Lectura de ficheros json con pyspark

Spark ofrece 2 alternativas para la carga de ficheros json:

1.cargar los datos como ficheros de texto y luego convertirlos a la estructura de datos deseada,mediante un analizador sintáctico de json

2.usar el módulo sql que ofrece spark.el módulo sql de spark tiene una función **read.json** para cargar un fichero en este formato.Esta opción aporta más flexibilidad ya que permite buscar por atributos concretos y realizar consultas SQL sobre el conjunto de datos.

```
from pyspark.sql import SQLContext
sqlCtx = SQLContext(sc)
data = sqlCtx.read.json("data.json")
```

Spark SQL puede deducir automáticamente el esquema de un conjunto de datos JSON y cargarlo como un DataFrame. Esta conversión se puede realizar utilizando SQLContext.read.json en un archivo JSON.

Tenga en cuenta que el archivo que se ofrece como un archivo json no es un archivo JSON típico. Cada línea debe contener un objeto JSON válido.

```
# sc is an existing SparkContext.
from pyspark.sql import SQLContext
sqlContext = SQLContext(sc)

# A JSON dataset is pointed to by path.
# The path can be either a single text file or a directory storing text files.
people = sqlContext.read.json("examples/src/main/resources/people.json")

# The inferred schema can be visualized using the printSchema() method.
people.printSchema()
# root
#  |-- age: integer (nullable = true)
#  |-- name: string (nullable = true)

# Register this DataFrame as a table.
people.registerTempTable("people")

# SQL statements can be run by using the sql methods provided by `sqlContext`.
teenagers = sqlContext.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")

# Alternatively, a DataFrame can be created for a JSON dataset represented by
# an RDD[String] storing one JSON object per string.
anotherPeopleRDD = sc.parallelize([
  '{"name":"Yin","address":{"city":"Columbus","state":"Ohio"}}'])
anotherPeople = sqlContext.jsonRDD(anotherPeopleRDD)
```