

Práctica 2

Archivo 1 - "GuatemalaExportsTo.csv"

1. Gráfica de barras que muestre el país con el mayor valor por exportaciones.

The screenshot shows the JupyterLab interface with a file browser on the left and a code editor on the right. The file browser displays a list of files in the '/work/' directory, including 'Covid19.csv', 'GuatemalaExportsTo.csv', and several '.ipynb' files. The code editor shows the following code:

```
[1]: import findspark
findspark.init()
from pyspark import SparkContext
from pyspark.sql.session import SparkSession
spark_context = SparkContext("local", "first_app")
spark = SparkSession(spark_context)
import plotly.graph_objects as go
from pyspark.sql import SQLContext
sql_context = SQLContext(spark_context)

[2]: text_file = sql_context.read.format("com.databricks.spark.csv").options(header='true', inferschema='true', quote='', delimiter=',').load("/home/joyvan/work/GuatemalaExportsTo.csv")
rdd_filter = text_file.rdd.map(tuple)
rdd_countries = rdd_filter.map(lambda word: (word[4], word[1]))
rdd_countries.take(5)

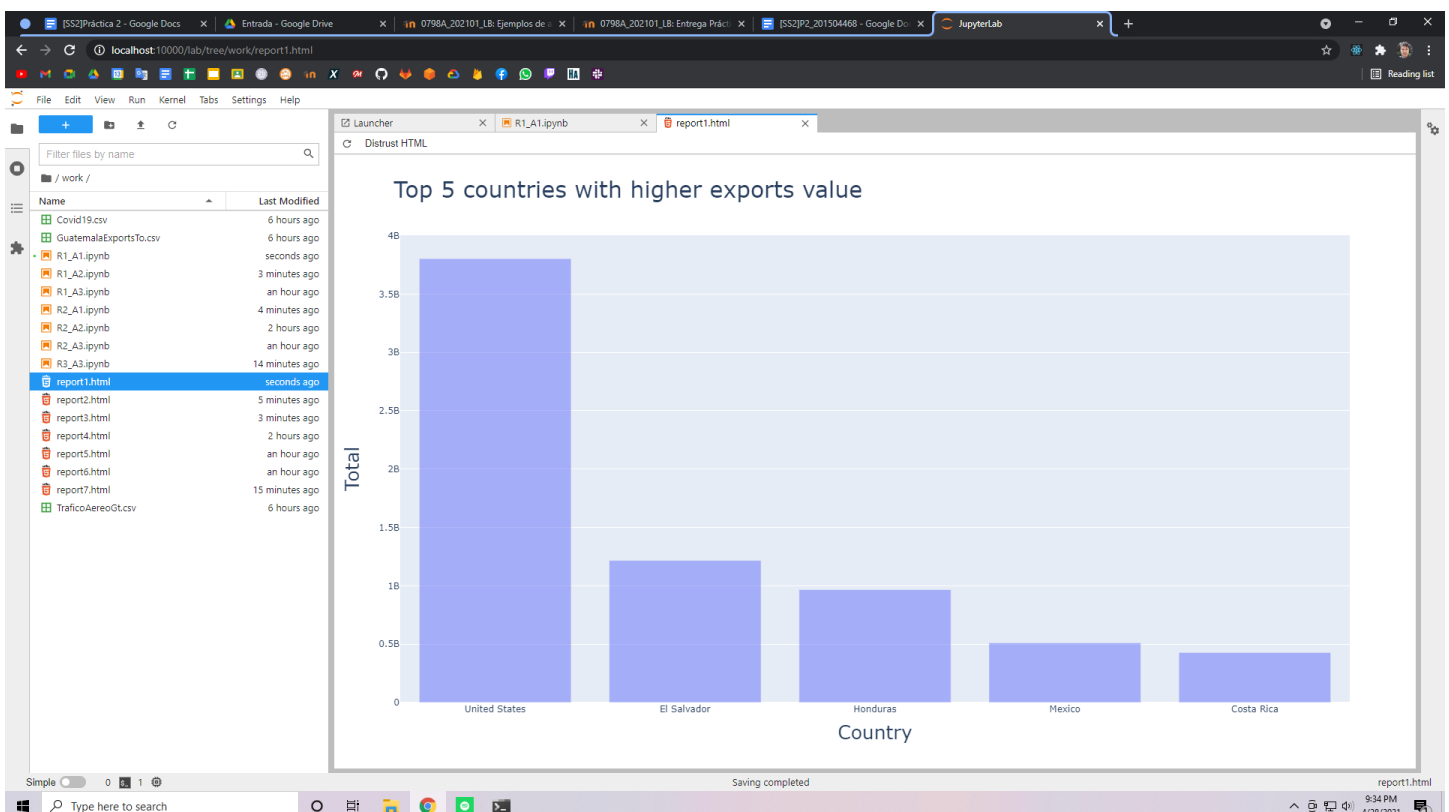
[3]: [ ('China', 18490000.0),
      ('China', 9830000.0),
      ('China', 8360000.0),
      ('China', 7110000.0),
      ('China', 4070000.0) ]

[4]: rdd_countries_export = rdd_countries.reduceByKey(lambda a,b: a+b)
print(rdd_countries_export.collect())
[ ('China', 64176766.0), ('Mexico', 511215129.0), ('Canada', 222575942.0), ('Japan', 164579976.0), ('United Kingdom', 101910872.0), ('Costa Rica', 428419775.0), ('Saudi Arabia', 111274535.0), ('Gabon', 6420.0), ('Italy', 166680831.0), ('France', 34023574.0), ('Brazil', 39146101.0), ('Germany', 144422819.0), ('Honduras', 967311616.0), ('Venezuela', 14990293.0), ('El Salvador', 1217138626.0), ('United States', 3803123168.0) ]

[5]: rdd_countries_export_sort = spark_context.parallelize(rdd_countries_export.sortBy(lambda a: a[1], False).take(5))
print(rdd_countries_export_sort.collect())
[ ('United States', 3803123168.0), ('El Salvador', 1217138626.0), ('Honduras', 967311616.0), ('Mexico', 511215129.0), ('Costa Rica', 428419775.0) ]

[6]: countries_name = rdd_countries_export_sort.map(lambda x: x[0])
countries_quantity = rdd_countries_export_sort.map(lambda x: x[1])
print(countries_name.collect())
print(countries_quantity.collect())
['United States', 'El Salvador', 'Honduras', 'Mexico', 'Costa Rica']
[3803123168.0, 1217138626.0, 967311616.0, 511215129.0, 428419775.0]

[7]: fig = go.Figure(data=go.Bar(x=countries_name.collect(), y=countries_quantity.collect()))
fig.update_layout(title_text="Top 5 countries with higher exports value", title_font_size=30,
                  yaxis=dict(title="Total", title_font_size=25),
                  xaxis=dict(title="Country", title_font_size=25))
fig.update_traces(overwrite=True, marker=dict(opacity=0.5))
fig.write_html("report1.html", auto_open=True)
```



2. Gráfica de pie de los 5 países con menos valor por exportaciones.

The screenshot shows a JupyterLab interface with a file browser on the left and a code editor on the right. The file browser displays a list of files including CSVs, HTML reports, and IPYNB files. The code editor contains a Python script that uses PySpark to read data from a CSV file, filter it, and generate a pie chart showing the top 5 countries with lower exports value. The script includes comments and prints the results of each step.

```
[1]: import findspark
findspark.init()
from pyspark import SparkContext
from pyspark.sql.session import SparkSession
spark_context = SparkContext("local", "first_app")
spark = SparkSession(spark_context)
import plotly.graph_objects as go
from pyspark.sql import SQLContext
sql_context = SQLContext(spark_context)

[2]: text_file = sql_context.read.format("com.databricks.spark.csv").options(header='true', inferschema='true', quote='', delimiter=',').load("/home/jovyan/work/GuatemalaExportsTo.csv")
rdd_filter = text_file.rdd.map(tuple)
rdd_countries = rdd_filter.map(lambda word: (word[4], word[1]))
rdd_countries.take(5)

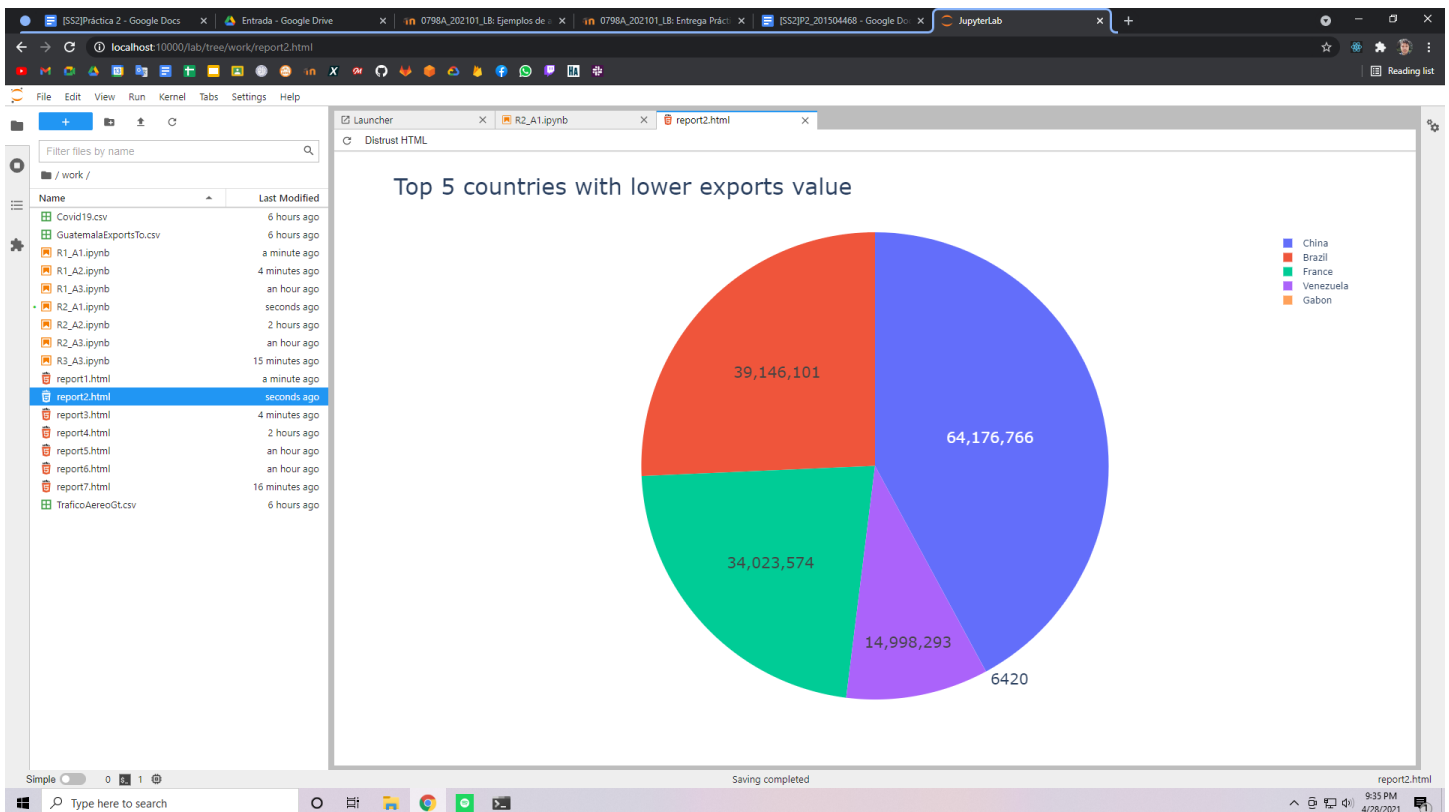
[2]: [('China', 18400000.0),
('China', 9830000.0),
('China', 8360000.0),
('China', 7110000.0),
('China', 4070000.0)]

[3]: rdd_countries_export = rdd_countries.reduceByKey(lambda a,b: a+b)
print(rdd_countries_export.collect())

[4]: rdd_countries_export_sort = spark_context.parallelize(rdd_countries_export.sortBy(lambda a: a[1], True).take(5))
print(rdd_countries_export_sort.collect())

[5]: countries_name = rdd_countries_export_sort.map(lambda x: x[0])
countries_quantity = rdd_countries_export_sort.map(lambda x: x[1])
print(countries_name.collect())
print(countries_quantity.collect())

[6]: fig = go.Figure(data=go.Pie(labels=countries_name.collect(), values=countries_quantity.collect()))
fig.update_layout(title_text='Top 5 countries with lower exports value', title_font_size=30)
fig.update_traces(hoverinfo='labelpercent', textinfo='value', textfont_size=20)
fig.write_html("report2.html", auto_open=True)
```



Archivo 2 - "TraficoAereoGt.csv"

1. Gráfica de barra con el total de aterrizajes por Aeropuerto.

```

[1]: import findspark
findspark.init()
from pyspark import SparkContext
from pyspark.sql.session import SparkSession
spark_context = SparkContext("local", "first_app")
spark = SparkSession(spark_context)
import plotly.graph_objects as go
from pyspark.sql import SQLContext
sql_context = SQLContext(spark_context)

[2]: text_file = sql_context.read.format("com.databricks.spark.csv").options(header="true", inferschema="true", quote="'", delimiter=',').load("/home/jovyan/work/TraficoAereoGt.csv")
rdd_filter = text_file.rdd.map(tuple)
rdd_landings = rdd_filter.map(lambda word: (word[2], word[3]))
rdd_landings.take(5)

[2]: [('Coatepeque', 3607),
('Coatepeque', 5261),
('Coatepeque', 629),
('Coatepeque', 3690),
('Coatepeque', 307)]

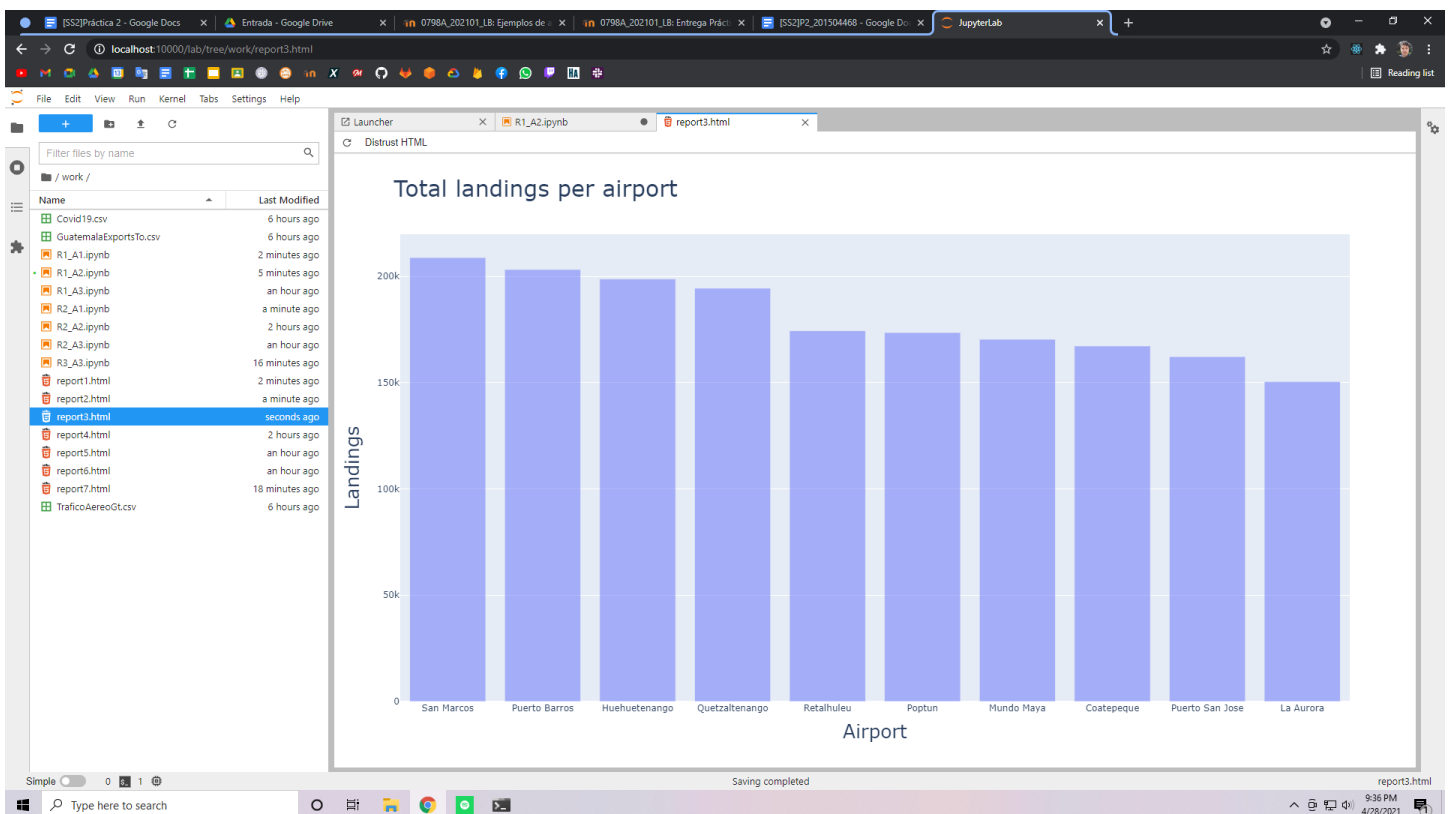
[3]: rdd_landings = rdd_landings.reduceByKey(lambda a,b: a+b)
print(rdd_landings.collect())

[4]: rdd_landings = spark_context.parallelize(rdd_landings.sortBy(lambda a: a[1], False).collect())
print(rdd_landings.collect())

[5]: airports_name = rdd_landings.map(lambda x: x[0])
airports_landings = rdd_landings.map(lambda x: x[1])
print(airports_name.collect())
print(airports_landings.collect())

[6]: fig = go.Figure(data=go.Bar(x=airports_name.collect(), y=airports_landings.collect()))
fig.update_layout(title="Total landings per airport", title_font_size=30,
yaxis=dict(title="Landings", title_font_size=25),
xaxis=dict(title="Airport", title_font_size=25))
fig.update_traces(overwrite=True, marker={"opacity": 0.5})
fig.write_html('report3.html', auto_open=True)

```



2. Gráfica de pie con los 3 meses con mayor número de pasajeros de salida.

The screenshot shows a JupyterLab interface with a file browser on the left and a code editor on the right. The file browser shows a directory structure with files like Covid19.csv, GuatemalaExportsTo.csv, and various R2_A1.ipynb, R2_A2.ipynb, R2_A3.ipynb files. The code editor shows the following code:

```
[1]: import findspark
findspark.init()
from pyspark import SparkContext
from pyspark.sql.session import SparkSession
spark_context = SparkContext("local", "first_app")
spark = SparkSession(spark_context)
import plotly.graph_objects as go
from pyspark.sql import SQLContext
sql_context = SQLContext(spark_context)

[2]: text_file = sql_context.read.format("com.databricks.spark.csv").options(header='true', inferschema='true', quote='', delimiter=',').load("/home/jovyan/work/TraficoAereoOt.csv")
rdd_filter = text_file.rdd.map(tuple)
rdd_landings = rdd_filter.map(lambda word: (word[0].split("/") [1], word[5]))
rdd_landings.take(5)

[3]: [ ('01', 195), ('02', 112), ('03', 159), ('04', 193), ('05', 82) ]

[4]: rdd_landings = rdd_landings.reduceByKey(lambda a,b: a+b)
print(rdd_landings.collect())

[5]: [ ('01', 6034), ('02', 5912), ('03', 5600), ('04', 5723), ('05', 5690), ('06', 5476), ('07', 5972), ('08', 5850), ('09', 5425), ('10', 5347), ('11', 6467), ('12', 5862) ]

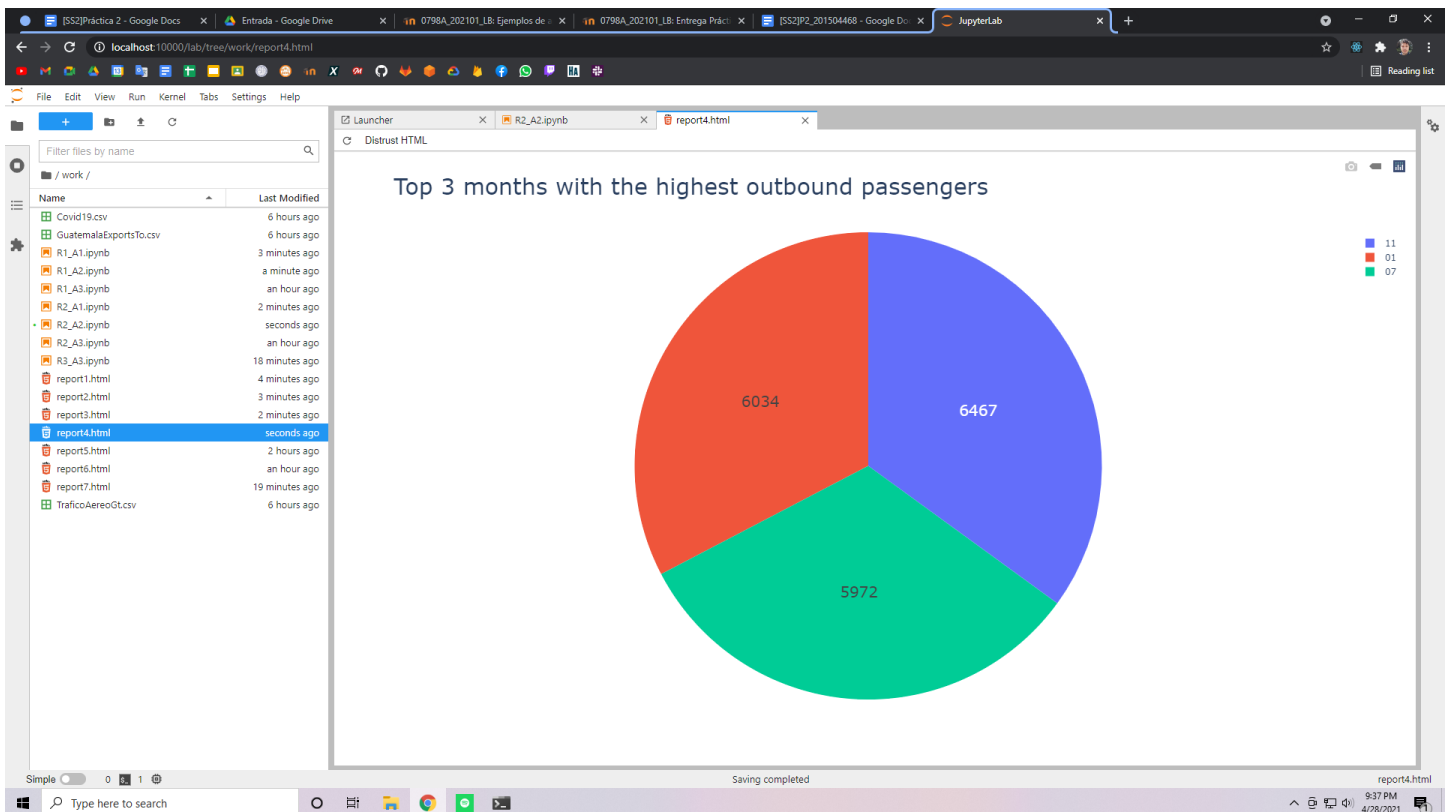
[6]: rdd_landings = spark_context.parallelize(rdd_landings.sortBy(lambda a: a[1], False).take(3))
print(rdd_landings.collect())

[7]: [ ('11', 6467), ('01', 6034), ('07', 5972) ]

[8]: landing_month = rdd_landings.map(lambda x: x[0])
landing_quantity = rdd_landings.map(lambda x: x[1])
print(landing_month.collect())
print(landing_quantity.collect())

[9]: [ '11', '01', '07' ]
[ 6467, 6034, 5972 ]

[10]: fig = go.Figure(data=go.Pie(labels=landing_month.collect(), values=landing_quantity.collect()))
fig.update_layout(title_text='Top 3 months with the highest outbound passengers', title_font_size=30)
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=20)
fig.write_html('report4.html', auto_open=True)
```



Archivo 3 - "Covid19"

1. Gráfica de barras con el número total de casos de Covid-19 en
 - a. Cuba
 - b. France
 - c. Canada
 - d. Singapore
 - e. South_Korea

```

[1]: import findspark
findspark.init()
from pyspark import SparkContext
from pyspark.sql.session import SparkSession
spark_context = SparkContext("local", "first_app")
spark = SparkSession(spark_context)
import plotly.graph_objects as go
from pyspark.sql import SQLContext
sql_context = SQLContext(spark_context)

[2]: text_file = sql_context.read.format("com.databricks.spark.csv").options(header="true", inferSchema="true", quote="").load("/home/jovyan/work/Covid19.csv")
rdd_filter = text_file.rdd.map(tuple)
rdd_covid = rdd_filter.map(lambda word: (word[7], word[5]))
rdd_covid.take(5)

[2]: [('Afghanistan', 0),
('Afghanistan', 125),
('Afghanistan', 47),
('Afghanistan', 0),
('Afghanistan', 17)]

[3]: rdd_covid = rdd_covid.reduceByKey(lambda a,b: a+b)
print(rdd_covid.top(3))

[('Vietnam', 716), ('Venezuela', 14263), ('Thailand', 1248)]

[4]: rdd_covid = rdd_covid.filter(lambda word: word[0] in ["Cuba", "France", "Canada", "Singapore", "South_Korea"])
print(rdd_covid.collect())

[('Canada', 143649), ('Cuba', 5091), ('France', 453763), ('Singapore', 57576), ('South_Korea', 23045)]

[5]: covid_country = rdd_covid.map(lambda x: x[0])
covid_cases = rdd_covid.map(lambda x: x[1])
print(covid_country.collect())
print(covid_cases.collect())

['Canada', 'Cuba', 'France', 'Singapore', 'South_Korea']
[143649, 5091, 453763, 57576, 23045]

[6]: fig = go.Figure(data=go.Bar(x=covid_country.collect(), y=covid_cases.collect()))
fig.update_layout(title_text="Covid cases per country", title_font_size=30,
axis=dict(title="Total", title_font_size=25),
axis=dict(title="Country", title_font_size=25))
fig.update_traces(overwrite=True, marker={"opacity": 0.5})
fig.write_html('report5.html', auto_open=True)

```



2. Gráfica de pie con los 5 meses con menos casos de Covid-19.

The screenshot shows a JupyterLab environment with a file browser on the left and a code editor on the right. The file browser displays a list of files in the '/work/' directory, including 'Covid19.csv', 'GuatemalaExportsTo.csv', and several '.ipynb' files. The code editor shows a Python script that uses PySpark to read a CSV file, filter data, and generate a bar chart titled 'Total cases per month'.

```
[1]: import findspark
findspark.init()
from pyspark import SparkContext
from pyspark.sql.session import SparkSession
spark_context = SparkContext("local", "first_app")
spark = SparkSession(spark_context)
import plotly.graph_objects as go
from pyspark.sql import SQLContext
sql_context = SQLContext(spark_context)

[2]: text_file = sql_context.read.format("com.databricks.spark.csv").options(header='true', inferschema='true', quote='', delimiter=',').load("/home/jovyan/work/Covid19.csv")
rdd_filter = text_file.rdd.map(tuple)
rdd_covid = rdd_filter.map(lambda word: (word[2], word[5]))
rdd_covid.take(5)

[2]: [(9, 0), (9, 125), (9, 47), (9, 0), (9, 17)]

[3]: rdd_covid = rdd_covid.reduceByKey(lambda a,b: a+b)
print(rdd_covid.collect())

[(9, 2374517), (8, 3584040), (7, 3288723), (6, 2527885), (5, 1555025), (3, 324799), (4, 868927), (2, 73635), (1, 9758)]

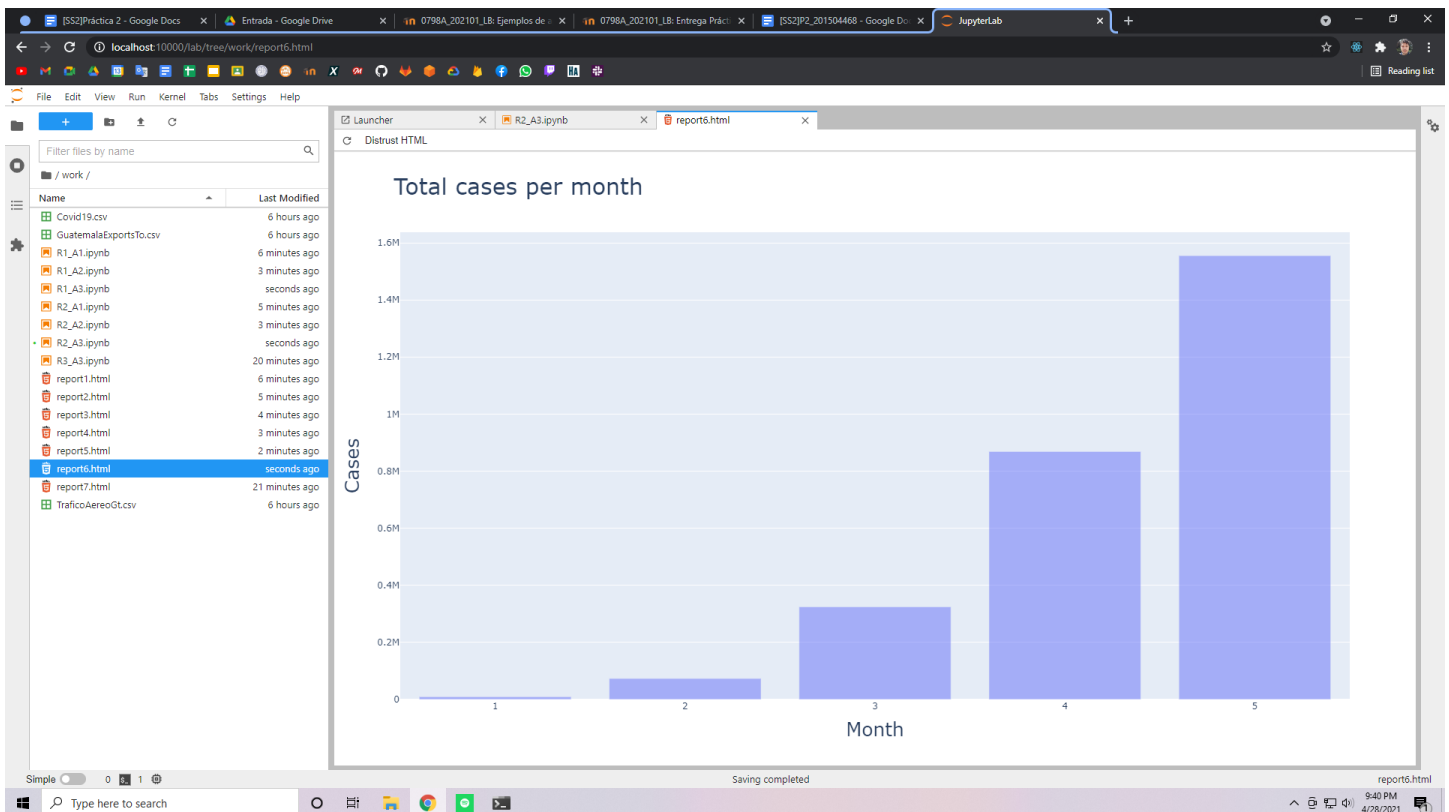
[4]: rdd_covid = spark_context.parallelize(rdd_covid.sortBy(lambda a: a[1], True).take(5))
print(rdd_covid.collect())

[(1, 9758), (2, 73635), (3, 324799), (4, 868927), (5, 1555025)]

[5]: covid_month = rdd_covid.map(lambda x: x[0])
covid_cases = rdd_covid.map(lambda x: x[1])
print(covid_month.collect())
print(covid_cases.collect())

[1, 2, 3, 4, 5]
[9758, 73635, 324799, 868927, 1555025]

[6]: fig = go.Figure(data=go.Bar(x=covid_month.collect(), y=covid_cases.collect()))
fig.update_layout(title_text='Total cases per month', title_font_size=30,
                    yaxis=dict(title='Cases', title_font_size=25),
                    xaxis=dict(title='Month', title_font_size=25))
fig.update_traces(overwrite=True, marker=dict(opacity=0.5))
fig.write_html("report6.html", auto_open=True)
```



3. Gráfica a su elección del total de casos y muertes en Guatemala en el mes de Agosto.

The screenshot shows a JupyterLab environment with a file browser on the left and a code editor on the right. The file browser lists files like Covid19.csv, GuatemalaExportsTo.csv, and various report files. The code editor contains a Python script that uses PySpark to process a CSV file, filter data for Guatemala, and generate a bar chart titled 'Covid-19 cases and deaths in Guatemala in august'.

```
[1]: import findspark
findspark.init()
from pyspark import SparkContext
from pyspark.sql.session import SparkSession
spark_context = SparkContext("local", "first_app")
spark = SparkSession(spark_context)
import plotly.graph_objects as go
from pyspark.sql import SQLContext
sql_context = SQLContext(spark_context)

[2]: text_file = sql_context.read.format("com.databricks.spark.csv").options(header='true', inferschema='true', quote='', delimiter=',').load("/home/jovyan/work/Covid19.csv")
rdd_filter = text_file.rdd.map(tuple)
rdd_covid = rdd_filter.map(lambda word: (word[7], (word[5], word[6], word[2])))
rdd_covid = rdd_covid.filter(lambda word: word[1][2] == 8)
rdd_covid = rdd_covid.map(lambda word: (word[0], (word[1][0], word[1][1])))
rdd_covid.take(5)

[2]: [('Afghanistan', (19, 0)),
('afghanistan', (3, 0)),
('Afghanistan', (11, 1)),
('Afghanistan', (3, 0)),
('Afghanistan', (55, 4))]

[3]: rdd_covid = rdd_covid.reduceByKey(lambda a,b: (a[0]+b[0],a[1]+b[1]))
print(rdd_covid.take(5))

[3]: [('Afghanistan', (1620, 131)), ('Argentina', (222243, 5090)), ('Armenia', (5200, 139)), ('Aruba', (1877, 7)), ('Australia', (9367, 422))]

[4]: rdd_covid = rdd_covid.filter(lambda word: word[0] in ["Guatemala"])
print(rdd_covid.collect())

[4]: [('Guatemala', (25086, 873))]

[5]: covid_cases = rdd_covid.map(lambda x: x[1][0])
covid_deaths = rdd_covid.map(lambda x: x[1][1])
covid_data = [covid_cases.collect()[0], covid_deaths.collect()[0]]
covid_data

[5]: [25086, 873]

[6]: fig = go.Figure(data=go.Bar(x=["Cases", "Deaths"], y=covid_data))
fig.update_layout(title_text="Covid-19 cases and deaths in Guatemala in august", title_font_size=30,
yaxis=dict(title="Total", title_font_size=25),
xaxis=dict(title="Type", title_font_size=25))
fig.update_traces(overwrite=True, marker=dict(opacity=0.5))
fig.write_html('report7.html', auto_open=True)
```

