EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

NATIONAL TECHNOLOGICAL INSTITUTE OF MEXICO
TECHNOLOGICAL INSTITUTE OF TIJUANA

ACADEMIC SUBDIRECTION
Systems and Computing Department

SEMESTER
agosto - diciembre 2021

ACADEMIC CAREER
Eng. Information and communication technologies

MATERIA Y CLAVE:
Datos Masivos BDD-1704TI9A

STUDENT'S NAME AND REGISTRATION:
Velázquez Farrera César Alejandro 17212937

NAME OF THE JOB:
Evaluation #1

UNIT:
Unidad I

NAME OF THE TEACHER:
M.C. José Christian Romero Hernández

Tijuana, Baja California **Due date:** September 28, 2021

Introduction

This evaluation aims to evaluate the student of the bases of the Scala and Apache Spark language for handling Big Data.

The following questions are answered below with Spark DataFrames and Scala using a "CSV" file named Netflix_2011_2016.csv found in the spark-dataframes folder.

1. Initialize a simple Spark session

To initialize a simple session in Spark it is necessary to import a library called SparkSession, use the import instruction followed by the name of the class.

```
import org.apache.spark.sql.SparkSession
val session = SparkSession.builder().getOrCreate
```

2. Upload Netflix Stock CSV file, have spark infer data types

To load the session the data from the provided CSV file, the data in the CSV file must be assigned a value along with its data types and headers.

```
val df_netflix = session.read.option("header",
"true").option("inferSchema", true).csv("Netflix_2011_2016.csv")
```

3. What are the names of the columns?

To learn the names of the columns in the CSV file, you only need to refer to the value that was provided in the previous step and execute the columns.

```
df_netflix.columns
```

4. To learn the names of the columns in the CSV file, you only need to refer to the value that was provided in the previous step and execute the columns.

The schema of a table will show the name of the columns and the type of data that corresponds to each one. The printSchema method is used.

```
df_netflix.printSchema()
```

5. Print the first 5 columns

Like other languages like R or Pandas syntax in Python, you need to use the head () method, its attribute will be the number of lines you want to display.

```
df_netflix.head(5)
```

6. Use describe () to learn about the Data Frame.

To show the result of the describe () method, you need to execute a show method to print the results of the previous method to the console.

```
df_netflix.describe().show
```

7. Create a new data frame with a new column called "HV Ratio" which is the relationship between the price in the "High" column versus the "Volume" column of shares traded for a day. Hint is an operation.

To get the relationship between the columns "High" and "Volume" you need to use division. To save the results in a dataframe you need to create a copy and specify a new column "HV Ratio".

```
val df_netflix2 = df_netflix.withColumn("HV Ratio",
df_netflix("High")/df_netflix("Volume"))
```

8. What day had the highest peak in the “Open” column?

To obtain the highest day, the select function is used together with the max method as an attribute.

```
df_netflix.select(mean("Open")).show()
```

9. What is the meaning of the Close column “Close” in the context of financial information?

The close column in the world of finance means that at the end of each day the transaction books are closed. Take into account the profits and losses of companies. Most likely, in the context of this assessment, it will be used on the stock market.

```
df_netflix.select(mean("Close")).show()
```

10. What is the maximum and minimum in the “Volume” column?

To display the data of a table in SQL, in Scala the "select" statement is used and together with the "max" and "min" functions we can display the maximum and minimum values of a column.

```
df_netflix.select(max("Volume")).show()
df_netflix.select(min("Volume")).show()
```

11. With Scala / Spark $ syntax, answer the following:

a. How many days was the “Close” column less than $ 600?

```
val Day = df_netflix.where($"Close" < 600).count()
```

b. What percentage of the time was the "High" column greater than $ 500?

```
val Day = df_netflix.where($"High" > 500).count().toFloat
```

c. What is the Pearson correlation between the “High” column and the “Volume” column?

```
df_netflix.select(corr("High", "Volume")).show()
```

d. What is the maximum in the “High” column per year?

```
df_netflix.groupBy)yrar($"Date")).max("High").show()
```

e. What is the average in the “Close” column for each calendar month?

```
val df_netflix3 = df_netflix.groupBy(year($"Date"),
month($"Date")).mean("Close"). toDF("Year","Month","Mean")

df_netflix3.orderBy($"Year",$"Month").show()
```

Conclusion

In this exam we were able to observe how efficient the Apache Spark program is for querying and analyzing, in the case of a CSV file such as the data in the file “Netflix_2011_2016.csv” such as the duration, volume and date of the data that they are within it.