CHALLENGE - DATA ENGINEER

Cirino Martinez

Objective

The objective of this challenge is to assess your skills using distributed programming.

Problem

The challenge consists in computing in a distributed way the 7-day moving average for historical stock prices using either Spark or MapReduce (in the Appendix below we provide a definition of a moving average).

You will use the stock prices data contained here:

https://www.kaggle.com/ehallmar/daily-historical-stock-prices-1970-2018#historical_stock_price s.csv (you need to register/log in in order to download the data file).

You have to compute the 7-day moving average for at least two stocks, using all the available information for those stocks. There is not a unique way to do this, and you are free to implement the solution you find the best.

Solutions

For the solution of the problem, I used the following tools:

- java version "1.8.0_221"
- spark 2.4.0
- Scala 2.11.12
- Maven
- IntelliJ
- Postgres
- Github

+

- S3
- EC2
- EMR
- Amazon RDS (Postgres)
- Tableau Desktop.

About the file historical_stock_prices

Daily Historical Stock Prices (1970 - 2018)
Historical stock prices for several thousand unique stock tickers

Daily stock prices for a selection of several thousand stock tickers from NYSE and NASDAQ. Unfortunately, it was not possible to parse the data in a manner that allowed exact decimal calculations, so floating-point numbers were required.

The file contains the following columns:

- **Ticker**: The symbol for the stock.
- **Open**: The open price.
- **Close**: The close price.
- **Adj**_close: The adjusted close price.
- **Low:** The low price.
- **High:** The high prices.
- **Volume:** The volume.
- **Date:** The date.

- Download the data historical_stock_prices.csv (1.87 GB).

Fig. 1.0 – historical_stock_prices.csv

- Create an application with Spark 2.4.0 and Scala language, to obtain the calculation of the required key indicators.

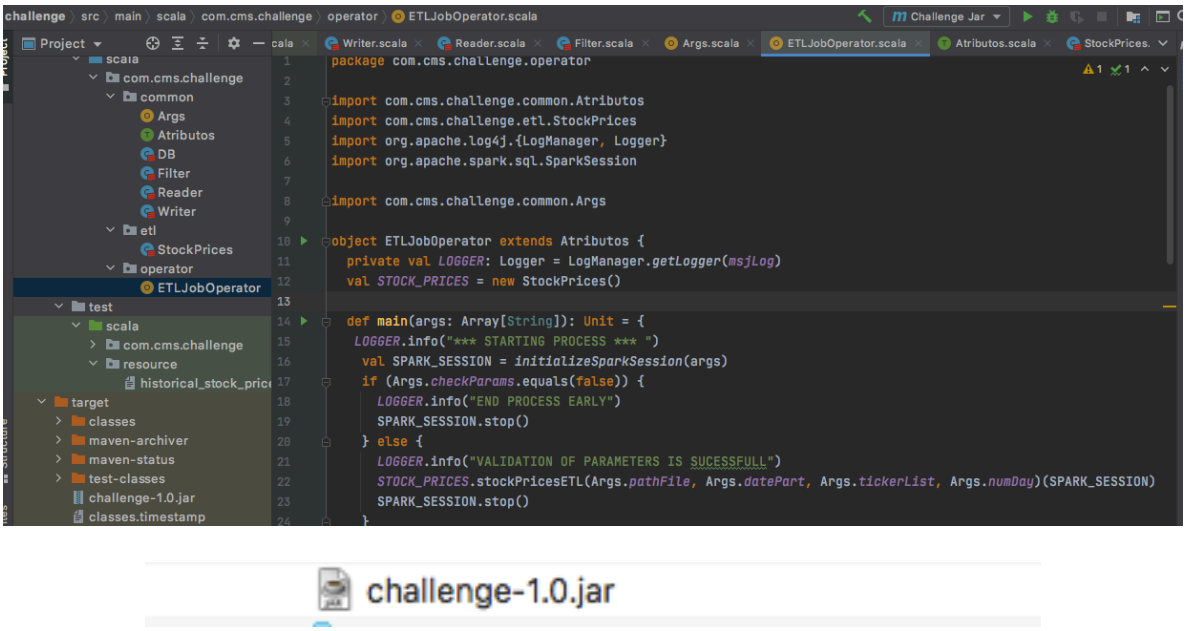

Fig. 2.0 – Overview of the app code.

- Link: https://github.com/cmartinezsa/challenge

- Execute the application for the compute the results.
  The process Challenge Operator receive the follow arguments:

  1. Source path of the csv file with contain the data.
  2. Target date to compute the results.
  3. Tickers list to filter results.
  4. Number of days required to compute the data.
  5. The App name for the Spark process.
  6. Mode execution of Spark (Local or Cluster)
  7. Target table to save the results.

Example of Spark-submit.

```
spark-submit  --conf "spark.service.user.postgresql.pass=*"\
--conf "spark.service.user.postgresql.user=postgres"\
--conf "spark.service.user.postgresql.database=db_challenge"\
--conf "spark.service.user.postgresql.port=5432"\
--conf "spark.service.user.postgresql.host=challengedb.cxbrv63skb3q.us-east-
1.rds.amazonaws.com"\
--class com.cms.challenge.operator.ETLJobOperator challenge-1.0.jar\
s3://changedb/historical_stock_prices.csv 2018-12-31 AHH,APO,PEZ,CRCM,FLWS 365\
AppChallenge local[*] hist_stock_prices_mov_hist
```

- The input data from S3 AWS.

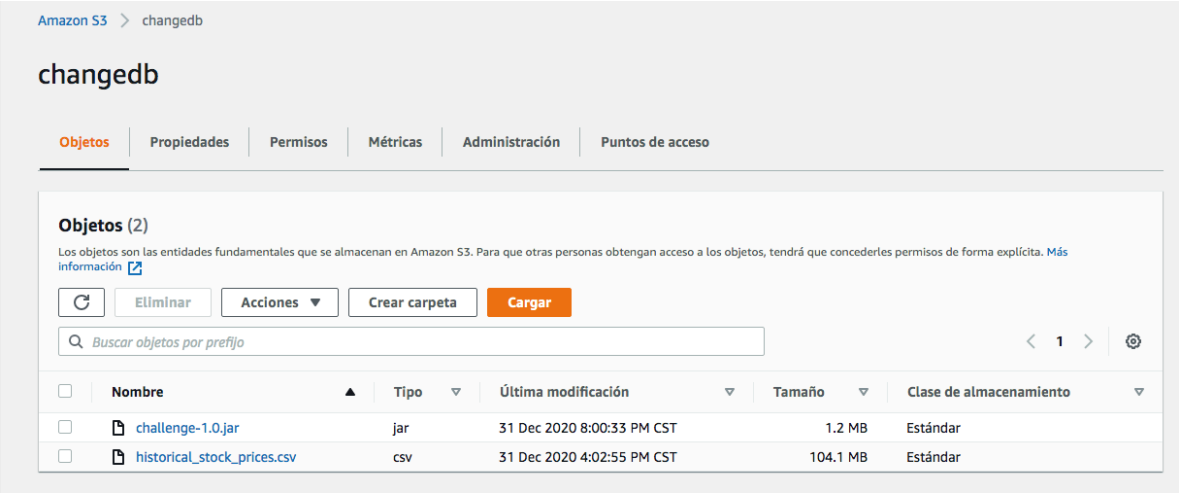

Fig. 3.0 – Amazon S3 – Bucket changedb.

- Execution with Amazon EMR

Fig. 4.0 – Amazon EMR – Spark-Submit – AppChallenge.

- Storage out in Amazon RDS Postgres database.

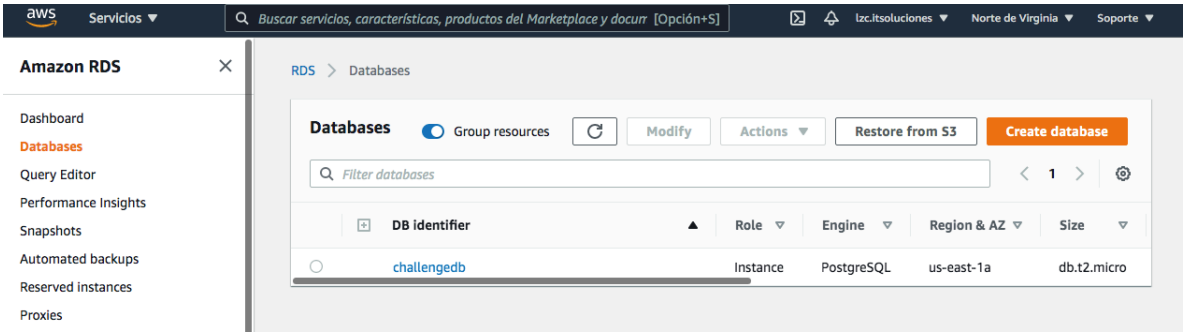

Fig. 5.0 – Amazon RDS – Database PostgreSQL.

- Show data out process.

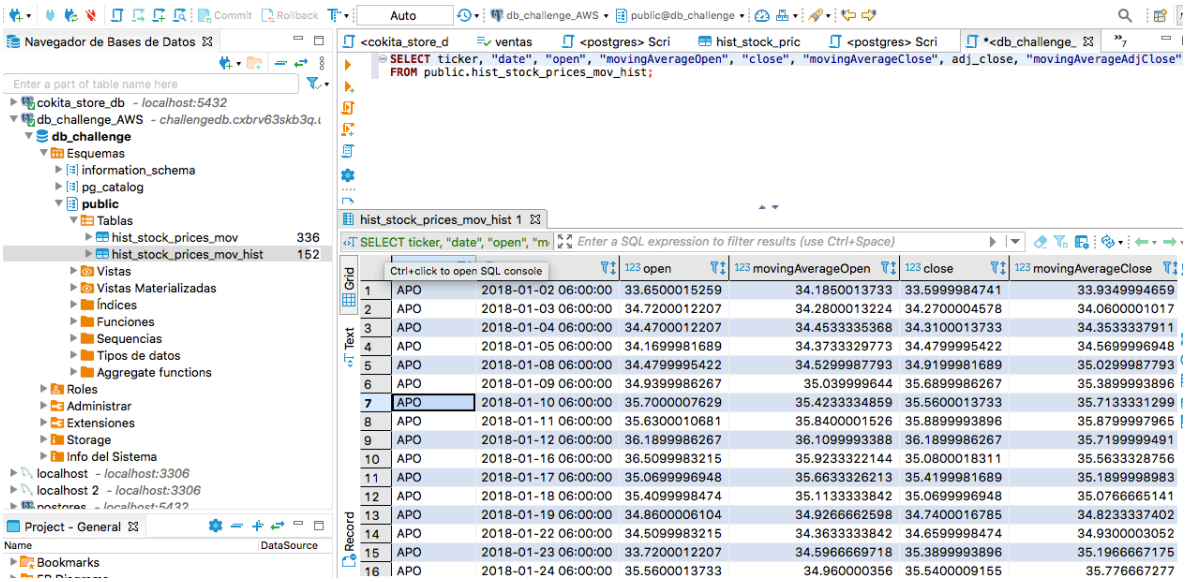

Fig. 6.0 – db_challenge – table. Hist_stoctk_prices_mov_hist.

- Show data out process with Tableau Desktop.

Fig. 7.0 – Slide presentation.

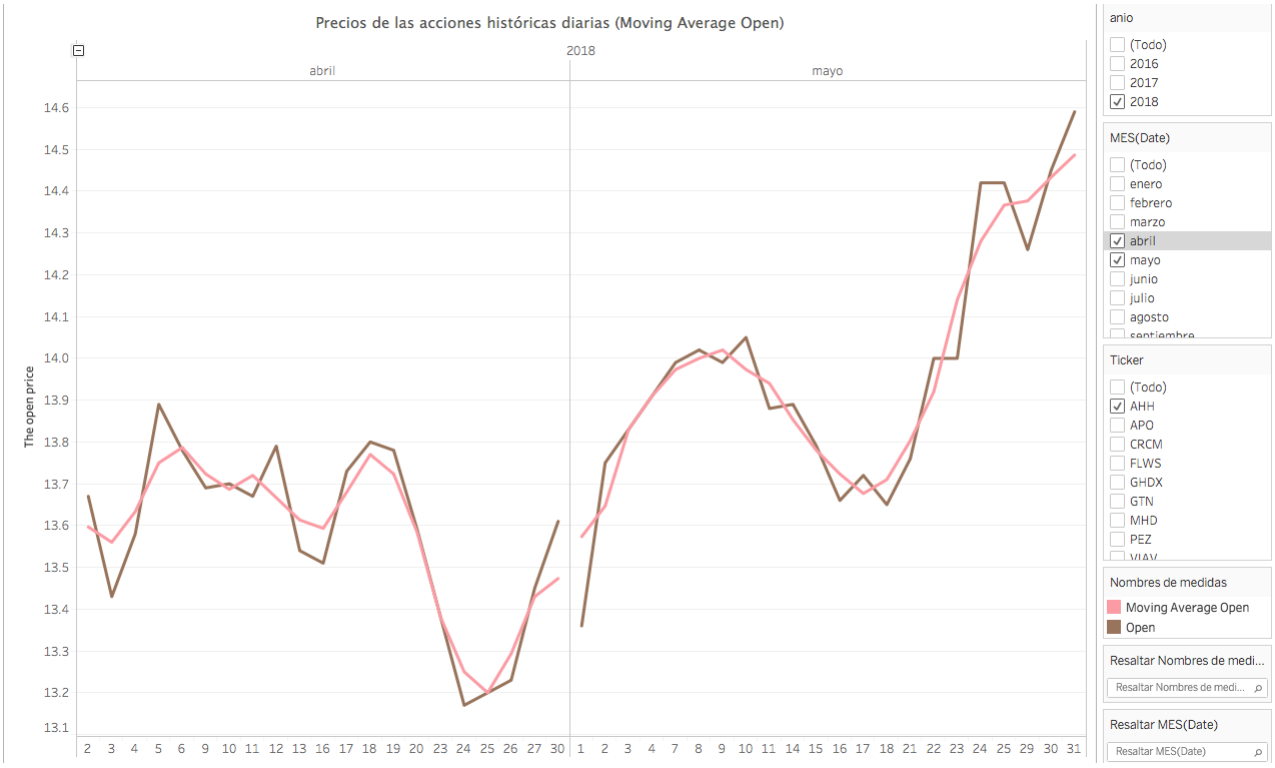

Fig. 8.0 – Slice Moving Average Open.

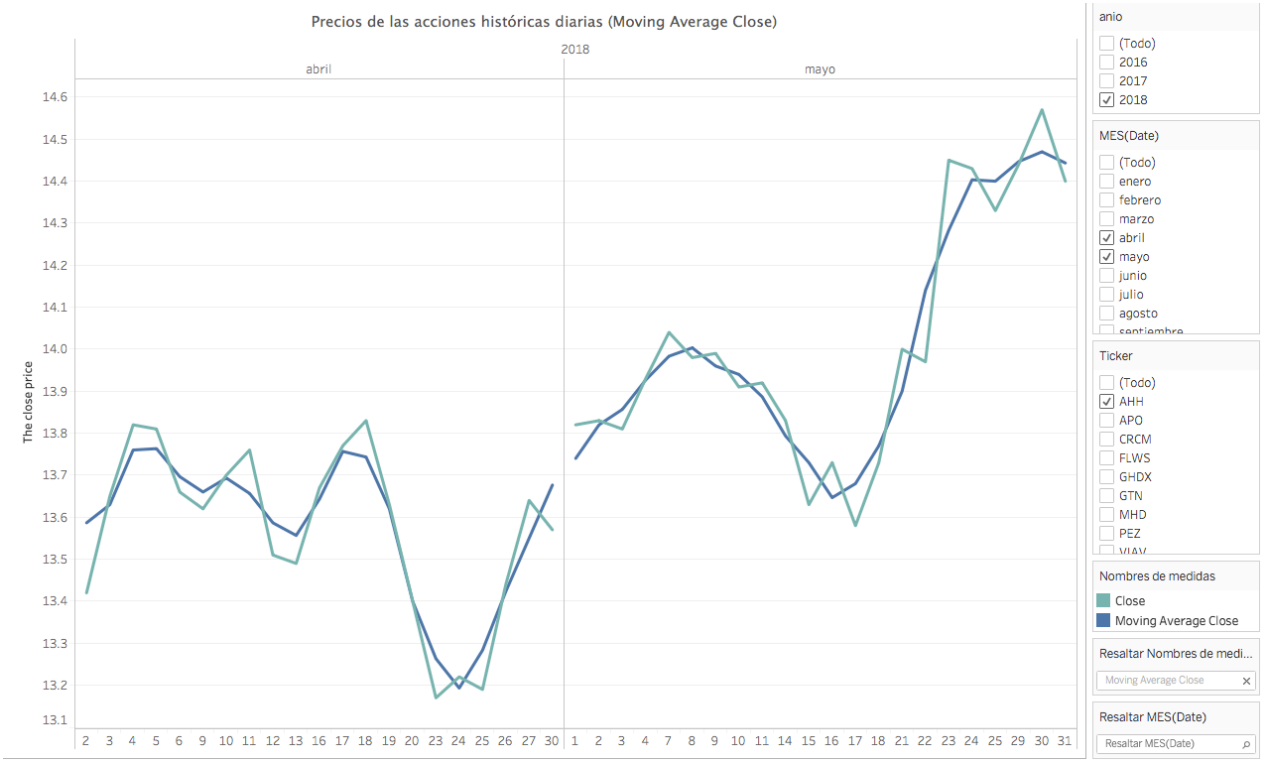

Fig. 9.0 – Slice Moving Average Close.

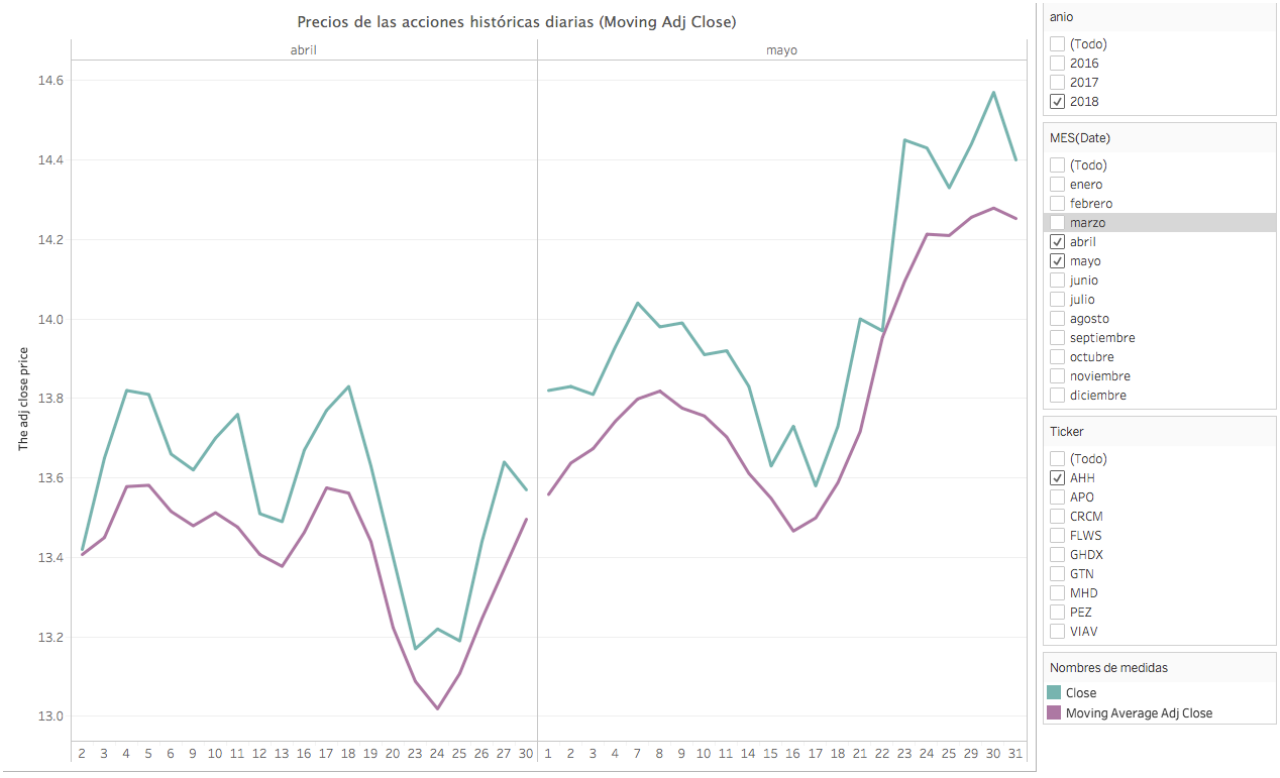

Fig. 10.0 – Slice Moving Adj. Close.

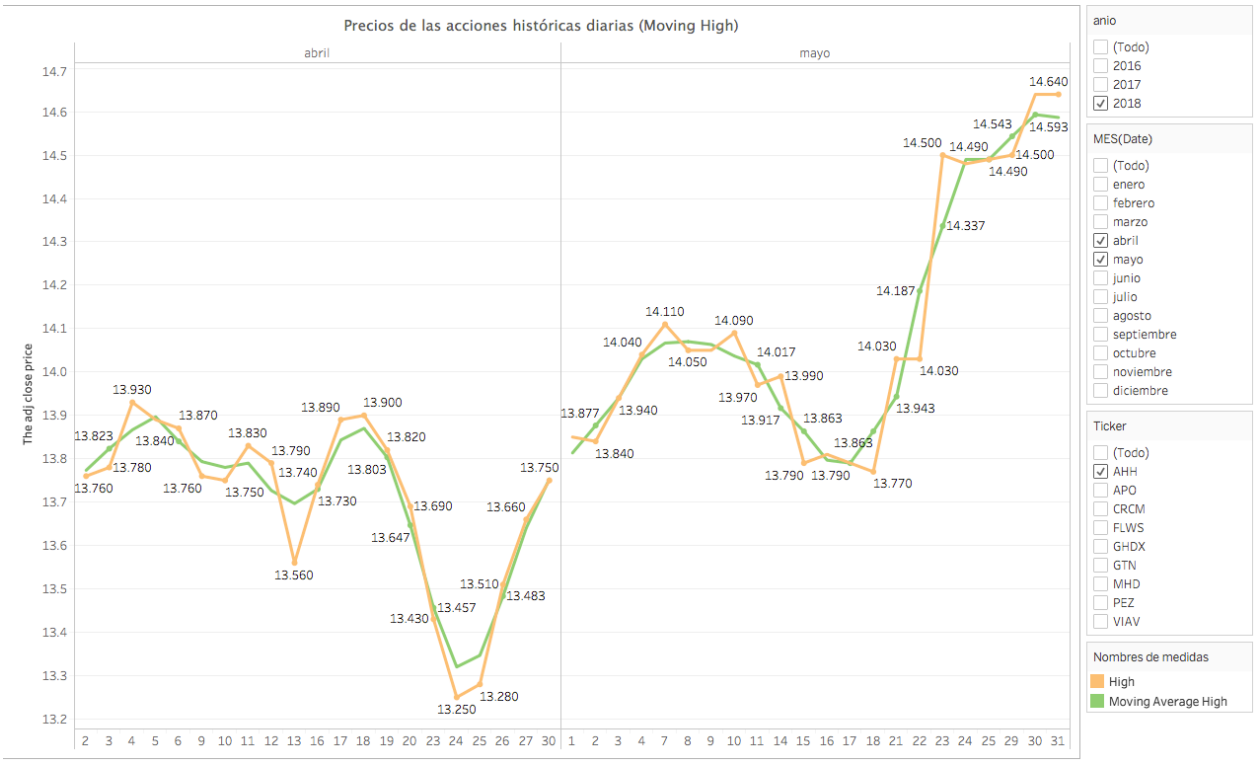

Fig. 11.0 – Slice Moving Average high.

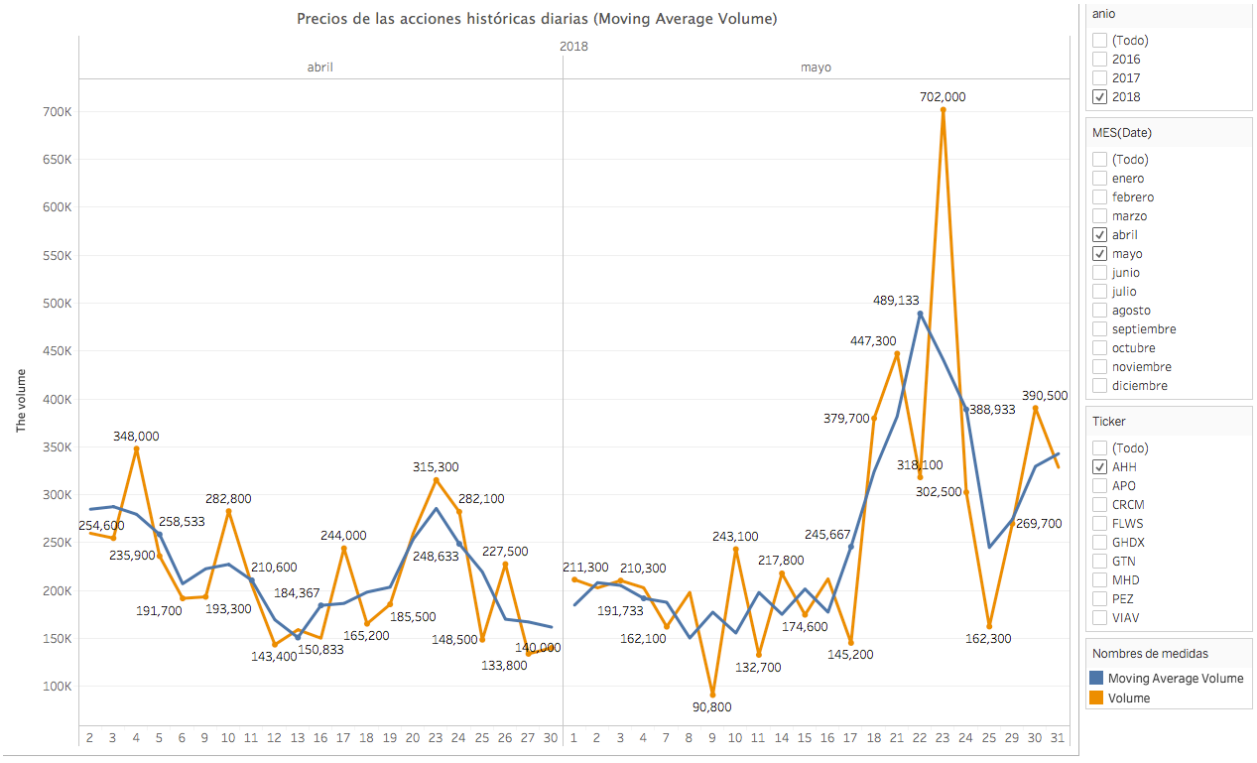

Fig. 12.0 – Slice Moving Average Volume.

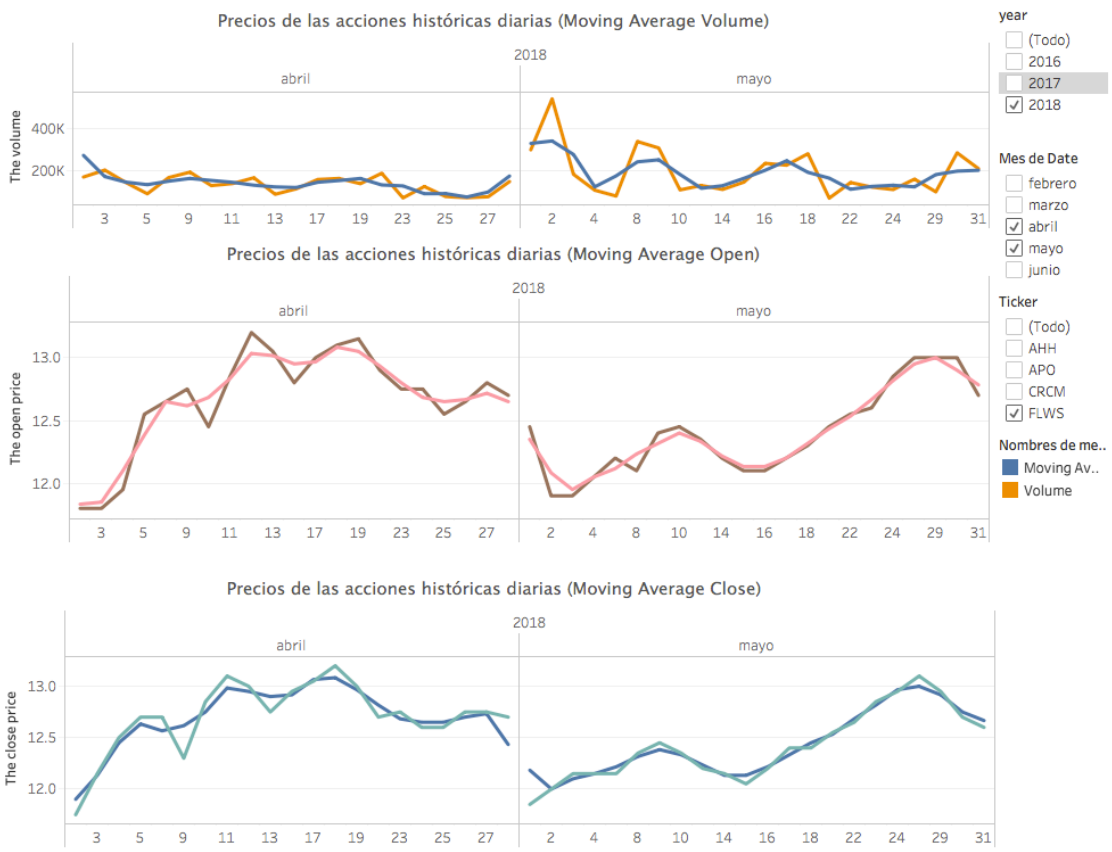

Fig. 13.0 – Slice Moving Average Volume / Open / Close.

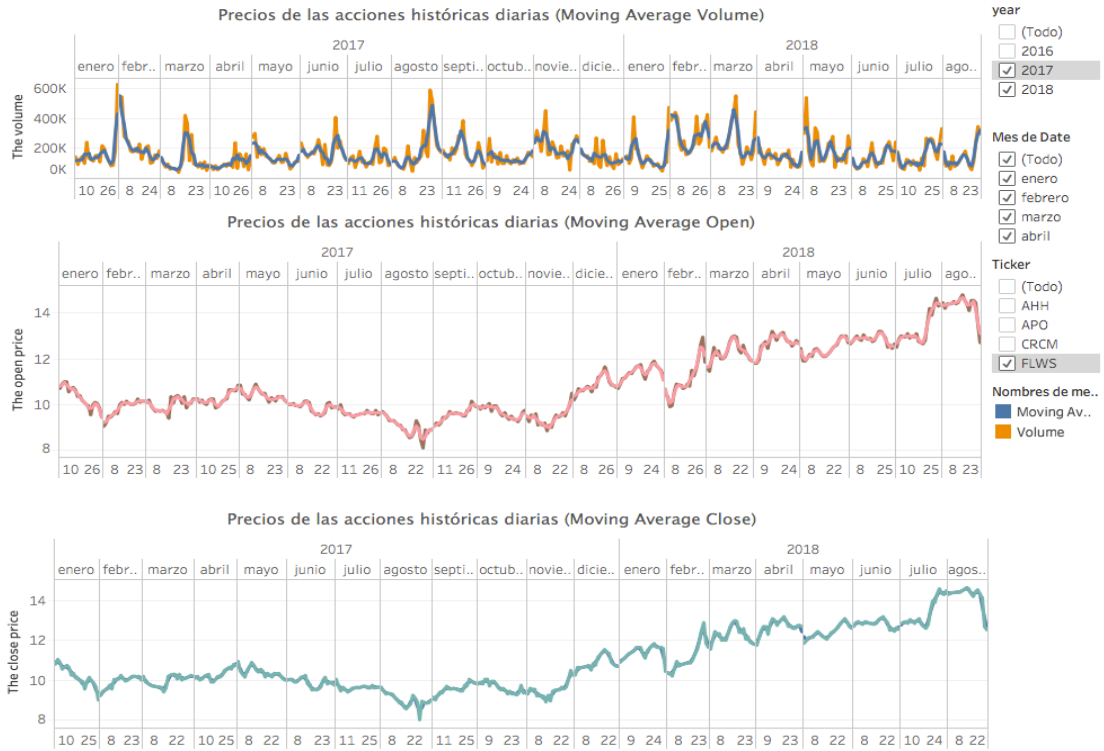

Fig. 14.0 – Slice Moving Average Volume / Open / Close – 2017 -2018