

Spark Streaming: Ejemplo con Twint [2/2]

Para la realización de esta actividad será necesario lanzar el script anterior (Tema_107_Spark_streaming) para poder realizar la conexión.

Importamos las dependencias

```
In [1]: from pyspark import SparkConf, SparkContext
from pyspark.streaming import StreamingContext
```

Creamos el contexto de streaming

```
In [2]: # Crear la configuración de Spark
conf = SparkConf()
conf.setAppName("TwitterStreamingApp")

# Creamos el contexto con la configuración anterior:
sc = SparkContext(conf=conf)
sc.setLogLevel("WARN")

# Creamos el contexto de Streaming desde el contexto de spark visto arriba
# con intervalo de tiempo cada 2 segundos.
ssc = StreamingContext(sc, 2)

# establece un punto de control para permitir la recuperación de RDD
ssc.checkpoint("checkpoint_TwitterAPP")

dataStream = ssc.socketTextStream("localhost", 9009)
```

```
In [3]: # recogida de la información
dataStream.pprint()

# comienza la computación streaming
ssc.start()

## espera que la transmisión termine
ssc.awaitTermination()
ssc.stop()
```

Time: 2022-11-30 10:47:26

Time: 2022-11-30 10:47:28

```
{"tweet":"Las cuentas de Argentina para clasificarse para octavos ¿qué n  
ecesita el equipo de Messi para seguir adelante en el Mundial de Qatar?  
https://t.co/Nn2Aq3mSSB 🤖 @ngomez_7 https://t.co/Uqv1Jys4V3"}  
{"tweet":"Risto Mejide, sin palabras al recibir la sorpresa más inespera  
da por su cumpleaños en 'Got Talent' https://t.co/VBNC7AXCjK"}  
{"tweet":"\"Barrilete\", \"vago grotesco\" o \"engendro\" los insultos e  
n Twitter por los que Cristina Seguí tendrá pagar 6.000 euros al exminis  
tro socialista y diputado en el Congreso José Luis Ábalos https://t.co/p  
i8DzMoX62"}  
{"tweet":"Citan a declarar a Quequé por posible delito contra los sentim  
ientos religiosos https://t.co/kwYKLcHtv6"}  
{"tweet":"Tamara Gorro presenta 'La princesa y la mariposa', un cuento b  
enéfico para que los niños entiendan la pérdida y el duelo https://t.co/  
aSRcCQy93H"}  
{"tweet":"Albares, condecorado por el Gobierno de Zelenski con la medall  
a 'Yaroslav, el Sabio' por \"su apoyo y defensa a Ucrania\" https://t.c  
o/uG0UXw7rIj"}  
{"tweet":"Al menos 42 condenados se han beneficiado de la rebaja de pena  
s de ley del 'sólo sí es sí' y 10 han sido excarcelados https://t.co/wS2  
ETZ79Dw"}  
{"tweet":"Desabastecimiento en España de amoxicilina pediátrica 📌 Adv  
ertencia de la Agencia Española del Medicamento https://t.co/xVwPtwDVi  
h"}  
{"tweet":"Cómo se hace la maniobra de Heimlich los pasos que pueden salv  
ar la vida en caso de atragantamiento https://t.co/eiirEzzhil"}  
{"tweet":"Desarticulada una banda dedicada al tráfico de cocaína rosa, l  
a nueva droga de los ricos https://t.co/6YsG0rw39K"}  
...
```

Time: 2022-11-30 10:47:30

Time: 2022-11-30 10:47:32

Time: 2022-11-30 10:47:34

Time: 2022-11-30 10:47:36

Time: 2022-11-30 10:47:38

Time: 2022-11-30 10:47:40

Time: 2022-11-30 10:47:42

Time: 2022-11-30 10:47:44

Time: 2022-11-30 10:47:46

Time: 2022-11-30 10:47:48

Time: 2022-11-30 10:47:50

Time: 2022-11-30 10:47:52

Time: 2022-11-30 10:47:54

Time: 2022-11-30 10:47:56

Time: 2022-11-30 10:47:58

Time: 2022-11-30 10:48:00

Time: 2022-11-30 10:48:02

Time: 2022-11-30 10:48:04

Time: 2022-11-30 10:48:06

Time: 2022-11-30 10:48:08

Time: 2022-11-30 10:48:10

```
ERROR:root:KeyboardInterrupt while sending command.  
Traceback (most recent call last):  
  File "/usr/local/spark/python/lib/py4j-0.10.9.5-src.zip/py4j/java_gateway.py", line 1038, in send_command  
    response = connection.send_command(command)  
  File "/usr/local/spark/python/lib/py4j-0.10.9.5-src.zip/py4j/clientserver.py", line 511, in send_command  
    answer = smart_decode(self.stream.readline()[:-1])  
  File "/opt/conda/lib/python3.10/socket.py", line 705, in readinto  
    return self._sock.recv_into(b)  
KeyboardInterrupt
```

```

-----
KeyboardInterrupt                                Traceback (most recent call last)
Cell In [3], line 8
      5 ssc.start()
      7 ## espera que la transmisión termine
----> 8 ssc.awaitTermination()
      9 ssc.stop()

File /usr/local/spark/python/pyspark/streaming/context.py:227, in StreamingContext.awaitTermination(self, timeout)
    218 """
    219 Wait for the execution to stop.
    220
    (...)
    224 time to wait in seconds
    225 """
    226 if timeout is None:
-> 227     self._jssc.awaitTermination()
    228 else:
    229     self._jssc.awaitTerminationOrTimeout(int(timeout * 1000))

File /usr/local/spark/python/lib/py4j-0.10.9.5-src.zip/py4j/java_gateway.py:1320, in JavaMember.__call__(self, *args)
    1313 args_command, temp_args = self._build_args(*args)
    1315 command = proto.CALL_COMMAND_NAME + \
    1316     self.command_header + \
    1317     args_command + \
    1318     proto.END_COMMAND_PART
-> 1320 answer = self.gateway_client.send_command(command)
    1321 return_value = get_return_value(
    1322     answer, self.gateway_client, self.target_id, self.name)
    1324 for temp_arg in temp_args:

File /usr/local/spark/python/lib/py4j-0.10.9.5-src.zip/py4j/java_gateway.py:1038, in GatewayClient.send_command(self, command, retry, binary)
    1036 connection = self._get_connection()
    1037 try:
-> 1038     response = connection.send_command(command)
    1039     if binary:
    1040         return response, self._create_connection_guard(connection)

File /usr/local/spark/python/lib/py4j-0.10.9.5-src.zip/py4j/clientserver.py:511, in ClientServerConnection.send_command(self, command)
    509 try:
    510     while True:
-> 511         answer = smart_decode(self.stream.readline()[:-1])
    512         logger.debug("Answer received: {0}".format(answer))
    513         # Happens when a the other end is dead. There might be a
n empty
    514         # answer before the socket raises an error.

File /opt/conda/lib/python3.10/socket.py:705, in SocketIO.readinto(self, b)
    703 while True:
    704     try:
-> 705         return self._sock.recv_into(b)
    706     except timeout:

```

```
707         self._timeout_occurred = True
```

```
KeyboardInterrupt:
```