Examen Final

Nombre: Micaela Jhoselin Saenz Molina

C.I: 8464832 LP

1. En una máquina virtual realice la configuración de apache spark, puede guiarse en cualquier tutorial o el proporcionado por el docente.
   url: https://computingforgeeks.com/how-to-install-apache-spark-on-ubuntu-debian/

```
tar: Error is not recoverable: exiting now
user@user-virtual-machine:~$ wget https://dlcdn.apache.org/spark/spark-3.3.1/spa
rk-3.3.1-bin-hadoop3.tgz
--2022-12-03 14:19:31--  https://dlcdn.apache.org/spark/spark-3.3.1/spark-3.3.1-
bin-hadoop3.tgz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 299350810 (285M) [application/x-gzip]
Saving to: ‘spark-3.3.1-bin-hadoop3.tgz’

spark-3.3.1-bin-had 100%[===================>] 285,48M  10,1MB/s    in 28s

2022-12-03 14:20:00 (10,1 MB/s) - ‘spark-3.3.1-bin-hadoop3.tgz’ saved [299350810
/299350810]

user@user-virtual-machine:~$
```

Permanent link

```
HTTP request sent, awaiting response... 200 OK
Length: 299350810 (285M) [application/x-gzip]
Saving to: ‘spark-3.3.1-bin-hadoop3.tgz’

spark-3.3.1-bin-had 100%[===================>] 285,48M  10,1MB/s    in 28s

2022-12-03 14:20:00 (10,1 MB/s) - ‘spark-3.3.1-bin-hadoop3.tgz’ saved [299350810
/299350810]

user@user-virtual-machine:~$ tar xvf spark-3.3.1-bin-hadoop3.tgz
spark-3.3.1-bin-hadoop3/
spark-3.3.1-bin-hadoop3/LICENSE
spark-3.3.1-bin-hadoop3/NOTICE
spark-3.3.1-bin-hadoop3/R/
spark-3.3.1-bin-hadoop3/R/lib/
spark-3.3.1-bin-hadoop3/R/lib/SparkR/
spark-3.3.1-bin-hadoop3/R/lib/SparkR/DESCRIPTION
spark-3.3.1-bin-hadoop3/R/lib/SparkR/INDEX
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/Rd.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/features.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/hsearch.rds
spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/links.rds
```

```
spark-3.3.1-bin-hadoop3/yarn/
spark-3.3.1-bin-hadoop3/yarn/spark-3.3.1-yarn-shuffle.jar
user@user-virtual-machine:~$ cd ./spark-3.3.1-bin-hadoop3
user@user-virtual-machine:~/spark-3.3.1-bin-hadoop3$ ls
bin  conf  data  examples  jars  kubernetes  LICENSE  licenses  NOTICE  python  R  README.md  RELEASE  sbin  yarn
user@user-virtual-machine:~/spark-3.3.1-bin-hadoop3$ pwd
/home/user/spark-3.3.1-bin-hadoop3
user@user-virtual-machine:~/spark-3.3.1-bin-hadoop3$ sudo nano ~/.bashrc
user@user-virtual-machine:~/spark-3.3.1-bin-hadoop3$ source ~/.bashrc
user@user-virtual-machine:~/spark-3.3.1-bin-hadoop3$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/12/05 09:42:30 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicabl
e
Spark context Web UI available at http://localhost:4040
Spark context available as 'sc' (master = local[*], app id = local-1670247751727).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 11.0.17)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

Cambios previos en nano

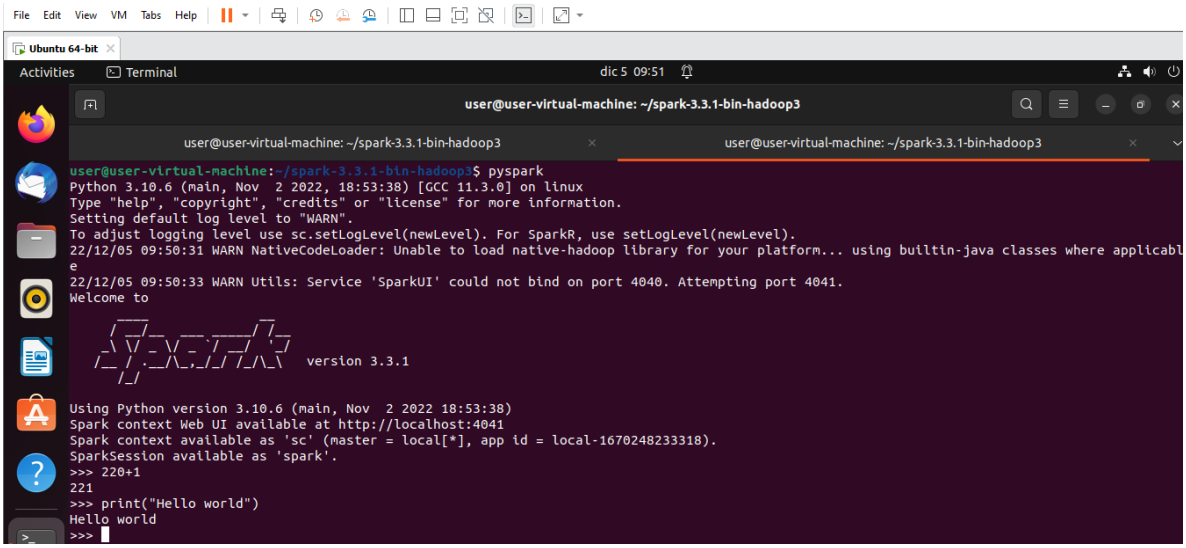
```
  GNU nano 6.2                                    /home/user/.bashrc

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

#[ -f "/home/user/.ghcup/env" ] && source "/home/user/.ghcup/env" # ghcup-env
[ -f "/home/user/.ghcup/env" ] && source "/home/user/.ghcup/env" # ghcup-env

export SPARK_HOME=/home/user/spark-3.3.1-bin-hadoop3

export PATH=$PATH:$SPARK_HOME/bin

export SPARK_LOCAL_IP=localhost

export PYSPARK_PYTHON=/usr/bin/python3

export PYTHONPATH=$(ZIPS=("$SPARK_HOME"/python/lib/*.zip); IFS=:; echo "${ZIPS[*]}"):$PYTHONPATH

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location  M-U Undo      M-A Set Mark  M-] To Brac
```

Con el shell podra ejecutar scala por defecto

Instale Python para spark

Se uso el siguiente tutorial:

https://dev.to/kinyungu_denis/to-install-apache-spark-and-run-pyspark-in-ubuntu-2204-4i79



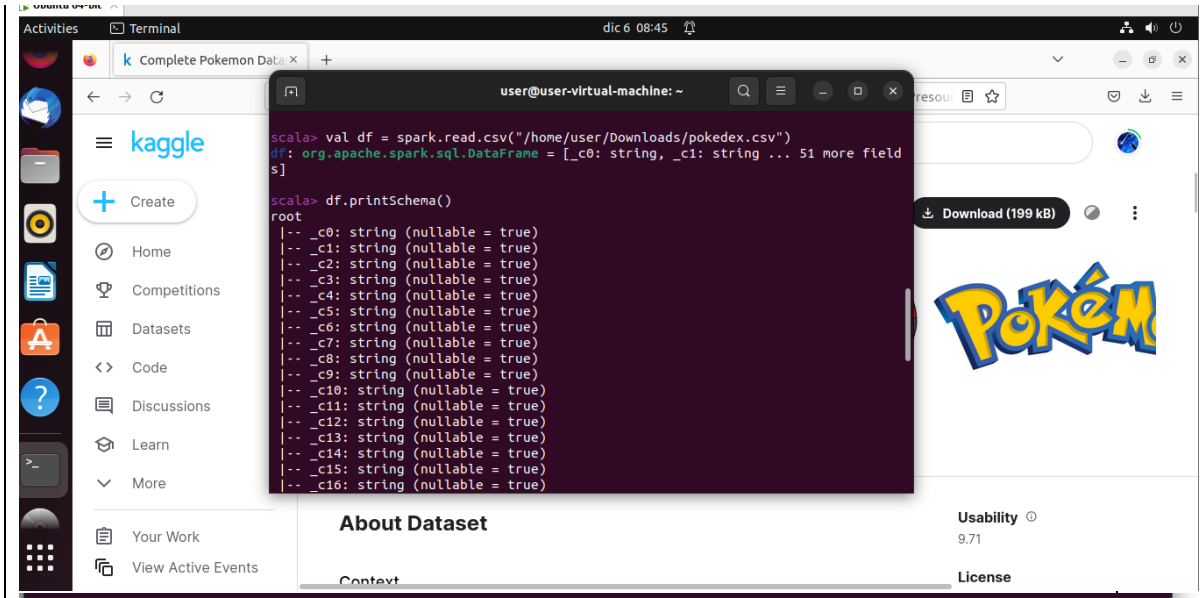
2. Realice el siguiente código, documente su funcionamiento en apache spark

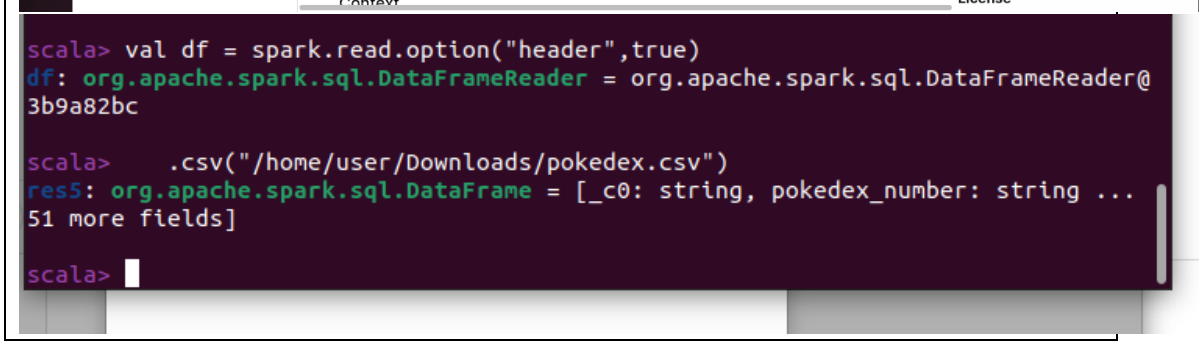
Sesiones

```
val spark: SparkSession = SparkSession.builder()
  .master("local[*]")
  .appName("simple-app")
  .getOrCreate()
```

```
scala>   .builder()
res5: spark.Builder = org.apache.spark.sql.SparkSession$Builder@422b5005

scala>   .appName("Spark SQL basic example")
res6: spark.Builder = org.apache.spark.sql.SparkSession$Builder@422b5005

scala>   .config("spark.some.config.option", "some-value")
res7: spark.Builder = org.apache.spark.sql.SparkSession$Builder@422b5005

scala>   .getOrCreate()
22/12/05 09:56:49 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
res8: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@31adc338
```

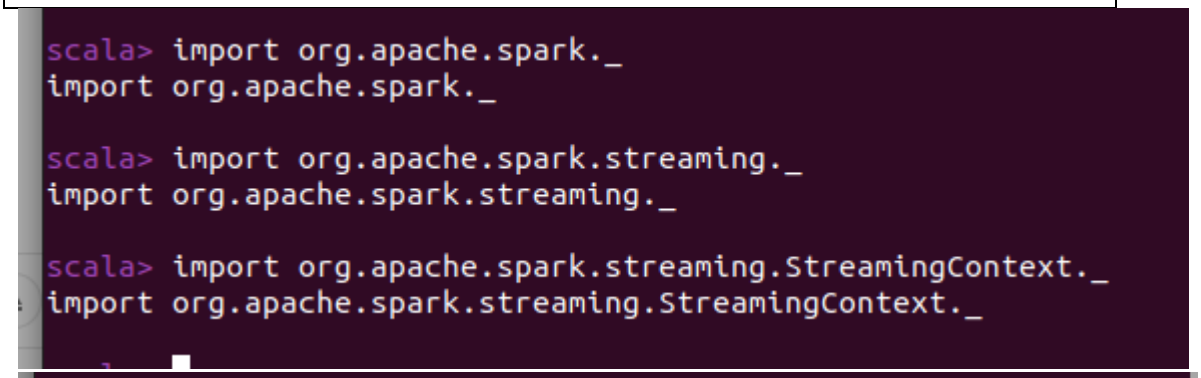
```
val dataSet: Dataset[String] = spark.read.textFile("textfile.csv")
val df: DataFrame = dataSet.toDF()
```

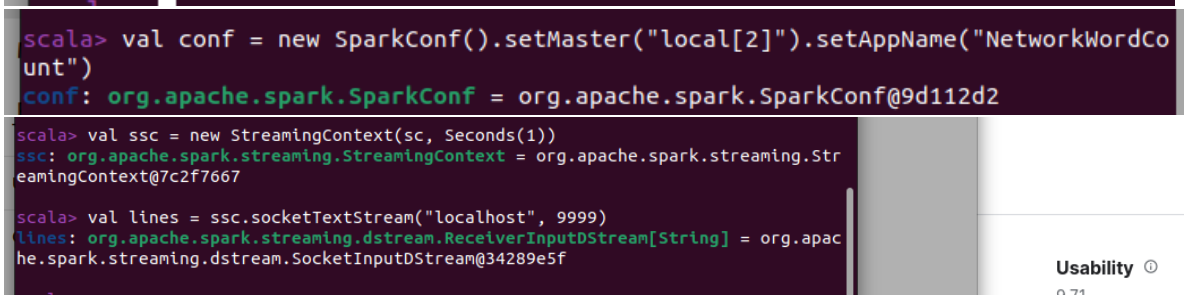




Streaming

```
val streamingContext: StreamingContext = new StreamingContext(sparkContext, Seconds(20))
val lines: ReceiverInputDStream[String] = streamingContext.socketTextStream("localhost", 9999)
```





```
scala> val conf = new SparkConf().setMaster("local[2]").setAppName("NetworkWordC
ount")
conf: org.apache.spark.SparkConf = org.apache.spark.SparkConf@358720a0

scala> val ssc = new StreamingContext(conf, Seconds(1))
org.apache.spark.SparkException: Only one SparkContext should be running in this
 JVM (see SPARK-2243).The currently running SparkContext was created at:
org.apache.spark.sql.SparkSession$Builder.getOrCreate(SparkSession.scala:947)
org.apache.spark.repl.Main$.createSparkSession(Main.scala:106)
<init>(<console>:15)
<init>(<console>:42)
<init>(<console>:44)
.<init>(<console>:48)
```

RDD

```
val cadenas = Array("Docentes", "inteligenciaArtificial", "quefinal")
val cadenasRDD = sc . parallelize (cadenas)
cadenasRDD.collect()
file.collect()
```

```
scala> val rdd = sc.parallelize(Array("Docentes", "IA", "Final"))
rdd: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[4] at parallelize a
t <console>:23

scala> rdd.take(3)
res3: Array[String] = Array(Docentes, IA, Final)

scala> val df = rdd.toDF()
df: org.apache.spark.sql.DataFrame = [value: string]

scala> df.show
+--------+
|   value|
+--------+
|Docentes|
|      IA|
|   Final|
+--------+
```

```
val filtro = cadenasRDD.filter(line => line.contains("quefinal"))
```

```
scala> import org.apache.spark.sql.functions.col
import org.apache.spark.sql.functions.col

scala> df.filter(col("value").contains("inal")).show()
+-----+
|value|
+-----+
|Final|
+-----+


scala> df.filter(col("value").contains("centes")).show()
+--------+
|   value|
+--------+
|Docentes|
+--------+
```

```
val fileNotFound = sc.textFile("/7añljdlsjd/alkls/", 6)
fileNotFound.collect()
```

```
scala> val fileNotFound = sc.textFile("user/jkdjksdj/",6)
fileNotFound: org.apache.spark.rdd.RDD[String] = user/jkdjksdj/ MapPartitionsRDD[
17] at textFile at <console>:24

scala> fileNotFound.collect()
org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: file:/
home/user/user/jkdjksdj
  at org.apache.hadoop.mapred.FileInputFormat.singleThreadedListStatus(FileInputF
ormat.java:304)
  at org.apache.hadoop.mapred.FileInputFormat.listStatus(FileInputFormat.java:244
)
```

En github tienen que subir en un repositorio los códigos de cada pregunta(carpeta), darle mínimamente acceso a msilva@fcpn.edu.bo, mandar al correo con referencia "2o parcial 319", notificar al mismo correo hasta el día 12 de diciembre a horas 12:00.