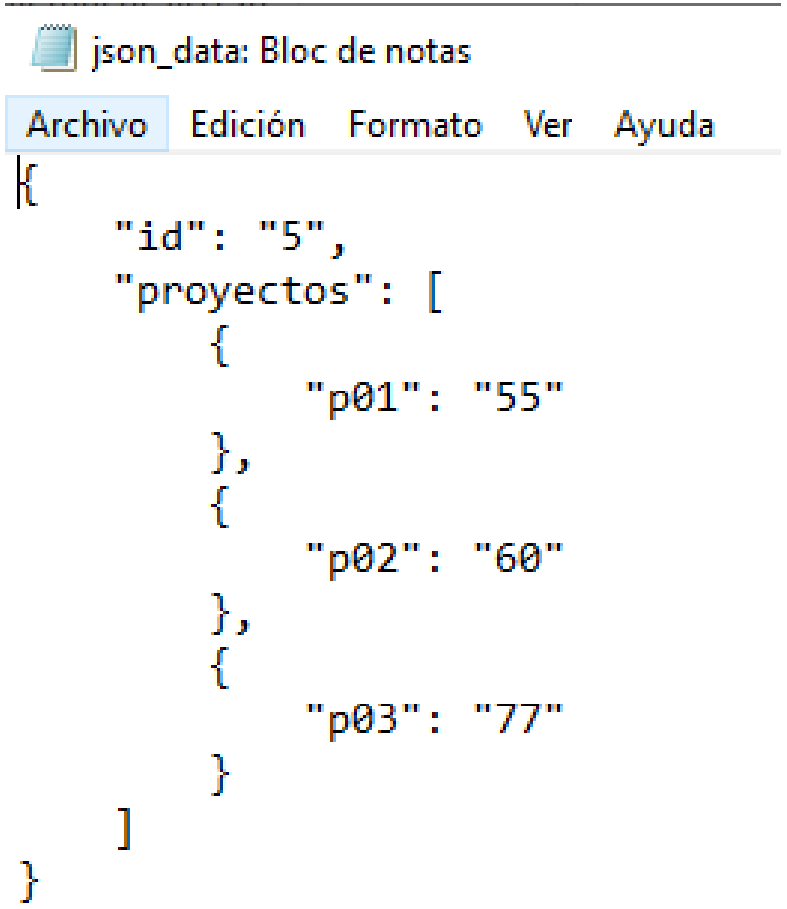
¿Qué es JSON?

Para responder qué es **JSON**, debemos empezar por decir que sus siglas en inglés son por JavaScript Object Notation. Se trata de un formato para guardar e intercambiar información que cualquier persona pueda leer. Los archivos json contienen solo texto y usan la extensión **.json**.



```python
from pyspark.sql.types import StructField,StringType,IntegerType,StructType
```

```python
data_schema = [StructField("age", IntegerType(), True),StructField("name", StringType(), True)]
```

```python
final_struc = StructType(fields=data_schema)
```

```python
df = spark.read.json('people.json', schema=final_struc)
```

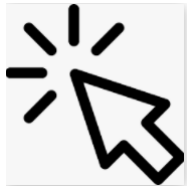
```python
df.printSchema()
```

```
root
 |-- age: integer (nullable = true)
 |-- name: string (nullable = true)
```

C: > LuisAlex > CURSO_BigData > Lec2 > {} people.json > ...

```
1   {"name":"Michael"}
2   {"name":"Andy", "age":30}
3   {"name":"Justin", "age":19}
4
```

pyspark.sql.DataFrameStatFunctions



approxQuantile(col, probabilities, relativeError)	Calculates the approximate quantiles of numerical columns of a **DataFrame**.
corr(col1, col2[, method])	Calculates the correlation of two columns of a **DataFrame** as a double value.
cov(col1, col2)	Calculate the sample covariance for the given columns, specified by their names, as a double value.
crosstab(col1, col2)	Computes a pair-wise frequency table of the given columns.
freqItems(cols[, support])	Finding frequent items for columns, possibly with false positives.
sampleBy(col, fractions[, seed])	Returns a stratified sample without replacement based on the fraction given on each stratum.

pyspark.sql.DataFrameWriter.csv¶

pyspark.sql.DataFrameWriter.csv¶

DataFrameWriter.**CSV**(*path*, *mode=None*, *compression=None*, *sep=None*, *quote=None*, *escape=None*, *header=None*, *nullValue=None*, *escapeQuotes=None*, *quoteAll=None*, *dateFormat=None*, *timestampFormat=None*, *ignoreLeadingWhiteSpace=None*, *ignoreTrailingWhiteSpace=None*, *charToEscapeQuoteEscaping=None*, *encoding=None*, *emptyValue=None*, *lineSep=None*) [source]

Saves the content of the DataFrame in CSV format at the specified path.

New in version 2.0.0.

```
from pyspark.sql.functions import format_number
```

```
sales_std = df.select(stddev("Sales").alias('std'))
```

```
sales_std.show()
```

```
+------------------+
|               std|
+------------------+
|250.08742410799007|
+------------------+
```

```
sales_std.select(format_number('std',2)).show()
```

```
+----------------------+
|format_number(std, 2)|
+----------------------+
|                250.09|
+----------------------+
```