UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

título del TFG
Documentación Técnica

Presentado por nombre alumno
en Universidad de Burgos — 27 de noviembre
de 2015
Tutor: nombre tutor

Índice general

Índice de figuras

Apéndice A

Manuales

A.1. Introducción

A.2. Planificación temporal

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

B.2. Objetivos generales

B.3. Catalogo de requisitos

B.4. Especificación de requisitos

Apéndice C

Especificación de diseño

C.1. Introducción

C.2. Diseño de datos

C.3. Diseño procedimental

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

Apéndice E

Documentación de usuario

E.1. Introducción

A lo largo de este apéndice vamos a indicar el material necesario para poder ejecutar el proyecto, así como la manera de instalarlo, y la forma de hacer funcionar el proyecto una vez contemos con todos los elementos requeridos.

E.2. Requisitos de usuarios

E.3. Instalación

A continuación se van a definir los métodos de instalación de todos los componentes necesarios para ejecutar el proyecto, por si pudiera resultar de utilidad.

Estos métodos de instalación hacen referencia a la instalación en máquinas con un sistema operativo basado en alguna distribución de Debian. En nuestro caso, esta distribución ha sido Ubuntu 14.04.

Oracle Java 8

Aunque existen distribuciones libres de Java, usaremos la que proporciona Oracle (`http://www.java.com`) por el hecho de que incluye una serie de programas que pueden usarse para medir el rendimiento de aplicaciones Java (JConsole y JVisualVM). Cualquier otra distribución será igualmente válida, pero es posible que no incluya estas herramientas. Podemos acceder tanto al JRE como al JDK desde la siguiente dirección: `http://www.oracle.com/technetwork/java/javase/downloads/index.html`.

Sin embargo, Oracle no proporciona una instalación automática de Java para Linux. Es por esta razón vamos a utilizar un PPA (Personal Packa-

ge Archive) que proporciona un instalador para diferentes versiones de Java (https://launchpad.net/~webupd8team/+archive/ubuntu/java). Este instalador no contiene ningún archivo Java, pero será el encargado de descargarlos en nuestra máquina (de igual manera que podríamos hacerlo manualmente) y realizar la instalación automáticamente, facilitando el proceso de instalación.

Una vez entendido esto, ejecutaremos los siguientes comandos:

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer
```

Podemos comprobar si la instalación ha sido correcta ejecutando el comando `java -version` en la terminal. Si todo ha salido bien deberíamos recibir una salida similar a la siguiente:

```
$ java -version
java version "1.8.0_66"
Java(TM) SE Runtime Environment (build 1.8.0_66-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17,
mixed mode)
```

Finalmente, podemos ejecutar el siguiente comando para configurar correctamente las variables del sistema que Java necesita:

```
$ sudo apt-get install oracle-java8-set-default
```

Scala 2.11.7

Aunque podemos descargar la última versión de Scala desde la página oficial (http://www.scala-lang.org/), la descarga que se nos ofrece por defecto es un archivo .tar.gz que nos obligaría a realizar toda la instalación manualmente (suponiendo que estamos en un sistema Ubuntu como el utilizado durante el proyecto).

Por ello, vamos a realizar la instalación mediante un archivo .deb (paquete Debian) que también puede ser utilizado por nuestro sistema y que nos ahorrará realizar la instalación manualmente. Podemos acceder al repositorio que guarda este paquete desde un navegador (http://www.scala-lang.org/files/archive/) o descargarlo mediante el siguiente comando:

```
$ sudo wget www.scala-lang.org/files/archive
 /scala-2.11.7.deb
```

Independientemente del método seguido, una vez tengamos el archivo en nuestro ordenador ejecutamos el siguiente comando desde la carpeta que contenga el paquete .deb:

```
$ sudo dpkg -i scala-2.11.7.deb
```

Si todo ha salido correctamente, una vez termine de ejecutarse la orden anterior podemos ejecutar el comando `scala -version` y esperar una salida similar a esta:

```
$ scala -version
Scala code runner version 2.11.7 -- Copyright 2002-2013
```

Recordar que, en el caso de utilizar cualquier otra versión de Scala, esta tiene que ser compatible con la versión Java que se encuentre instalada. Por ejemplo, Java 8 solo puede ser utilizado a partir de versiones 2.11 y será obligatorio a partir de la versión de Scala 2.12. [2]

Apache Maven

Lo primero que debemos hacer es descargarnos el paquete que contiene la herramienta. Esto podemos hacerlo desde el navegador en la página oficial de Apache Maven (https://maven.apache.org/) o mediante el siguiente comando en consola:

```
wget http://apache.rediris.es/maven/maven-3/3.3.3
 /binaries/apache-maven-3.3.3-bin.tar.gz
```

Recordar que el código de arriba es orientativo, pudiéndose seleccionar otro lugar desde donde realizar la descarga u otra versión del producto.

Descomprimimos el paquete descargado y lo movemos a la carpeta `/usr/local`:

```
$ tar -zxf apache-maven-3.3.3-bin.tar.gz
$ sudo mv -R apache-maven-3.3.3 /usr/local
$ sudo ln -s /usr/local/apache-maven-3.3.3/bin/mvn
  /usr/bin/mvn
```

Podemos comprobar que la instalación ha sido realizada correctamente si al escribir el comando `mvn --version` recibimos una salida parecida a la siguiente:

```
$ mvn --version
Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06;
 2015-04-22T13:57:37+02:00)
Maven home: /usr/local/apache-maven-3.3.3
Java version: 1.8.0_66, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-oracle/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.19.0-25-generic", arch:
 "amd64", family: "unix"
```

Apache Spark 1.5.1

Nos dirigiremos a la página oficial de Apache Spark (http://spark.apache.org/). Elegiremos la versión 1.5.1 por ser la utilizada a lo largo de la práctica, y descargaremos el código fuente. Igualmente, y como hemos mencionado en otras instalaciones, podemos ejecutar el siguiente comando para hacernos con el paquete:

```
$ wget http://apache.rediris.es/spark/spark-1.5.1
  /spark-1.5.1.tgz
```

Una vez tengamos el archivo en nuestra máquina lo descomprimimos y movemos a la carpeta que deseemos lanzando estos comandos desde el directorio que contenga el paquete descargado:

```
$ tar -xvf spark-1.5.1.tgz
$ sudo mv –R spark-1.5.1 /opt
```

Finalmente vamos a construir Spark utilizando los siguientes comandos desde la carpeta donde lo hemos ubicado. En nuestro caso, estará en `/opt/spark-1.5.1`:

```
$ sudo ./dev/change-scala-version.sh 2.11
$ sudo mvn -Pyarn -Phadoop-2.6 -Dscala-2.11 -Pnetlib-lgpl
  -DskipTests clean package
```

Es importante incluir la opción `-Pnetlib-lgpl` para que Spark incluya una serie de clases utilizadas por su librería MLlib[1].

Esta última operación llevará un tiempo.

Finalmente, si deseamos comprobar que Spark se ha desplegado correctamente podemos ejecutar, por ejemplo, el intérprete de comandos de Scala para Spark:

```
$ ./bin/spark-shell
```

Aunque recibiremos una serie de avisos (*warnings*) debido a que no hemos configurado ciertos aspectos de Spark, el intérprete debería poder lanzarse y funcionar sin problemas.

Algoritmos de selección de instancias

Para descargar el resultado final de nuestro proyecto

E.4. Instalaciones Opcionales

Weka

No se trata de un programa necesario para la ejecución del proyecto, pero, dado que parte del mismo ha sido destinado a comparar los resultados entre Apache Spark y Weka, lo consideraremos como un programa a incluir en este apéndice.

Durante la realización de la práctica se han utilizado dos versiones diferentes de este software: Weka-3.6.13 y Weka-3.7.13. La instalación de ambos es similar, por lo que para no tener información duplicada en este anexo explicaremos la instalación refiriéndonos siempre a la versión 3.7.13

Lo primero que haremos será descargar Weka desde su página oficial (http://www.cs.waikato.ac.nz/ml/weka/). De entre las posibles opciones que nos ofrecen, hemos de seleccionar la que nos permite descargar un archivo de extensión .zip destinado a "*Other platforms (Linux, etc.)*"

Una vez tengamos el archivo, escribiremos en la terminal el siguiente comando, que nos permitirá descomprimir el directorio y colocarlo en la carpeta `/opt`:

```
$ sudo unzip ~/Downloads/weka-3-7-13.zip -d /opt
```

Para comprobar el correcto funcionamiento de Weka podemos dirigirnos a la carpeta de instalación y ejecutar el siguiente comando. Si todo ha sido realizado correctamente Weka debería iniciarse:

```
$ cd /opt/weka-3-7-13
$ java -jar weka.jar
```

E.5. Manual del usuario

Bibliografía

[1] Spark. Machine Learning Library (MLlib) Guide - Dependencies, 2015. http://spark.apache.org/docs/latest/mllib-guide.html#dependencies.

[2] École Polytechnique Fédérale de Lausanne (EPFL). Scala 2.12 roadmap, 2015. http://www.scala-lang.org/news/2.12-roadmap.