posterior análisis, de ahí la cuarta V: Veracidad. Si partimos de datos falsos, por muy bueno que sea el análisis de estos, la conclusión a la que llegaremos será falsa.

Por otro lado, la información recogida puede ser mucha, pero solo sirve de algo si aporta un Valor, por este motivo considero necesaria la introducción de esta quinta V.

Toda esta información puede surgir de los siguientes grupos [15]:

- **Datos generados por las personas** en su día a día mediante las interacciones con dispositivos electrónicos (email, mensajes de texto, registro de llamadas...).
- **Web y Social Media**: referente a la web y a las redes sociales.
- **Machine to Machine (M2M)**: tecnologías que permiten conectarse a otros dispositivos como pueden ser los sensores.
- **Grandes transacciones de datos**.
- **Información biométrica**.

Como se puede observar, no todos los datos los generamos los humanos, hay otras fuentes igual de importantes.

Este fenómeno llamado Big Data está cobrando cada vez más importancia. Cuanta mayor cantidad de datos se tienen es más sencillo tomar decisiones, el problema es que manejar grandes datos tiene su complejidad. Por tanto, saber analizar esta información es un punto clave y lo será cada vez más.



### 2.2.3 Extracción, Transformación y Carga (ETL)

Son tres procesos importantes que se suelen realizar dentro de la minería de datos. Este proceso permite mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y finalmente cargarlos. A continuación, se explican las tres fases [16]:

- **Extracción** (*Extract*): Es la primera parte del proceso y consiste en extraer los datos desde el sistema o los sistemas de origen.
- **Transformación** (*Transform*): Conjunto de modificaciones que se realizan a los datos para amoldarlos a la lógica de negocio de la aplicación.
- **Carga** (*Load*): En este proceso los datos serán cargados en el sistema, bien para ser almacenados en una base de datos, o bien para ser procesados y poder obtener información útil para la toma de decisiones.

### 2.2.4 Apache Hadoop

Apache Hadoop es un framework de código abierto que permite el almacenamiento distribuido y el procesamiento de grandes conjuntos de datos. Hadoop está diseñado para escalar desde un único servidor a miles de máquinas, cada una de las cuales ofrece cómputo y almacenamiento local [17].

Apache Hadoop incluye estos módulos [18]:

- **Hadoop Common**: Utilidades comunes que son compatibles con los otros módulos de Hadoop.
- **Sistema de archivos distribuidos de Hadoop** (HDFS): Sistema de archivos distribuido que proporciona acceso de alto rendimiento a los datos de las aplicaciones.
- **HADOOP YARN**: Framework para la programación de trabajos y la administración de los recursos de clúster.
- **Hadoop MapReduce**: Sistema basado en YARN para el procesamiento paralelo de grandes conjuntos de datos.

Aunque los anteriores puntos son los módulos principales, también incluye otros módulos relacionados: *Ambari*, *Avro*, *Cassandra*, *Chukwa*, *HBase*, *Hive*, *Mahout*, *Pig*, *Spark*, *Tez* y *ZooKeeper*.

### 2.2.5 MapReduce

MapReduce [19] es un framework que proporciona un sistema de procesamiento de datos paralelo y distribuido. Su nombre se debe a sus funciones principales que son Map y Reduce. MapReduce está pensado para la solución práctica de algunos problemas que pueden ser paralelizados.

Se componen de dos funciones:

- **Map:** se encarga de procesar los datos de entrada. Estos datos serán archivos o directorios y serán almacenados en el sistema de archivos HDFS. Estos datos se pasan línea a línea por la función Map, que los procesa y crea pequeños fragmentos de datos formados por un conjunto de clave/valor. La finalidad es producir una colección de datos en la que se identifica un único registro por cada valor utilizando la clave. Todos estos registros quedan ordenados por su clave.
- **Reduce:** procesa los datos que vienen de la etapa anterior. Realiza uno o varios procesos y genera un nuevo conjunto de datos a su salida que son almacenados en HDFS.

## **2.2.6 Data Mining**

Data Mining es una etapa dentro de un proceso mayor llamado extracción de conocimiento en bases de datos (*Knowledge Discovery in Databases* o KDD). La utilidad del Data Mining es que junta las ventajas de varias áreas como la Estadística, la Inteligencia Artificial, la Computación Gráfica, las Bases de Datos y el Procesamiento Masivo, principalmente usando como materia prima las bases de datos. Una buena definición sería "la integración de un conjunto de áreas que tienen como propósito la identificación de un conocimiento obtenido a partir de las bases de datos que aporten un sesgo hacia la toma de decisión" [20] [21].

Concretamente la fase de minado de datos es donde se implementan las técnicas concebidas para extraer patrones, establecer relaciones y formular modelos para el estudio de datos.

## **2.3 Materiales, entornos, herramientas y librerías**

En este apartado se explican todas las herramientas que se han utilizado en el desarrollo del proyecto. La descarga, instalación y configuración de estas herramientas está incluida en el [Anexo 2](#).

### **2.3.1 Software**

En este apartado se van a explicar las plataformas y los sistemas operativos, así como las herramientas y lenguajes de programación utilizados en el proyecto.

#### **2.3.1.1 Sistemas operativos**

Para el desarrollo de este proyecto se han usado dos sistemas operativos, Windows 10 y Ubuntu.

En Windows 10 se ha realizado el proceso ETL completo. Por lo tanto, en él se ha instalado la herramienta FileZilla, para la descarga de los datos, y Anaconda, para la transformación de los mismos.

Tras tener los datos descargados y formateados se ha usado una máquina virtual de SpatialHadoop con Ubuntu. En ella se han procesado los datos a través del proceso MapReduce. Posteriormente se ha utilizado Weka para analizar y mostrar los resultados.

#### **2.3.1.2 Máquina Virtual con SpatialHadoop**

Como se ha mencionado en el punto anterior, ha sido necesaria la instalación de una máquina virtual para la instalación de SpatialHadoop. Para ello, se ha utilizado la herramienta VirtualBox que permite ejecutar un sistema operativo dentro de otro. En ella, se ha instalado la máquina proporcionada por SpatialHadoop.

SpatialHadoop [22] es una extensión de código abierto MapReduce diseñada específicamente para manejar grandes conjuntos de datos de datos espaciales en Apache Hadoop. Aunque en este caso no se iban a usar las directivas de SpatialHadoop, se ha utilizado ya que proporciona la instalación de Hadoop por defecto, evitando así la compleja instalación del mismo.

#### **2.3.1.3 Java**

La versión del JDK (*Java Development Kit*) que se ha instalado es la versión 1.8 aunque también se ha usado la versión 1.7, ya que es la que viene instalada por defecto en la máquina de SpatialHadoop.

#### **2.3.1.4 Python**

Se utiliza Python para la transformación de los datos de los satélites RapidSCAT y ASCAT, ya que se obtienen en formato netCDF-4 y necesitan ser trasladados a ficheros de formato CSV (*Comma-Separated Values*). Para esto se utiliza la versión 2.7, ya que es la última de las versiones de 2.X y se realizó para que funcionase a largo plazo. Se han incluido en esta versión algunas características de versiones 3.X para validar futuras migraciones, incrementando así el tiempo de vida de esta versión y haciéndola bastante estable.

Para realizar la transformación de los datos de los satélites se utiliza la versión 2 de Python, y no la 3, porque tanto la NASA como ASCAT proporcionan programas base implementados en esta versión, que son incompatibles con la 3.



### **2.3.1.5 Librerías para la transformación de los datos**

Para que la transformación de datos se pueda realizar con éxito es importante tener una serie de librerías para el tratamiento de los mismos. Estas librerías son llamadas desde el programa de transformación “transformación.py”, por lo que, de no tenerlas instaladas, fallará en su ejecución. Estas librerías son:

- **netCDF4** [10]: Librería creada por UNIDATA capaz de realizar cálculos científicos con cualquier tipo de datos que se encuentren dentro del fichero, sin necesitar un fichero adicional que le interprete estos datos. Se utiliza en el proyecto, ya que permite tratar ficheros de tipo NetCDF, formato en el que se encuentran los datos del satélite RapidSCAT.
- **Numpy** [23]: Paquete fundamental para la computación científica en Python. Además de sus usos científicos, puede utilizarse como un eficiente contenedor multidimensional de datos genéricos.

### **2.3.1.6 Weka**

Weka [24] es un software de código abierto emitido bajo la Licencia Pública General de GNU. Esta herramienta contiene una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Los algoritmos que contiene pueden aplicarse directamente a un conjunto de datos o pueden ser llamados desde su propio código Java. Weka también posee herramientas para el procesamiento previo de datos, clasificación, regresión, agrupación, reglas de asociación y visualización, aunque además permite instalar otras extensiones.

Weka, por lo tanto, se usará para el análisis de los datos en la máquina de SpatialHadoop. Será la última parte de este proyecto, ya que será ejecutada al terminar el proceso MapReduce.

### **2.3.1.7 Eclipse**

Eclipse es un IDE (*Integrated Development Environment*) utilizado para el desarrollo de este proyecto. Aunque es un IDE genérico, está muy extendido en el uso por la comunidad de desarrolladores Java. Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones.

Este IDE ha sido utilizado con su versión *Mars* en SpatialHadoop ya que es la que viene instalada por defecto. También se ha usado su última versión (*Oxygen*) sobre Windows para la realización de parte de la interfaz gráfica.

### **2.3.1.8 WindowBuilder**

WindowBuilder es un plugin desarrollado por Eclipse para el desarrollo de aplicaciones GUI (*Graphical User Interface*) de forma sencilla. Este plugin incluye unas vistas que permiten desarrollar una interfaz arrastrando y soltando objetos.

WindowBuilder se ha utilizado para el desarrollo de la interfaz de usuario de este proyecto.

### **2.3.1.9 FileZilla**

FileZilla [25] es un cliente FTP que permite descargar datos desde servidores remotos a través de su dirección, un nombre de usuario y una contraseña. Además, incorpora un administrador de servidores para poder guardar las direcciones más utilizadas, evitando introducirlas de nuevo.

La ventaja principal de esta herramienta es que permite conexiones a través de servidores proxy y cortafuegos. Esto facilita mucho el trabajo al usuario, ya que permite continuar con las descargas interrumpidas y poner a subir o descargar archivos en cola, sin necesidad de empezar de cero.

En este TFG, FileZilla se ha utilizado para la descarga inicial de los datos a través del servidor FTP de la NASA.

## **2.3.2 Hardware**

Este proyecto se ha desarrollado sobre una máquina con las siguientes características:

- Procesador: i7-4790k.
- RAM 16GB.
- Almacenamiento: 500GB SSD y 1TB HD.
- Tarjeta gráfica: NVIDIA GeForce GTX 970.

Aunque las características de la máquina son elevadas y con ellas el proyecto se ha podido realizar sin ningún problema, no son ni de cerca las ideales para la realización de un proyecto de Big Data.

### **3. Desarrollo del proyecto**

En este capítulo se hablará sobre los pasos realizados en el desarrollo del proyecto. Se comenzará hablando sobre el estudio de viabilidad necesario para realizar el proyecto, seguido del reparto del tiempo estimado y real. Tras ello, se explicará el desarrollo de todo el proyecto y, por último, cómo se realiza el análisis de los datos obtenidos.

#### **3.1 Estudio de viabilidad**

El estudio de viabilidad de este proyecto se ha comprobado de la siguiente manera:

El objetivo de este proyecto es estudiar las velocidades del viento máximas y medias en determinadas regiones del planeta. Por ello, lo primero será obtener los datos meteorológicos. Esto es posible gracias a un repositorio FTP de la NASA. En él se almacenan libremente los datos de varios satélites. Estos datos se pueden descargar con herramientas como FileZilla.

Una vez descargados sería necesaria la transformación de los mismos a un formato más cómodo. Esto es posible gracias a un programa proporcionado por Daniel Teomiro Villa en su TFG [1].

Una vez descargados y transformados, lo siguiente sería procesarlos para obtener los valores de las velocidades en las regiones que se le indiquen. Este proceso conllevaría un procesamiento de grandes cantidades de datos. Para este procesamiento se puede usar la herramienta Hadoop. Concretamente, se puede utilizar un proceso MapReduce, donde el Map sea el encargado de filtrar todos los datos según las regiones y el Reduce calcule los valores máximos y medios de cada región. Este proceso puede conllevar una gran cantidad de tiempo por las condiciones de la máquina, aun así, se puede ejecutar de igual manera.

La última parte que quedaría sería el análisis de estos datos. Esto es posible utilizando la herramienta de Weka que permite aplicar ajustes de regresión lineal sobre un conjunto de datos.

### 3.2 Previsión del reparto del tiempo

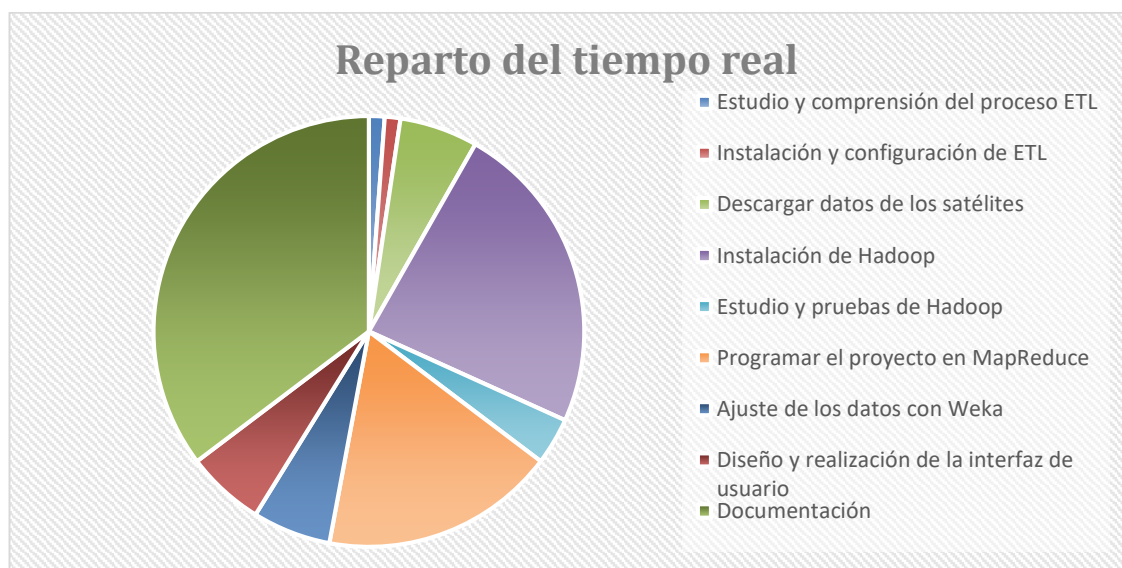
A continuación, se muestran dos tablas con la previsión del reparto del tiempo. En primer lugar, se muestra la previsión inicial estimada y luego la real.

Tareas	Estimación en días
Estudio y comprensión del proceso ETL	2
Instalación y configuración de ETL	2
Descargar datos de los satélites	1
Instalación de Hadoop	10
Estudio y pruebas de Hadoop	3
Programar el proyecto en MapReduce	10
Ajuste de los datos con Weka	5
Diseño y realización de la interfaz de usuario	10
Documentación	30
<b>Total</b>	<b>69</b>

**Tabla 1: Tareas propuestas y estimación del tiempo de realización**

Tareas	Estimación en días
Estudio y comprensión del proceso ETL	1
Instalación y configuración de ETL	1
Descargar datos de los satélites	5
Instalación de Hadoop	20
Estudio y pruebas de Hadoop	3
Programar el proyecto en MapReduce	15
Ajuste de los datos con Weka	5
Diseño y realización de la interfaz de usuario	5
Documentación	30
<b>Total</b>	<b>85</b>

**Tabla 2: Tareas realizadas y tiempo utilizado en las mismas**



**Figura 4 Gráfico de la distribución del tiempo**

Como se puede observar en la Tabla 2, se ha empleado más tiempo del previsto en general. Principalmente provocado por las tareas de “Descargar datos de los satélites”, “Instalación de Hadoop” y “Programar el proyecto en MapReduce”. A continuación, se desglosa un comentario sobre cada tarea realizada.

En primer lugar, las dos primeras tareas tienen relación con el proceso ETL. La realización de ese proceso es ajena a este proyecto, por lo tanto, se estimaba un coste de tiempo elevado para su entendimiento y ejecución. A pesar de lo estimado, al final resultó ser bastante sencillo de entender.

La tarea de “Descargar datos de los satélites” se estimaba un tiempo menor ya que solo consistía en descargar unos datos sin tratamiento alguno. Esta tarea se ha llevado bastante más tiempo del previsto debido a la dimensión de los datos. Se han descargado cientos de gigabytes y el tiempo para ello ha sido elevado debido principalmente a la velocidad del servidor que los provee.

La tarea de “Instalación de Hadoop” ha sido sin duda la que más ha retrasado este proyecto. Se comenzó realizando en Windows, donde fue bastante complicado de instalar debido a la falta de documentación actualizada. Tras tenerlo instalado y configurado en Windows unos fallos con librerías en Eclipse hicieron que se pasara a instalar en una máquina virtual de Ubuntu. Tras varios intentos de instalación infructuosos se pasó a realizarlo usando la máquina virtual de SpatialHadoop que ya lo tiene instalado y configurado.

Una vez instalada la máquina de SpatialHadoop, el estudio y pruebas de Hadoop se realizó según lo planificado inicialmente.

La tarea “Programar el proyecto en MapReduce” también retrasó la previsión inicial del proyecto. Aunque se conocía el funcionamiento teórico del proceso MapReduce, se necesitó un tiempo de estudio extra para comprender el funcionamiento del código del mismo.

La tarea “Ajuste de los datos con Weka” ha seguido en torno a lo planificado. Aunque se ha necesitado un tiempo adicional para la ejecución de varias pruebas, el tiempo invertido en el ajuste se ha mantenido correcto.

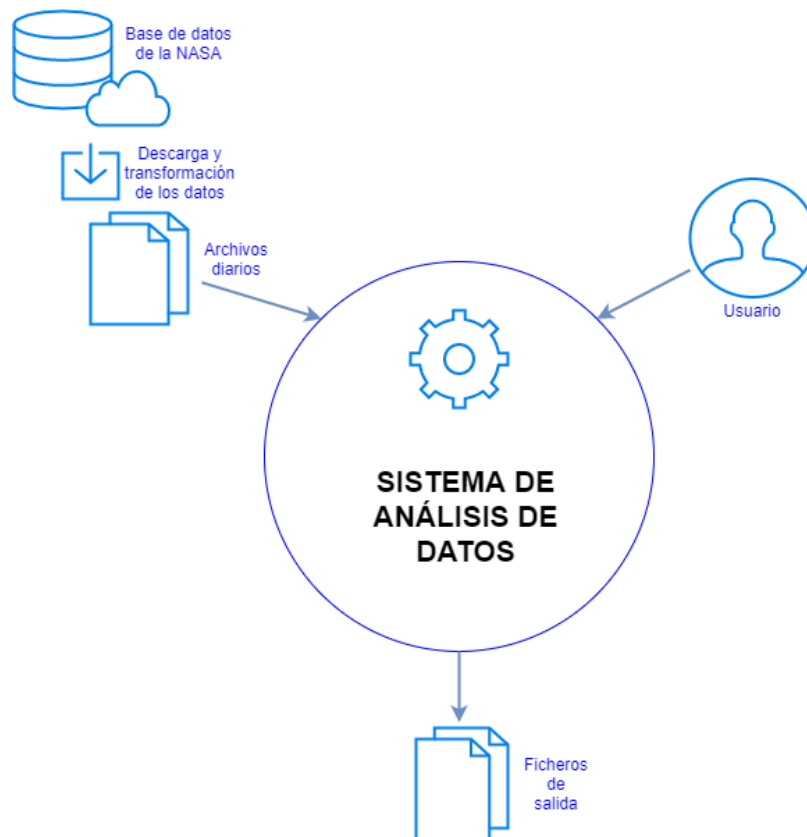
Como tarea personal se propuso la realización de una interfaz para ejecutar todo el proceso. En principio se estimó que se necesitaría bastante tiempo para comparar diferentes alternativas software y el aprendizaje y uso de herramienta elegida. Aunque se

han realizado varias modificaciones de la interfaz inicial, el tiempo requerido para el aprendizaje y creación ha sido bastante inferior al previsto.

Por último, el tiempo de la documentación ha sido en torno al establecido inicialmente, aunque se han ido documentando ciertas partes a lo largo de todo el proceso.

### 3.3 Funcionamiento general

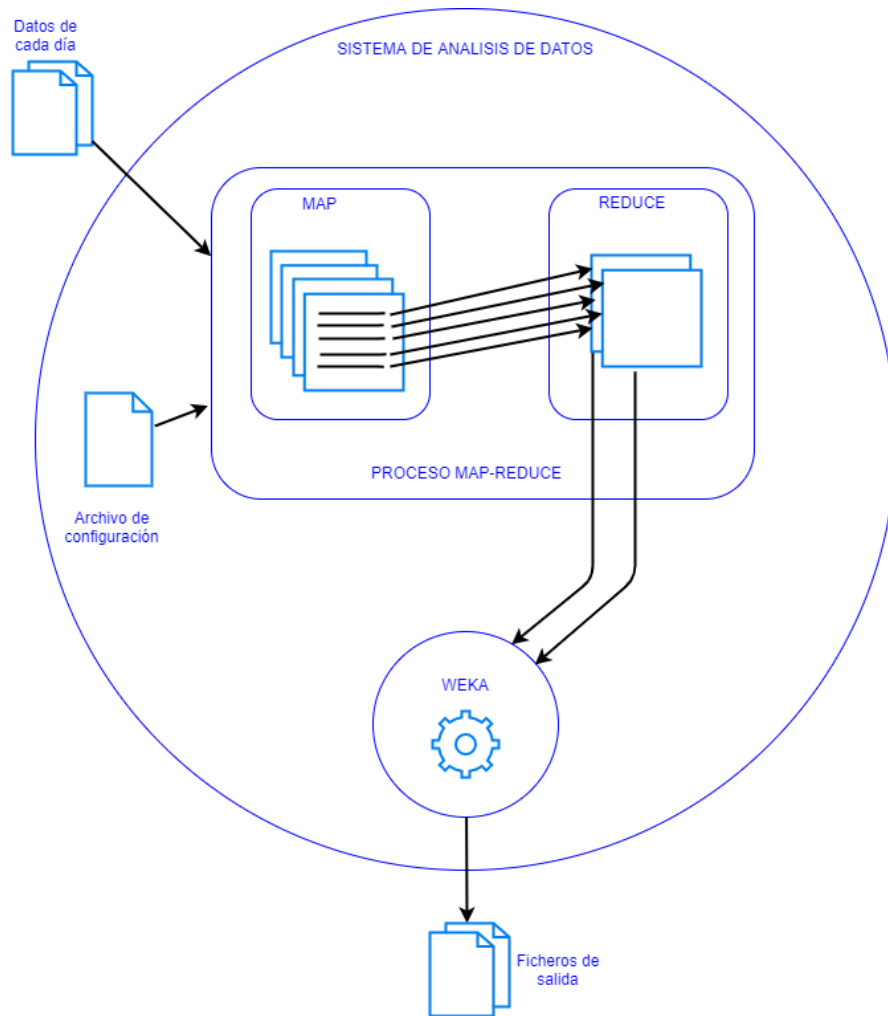
En este apartado se va a explicar cómo funciona el proyecto completo. En primer lugar, se explicarán dos diagramas para comprender el funcionamiento general y después, se detallará cada una de las partes involucradas.



**Figura 5 Esquema general del funcionamiento del sistema**

Como se puede observar en la Figura 5, el sistema recibirá como parámetros de entrada los archivos .csv asociados a los datos meteorológicos de cada día. Por otra parte, también será necesaria la intervención de un usuario que ejecute el sistema. Como salida, se obtendrán unos ficheros con el resultado del procesado de los datos anteriormente mencionados.

Estos datos de entrada deberán haber sido procesados con anterioridad debido a que los servidores de la NASA proporcionan los datos de cada día como un conjunto de archivos y un número de atributos innecesarios para este proyecto.



**Figura 6 Esquema específico del funcionamiento del sistema**

La Figura 6 muestra con más detalle el funcionamiento del sistema. Está formado por dos partes fundamentales, el procedimiento MapReduce y Weka:

El proceso **MapReduce** va a recibir los archivos .csv con la información de cada día y un archivo de configuración. Este proceso tiene dos partes: el proceso Map y el proceso Reduce. El proceso Map será el encargado de gestionar un único punto y mandárselo al proceso Reduce que gestione la región donde se encuentra dicho punto. El proceso Reduce calculará los valores de la velocidad del viento máxima y media de la región y hará una llamada a **Weka** para que analice esos valores.

El número de procesos Map será equivalente a la cantidad de puntos que existan en los archivos. El número de procesos Reduce, en cambio, será igual al número de regiones insertadas en el archivo de configuración. La implementación de estas partes se explicará en el apartado [3.3.2](#).

### 3.3.1 ETL

El proceso ETL, es la primera parte de este proyecto. Su objetivo es descargar los datos de los servidores de la NASA, darles un formato adecuado y posteriormente cargarlos para su uso. En los siguientes apartados, se explicarán cada una de las partes.

#### 3.3.1.1 Extracción

Para la extracción de los datos se va a acceder a un servidor FTP de la NASA. En este repositorio no sólo se almacenan datos del viento de los océanos, sino que también se almacenan otros como la temperatura del océano o el hielo marino. La figura 7 muestra el contenido del directorio FTP donde aparecen los diferentes tipos de datos proporcionados.

### Índice de /

Nombre	Tamaño	Fecha de modificación
GeodeticsGravity/		15/6/17 15:37:00
OceanCirculation/		15/6/17 15:39:00
OceanTemperature/		15/6/17 22:37:00
OceanWinds/		20/6/17 0:01:00
README	1.0 kB	25/10/16 2:00:00
README.txt	866 B	25/10/16 2:00:00
SalinityDensity/		15/6/17 15:46:00
SeaIce/		15/6/17 15:47:00
SeaSurfaceTopography/		15/6/17 15:50:00
allData/		22/5/17 5:01:00
common	0 B	18/1/17 1:00:00
misc/		13/7/16 2:00:00
quikscat	1.3 kB	24/2/17 1:00:00
seawinds	1.1 kB	24/2/17 1:00:00

Figura 7 Repositorio FTP de la NASA

Para descargar estos datos se ha usado la herramienta FileZilla, ya que permite la pausa y la reanudación de las descargas, así como realizar conexiones rápidas y fiables, controlando de esta manera los posibles fallos del sistema. La utilización de esta herramienta se explica en el [Anexo 1](#), concretamente en el apartado [1.2.1](#).

Los datos que se van a utilizar en este proyecto proceden principalmente del satélite RapidSCAT, aunque también funciona de igual manera con los de QuikSCAT y el ASCAT.

Para la descarga de los datos del satélite RapidSCAT, se accede a la dirección del repositorio FTP de PODAAC (<ftp://podaac-ftp.jpl.nasa.gov/>), se navega hasta el directorio OceanWinds, y a continuación se accede al directorio de RapidSCAT.

Una vez dentro se muestran diferentes formatos de los datos, en este caso, se han descargado los datos con densidad 12,5 km<sup>2</sup> del año 2015.



Si se quisieran descargar los datos de otro satélite se realizaría de la misma forma que para el RapidSCAT, pero accediendo al directorio correspondiente.

### **3.3.1.2 Transformación**

Una vez descargados los datos es necesaria una transformación de los mismos para eliminar los campos que no interesan y para juntar los datos en un mismo archivo de menor tamaño.

Debido a que este proyecto es consecuencia de otros, la transformación de estos archivos se realiza usando un programa en Python creado por Daniel Teomiro Villa en su TFG “*Integración de datos de vientos marinos en Oracle NoSQL*” [1]. A continuación, se va a explicar a grandes rasgos el funcionamiento del mencionado programa. Para más información se puede acudir a su TFG.

El programa se ejecuta desde consola y una vez abierto muestra 3 opciones de transformación. Permite transformar datos de tipo L2B de QuikSCAT, datos de tipo L2B12 de RapidSCAT o datos de tipo L2 de ASCAT.

El hilo de ejecución del programa varía un poco según el tipo de dato que se le indique debido a que los archivos tienen composiciones diferentes, pero esto es transparente al usuario.

Una vez elegido el satélite, el programa pide la ruta donde se encuentran los archivos descargados previamente. Acto seguido, pide la ruta donde va a dejar los archivos .csv resultantes. Tras ello, el programa procede a transformar los datos.

La importancia de la transformación reside en el filtrado masivo que los campos, en el que finalmente deja solo 4 de interés: Latitud, longitud, dirección del viento y velocidad del viento. Así mismo, aligera mucho el tamaño de datos a tratar, por ejemplo, los datos del año 2015 en bruto ocupan 363 GB y una vez transformados ocupan 76GB. El tamaño de los archivos está directamente relacionado con los tiempos de ejecución, por lo que es un parámetro de suma importancia.

### **3.3.1.3 Carga**

La fase de carga se realiza al usar los ficheros .csv resultantes para el procesamiento de los mismos. Este proceso se va a usar en el siguiente apartado ya que el proceso MapReduce tendrá que leer los datos de todos los días para poder funcionar.

Debido al gran tamaño de los datos es importante realizar una ampliación de la memoria de la máquina virtual donde se usa Hadoop o bien generar carpetas compartidas. Ambos casos se explican en el [Anexo 2](#), en los apartados [2.4.2.2](#) y [2.7](#).

### 3.3.2 Procedimiento MapReduce

En este apartado se va a explicar el funcionamiento del proyecto usando Hadoop con Eclipse. Así mismo, se explicarán todas las clases creadas y su interacción.

El objetivo de este procedimiento es seleccionar unas regiones de interés y por cada una de ellas obtener el valor máximo y medio de la velocidad del viento por cada día. Es importante recordar que este procedimiento recibe un conjunto de archivos *.csv* donde cada uno tiene la información de un día y en él, todos los puntos del planeta con sus valores. La salida de este procedimiento serán unos archivos que leerá Weka para su análisis.

El proyecto creado llamado *TFG\_Jaime* contiene dos paquetes, UI y MapReduce. El paquete MapReduce tiene 5 clases encargadas de gestionar este proceso. El paquete UI contiene 3 clases que gestionan la interfaz de usuario. A continuación, se explicarán las clases del paquete MapReduce y después las del paquete UI.

#### Paquete MapReduce

De las 5 clases del paquete MapReduce hay 3 que son principales, *Main\_MapReduce*, *MapClass*, *ReduceClass*. Por otra parte, están las clases *Región* y *Punto* que son utilizadas por las otras.

##### Clase Main MapReduce

Esta clase es la encargada de gestionar todo el proceso MapReduce. Genera un trabajo y le indica qué clase es la que va a ejecutar el proceso Map (*MapClass*) y qué clase es la que ejecuta el proceso Reduce (*ReduceClass*). Además, establece cuál es el tipo de datos que entran y salen en el Map y lo mismo con el Reduce (Text en ambos casos).

Solo tiene un método denominado “*ejecutar*” que recibe los parámetros que se le pasan desde la interfaz. Estos parámetros serán explicados en los apartados referentes a la interfaz.

##### Clase MapClass

Esta clase, como se ha comentado en el punto anterior, es la encargada de ejecutar el proceso Map. Para este proyecto se van a generar tantos procesos Maps como puntos se tengan, es decir, el gestor va a coger todos los archivos *.csv* con la información de cada día. Tras ello, por cada línea del *.csv*, que corresponde a un punto, va a llamar a un proceso Map que será el encargado de decidir qué pasará con este punto.

El objetivo del Map será realizar un filtrado de los datos. En primer lugar, cogerá un punto y comprobará si pertenece a alguna de las regiones. En caso de pertenecer a alguna se lo pasará al proceso Reduce asociado a esa región.

La lista de regiones de interés la recibe mediante un parámetro. Este parámetro contiene la ruta de un fichero de configuración en el que se han establecido previamente las regiones que se quieran analizar. Este fichero está estructurado de la siguiente manera:

Cada región será establecida con 4 coordenadas, separadas por comas, asociadas a los dos puntos extremos de la región. El primer punto será el inferior izquierdo y el segundo el superior derecho.

Por lo tanto, esta clase tiene una única función denominada “*map*”, que va a recibir por parámetro un objeto de tipo Text. El proceso que va a realizar dicha función es el siguiente: En primer lugar, va a transformar los datos de tipo Text a la clase *Punto* mediante un *StringTokenizer* para dividir el tipo Text. Tras ello, genera la lista de regiones leyendo el archivo de configuración. Por último, recorre esa lista y comprueba si ese *Punto* está dentro de una región. En caso afirmativo llama a un proceso Reduce poniendo como clave el número de la región y como valor el punto. En caso negativo descarta el punto.

#### Clase *ReduceClass*

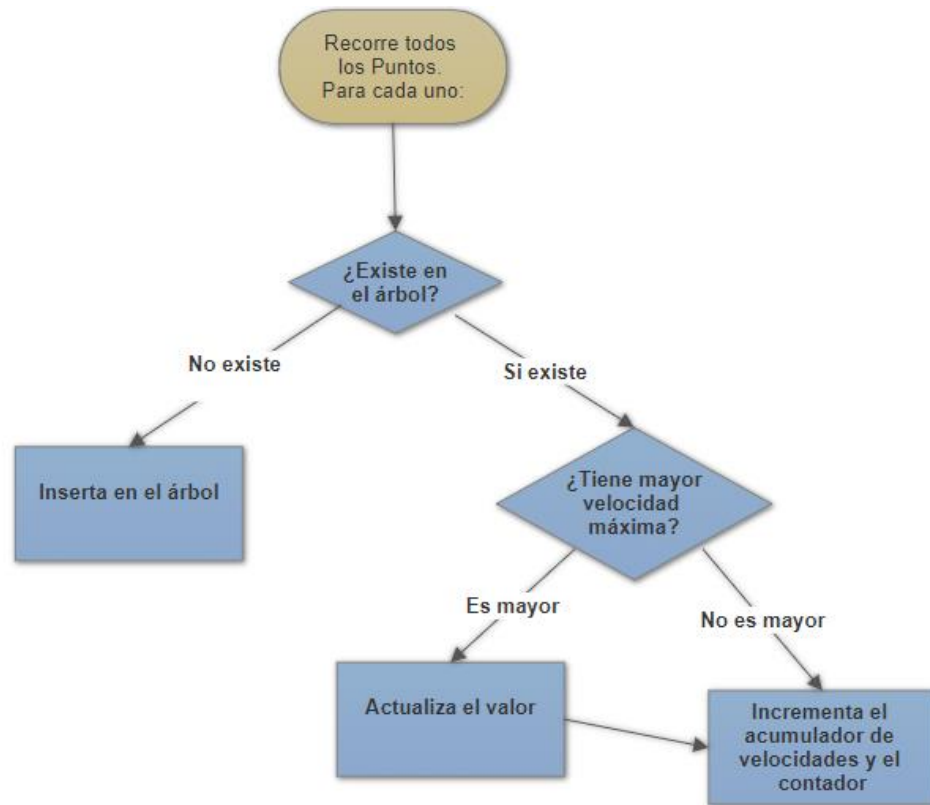
Como su nombre indica, esta clase es la encargada de realizar el proceso Reduce. Se van a generar tantos procesos Reduce como regiones existan. Cada Reduce recibe los datos de una única región.

Es importante fijarse en que los puntos que recibe pertenecen a diferentes días, y que un día puede tener muchísimos puntos en función del tamaño de la región.

El objetivo del Reduce es quedarse con el valor máximo y el valor medio de cada día. Para ello deberá realizar una organización por días y calcular dichos valores. Para realizar esto, se utiliza una estructura de árbol debido a su eficiencia y velocidad para el acceso a los datos. Este árbol tendrá como clave la fecha del día, y como valor un *Punto*.

En su ejecución recorre todos los puntos que tiene y por cada uno lo introduce en el árbol. Si el día no ha sido insertado con anterioridad lo inserta, si, por el contrario, el día existe, compara los valores máximos de la velocidad. En caso de tener una velocidad mayor actualiza la velocidad máxima en el *Punto*. En ambos casos se van modificando dos atributos dentro del punto que servirán para mostrar la velocidad media al final de la ejecución. Estos campos serán un acumulador de velocidades y un contador de número de

puntos, de tal manera que la velocidad media sea el resultado de la división de los dos. Esta ejecución se puede ver mejor en la Figura 8.



**Figura 8 Diagrama de flujo del proceso Reduce**

Una vez que el árbol está completo se procede a generar un archivo *.arff* (*Attribute-Relation File Format*) que será leído por Weka. Este archivo tiene una cabecera especial donde Weka identificará qué es cada campo y el tipo del dato. Para poner dicha cabecera se llama al método *insertHeadARFF*. Tras ello, se recorre el árbol y se escriben los valores en el fichero.

Por último, con el archivo *.arff* completo, se procederá a realizar una llamada a Weka para realizar el análisis. Se ejecutará el método *runWeka* que crea el comando y llama al método *executeCommand* para realizar la llamada al sistema y ejecutarlo. La salida generada por Weka se guardará en el mismo directorio donde se ha generado el archivo *.arff*.

### Clase Punto

Esta clase representa la información de unas coordenadas concretas. Está compuesta por 8 atributos: 6 de ellos vienen dados por la información en los archivos *.csv* (tipo del satélite, día en fecha juliana, latitud, longitud, velocidad del viento y dirección del viento)

y los otros dos sirven para calcular la velocidad media (sumatorio de velocidades y contador).

Además de los métodos *sets* y *gets* asociados a los atributos y tiene dos más de especial interés:

El primero es un método denominado *incrementarVelocidadVientoMedia*. Este método será llamado por el proceso Reduce cada vez que se encuentre con un punto ya existente en el árbol. Recibe por parámetro una nueva velocidad, la cual sumará al acumulador de velocidades e incrementará en 1 el contador.

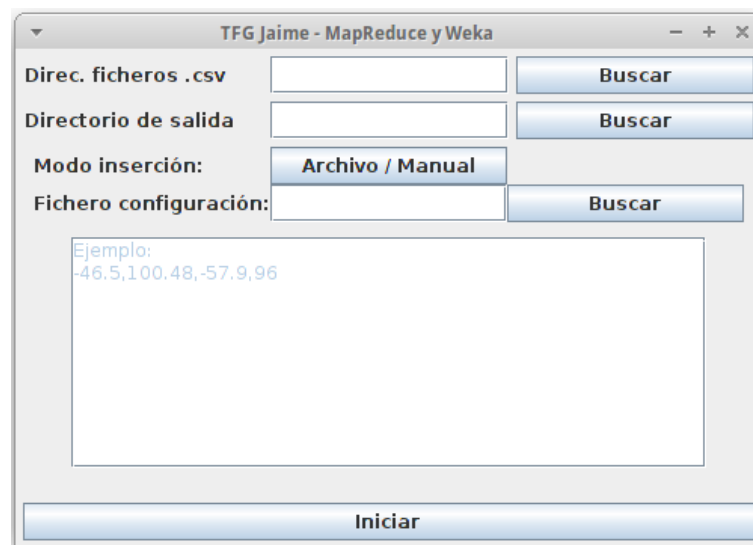
El segundo es el método *toString*. Este método pintará el punto en el archivo *.arff*. También será el encargado de convertir la fecha de tipo Juliana (YYYYDDD) a Gregoriana (YYYY-MM-DD) y calcular la velocidad media.

### Clase Region

Representa una región formada por dos puntos. Esta clase es utilizada por el proceso Map para establecer las regiones del archivo de configuración y generar la lista de regiones.

### **Paquete UI**

A continuación, se van a explicar las 3 clases asociadas al paquete UI. Estas tres clases son las encargadas de generar la interfaz gráfica mostrada en la Figura 9. La interfaz de usuario se ha generado usando WindowBuilder, que se ha mencionado en el capítulo anterior. Con esta herramienta se ha generado un *JFrame* y dos *JPanel* contenidos dentro del anterior que corresponden a las siguientes clases:



**Figura 9** Interfaz gráfica del proyecto

### Clase MainWindow

Esta clase es la encargada de gestionar la ventana completa. Es un *JFrame* y por lo tanto tiene de atributos los dos *JPanel*, uno encargado de la parte de los directorios y el otro encargado del modo de inserción del archivo de configuración.

Por lo tanto, recibirá los datos de los campos y llamará al proceso *ejecutar* de la clase *Main\_MapReduce* para lanzar el proyecto.

### Clase Rutas

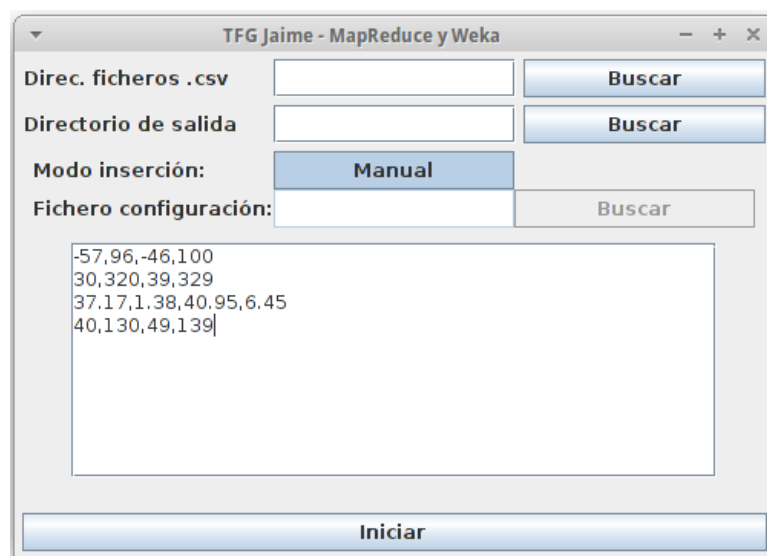
Esta clase será la encargada de gestionar los directorios, concretamente el directorio donde se encuentran los archivos *.csv* y el directorio de salida donde se situarán los archivos *.arff* para que Weka los analice.

Permite la inserción de rutas de forma manual o a través de un explorador de archivos al pulsar sobre los botones “Buscar” correspondientes.

### Clase SeleccionManual

Esta clase es la encargada de buscar o generar el archivo de configuración con las regiones de interés.

En el panel hay un botón que indica el modo de inserción, al comienzo muestra el texto “Archivo / Manual”. Por defecto solo dejará buscar una ruta con el explorador o insertando la ruta. En caso de ser pulsado, cambiará el texto a “Manual” o “Archivo”, bloqueando/activando los campos asociados al mismo. En caso de dejarlo en modo manual habrá que insertar las regiones en el campo preparado para ello, tal como se muestra en la Figura 10.



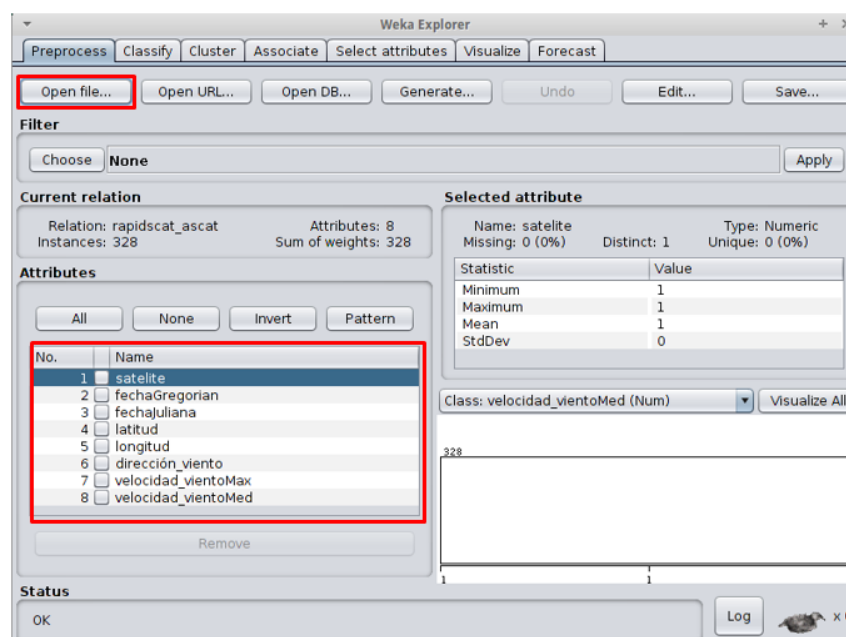
The screenshot shows a Java Swing window titled "TFG Jaime - MapReduce y Weka". It contains several input fields and buttons. The "Modo inserción:" field is set to "Manual". The "Direc. ficheros .csv" and "Directorio de salida" fields have "Buscar" buttons next to them. The "Fichero configuración:" field has a "Buscar" button. Below these fields is a large text area containing the following coordinates: -57,96,-46,100; 30,320,39,329; 37,17,1,38,40,95,6,45; 40,130,49,139. At the bottom of the window is a large "Iniciar" button.

Figura 10 Ejemplo de la interfaz con inserción manual

### 3.3.3 Weka

Weka realizará el análisis de los datos de cada región ya procesados. Cada proceso reduce genera un archivo *.arff* que contiene la información de las velocidades del viento máximas y medias de una región. El proceso reduce creará estos ficheros en el directorio que se le haya indicado desde la interfaz. Cada uno de esos archivos, será ejecutado con Weka desde la llamada a sistema del Reduce. Asimismo, también se puede llamar de forma manual para poder visualizar las gráficas, este proceso se realizará de la siguiente manera:

Se ejecuta Weka y se abre el explorador. En él, se abre el fichero *.arff* correspondiente mediante el botón “Open file...”. Tras abrirlo se pueden ver los atributos del archivo tal y como se muestra en la Figura 11.



**Figura 11 Pantalla de Weka tras insertarle un archivo**

A continuación, se abre la pestaña de “Forecast” que es paquete de Weka que permite analizar series temporales. Ahora hay que realizar una serie de configuraciones para ajustar la regresión lineal:

El primer paso se realiza en la pestaña de “Basic configuration”. En ella se selecciona el atributo que vamos a analizar, en este caso se va a poner la velocidad del viento máxima, aunque también se podría ejecutar con la velocidad media. Tras ello, habrá que cambiar unos parámetros: el número de unidades de tiempo a predecir se subirá de 1 a 10, el periodo de tiempo se cambiará a desconocido y, por último, se activará la evaluación para poder ver los errores que se producen. Estos ajustes se pueden ver en la Figura 12.

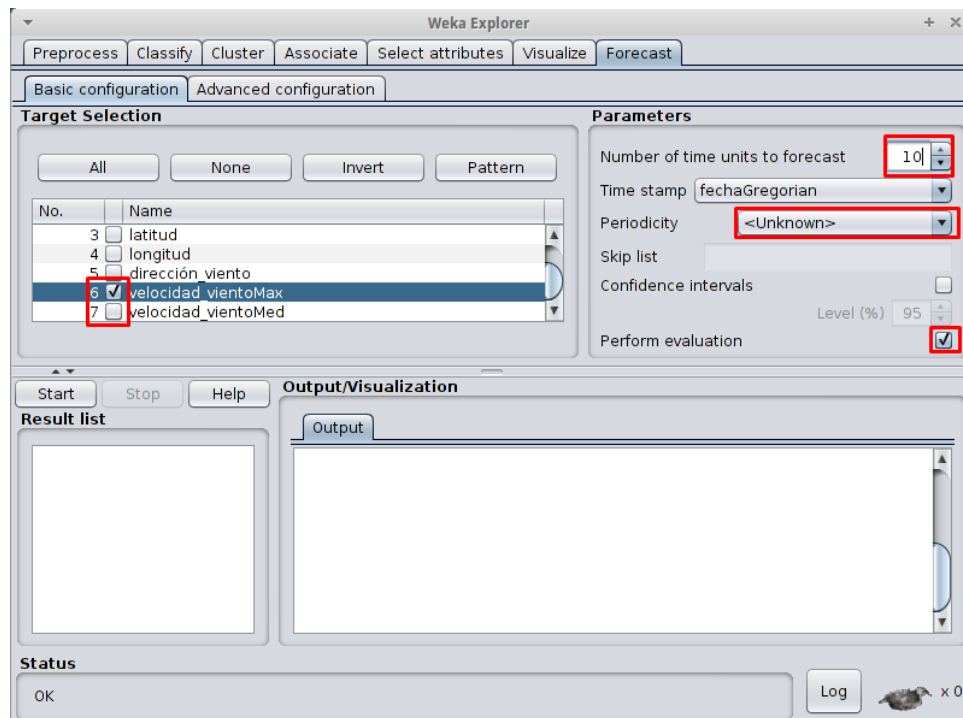


Figura 12 Configuración básica de Weka (paquete Forecast)

Los siguientes pasos se realizarán sobre la configuración avanzada. En la pestaña de “Lag creation” habrá que añadir un retraso personalizado. Esto hará que se desechen los primeros valores para ajustar la recta. Esta configuración se muestra en la Figura 13.

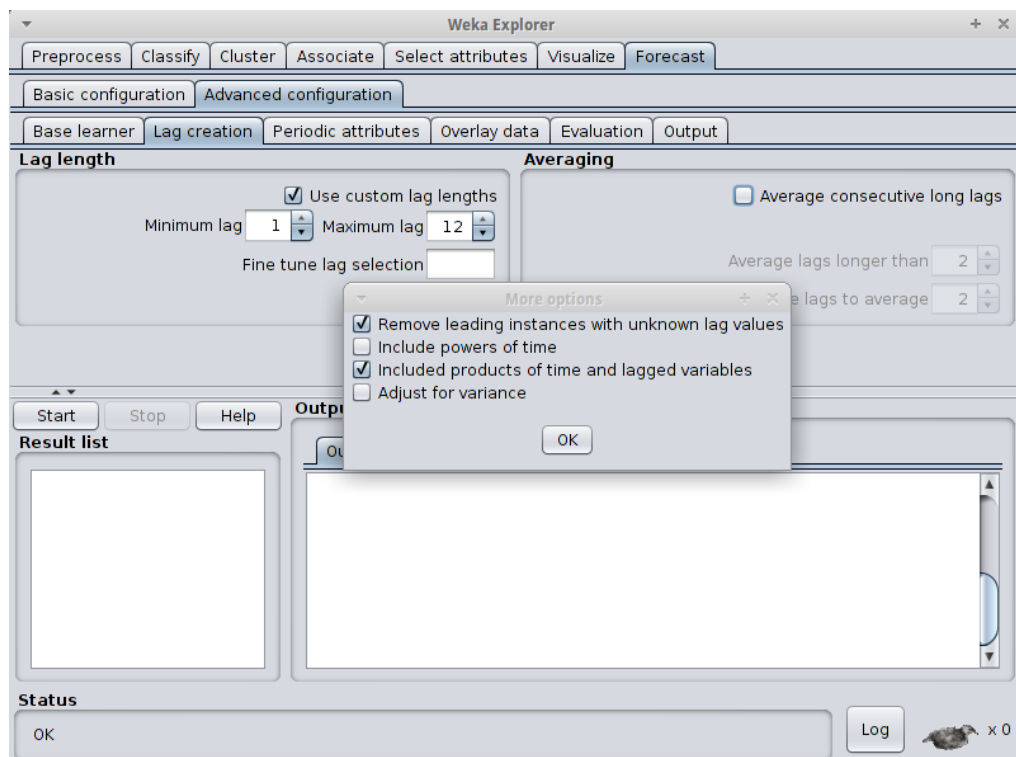
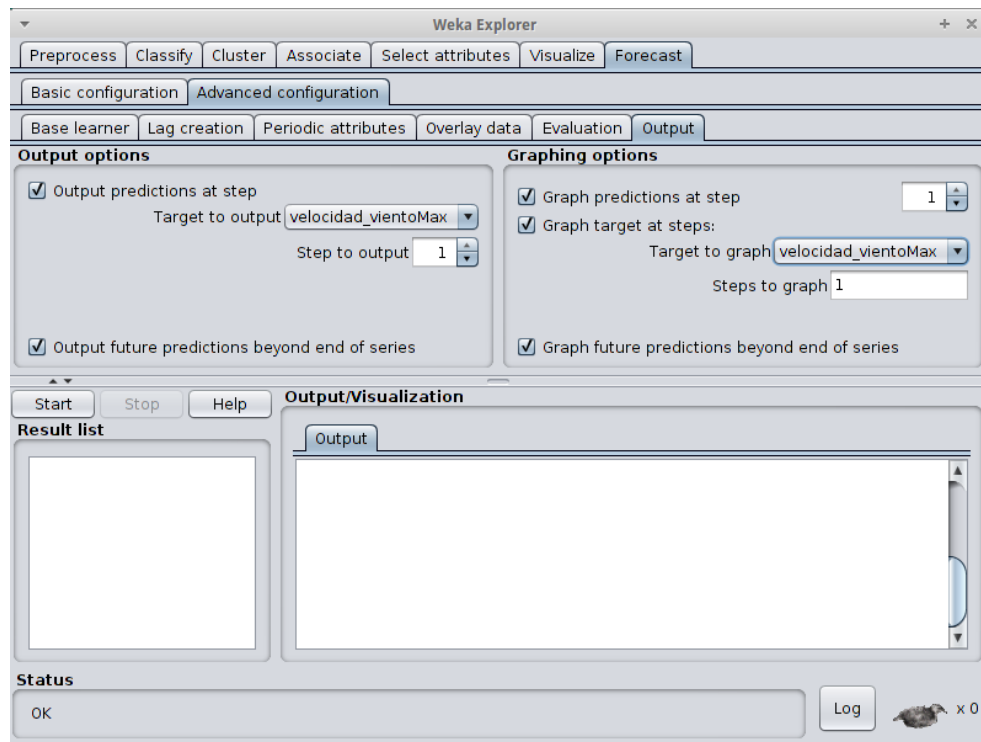


Figura 13 Configuración avanzada de Weka - Lag creation



Por último, en la pestaña Output habrá que activar los gráficos sobre el atributo de velocidades máximas. Este ajuste se muestra en la Figura 14.

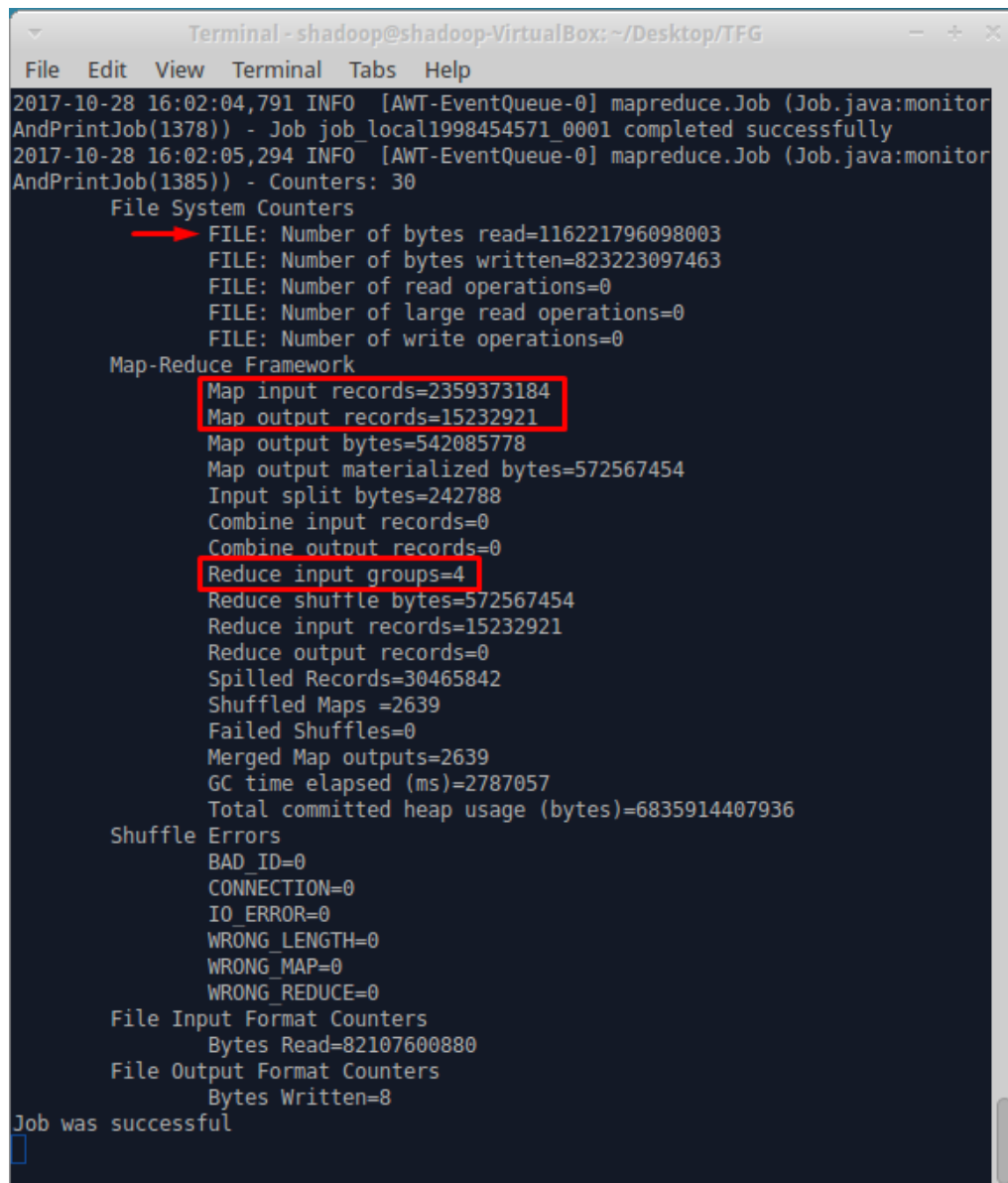


**Figura 14 Configuración avanzada de Weka – Output**



## 4. Resultados

En este capítulo se mostrarán los resultados obtenidos al ejecutar el TFG con los datos del año 2015 del satélite RapidSCAT. Se ha ejecutado el proceso MapReduce con 4 regiones (Océano Índico cerca de Australia, Océano Atlántico norte, Islas Baleares y el Mar de Japón). Al terminar el proceso se han generado 4 archivos *.arff*, con los datos calculados de cada región, y otros 4 archivos *.txt* con las salidas de la ejecución de Weka. En la Figura 16 se pueden observar estos archivos, pero antes de verlos, vamos a centrarnos en la salida por consola mostrada en la Figura 15:

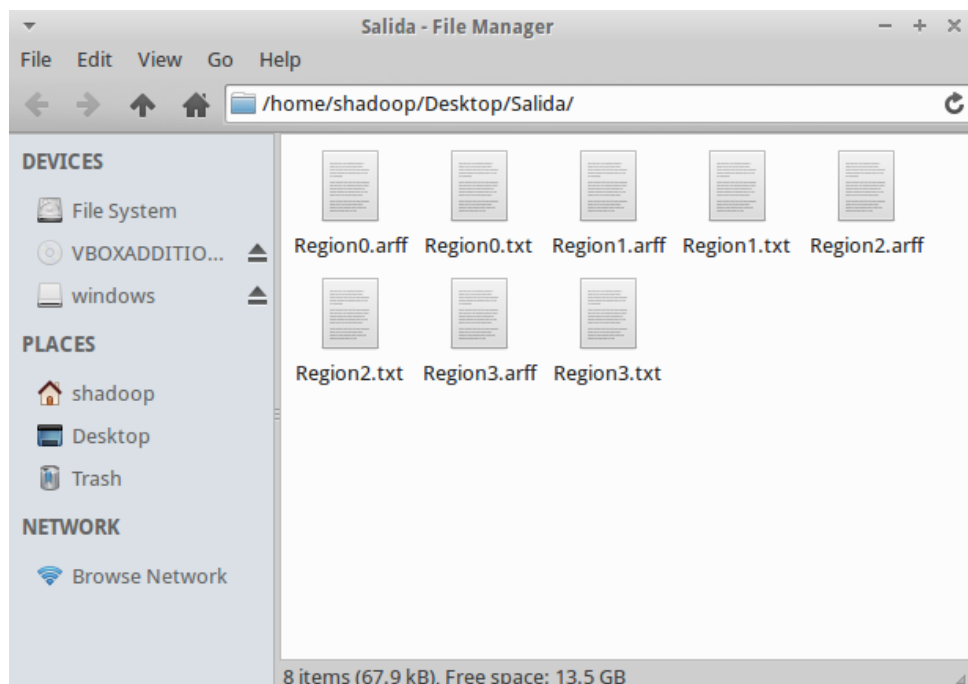


```
Terminal - shadoop@shadoop-VirtualBox: ~/Desktop/TFG
File Edit View Terminal Tabs Help
2017-10-28 16:02:04,791 INFO [AWT-EventQueue-0] mapreduce.Job (Job.java:monitor
AndPrintJob(1378)) - Job job_local1998454571_0001 completed successfully
2017-10-28 16:02:05,294 INFO [AWT-EventQueue-0] mapreduce.Job (Job.java:monitor
AndPrintJob(1385)) - Counters: 30
File System Counters
  FILE: Number of bytes read=116221796098003
  FILE: Number of bytes written=823223097463
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
Map-Reduce Framework
  Map input records=2359373184
  Map output records=15232921
  Map output bytes=542085778
  Map output materialized bytes=572567454
  Input split bytes=242788
  Combine input records=0
  Combine output records=0
  Reduce input groups=4
  Reduce shuffle bytes=572567454
  Reduce input records=15232921
  Reduce output records=0
  Spilled Records=30465842
  Shuffled Maps =2639
  Failed Shuffles=0
  Merged Map outputs=2639
  GC time elapsed (ms)=2787057
  Total committed heap usage (bytes)=6835914407936
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=82107600880
File Output Format Counters
  Bytes Written=8
Job was successful
```

Figura 15 Salida por consola tras realizar la ejecución

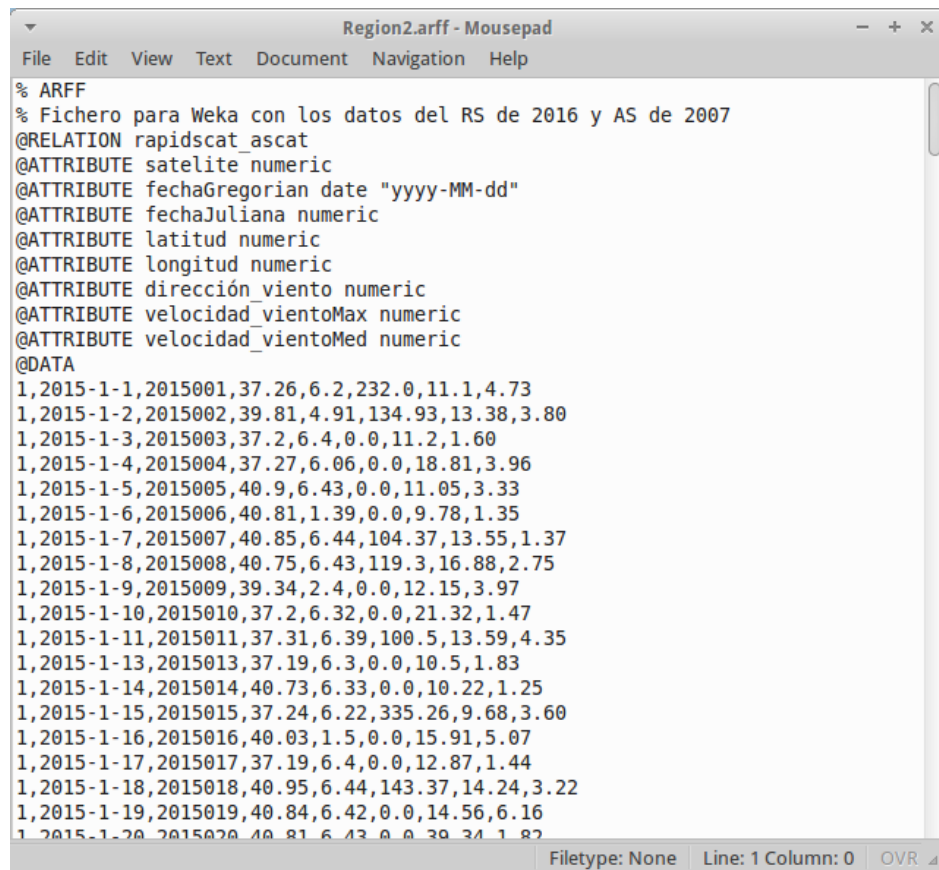
En la figura anterior se pueden observar bastantes campos que se generan automáticamente al terminar el MapReduce. De especial interés son aquellos que aparecen remarcados:

- Sobre los archivos leídos llama la atención el gran número de bytes leídos. Este número se corresponde a los 82,1Gb que ocupan los datos del año 2015.
- Lo siguiente que destaca son los datos de entrada y salida del Map. El primero representa la cantidad de puntos que se han tratado y, el segundo, la cantidad de ellos que se encontraban dentro de las regiones y que por lo tanto se han mandado al proceso Reduce. Con estos datos se puede analizar el filtrado masivo de datos que hace el proceso Map.
- El último dato que llama la atención es la cantidad de grupos Reduce generados (4). Este número se corresponde a las regiones que se insertaron al iniciar el programa.



**Figura 16** Carpeta con los archivos .arff y .txt generados tras la ejecución

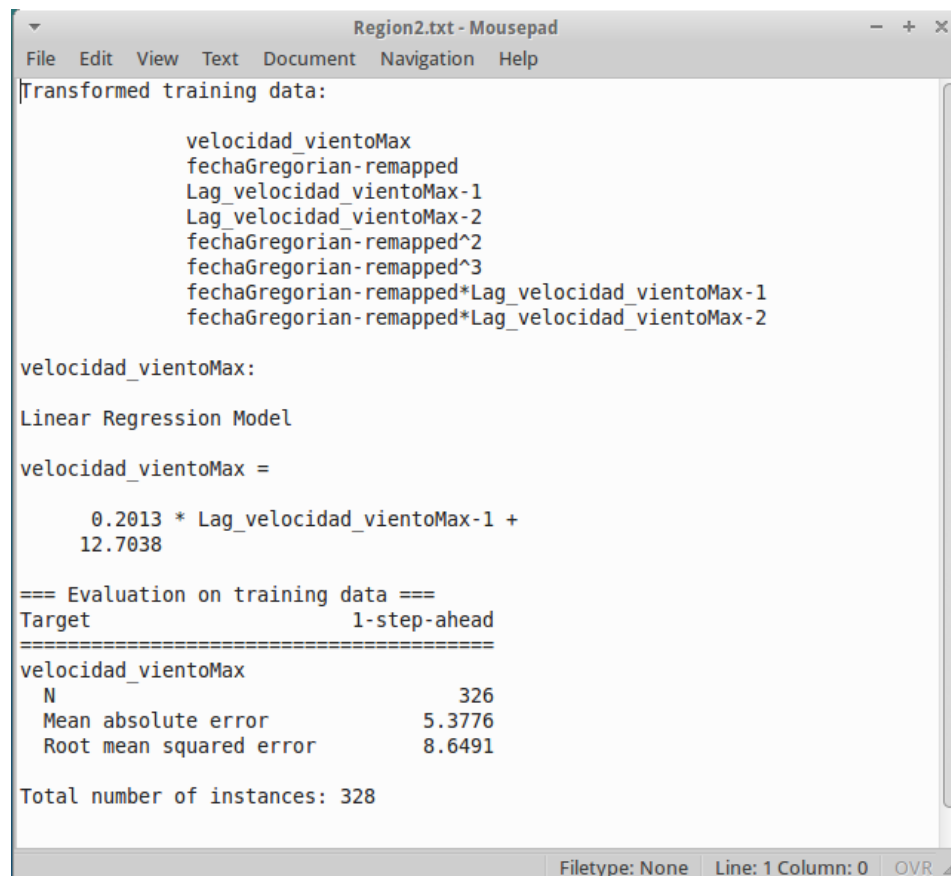
A continuación, nos centraremos en los archivos referentes a las Islas Baleares (región 2), los demás archivos generados siguen la misma estructura que estos:



```

% ARFF
% Fichero para Weka con los datos del RS de 2016 y AS de 2007
@RELATION rapidscat_ascat
@ATTRIBUTE satelite numeric
@ATTRIBUTE fechaGregorian date "yyyy-MM-dd"
@ATTRIBUTE fechaJuliana numeric
@ATTRIBUTE latitud numeric
@ATTRIBUTE longitud numeric
@ATTRIBUTE dirección_viento numeric
@ATTRIBUTE velocidad_vientoMax numeric
@ATTRIBUTE velocidad_vientoMed numeric
@DATA
1,2015-1-1,2015001,37.26,6.2,232.0,11.1,4.73
1,2015-1-2,2015002,39.81,4.91,134.93,13.38,3.80
1,2015-1-3,2015003,37.2,6.4,0.0,11.2,1.60
1,2015-1-4,2015004,37.27,6.06,0.0,18.81,3.96
1,2015-1-5,2015005,40.9,6.43,0.0,11.05,3.33
1,2015-1-6,2015006,40.81,1.39,0.0,9.78,1.35
1,2015-1-7,2015007,40.85,6.44,104.37,13.55,1.37
1,2015-1-8,2015008,40.75,6.43,119.3,16.88,2.75
1,2015-1-9,2015009,39.34,2.4,0.0,12.15,3.97
1,2015-1-10,2015010,37.2,6.32,0.0,21.32,1.47
1,2015-1-11,2015011,37.31,6.39,100.5,13.59,4.35
1,2015-1-13,2015013,37.19,6.3,0.0,10.5,1.83
1,2015-1-14,2015014,40.73,6.33,0.0,10.22,1.25
1,2015-1-15,2015015,37.24,6.22,335.26,9.68,3.60
1,2015-1-16,2015016,40.03,1.5,0.0,15.91,5.07
1,2015-1-17,2015017,37.19,6.4,0.0,12.87,1.44
1,2015-1-18,2015018,40.95,6.44,143.37,14.24,3.22
1,2015-1-19,2015019,40.84,6.42,0.0,14.56,6.16
1,2015-1-20,2015020,40.81,6.43,0.0,39.34,1.82
    
```

Figura 17 Archivo .arff de las Islas Baleares de 2015



```

Transformed training data:

    velocidad_vientoMax
    fechaGregorian-remapped
    Lag_velocidad_vientoMax-1
    Lag_velocidad_vientoMax-2
    fechaGregorian-remapped^2
    fechaGregorian-remapped^3
    fechaGregorian-remapped*Lag_velocidad_vientoMax-1
    fechaGregorian-remapped*Lag_velocidad_vientoMax-2

velocidad_vientoMax:

Linear Regression Model

velocidad_vientoMax =

    0.2013 * Lag_velocidad_vientoMax-1 +
    12.7038

=== Evaluation on training data ===
Target          1-step-ahead
=====
velocidad_vientoMax
N                326
Mean absolute error    5.3776
Root mean squared error 8.6491

Total number of instances: 328
    
```

Figura 18 Archivo .txt generado por Weka de las Islas Baleares de 2015

Como se puede ver en la Figura 17, cada línea tiene la información de un único día. Estos días se pueden ver en el segundo y tercer atributo (fechas gregoriana y juliana, respectivamente). Asimismo, los últimos dos atributos son los de especial interés ya que son los relacionados con las velocidades medias y máximas.

En la Figura 18 se puede ver la salida de la ejecución de Weka. Esta ejecución se centra en la velocidad del viento máxima. En este archivo se puede ver el conjunto transformado de los datos y el modelo de regresión que usa. Una parte interesante es la de *Evaluation*, en ella se pueden ver los errores de predicción generados. También llama la atención el número total de instancias tratadas, en este caso 328, deberían ser 365 ya que se le ha insertado todo el año 2015. Esto se debe a que el satélite no cubre todos los puntos del planeta todos los días, por lo tanto, es posible que existan varios días donde el satélite no ha pasado por la región marcada.

La salida de Weka genera bastante información, aun así, es más interesante realizar una ejecución manual de Weka para ver la regresión de forma gráfica. Para ello, se configura Weka como se explicó en el apartado [3.3.3](#) y se obtienen los resultados de las Figuras 19 y 20.

La interpretación de estos resultados queda fuera de los objetivos de este proyecto por ser necesario alguien que domine el campo del estudio. A pesar de ello, se puede observar que con las velocidades máximas (Figura 19) la recta de regresión se ajusta más al final que al principio. Asimismo, llama la atención los datos atípicos que se visualizan sobretodo en el mes de septiembre, también se puede ver como la recta de regresión también se ajusta en su propia escala a estos datos. Con las velocidades medias (Figura 20) también ocurre algo parecido pero esta gráfica sigue una distribución bastante más formal al usar los valores medios.

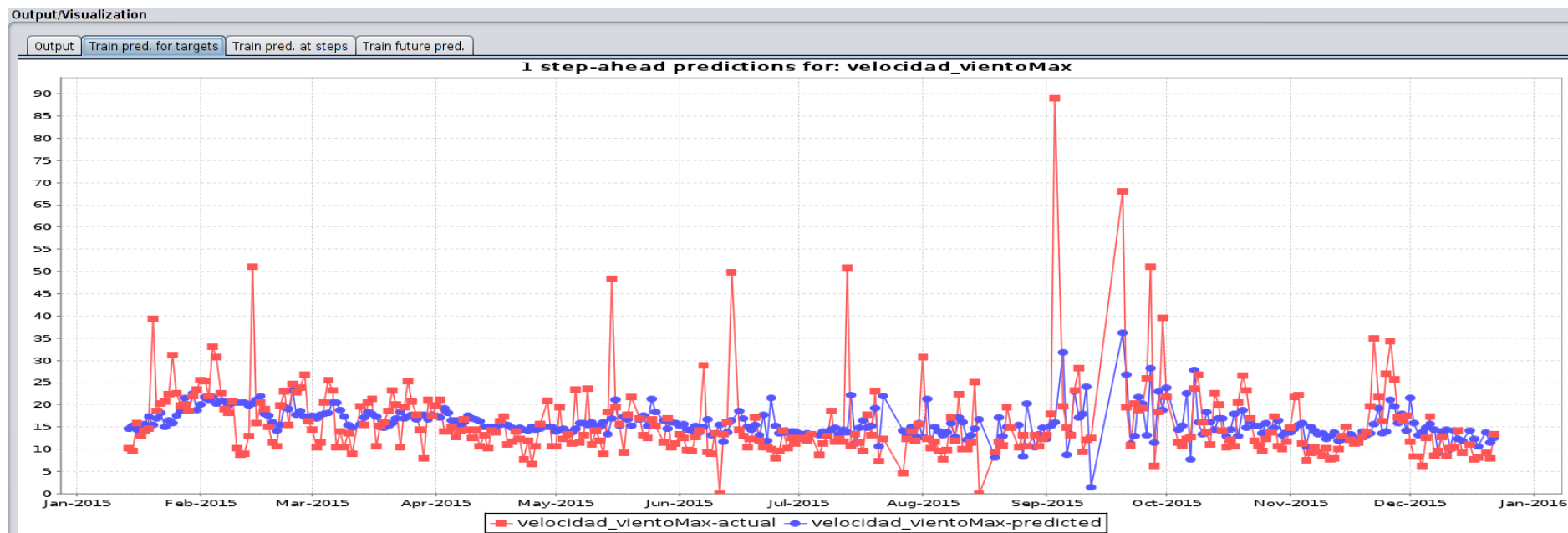


Figura 19 Resultados gráficos sobre la velocidad máxima

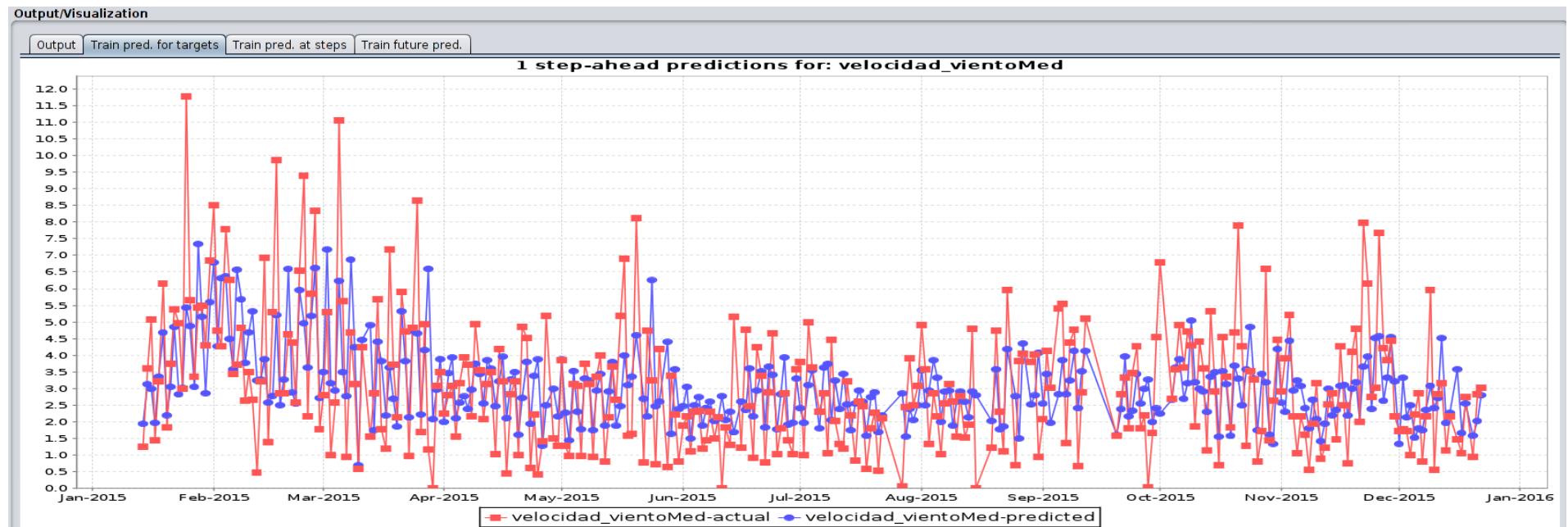


Figura 20 Resultados gráficos sobre la velocidad media



## 5. Conclusiones y Trabajos futuros

En este capítulo se reflexionará sobre el proyecto en general. En primer lugar, se comentarán las conclusiones del proyecto tanto generales como propias. Tras ello, se comentarán cuáles han sido los principales problemas que se han encontrado durante el desarrollo del mismo. Por último, se propondrán una serie de trabajos futuros por si alguien está interesado en continuar con el proyecto.

### 5.1 Conclusiones del proyecto

Tras finalizar el proyecto se puede realizar un repaso sobre los objetivos que se plantearon inicialmente. Se puede observar que se han cumplido todos:

- 1- Se ha identificado la necesidad del análisis con los datos meteorológicos. Así mismo, también se ha comprobado el potencial que puede tener tratar otros tipos de datos.
- 2- Se han estudiado las soluciones existentes al problema y se ha desarrollado una que lo cumpla.
- 3-4 Se ha obtenido información sobre cómo realizar un proceso ETL y se ha realizado.
- 5- Se ha diseñado e implementado un proceso MapReduce que genere unos archivos con la información meteorológica de unas regiones.
- 6- Se ha usado Weka para realizar el análisis de los datos.
- 7- Se han probado diferentes configuraciones para mejorar la precisión de los resultados.
- 8- Se ha implementado una interfaz de usuario para lanzar todo el proceso MapReduce.
- 9- En este apartado se están comprobado si se han cumplido todos los objetivos iniciales.
- 10- Se han planteado diferentes trabajos futuros por si alguien quisiera continuar o ampliar el proyecto

## 5.2 Conclusiones personales

A pesar de las dificultades que se han encontrado a lo largo del proyecto, más las intrínsecas al proyecto en sí, puedo decir con orgullo que he sido capaz de llevarlo a cabo con los recursos que tenía disponibles, no los más adecuados para este tipo de trabajo (ordenador personal). Tenía muchas ganas de realizar un TFG que realizase un proceso entero de Big Data, principalmente por dos motivos: En primer lugar, saber el potencial que tienen estas herramientas para obtener información útil que pueda ser usada en múltiples áreas. En segundo lugar, quería saber si me gustaba esta rama de la informática con el fin de poder dedicarme a ella en el futuro, así ha sido.

Durante la realización de este proyecto he aprendido bastantes cosas, por ejemplo, la realización del proceso ETL, cómo ejecutar un proyecto MapReduce y usar Weka para realizar regresiones lineales. Igualmente hay una parte con la que me siento bastante satisfecho, la interfaz gráfica. Inicialmente no era un objetivo de este proyecto, pero era una espina que tenía clavada durante toda la carrera y quise realizarla para aprender cómo funcionaban. Pensaba que iba a suponer un gran reto, pero al final no fue para tanto.

## 5.3 Principales problemas encontrados

Esta memoria podría haber sido organizada en función de los problemas encontrados y las soluciones que se le han dado a los mismos. Como no ha sido así, a continuación, se van a comentar los tres principales problemas encontrados:

1. **Instalaciones de Hadoop:** Este problema ha sido sin duda el peor de todos, llegando incluso a ocupar 3 semanas de trabajo consecutivo. Inicialmente la instalación de Hadoop no parecía compleja, descargar un archivo y sobre él modificar unos archivos *html* para realizar la configuración del nodo.

Esta instalación se empezó realizando en Windows 10, ya que desde la página oficial decían que era compatible, y era el sistema con el cual estoy acostumbrado a trabajar. El problema reside en la falta de documentación actualizada sobre Windows, dado que los usuarios están acostumbrados a ejecutarlo en Ubuntu, ya que fue en el sistema operativo en el que salió por primera vez. Tras varios días superando errores tras errores, se consiguió lanzar el servidor, y que funcionaran los comandos asociados. El problema vino cuando se quiso realizar el proyecto en Eclipse, ya que las directivas de Hadoop necesitaban librerías las cuales no se incluyen al descargártelo de la página

oficial. Una posible solución fue la ejecución usando *Maven*, pero al final fue descartada debido a que metía más niveles de complejidad al proyecto.

Tras ello, se procedió a realizar una instalación limpia en Ubuntu. Esto se planteaba más sencillo debido a que había más documentación. Se realizaron 3 intentos con 3 máquinas diferentes y siguiendo 3 manuales diferentes. A pesar de ello, no se consiguió que funcionara. Actualmente, creo que se debió a algún problema con el hardware de la máquina, o al desconocimiento del funcionamiento de los usuarios y grupos en Ubuntu, o al agotamiento acumulado tras varios intentos infructuosos. Como solución, se procedió a instalar la máquina de SpatialHadoop, la cual ya traía instalado Hadoop por defecto. Tras esta instalación, la dificultad del proyecto fue la típica para un proyecto de este calibre.

2. **Interfaz de filtrado de regiones.** Al comienzo del proyecto se estaba realizando el proceso ETL. Una parte de este proceso era la transformación de los datos que se habían descargados. Tras realizar la transformación el tamaño de los datos seguía siendo aún muy grande, por lo tanto, para mejorar la velocidad de procesamiento se realizó un programa con una interfaz que leyera todos los archivos .csv y generara uno por cada región de interés. La interfaz era más compleja que la actual ya que leía campos individuales para formar las regiones. De esta manera se obtendrían ficheros con los datos que realmente se iban a usar, ocupaban menos y el proceso sería más rápido.

El problema llegó cuando se desarrolló el Map, la tarea que debería realizar el Map ya la hacía el otro programa por lo tanto parecía que el Map no era necesario. Obviamente no era así. El filtrado que se realizaba con el programa creado se ejecutaba en monohilo, por lo tanto, su tiempo de ejecución era muy elevado. En cambio, la función del Map se podía ejecutar con varios hilos e incluso varias máquinas diferentes, haciendo este proceso infinitamente más rápido. La solución no fue muy complicada, ya que solo fue necesario copiar el código del programa al Map y simplificar la interfaz para que lanzara el proceso MapReduce en vez del filtrado.

3. **Funcionamiento del código MapReduce.** Una de las tareas que se ha llevado algo más de tiempo ha sido comprender cómo funcionaba el código del proceso MapReduce. Teóricamente se conocía cómo funcionaba, cuál era la tarea de cada uno y su hilo de ejecución, pero al plasmar el código de Java aparecieron muchos más conceptos que se desconocían y han necesitado un tiempo de estudio para comprenderlos.

## 5.4 Trabajos futuros

- **Ejecución del proyecto usando HPC (*High Performance Computing*).** Un trabajo futuro muy interesante sería la ejecución de este proyecto en un supercomputador (más potencia), o bien usar la estructura de Hadoop para distribuirlo en varios computadores (más nodos). Al contar con muchos más recursos, sería posible realizar el análisis de los datos de varios años. Esto sin duda sería mucho más rápido, y mejoraría el ajuste lineal de Weka (al contar con más datos para aprender), obteniendo así mejores resultados.
- **Efectuar un análisis completo con todos los datos de QuikSCAT, RapidSCAT y ASCAT disponibles para una mayor amplitud y veracidad del análisis.** Hay que recordar que los satélites no cubren todos los puntos del planeta todos los días, por ello, sería muy interesante juntar los datos de diferentes satélites para que se puedan complementar. Las órbitas serían diferentes en número y amplitud, por lo cual los agujeros sin datos diarios se podrían complementar.
- **Un estudio más afinado del algoritmo de serie temporal utilizado,** tanto en el modelo de regresión como en los parámetros aplicados. Esto queda fuera del alcance de este proyecto porque necesita de la participación de personal experto en este tipo de datos y análisis.

## 5. Anexos

En este apartado se incluyen dos anexos. El primero es el manual de usuario donde se muestra cómo usar todo el proyecto desde el principio hasta el final. El segundo es el manual del programador donde se explica cómo instalar y configurar todo.

### Anexo 1. Manual de usuario

#### 1.1 Introducción

Este manual contiene todos los pasos que debe realizar un usuario para ejecutar el proyecto completo. En primer lugar, se empezará explicando cómo descargar y transformar los datos. Tras ello, se explicará cómo ejecutar el procedimiento MapReduce y, por último, cómo visualizar el resultado de los mismos.

#### 1.2 Ejecución ETL

En este apartado se explica cómo descargar los datos del repositorio FTP de la NASA y cómo transformarlos a un formato más cómodo y ligero.

##### 1.2.1 Extracción de los datos con FileZilla

Para la descarga de los datos del repositorio FTP se usará la herramienta FileZilla. Una vez ejecutada se podrán introducir los siguientes datos: dirección del servidor, usuario, contraseña y puerto.

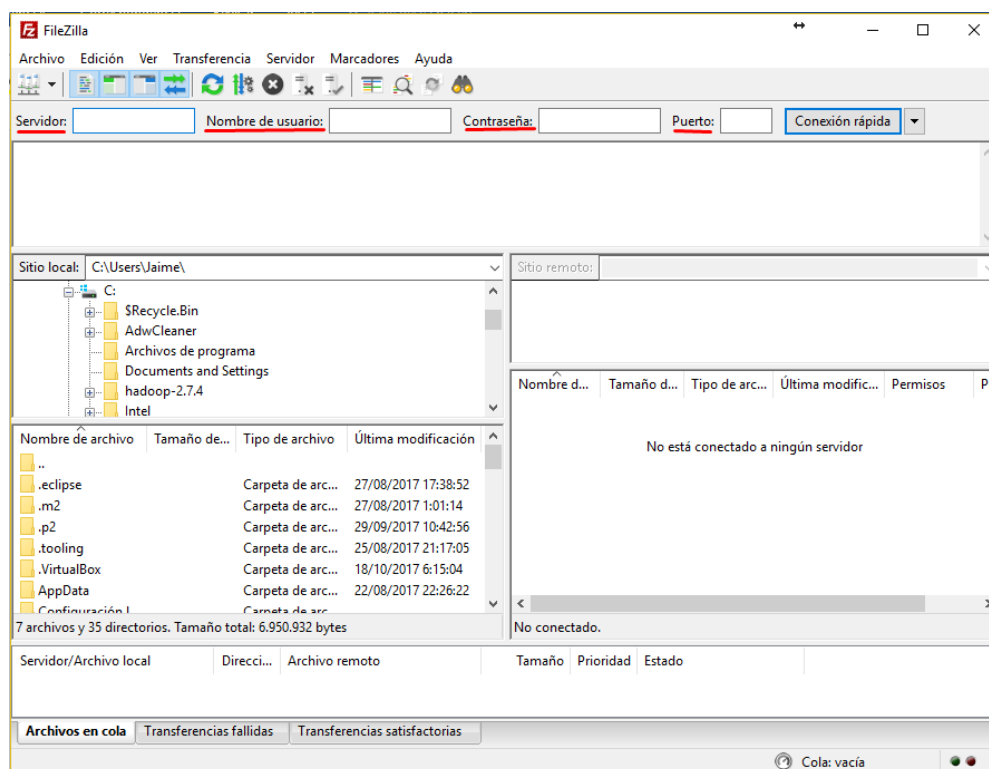


Figura 21 Pantalla de FileZilla una vez ejecutada

El único dato necesario para realizar esta descarga será la dirección del servidor donde se encuentran los datos. En este caso se ha usado el satélite RapidSCAT y se han descargado los datos del año 2015. Por lo tanto, se accede a la dirección del servidor, se puede acceder al directorio general o bien al subdirectorio con los datos de interés directamente (<ftp://podaac-ftp.jpl.nasa.gov/allData/RapidSCAT/L2B12/v1.1/2015/>).

Tras seleccionar el directorio donde se van a descargar los datos, se inicia la descarga tal y como se muestra en la Figura 22. Se hace clic con el botón derecho en las carpetas que representan los años, y seguidamente en “Descargar”.

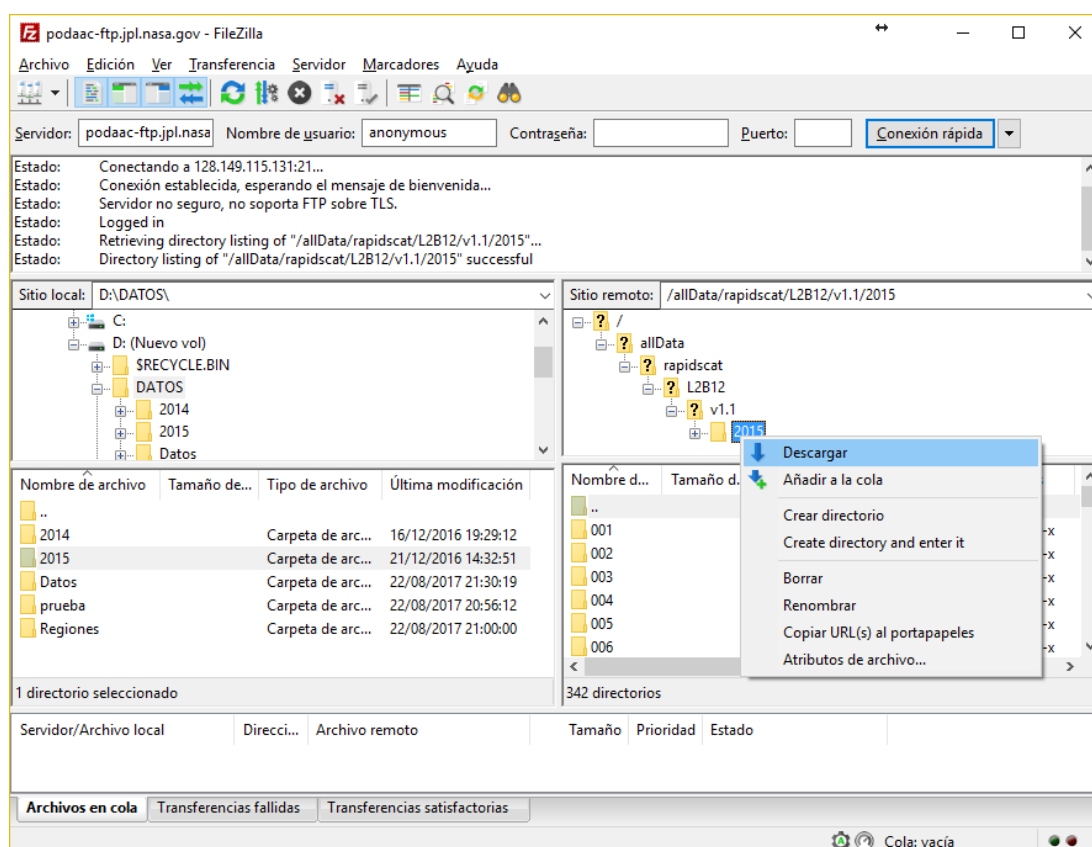


Figura 22 Inicialización de la descarga de los datos

### 1.2.2 Transformación

Una vez descargados los datos se va a proceder a la transformación de los mismos. Esta transformación consiste en convertirlos a archivos .csv. Originalmente los datos de un día están divididos en varios archivos, como se puede observar en la Figura 23.

Este equipo > Nuevo vol (D:) > DATOS > 001				
Nombre	Fecha de modifica...	Tipo	Tamaño	
rs_l2b_v1.1_01546_201504160922.nc	18/10/2017 22:30	WinRAR archive	16.292 KB	
rs_l2b_v1.1_01546_201504160922.nc.gz.md5	18/10/2017 22:28	Archivo MD5	1 KB	
rs_l2b_v1.1_01547_201504281944.nc	18/10/2017 22:29	WinRAR archive	15.086 KB	
rs_l2b_v1.1_01547_201504281944.nc.gz.md5	18/10/2017 22:27	Archivo MD5	1 KB	
rs_l2b_v1.1_01548_201504281939.nc	18/10/2017 22:28	WinRAR archive	12.815 KB	
rs_l2b_v1.1_01548_201504281939.nc.gz.md5	18/10/2017 22:26	Archivo MD5	1 KB	
rs_l2b_v1.1_01549_201504281940.nc	18/10/2017 22:27	WinRAR archive	13.481 KB	
rs_l2b_v1.1_01549_201504281940.nc.gz.md5	18/10/2017 22:25	Archivo MD5	1 KB	
rs_l2b_v1.1_01550_201504160924.nc	18/10/2017 22:26	WinRAR archive	13.633 KB	
rs_l2b_v1.1_01550_201504160924.nc.gz.md5	18/10/2017 22:24	Archivo MD5	1 KB	
rs_l2b_v1.1_01551_201504160924.nc	18/10/2017 22:25	WinRAR archive	11.078 KB	
rs_l2b_v1.1_01551_201504160924.nc.gz.md5	18/10/2017 22:23	Archivo MD5	1 KB	
rs_l2b_v1.1_01552_201504160925.nc	18/10/2017 22:23	WinRAR archive	12.059 KB	
rs_l2b_v1.1_01552_201504160925.nc.gz.md5	18/10/2017 22:22	Archivo MD5	1 KB	
rs_l2b_v1.1_01553_201504160927.nc	18/10/2017 22:24	WinRAR archive	15.516 KB	
rs_l2b_v1.1_01553_201504160927.nc.gz.md5	18/10/2017 22:22	Archivo MD5	1 KB	
rs_l2b_v1.1_01554_201504281942.nc	18/10/2017 22:22	WinRAR archive	16.106 KB	
rs_l2b_v1.1_01554_201504281942.nc.gz.md5	18/10/2017 22:21	Archivo MD5	1 KB	
rs_l2b_v1.1_01555_201504281941.nc	18/10/2017 22:22	WinRAR archive	14.127 KB	
rs_l2b_v1.1_01555_201504281941.nc.gz.md5	18/10/2017 22:20	Archivo MD5	1 KB	
rs_l2b_v1.1_01556_201504281941.nc	18/10/2017 22:21	WinRAR archive	13.561 KB	
rs_l2b_v1.1_01556_201504281941.nc.gz.md5	18/10/2017 22:19	Archivo MD5	1 KB	
rs_l2b_v1.1_01557_201504160929.nc	18/10/2017 22:20	WinRAR archive	16.175 KB	
rs_l2b_v1.1_01557_201504160929.nc.gz.md5	18/10/2017 22:19	Archivo MD5	1 KB	
rs_l2b_v1.1_01558_201504281944.nc	18/10/2017 22:19	WinRAR archive	16.090 KB	
rs_l2b_v1.1_01558_201504281944.nc.gz.md5	18/10/2017 22:18	Archivo MD5	1 KB	
rs_l2b_v1.1_01559_201504281943.nc	18/10/2017 22:19	WinRAR archive	14.460 KB	
rs_l2b_v1.1_01559_201504281943.nc.gz.md5	18/10/2017 22:17	Archivo MD5	1 KB	
rs_l2b_v1.1_01560_201504160930.nc	18/10/2017 22:18	WinRAR archive	14.726 KB	
rs_l2b_v1.1_01560_201504160930.nc.gz.md5	18/10/2017 22:16	Archivo MD5	1 KB	
rs_l2b_v1.1_01561_201504160930.nc	18/10/2017 22:17	WinRAR archive	16.315 KB	
rs_l2b_v1.1_01561_201504160930.nc.gz.md5	18/10/2017 22:16	Archivo MD5	1 KB	

**Figura 23 Archivos que contienen la información de un día**

Para la transformación de los mismos se va a usar un programa escrito en Python. Para la ejecución del mismo se accede a la carpeta donde está el programa a través de una consola. Tras ello se inserta el siguiente comando:

***python transformacion.py***

Una vez ejecutado se mostrará un menú con 4 opciones tal y como muestra la Figura 24.

```
C:\Windows\system32\cmd.exe - python transformacion.py

C:\Users\Jaime\Desktop\Programa de transformación>python transformacion.py

Bienvenido al programa de extraccion-modificacion de datos

=====
                        MENU
0.- Salir
1.- Obtener datos L2B de Quikscat
2.- Obtener datos L2B12 de Rapidscat
3.- Obtener datos L2 de Ascet

Introduzca su seleccion:
_
```

Figura 24 Menú del programa de transformación

Debido a que los datos descargados en el apartado [1.2.1](#) son del satélite RapidSCAT se introducirá un 2. En caso de tener los datos de otro satélite se podrá introducir el número correspondiente. Los siguientes pasos se realizarían de igual manera.

Tras ello, el programa pedirá la ruta donde están los ficheros descargados anteriormente. También pedirá la ruta donde pondrá la salida de los mismos. Un ejemplo de esta ejecución es el que se muestra en la Figura 23.

```
C:\Windows\system32\cmd.exe - python transformacion.py

C:\Users\Jaime\Desktop\Programa de transformación>python transformacion.py

Bienvenido al programa de extraccion-modificacion de datos

=====
                        MENU
0.- Salir
1.- Obtener datos L2B de Quikscat
2.- Obtener datos L2B12 de Rapidscat
3.- Obtener datos L2 de Ascet

Introduzca su seleccion:
2

Introduzca la ruta en la que se encuentran los directorios con ficheros gz:
D:\DATOS\2015
Introduzca la ruta en la que desea guardar los csv finales:
D:\DATOS\csvs
```

Figura 25 Ejemplo de ejecución de la transformación

Una vez termine la ejecución se pueden ver los archivos .csv resultantes en la carpeta destino.



## 1.3 MapReduce

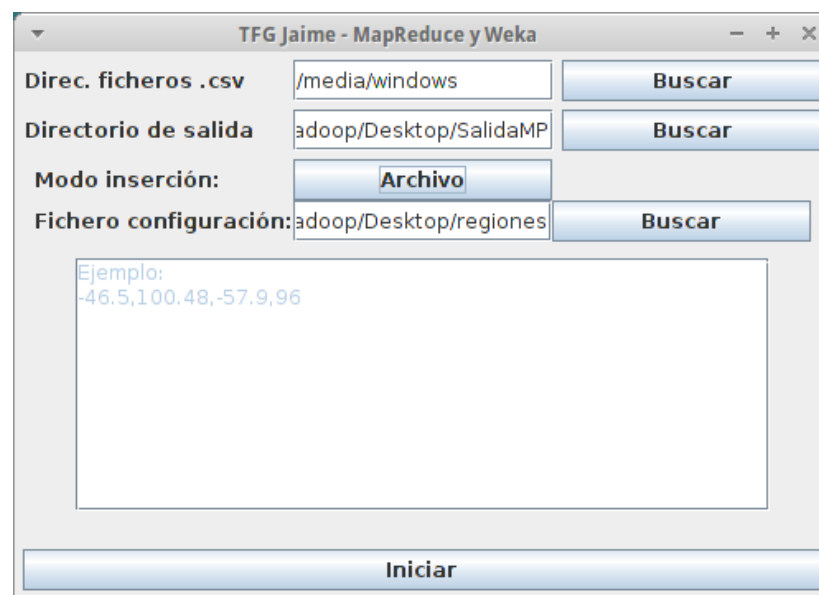
En este apartado se explicará cómo ejecutar el proceso MapReduce una vez obtenidos los archivos *.csv* correspondientes.

En primer se localiza el archivo *TFG\_Jaime.jar*. Tras ello se ejecuta desde consola a través del comando:

***java -jar TFG\_Jaime.jar***

Con este comando se ejecutará la interfaz gráfica. En ella se deberán poner las rutas donde se encuentran los archivos *.csv* y el directorio donde se depositarán los ficheros de salida.

Tras ello, habrá que añadir las regiones que se quieren analizar. Estas regiones se pueden añadir a través de un archivo de configuración, o escribirlas en la propia interfaz. Habrá que pulsar el botón del *Modo de inserción* para cambiar de una opción a la otra. Esta configuración se puede ver en la Figura 26.



**Figura 26 Configuración de la interfaz.**

Para ejecutar el proceso MapReduce solo habrá que pulsar sobre el botón *Iniciar*. Tras la ejecución en el directorio de salida aparecerán los archivos *.arff* correspondientes.

## 1.4 Weka

Se usará Weka para realizar el análisis de los datos. La configuración de Weka se realizará de igual manera que la que se ha explicado en el capítulo de Desarrollo del proyecto en el apartado [3.3.3.](#)

## Anexo 2. Manual del programador

### 2.1 Introducción

En este anexo se explican los pasos a seguir para llevar a cabo la instalación de todos los componentes del proyecto. Se empezará explicando cómo instalar FileZilla y Python, cuyo objetivo es la descarga y transformación de los datos, y se terminará hablando de cómo instalar Hadoop y Weka para llevar a cabo el procesamiento y análisis de los mismos.

### 2.2 Instalación de FileZilla

La primera herramienta usada en este proyecto es FileZilla, ya que con ella se podrán descargar los datos desde el servidor FTP de la NASA.

La descarga de esta herramienta se puede realizar desde la página oficial de FileZilla en <https://FileZilla-project.org/download.php>. En ella, pulsado sobre “Download FileZilla Cliente” se descarga un archivo .exe tras elegir entre la versión normal o la versión pro. En este caso, con la versión normal es más que suficiente.

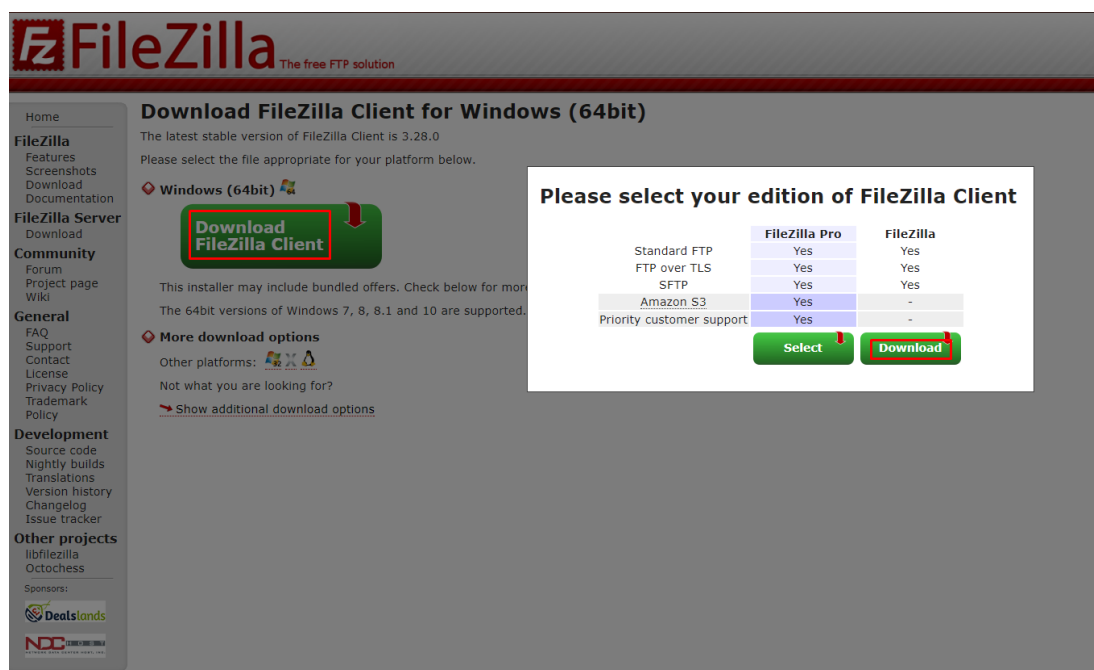


Figura 27 Descarga de la herramienta FileZilla

La instalación del ejecutable se realiza siguiendo las indicaciones del mismo sin ninguna configuración extra.

## 2.3 Instalación de Python

La instalación de Python se realizará a través de Anaconda. Si se instala Python, se tendrían que descargar muchas librerías que son necesarias para usar grandes volúmenes de datos, como se trabaja en este proyecto. Debido a esto, se utiliza Anaconda, que se caracteriza por ser libre, instala varias versiones, y lo mejor de todo, instala todos estos paquetes para trabajar con grandes volúmenes de datos, cálculos científicos y análisis de predicciones.

Para su descarga se accede a la página oficial de Anaconda, y desde allí se descarga el ejecutable. Una vez se tenga, se ejecuta y se siguen los pasos que va marcando la guía de instalación. Durante la instalación es importante marcar la opción de creación de variable en el *Path*, que aparece desmarcada por defecto, para que funcionen los siguientes pasos.

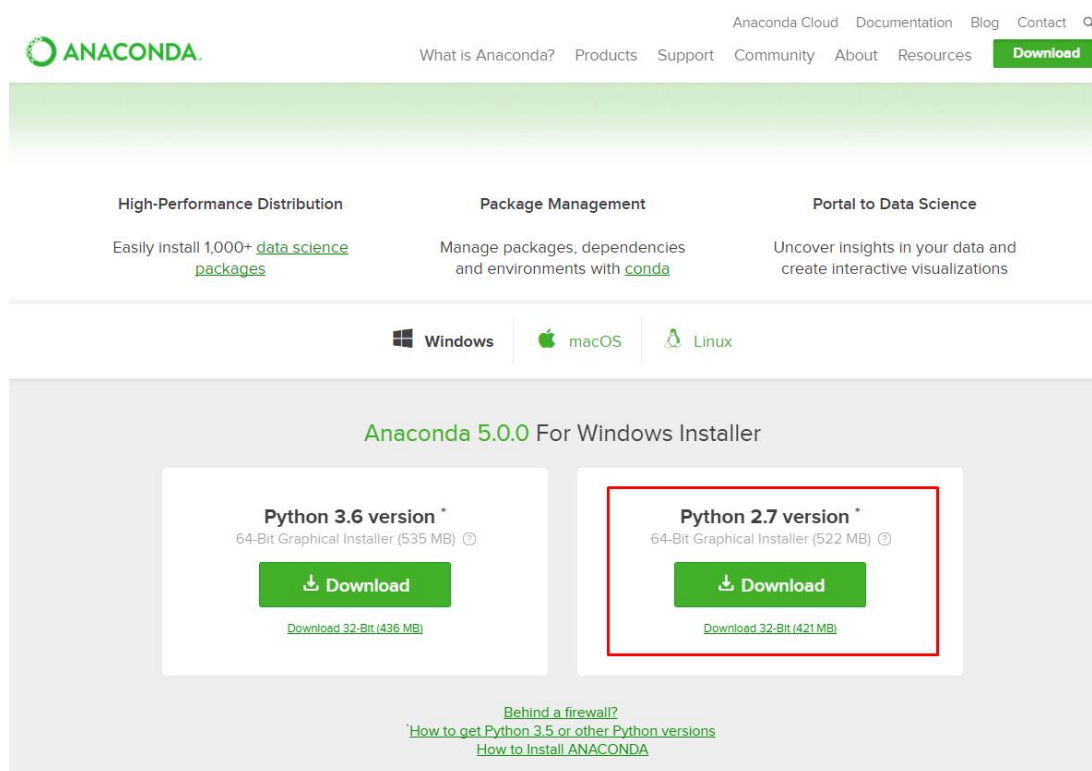


Figura 28 Descarga de la herramienta Anaconda

Tras su instalación, lo primero que se hace es instalar NumPy, que es el paquete fundamental para la computación científica con Python. Para ello, se ejecuta el siguiente comando:

***conda install numpy***

```
C:\Users\Jaime>conda install numpy
Fetching package metadata .....
Solving package specifications: .

# All requested packages already installed.
# packages in environment at C:\ProgramData\Anaconda2:
#
numpy                1.13.1                py27h0f1b411_2
```

**Figura 29 Comando de la instalación de numpy**

Para la instalación de paquetes, se recomienda tener instalado Pip, ya que es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes software de Python. Para esta instalación se necesitan hacer una serie de descargas:

1. Descargar “ez\_setup.py” desde el sitio web [https://pypi.python.org/pypi/ez\\_setup](https://pypi.python.org/pypi/ez_setup). Tras ello, ejecutar el siguiente comando:

*python ez\_setup.py*

2. Descargar “get-pip.py”. Se accede a la dirección <https://bootstrap.pypa.io/get-pip.py>, que muestra directamente este programa realizado en Python. Se guarda este fichero pulsando con el botón derecho y en la opción “Guardar como...” en el directorio que se prefiera. A continuación, se procede a la instalación del mismo igual que el punto anterior:

*python get-pip.py*

3. Actualizar el archivo “setuptools” para tener la versión más actual:

*python -m pip install -U pip setuptools*

4. Descargar Pydhf desde la dirección <http://www.lfd.uci.edu/~gohlke/pythonlibs/>, que es una interfaz de Python para los ficheros HDF4, y ejecutar el comando para su instalación:

```
C:\Users\Jaime\Downloads>pip install python_hdf4-0.9-cp27-cp27m-win_amd64.whl
Processing c:\users\jaime\downloads\python_hdf4-0.9-cp27-cp27m-win_amd64.whl
Installing collected packages: python-hdf4
Successfully installed python-hdf4-0.9
```

**Figura 30 Comando de la instalación de Pydhf**

5. Por último, se instala la librería netCDF4 que se necesita para la ejecución del fichero “transformación.py”. Esta librería se puede descargar desde el sitio web <http://www.lfd.uci.edu/~gohlke/pythonlibs/> :

```
C:\Users\Jaime\Downloads>pip install netCDF4-1.3.0-cp27-cp27m-win_amd64.whl
Processing c:\users\jaime\downloads\netcdf4-1.3.0-cp27-cp27m-win_amd64.whl
Requirement already satisfied: numpy>=1.7 in c:\programdata\anaconda2\lib\site-packages (from netCDF4==1.3.0)
Installing collected packages: netCDF4
Successfully installed netCDF4-1.3.0
```

Figura 31 Comando de la instalación de netCDF4

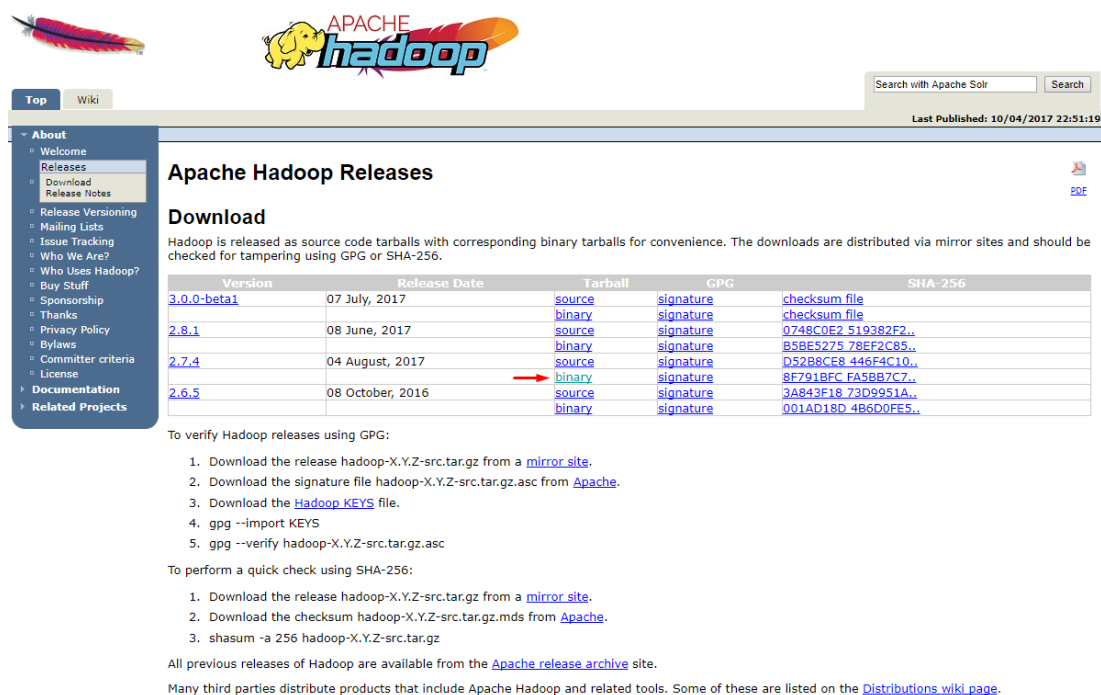
## 2.4 Instalación de Hadoop

Una parte fundamental de este proyecto, es la utilización de Hadoop para realizar el procesamiento de los datos a través del procedimiento MapReduce. Aunque la versión final de este proyecto se encuentra realizada en la máquina de SpatialHadoop, también se ha llevado a cabo la instalación en una máquina Windows. Esto último es bastante más complicado debido a la falta de documentación actualizada, y a la cantidad de pasos que hay que realizar.

### 2.4.1 Instalación en Windows

Para instalar Hadoop en Windows hay que seguir una serie de pasos explicados a continuación:

1. Primero hay descargar un archivo .tgz con el contenido del mismo desde su página oficial <http://hadoop.apache.org/releases.html>. La versión que se ha descargado en este caso es la 2.7.4 debido a fecha de actualización.



**Apache Hadoop Releases**

**Download**

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-256.

Version	Release Date	Tarball	GPG	SHA-256
<a href="#">3.0.0-beta1</a>	07 July, 2017	<a href="#">source</a>	<a href="#">signature</a>	<a href="#">checksum file</a>
<a href="#">2.8.1</a>	08 June, 2017	<a href="#">binary</a>	<a href="#">signature</a>	<a href="#">checksum file</a>
<a href="#">2.7.4</a>	04 August, 2017	<a href="#">source</a>	<a href="#">signature</a>	<a href="#">0748C0E2 519382F2...</a>
<a href="#">2.6.5</a>	08 October, 2016	<a href="#">binary</a>	<a href="#">signature</a>	<a href="#">B5B5275 78EF2C85...</a>
		<a href="#">source</a>	<a href="#">signature</a>	<a href="#">D52B8CE8 446F4C10...</a>
		<a href="#">binary</a>	<a href="#">signature</a>	<a href="#">8F791BFC FAS8B7C7...</a>
		<a href="#">source</a>	<a href="#">signature</a>	<a href="#">3A843F18 73D9951A...</a>
		<a href="#">binary</a>	<a href="#">signature</a>	<a href="#">001AD18D 4B6D0FE5...</a>

To verify Hadoop releases using GPG:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a [mirror site](#).
2. Download the signature file hadoop-X.Y.Z-src.tar.gz.asc from [Apache](#).
3. Download the [Hadoop KEYS](#) file.
4. `gpg --import KEYS`
5. `gpg --verify hadoop-X.Y.Z-src.tar.gz.asc`

To perform a quick check using SHA-256:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a [mirror site](#).
2. Download the checksum hadoop-X.Y.Z-src.tar.gz.mds from [Apache](#).
3. `shasum -a 256 hadoop-X.Y.Z-src.tar.gz`

All previous releases of Hadoop are available from the [Apache release archive](#) site.

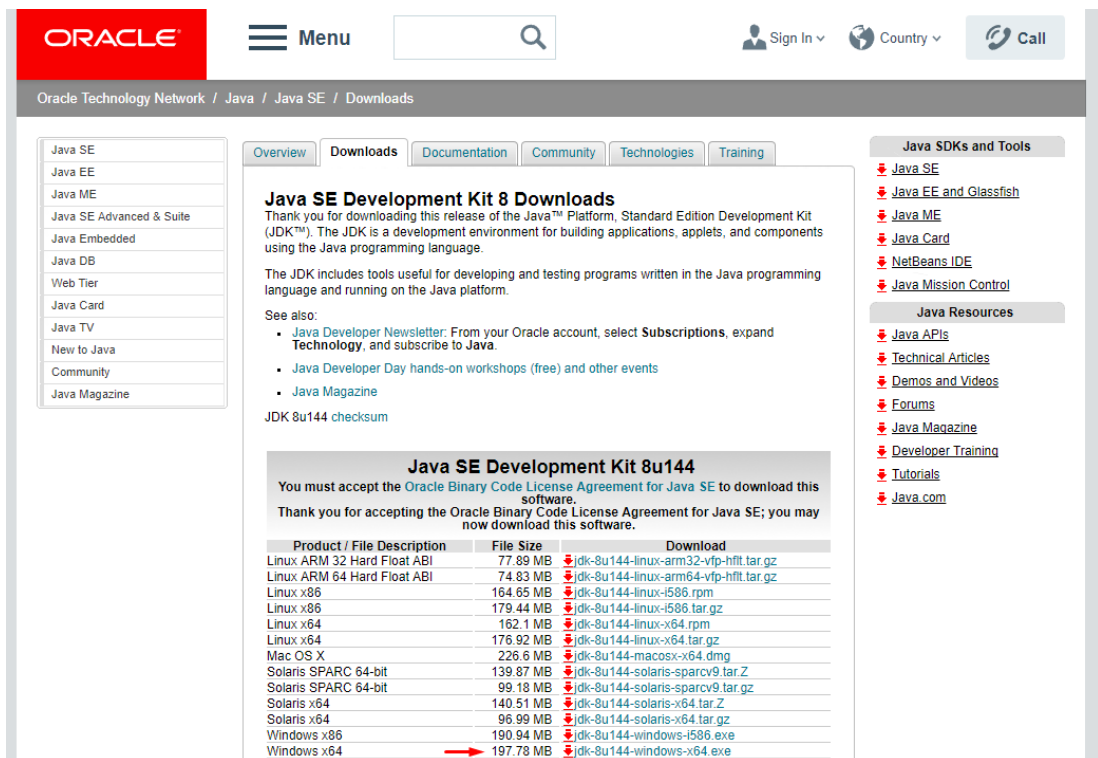
Many third parties distribute products that include Apache Hadoop and related tools. Some of these are listed on the [Distributions wiki page](#).

Figura 32 Descarga de la herramienta Hadoop

2. Tras la descarga se procede a descomprimir el archivo en la raíz.

Para seguir configurando Hadoop es necesario tener instalada una versión de JDK, en caso de tener ya una instalada se puede saltar al paso 4.

3. Instalar una versión de JDK, en este caso se ha instalado la versión 1.8.0. Para descargarla se puede realizar desde la página oficial de Oracle <http://www.oracle.com/technetwork/java/javase/downloads/>. Tras ello, se instala el ejecutable descargado y se siguen los pasos indicados en el mismo.

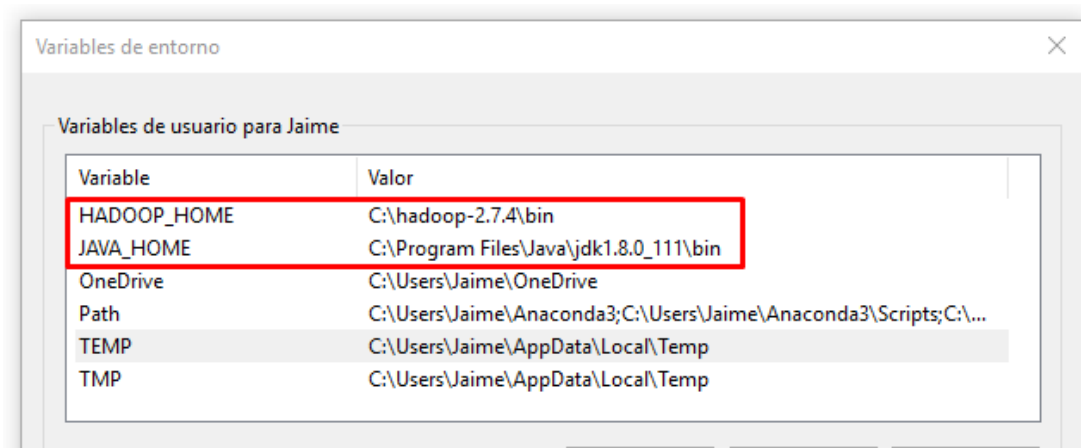


The screenshot shows the Oracle Java SE Development Kit 8 Downloads page. The page has a navigation menu on the left with links to Java SE, Java EE, Java ME, Java SE Advanced & Suite, Java Embedded, Java DB, Web Tier, Java Card, Java TV, New to Java, Community, and Java Magazine. The main content area is titled "Java SE Development Kit 8 Downloads" and includes a "Thank you" message for downloading the release. It also provides information about the JDK and links to the Java Developer Newsletter, Java Developer Day, and Java Magazine. A table lists the download links for various operating systems and architectures, with a red arrow pointing to the Windows x64 download link.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.89 MB	<a href="#">jdk-8u144-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.83 MB	<a href="#">jdk-8u144-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	164.65 MB	<a href="#">jdk-8u144-linux-i586.rpm</a>
Linux x86	179.44 MB	<a href="#">jdk-8u144-linux-i586.tar.gz</a>
Linux x64	162.1 MB	<a href="#">jdk-8u144-linux-x64.rpm</a>
Linux x64	176.92 MB	<a href="#">jdk-8u144-linux-x64.tar.gz</a>
Mac OS X	226.6 MB	<a href="#">jdk-8u144-macosx-x64.dmg</a>
Solaris SPARC 64-bit	139.87 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.18 MB	<a href="#">jdk-8u144-solaris-sparcv9.tar.gz</a>
Solaris x64	140.51 MB	<a href="#">jdk-8u144-solaris-x64.tar.Z</a>
Solaris x64	96.99 MB	<a href="#">jdk-8u144-solaris-x64.tar.gz</a>
Windows x86	190.94 MB	<a href="#">jdk-8u144-windows-i586.exe</a>
Windows x64	197.78 MB	<a href="#">jdk-8u144-windows-x64.exe</a>

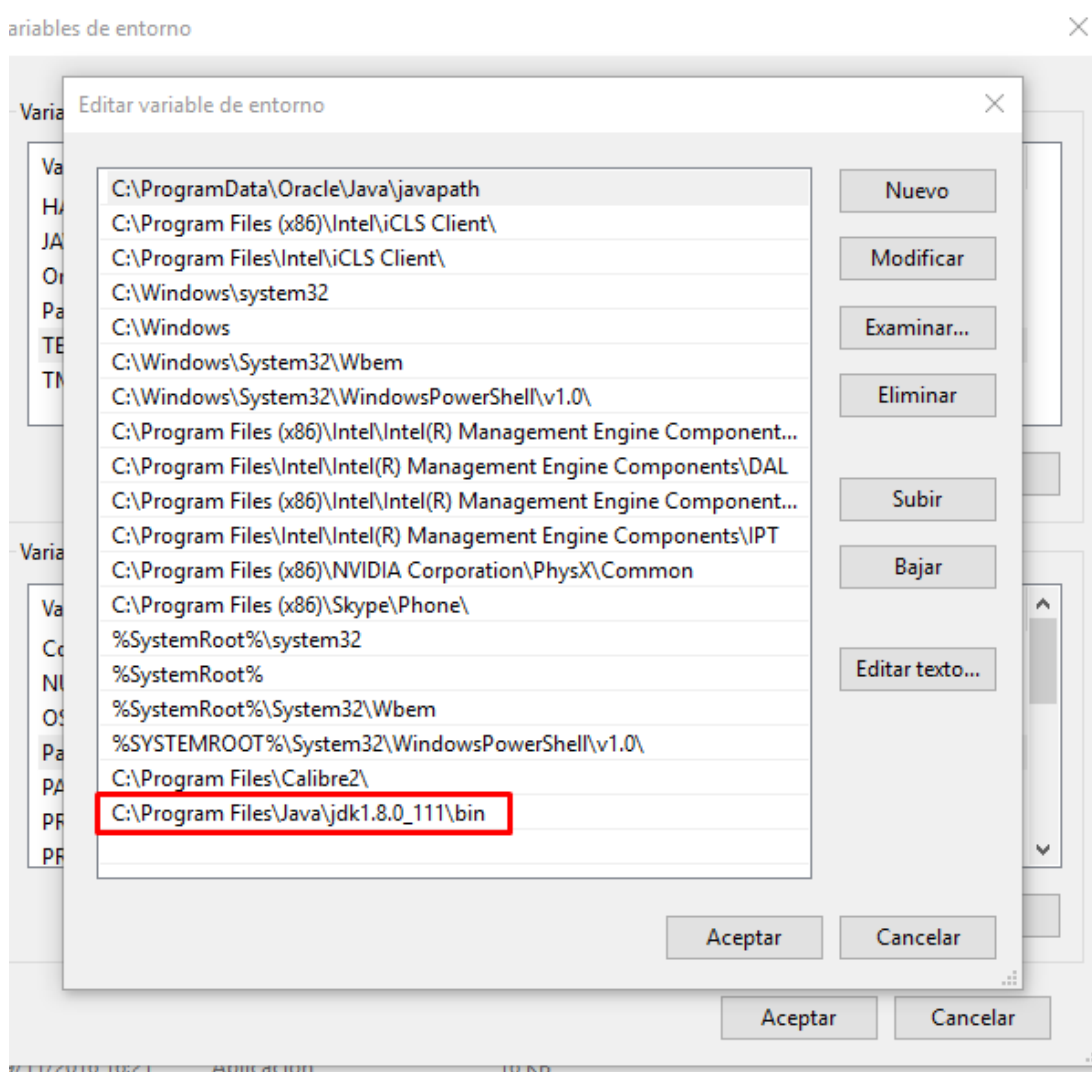
Figura 33 Descarga del JDK

4. Configuración de las variables de entorno. Para que funcione Hadoop es necesario tener 2 variables de entorno nuevas. Para añadirlas se accede a las Propiedades del sistema y una vez en él, se pulsa sobre “Variables de entorno...”. Una vez dentro se crean dos variables nuevas: HADOOP\_HOME y JAVA\_HOME, ambas con la ruta de sus respectivas carpetas *bin*.



**Figura 34 Variables de entorno añadidas**

También, es necesario editar la variable del Path para añadir la ruta de la carpeta bin donde se ha instalado el JDK.



**Figura 35 Modificación de la variable PATH**

5. Modificar los ficheros HTML. A continuación, hay que realizar una serie de modificaciones en los ficheros HTML incorporados dentro de la carpeta etc/hadoop:

- Editar `hadoop-2.7.4/etc/hadoop/core-site.xml`, y añadir:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

- Editar el fichero `hadoop-2.7.4/etc/hadoop/mapred-site.xml` y añadir:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

- Editar `hadoop-2.7.4/etc/hadoop/hdfs-site.xml` y añadir:



```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>C:\hadoop-2.7.4\data\namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>C:\hadoop-2.7.4\data\datanode</value>
  </property>
</configuration>
```

- Crear una carpeta “data” en el directorio de Hadoop:
- Editar el fichero `hadoop-2.7.4/etc/hadoop/yarn-site.xml`, añadiendo el código siguiente:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

- Editar el fichero `hadoop-2.7.4/etc/hadoop/hadoop-env.cmd`. En él, modificar la línea `"JAVA_HOME=%JAVA_HOME%"` y añadir el directorio donde está instalado el JDK. Hay que tener cuidado al poner la ruta ya que no puede contener espacios, por ello se puede usar `"~"` para completar la ruta, por ejemplo:  
Progra~1 correspondería a 'Program Files' y Progra~2 a 'Program Files(x86)'

```
@rem The java implementation to use. Required.
set JAVA_HOME="C:\Progra~1\Java\jdk1.8.0_144"
```

Figura 36 Modificación en `hadoop-env.cmd`

- Descargar el siguiente archivo del directorio de GitHub: <https://github.com/sardetushar/hadooponwindows/archive/master.zip>. Este archivo contiene algunas herramientas que no se proporcionan desde la página oficial. Se descarga y se descomprime en la ruta `C:\hadoop-2.7.4\bin` y se reemplazan los archivos duplicados.
- Abrir una consola y escribir: `hdfs namenode -format`
- Por último, se pueden lanzar todos los componentes de Hadoop desde el directorio `sbin`, en el ejecutar el archivo `"start-all.cmd"`. Se abrirán 4 consolas que lanzarán las diferentes partes.

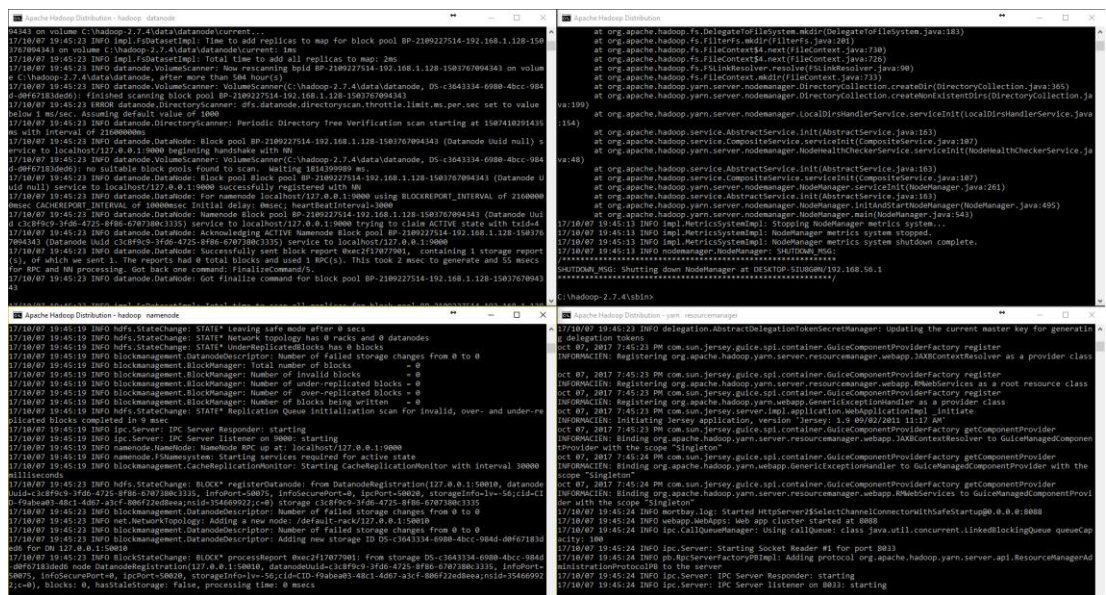


Figura 37 Ejecución de los componentes de Hadoop

- A continuación, se puede comprobar el correcto funcionamiento de Hadoop accediendo a <http://localhost:8088>. Desde esa dirección se pueden ver todos los parámetros del clúster, nodos, aplicaciones lanzadas, etc.

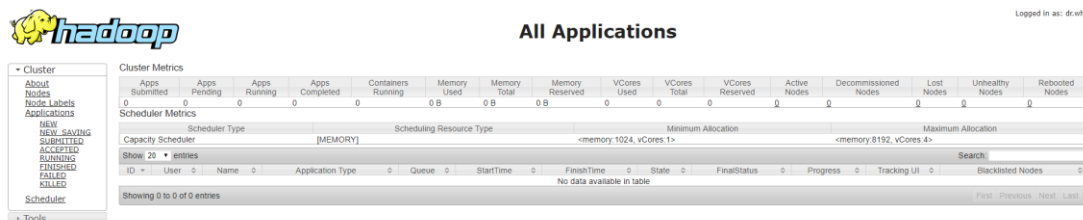


Figura 38 Interfaz de Hadoop en localhost:8088

- También, desde la dirección <http://localhost:50070>, se puede ver la información referida al namenode:

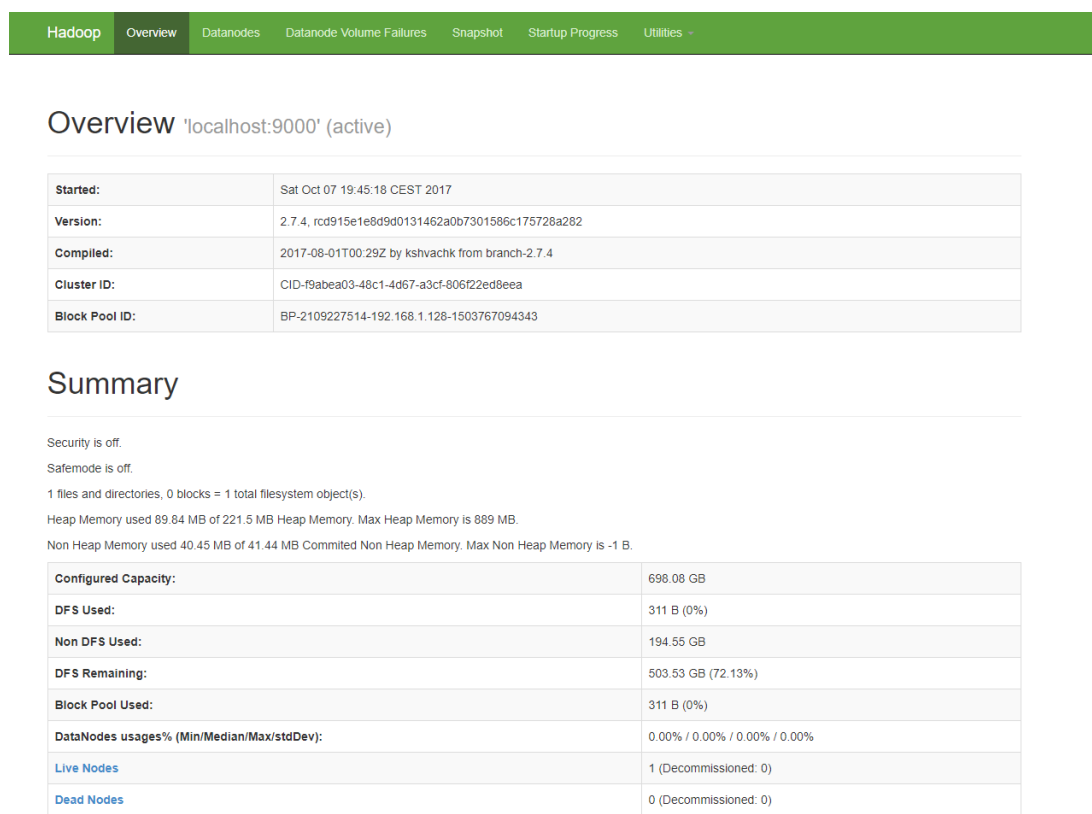


Figura 39 Interfaz de Hadoop en localhost:50070

## 2.4.2 Instalación usando SpatialHadoop

SpatialHadoop proporciona una máquina virtual en la cual viene ya instalado y configurado Hadoop, por esta razón, lo único necesario será la instalación de esta máquina virtual. Así mismo, también habrá que realizar algunas modificaciones sobre la misma para poder trabajar con comodidad desde ella, las cuales se explicarán en los apartados siguientes:

### 2.4.2.1 Instalación de la máquina virtual

Debido a que se va a trabajar con una máquina virtual, lo primero que se va a realizar es instalar VirtualBox, que permitirá realizar la instalación de la máquina de SpatialHadoop. Se puede descargar de forma sencilla accediendo a la página oficial <https://www.virtualbox.org/wiki/Downloads> y pulsando sobre “Windows hosts”. La instalación es muy sencilla y solo hay que seguir los pasos indicados por el asistente.

Para la descarga de la imagen de SpatialHadoop, habrá que acceder a su página oficial <http://spatialhadoop.cs.umn.edu>, y en ella, en la pestaña de descargas, se podrá proceder a la descarga de la misma, tal y como se muestra en la Figura 40.

The screenshot shows the SpatialHadoop website. At the top, there's a navigation bar with links: DATASETS, PUBLICATIONS, TUTORIALS, and DOWNLOAD. The DOWNLOAD link is highlighted. Below the navigation bar, there's a large banner with the text "Analyze your spatial data efficiently" and a blue elephant logo. To the right of the banner, there's a sidebar with links: SOURCE CODE, BINARY DISTRIBUTION, VIRTUAL MACHINE IMAGE (highlighted with a red box), and AMAZON EC2 IMAGE. Below the banner, there's a section with four columns: Spatial language, Spatial data types, Spatial indexes, and Spatial operations. Each column has a small image and a brief description. At the bottom, there's a section titled "SpatialHadoop Overview" with a description of the project, a "What is new" section listing updates for Version 2.4.2 and Version 2.4, a "Contact us" section with an email address and the SpatialHadoop Team members, and a "Recent Tweets" section.

**Figura 40** Descarga de SpatialHadoop

Una vez descargada la imagen, hay que importarla a VirtualBox. Para ello, se abre VirtualBox y se pulsa en el botón “Archivo” de la barra superior. Se abrirá un menú desplegable y se clicará en la opción “Importar servicio virtualizado...”. Saldrá un cuadro de diálogo, donde se tendrá que insertar la ruta donde se ha descargado la imagen de SpatialHadoop con extensión “.ova”, y realizaremos la importación de esta. Es importante darle los máximos parámetros que permita nuestra máquina, de tal manera que funcione lo mejor posible. Una vez importado, se ejecutará y estará listo para utilizarse.

#### 2.4.2.2 Ampliación de la capacidad de almacenamiento

La máquina instalada tiene una capacidad total de 8GB por defecto. Con las herramientas que ya tiene incorporadas, los ficheros con los que se va a trabajar, la instalación de Weka, etc., no se tiene espacio suficiente para poder trabajar. Por esto, se necesita expandir la capacidad de almacenamiento del disco. Para ello se deberán realizar los siguientes pasos:

1. Crear un nuevo disco de almacenamiento. Por medio del terminal, acceder al directorio donde está instalado VirtualBox, por defecto **C:\Program Files\Oracle\VirtualBox**. Tras ello, se escribe el comando que se observa en la Figura 41.

***VBoxManage createhd --format VMDK --size 20000 --filename  
"C:\Users\Jaime\VirtualBox VMs\SpatialHadoop\SH20GB.vmdk"***

```
C:\Program Files\Oracle\VirtualBox>VBoxManage createhd --format VMDK --size 20000 --filename "C:\Users\Jaime\VirtualBox VMs\SpatialHadoop\SH20gb.vmdk"
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Medium created. UUID: 3e5bdc21-dfa6-4675-9792-347875d999e9
```

**Figura 41 Comando de creación de disco duro virtual**

El disco que se ha creado ha sido de un tamaño de 20GB, como se indica con el argumento “**--size**”. Además, tendrá un formato VMDK (Virtual Machine Disk Format), indicado con el argumento “**--format**”, y con el nombre “spatialhadoop20GB”, como se indica mediante “**--filename**”. Por último, el disco será almacenado en la ruta “C:\Users\Jaime\VirtualBox VMs\SpatialHadoop”, ya que con el argumento “**--filename**”, se pone el PATH completo donde se almacenará el disco duro, incluyendo así, dirección y nombre.

2. Clonar el disco antiguo al nuevo. Este paso se realiza para que no se pierda nada de lo que se tiene ya instalado en la máquina virtual. Para hacer este paso, se ejecuta el siguiente comando:

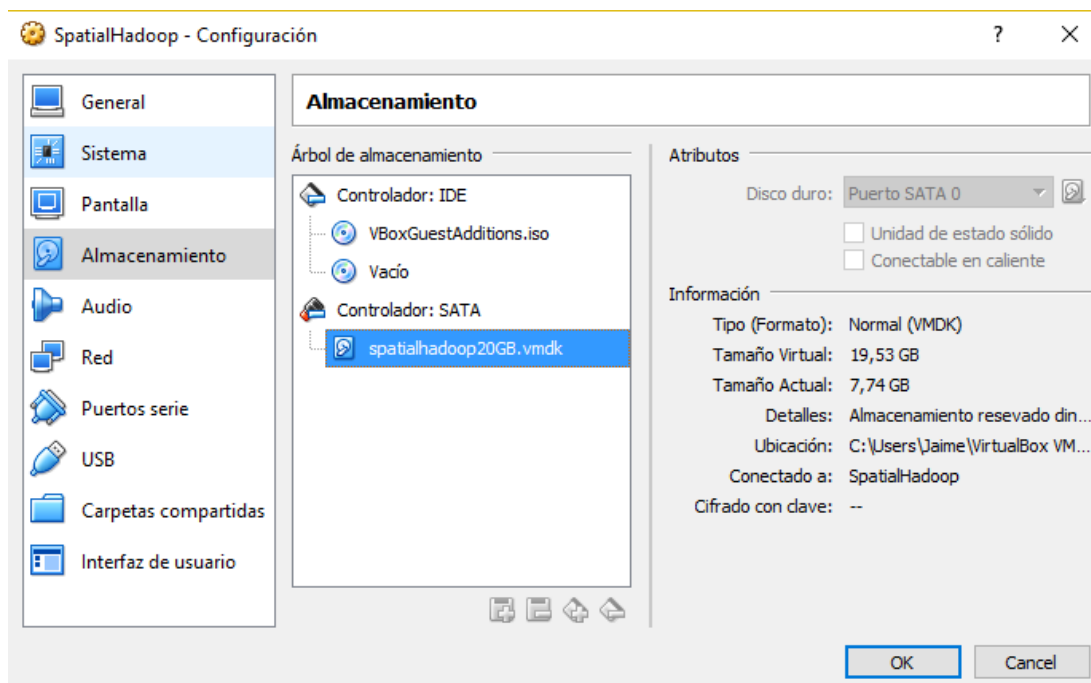
***VBoxManage clonehd "C:\Users\Jaime\VirtualBox  
VMs\SpatialHadoop\spatialhadoop-f3745a-disk1.vmdk" "C:\Users\Jaime\VirtualBox  
VMs\SpatialHadoop\SH20gb.vmdk" --existing***

```
C:\Program Files\Oracle\VirtualBox>VBoxManage clonehd "C:\Users\Jaime\VirtualBox VMs\SpatialHadoop\spatialhadoop-f3745a-disk1.vmdk" "C:\Users\Jaime\VirtualBox VMs\SpatialHadoop\SH20gb.vmdk" --existing
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Clone medium created in format 'VMDK'. UUID: 3e5bdc21-dfa6-4675-9792-347875d999e9
```

**Figura 42 Comando de clonación de disco duro virtual**

Este comando es el encargado de ejecutar la clonación (**clonehd**), y las dos direcciones representan la ubicación de ambos discos. Al final se inserta el argumento “**--existing**”, para que no de error al encontrar el disco en la ubicación y lo sobrescriba.

3. Por último, se procede a la inserción del nuevo disco. Para ello, se accede a la configuración de la máquina y en el apartado de Almacenamiento se elimina el controlador que viene por defecto (8GB) y en su lugar se pone el que se acaba de crear (20GB).



**Figura 43 Inserción del nuevo disco duro virtual**

Aunque ya está configurada con un nuevo disco duro, todavía falta organizar las particiones que existen dentro del disco. Hay que ampliar la partición de 8GB que tenía el antiguo disco y asignarle los 12GB restantes. Para ello utilizaremos la herramienta **GParted**.

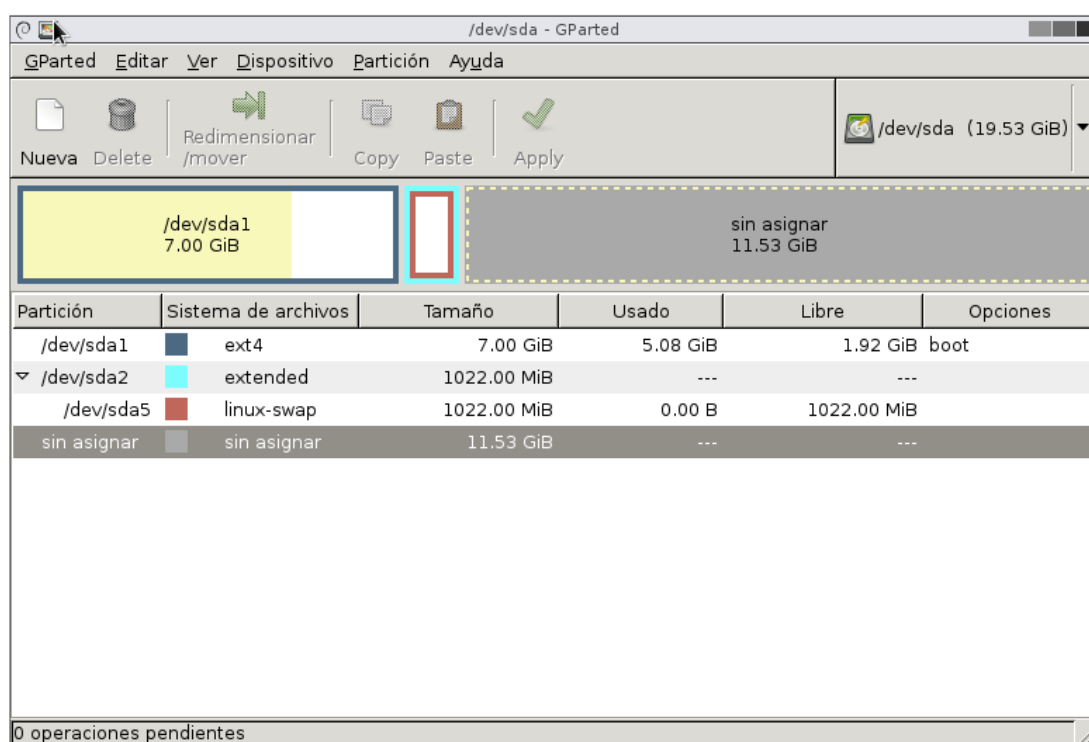
Si se instalase GParted dentro de la máquina virtual y se ejecutase, no se podría redimensionar la partición, ya que es con la que se ha arrancado el sistema y no puede ser modificada de ninguna forma. Se tiene que redimensionar la partición sin ejecutar el sistema y para eso utilizaremos una imagen ISO de GParted. La imagen será descargada de la página oficial de GParted <http://gparted.org/download.php>.

Una vez que se tiene la imagen de GParted descargada, se procede a insertar la imagen descargada. Esto se hace accediendo al apartado “Almacenamiento” de la configuración, en la parte del controlador, se pulsa en el disco vacío y se añade la *iso* descargada anteriormente.

Tras la inserción se procede a arrancar la máquina de forma normal. Tras ello, aparecerá un menú con diferentes opciones de GParted, donde se seleccionará la primera

(modo por defecto). De nuevo, se abre otro menú para configurar la máquina. Estos pasos no son importantes, puesto que lo único que interesa de esta máquina, es la herramienta GParted. En este segundo menú se pulsa en la opción “*Don’t touch keymap*”. Seguidamente, se pide el idioma con el que se quiere trabajar y se introduce el número 25, que representa al idioma español. Por último, pide un modo de ejecución, en el que introduciremos la opción 0, es decir, que se arranque GParted de forma normal.

Cuando ya está la máquina arrancada, se abre la herramienta GParted que está visible en el escritorio y se tendrá algo como se observa en la Figura 44.



**Figura 44 Particiones con la herramienta GParted**

Estas son las particiones que forman el nuevo disco duro, donde se encuentran destacando una de 7GB, que es la memoria del disco duro antiguo, y las 11,53GB que están sin asignar, que son los que se tienen que añadir a la anterior partición.

Lo primero que se ha de hacer es eliminar la segunda partición “/dev/sda2”, que no tiene función ninguna para nosotros. Para hacer esto, se selecciona la que tiene dentro de ella y se elimina. Seguidamente se hace lo mismo con esta, quedando así las dos particiones importantes que antes hemos destacado.

Por último, se pulsa en la partición de 7GB y se pulsa en redimensionar, donde se desplaza la barra hasta el final, aumentando la capacidad de 7GB a 20GB. Una vez hecho esto, se cierra la máquina virtual y se quita la imagen ISO de GParted del controlador



(apartado “Almacenamiento” de la configuración de la máquina virtual), dejando la máquina preparada para su uso con su almacenamiento de 20GB.

## 2.5 Instalación de Weka

La herramienta Weka será utilizada dentro de la máquina virtual instalada en el punto anterior. La descarga de esta herramienta se realizará desde su página oficial <https://www.cs.waikato.ac.nz/ml/Weka/downloading.html>. Tras la descarga se descomprime un *zip* en el directorio que se desee.

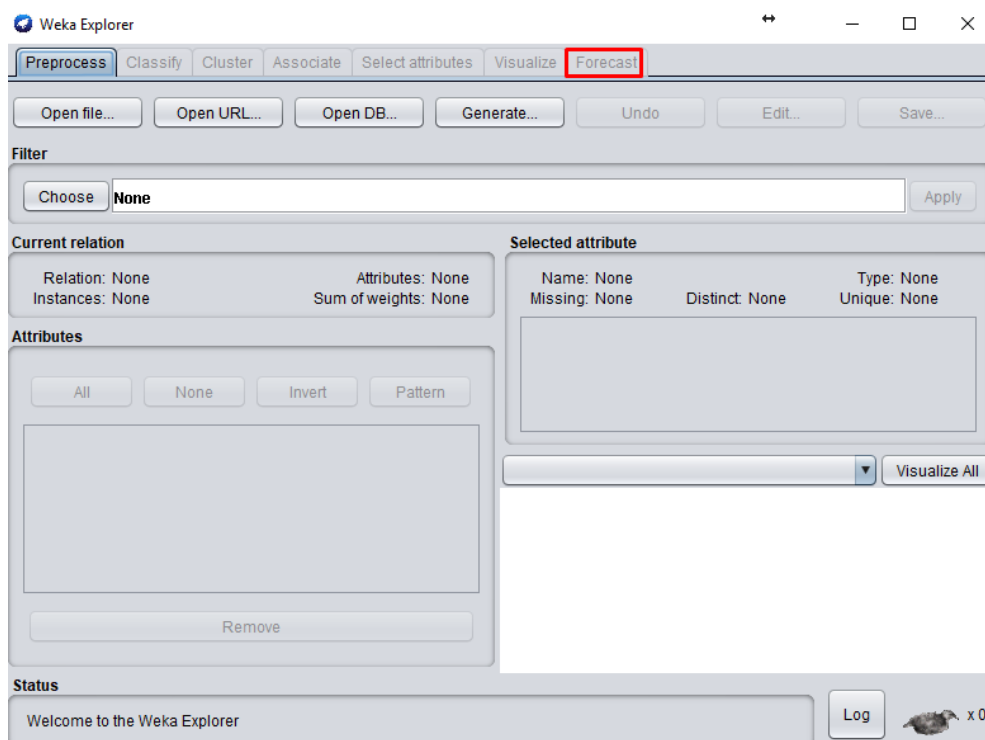
La ejecución de la Weka se realizará desde una consola. Se deberá estar en el directorio donde se encuentre instalada y habrá que poner el siguiente comando:

***java -jar Weka.jar***

Una vez Weka ha sido instalado y abierto, será necesaria la instalación del módulo de series temporales que permitirá realizar el análisis de los datos. Para ello, se abrirá el administrador de paquetes desde el menú *Tools* y luego en *Package Manager*.

Una vez abierto se desplegará una lista de paquetes varios, habrá que descender hasta encontrar el de **timeseriesForecasting**. Una vez seleccionado solo habrá que pulsar en el botón de Install para dar paso a la instalación del mismo.

Una vez instalado, en el explorador de Weka deberá aparecer una nueva pestaña denominada Forecast, tal y como se muestra en la Figura 45.



**Figura 45** Weka con el paquete TimeseriesForecasting instalado



## 2.6 Instalación WindowBuilder (Eclipse)

WindowBuilder permitirá la creación de una interfaz gráfica de manera más sencilla. Para su instalación se usará el instalador de software incluido en Eclipse. Para abrirlo se pulsará sobre Help > Install new Software...

Una vez dentro se tendrá que poner la dirección donde se sitúa el paquete, esta dirección varía según la versión de eclipse instalada, desde la página de eclipse se puede saber las diferentes versiones (<https://www.eclipse.org/windowbuilder/download.php>). Una vez conocida se pega en “Work with” y se instala.

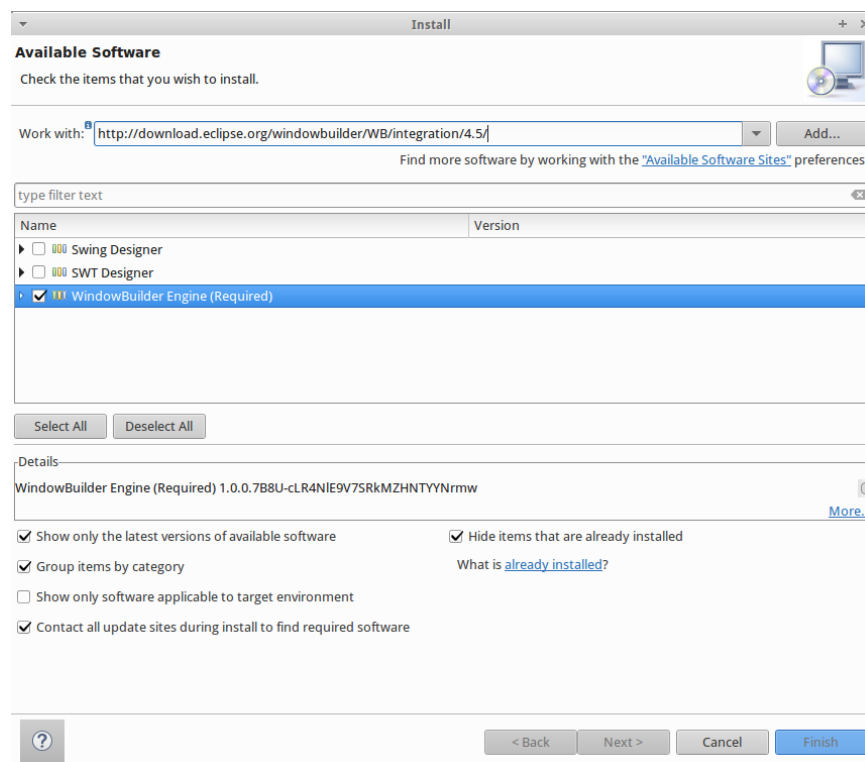


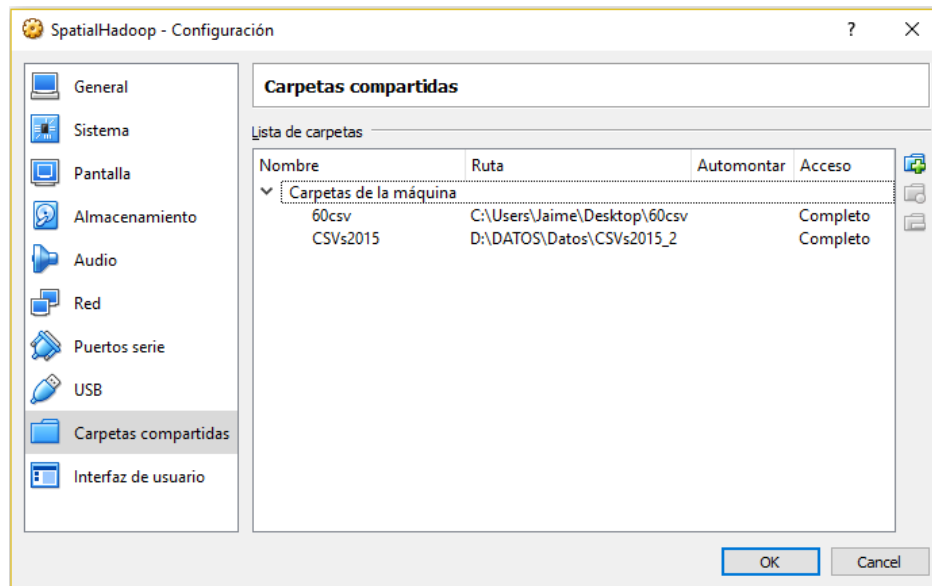
Figura 46 Instalación de WindowBuilder

## 2.7 Compartición de datos entre sistemas

Es importante destacar el tamaño de los datos que se van a analizar. En el punto 2.4.2 se ha ampliado el tamaño del disco de la máquina virtual a 20GB, pero es muy posible que los datos ocupen bastante más (un solo año de datos ocupa en torno a los 76GB). Para ello es importante la compartición de disco entre la máquina virtual y el sistema operativo base.

Para realizar esto se va a usar el sistema de carpetas compartidas que trae VirtualBox incluido. Para acceder a él se entrará en las propiedades de la máquina, y luego en el submenú de Carpetas compartidas. Una vez allí, se añadirá una nueva carpeta pulsando sobre el botón de la carpeta, una vez abierto se abrirá una ventana para establecer la ruta

y el nombre que se le va dar. Este nombre es importante ya que se usará en los pasos posteriores. Una vez completado se obtendrá algo como lo que se muestra en la Figura 47



**Figura 47 Configuración de las carpetas compartidas**

El siguiente paso se realizará desde la máquina de SpatialHadoop. En primer lugar, se creará un directorio donde aparecerán los archivos de la carpeta compartida. Se puede crear con el siguiente comando:

```
sudo mkdir /media/Windows
```

Lo siguiente que se va a realizar es montar la carpeta dentro de la máquina. Para ello se usará el comando:

```
sudo mount -t vboxsf Nombre_carpeta Ruta
```

Es importante destacar el que Nombre\_carpeta hace referencia al nombre que se le asignó a la carpeta al compartirla. Un ejemplo de este comando sería:

```
sudo mount -t vboxsf 60csv /media/Windows
```

Si se quiere que esta carpeta se auto monte cada vez que se inicia la máquina virtual hay que editar el archivo /etc/init.d/rc.local y añadir el comando anterior.

## **7. Bibliografía**

- [1] D. Teomiro Villa, «Integración de datos de vientos marinos en Oracle NoSQL,» Trabajo Fin de Grado del Grado en Ingeniería Informática en Ingeniería del Software de la Universidad de Extremadura, Director Félix R. Rodríguez, Septiembre, 2015.
- [2] C. Arroyo Fernández, «Técnicas Big Data con datos geoespaciales y hadoop,» Trabajo Fin de Grado del Grado en Ingeniería Informática en Ingeniería del Software de la Universidad de Extremadura, Octubre, 2016.
- [3] «Winds Measuring Ocean Winds from Space,» [En línea]. Available: <https://winds.jpl.nasa.gov/missions/seawinds/>. [Último acceso: Octubre 2017].
- [4] «National Aeronautics and Space Administration,» [En línea]. Available: <https://www.nasa.gov/>. [Último acceso: Octubre 2017].
- [5] «National Oceanic and Atmospheric Administration,» [En línea]. Available: <http://www.noaa.gov/>. [Último acceso: Octubre 2017].
- [6] «Winds Measuring Ocean Winds from Space,» [En línea]. Available: <https://winds.jpl.nasa.gov/missions/nscat/>. [Último acceso: Octubre 2017].
- [7] «Wave Life Cycle I: Generation,» [En línea]. Available: [http://wegc203116.uni-graz.at/metted/marine/mod2\\_wlc\\_gen/print.htm](http://wegc203116.uni-graz.at/metted/marine/mod2_wlc_gen/print.htm).
- [8] «Winds Measuring Ocean Winds from Space,» [En línea]. Available: <https://winds.jpl.nasa.gov/missions/quikscat/>. [Último acceso: Octubre 2017].
- [9] «The HDF group,» [En línea]. Available: <https://www.hdfgroup.org/>.
- [10] «What is NetCDF?,» [En línea]. Available: <http://www.unidata.ucar.edu/software/netcdf/docs/>. [Último acceso: Octubre 2017].
- [11] «NASA's ISS-RapidScat Earth Science Mission Ends,» [En línea]. Available: <https://www.jpl.nasa.gov/news/news.php?feature=6683>. [Último acceso: Octubre 2017].
- [12] «Winds Measuring Ocean Winds from Space,» [En línea]. Available: <https://winds.jpl.nasa.gov/missions/RapidScat/>. [Último acceso: Octubre 2017].

- [13] «ISS-RapidScat - Mission Specification,» [En línea]. Available: <https://podaac.jpl.nasa.gov/ISS-RapidScat>. [Último acceso: Octubre 2017].
- [14] «Big Data Analytics,» [En línea]. Available: <https://www.ibm.com/analytics/us/en/big-data/>. [Último acceso: Octubre 2017].
- [15] «¿Qué es Big Data? - ibm,» [En línea]. Available: <https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/index.html>. [Último acceso: Octubre 2017].
- [16] «Procesos ETL: Definición, Características, Beneficios y Retos,» [En línea]. Available: <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/312584/procesos-etl-definici-n-caracter-sticas-beneficios-y-retos>. [Último acceso: Octubre 2017].
- [17] «Start-Up Goes After Big Data With Hadoop Helper,» [En línea]. Available: <http://bits.blogs.nytimes.com/2010/04/22/start-up-goes-after-big-data-with-hadoop-helper/?dbk>. [Último acceso: Octubre 2017].
- [18] «What Is Apache Hadoop?,» [En línea]. Available: <http://hadoop.apache.org/#What+Is+Apache+Hadoop%3F>. [Último acceso: Octubre 2017].
- [19] «¿Qué es MapReduce? - solidq,» [En línea]. Available: <http://blogs.solidq.com/es/big-data/que-es-mapreduce>. [Último acceso: Octubre 2017].
- [20] L. Molina, «Data mining no processo de extração de conhecimento de bases de dados,» Tesis de máster. Instituto de Ciências Matemáticas e Computação. Universidad de São Paulo., Brasil, 1998.
- [21] «Data mining: torturando a los datos hasta que confiesen,» [En línea]. Available: <http://www.uoc.edu/web/esp/art/uoc/molina1102/molina1102.html>. [Último acceso: Octubre 2017].
- [22] «spatialhadoop,» [En línea]. Available: <http://spatialhadoop.cs.umn.edu/>. [Último acceso: Octubre 2017].
- [23] «Numpy,» [En línea]. Available: <http://www.numpy.org/>. [Último acceso: Octubre 2017].

- [24] «Attribute-Relation File Format,» [En línea]. Available: <https://www.cs.waikato.ac.nz/ml/weka/arff.html>. [Último acceso: Octubre 2017].
- [25] «FileZilla Features,» [En línea]. Available: [https://filezilla-project.org/client\\_features.php](https://filezilla-project.org/client_features.php). [Último acceso: Octubre 2017].
- [26] T. White, Hadoop. The Definitive Guide 4ª Edición, O'Reilly.
- [27] «Youtube,» WekaMOOC, 24 Abril 2016. [En línea]. Available: [https://www.youtube.com/watch?v=NDwn7G8zTOU&list=PLm4W7\\_iX\\_v4Msh-7lD0pSFWHRYU\\_6H5Kx&index=5](https://www.youtube.com/watch?v=NDwn7G8zTOU&list=PLm4W7_iX_v4Msh-7lD0pSFWHRYU_6H5Kx&index=5). [Último acceso: Octubre 2017].
- [28] «Youtube,» WekaMOOC, 24 Abril 2016. [En línea]. Available: [https://www.youtube.com/watch?v=NLTLUmt77-E&list=PLm4W7\\_iX\\_v4Msh-7lD0pSFWHRYU\\_6H5Kx&index=4](https://www.youtube.com/watch?v=NLTLUmt77-E&list=PLm4W7_iX_v4Msh-7lD0pSFWHRYU_6H5Kx&index=4). [Último acceso: Octubre 2017].
- [29] «Cómo compartir carpetas entre Windows y Ubuntu en VirtualBox,» [En línea]. Available: <https://blog.desdelinux.net/como-compartir-carpetas-entre-windows-y-ubuntu-en-virtualbox-ose/>. [Último acceso: Octubre 2017].
- [30] «Apache Hadoop: Single Node Setup,» [En línea]. Available: [http://hadoop.apache.org/docs/r1.2.1/single\\_node\\_setup.html](http://hadoop.apache.org/docs/r1.2.1/single_node_setup.html). [Último acceso: Octubre 2017].
- [31] «Data Mining: Practical Machine Learning Tools and Techniques,» [En línea]. Available: <https://www.cs.waikato.ac.nz/ml/weka/book.html>. [Último acceso: Octubre 2017].
- [32] «MetOp-A ASCAT Level 2 25.0 km Ocean Surface Wind Vectors,» [En línea]. Available: <https://podaac.jpl.nasa.gov/dataset/ASCATA-L2-25km?ids=Sensor:Platform&values=ASCAT:MetOp-A>. [Último acceso: Octubre 2017].
- [33] «ISS – RapidSCAT,» [En línea]. Available: <http://www.jpl.nasa.gov/missions/iss-rapidscat/>. [Último acceso: Octubre 2017].
- [34] «RapidScat Team Investigating Power System Anomaly,» [En línea]. Available: <http://www.nasa.gov/feature/jpl/rapidscat-team-investigating-power-system-anomaly>. [Último acceso: Octubre 2017].

- [35] «What is big data?,» [En línea]. Available: <https://www.ibm.com/analytics/us/en/big-data/>. [Último acceso: Octubre 2017].
- [36] «Start-Up Goes After Big Data With Hadoop Helper,» [En línea]. Available: <https://bits.blogs.nytimes.com/2010/04/22/start-up-goes-after-big-data-with-hadoop-helper/?dbk>. [Último acceso: Octubre 2017].