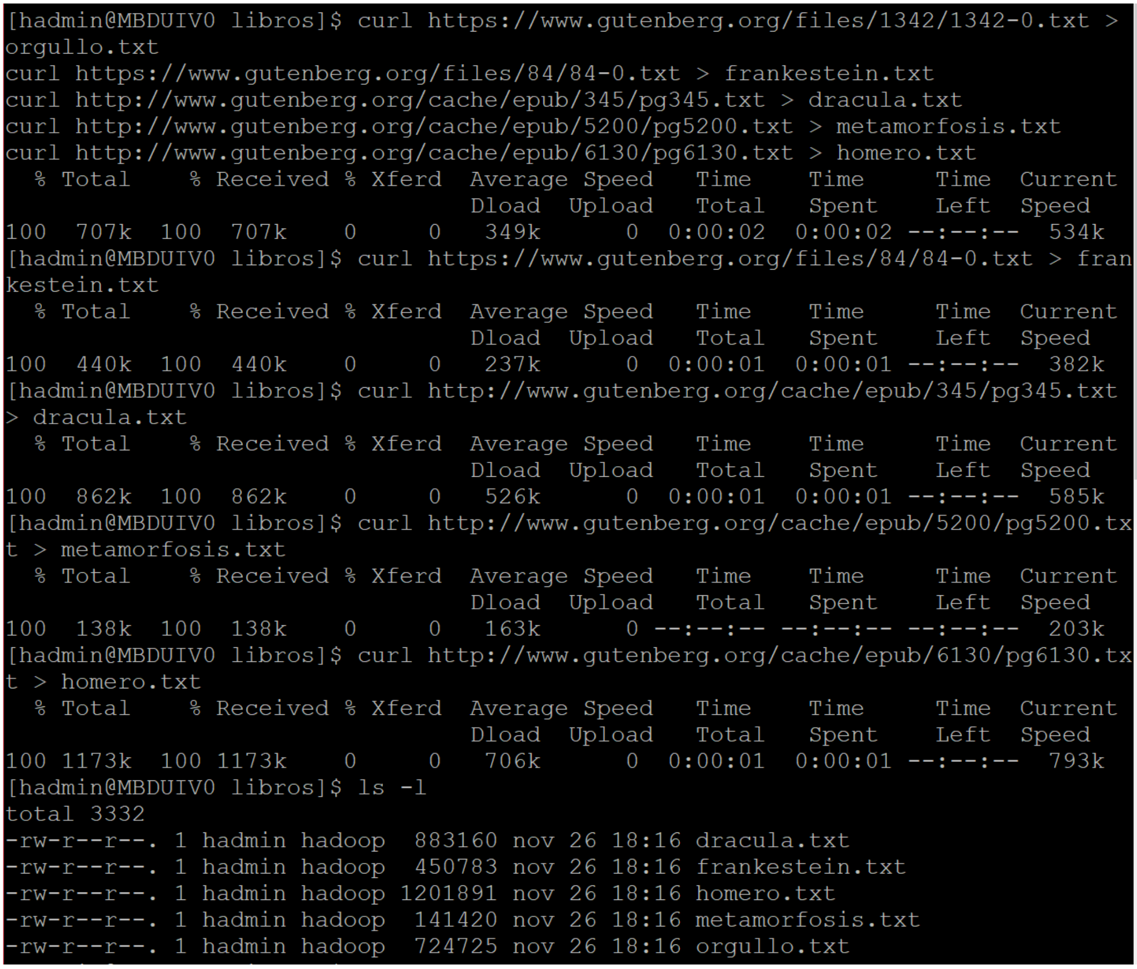
Preparación

Creamos varios archivos desde la terminal

```
mkdir libros
cd libros
curl https://www.gutenberg.org/files/1342/1342-0.txt > orgullo.txt
curl https://www.gutenberg.org/files/84/84-0.txt > frankestein.txt
curl http://www.gutenberg.org/cache/epub/345/pg345.txt > dracula.txt
curl http://www.gutenberg.org/cache/epub/5200/pg5200.txt >
metamorfosis.txt
curl http://www.gutenberg.org/files/epub/6130/pg6130.txt > homero.txt
```



```
[hadmin@MBDUIV0 libros]$ curl https://www.gutenberg.org/files/1342/1342-0.txt >
orgullo.txt
curl https://www.gutenberg.org/files/84/84-0.txt > frankestein.txt
curl http://www.gutenberg.org/cache/epub/345/pg345.txt > dracula.txt
curl http://www.gutenberg.org/cache/epub/5200/pg5200.txt > metamorfosis.txt
curl http://www.gutenberg.org/cache/epub/6130/pg6130.txt > homero.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  707k  100  707k    0     0   349k      0  0:00:02  0:00:02 --:--:--  534k
[hadmin@MBDUIV0 libros]$ curl https://www.gutenberg.org/files/84/84-0.txt > fran
kestein.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  440k  100  440k    0     0   237k      0  0:00:01  0:00:01 --:--:--  382k
[hadmin@MBDUIV0 libros]$ curl http://www.gutenberg.org/cache/epub/345/pg345.txt
> dracula.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  862k  100  862k    0     0   526k      0  0:00:01  0:00:01 --:--:--  585k
[hadmin@MBDUIV0 libros]$ curl http://www.gutenberg.org/cache/epub/5200/pg5200.tx
t > metamorfosis.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  138k  100  138k    0     0   163k      0 --:--:-- --:--:-- --:--:--  203k
[hadmin@MBDUIV0 libros]$ curl http://www.gutenberg.org/cache/epub/6130/pg6130.tx
t > homero.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1173k  100 1173k    0     0   706k      0  0:00:01  0:00:01 --:--:--  793k
[hadmin@MBDUIV0 libros]$ ls -l
total 3332
-rw-r--r--. 1 hadmin hadoop  883160 nov 26 18:16 dracula.txt
-rw-r--r--. 1 hadmin hadoop  450783 nov 26 18:16 frankestein.txt
-rw-r--r--. 1 hadmin hadoop 1201891 nov 26 18:16 homero.txt
-rw-r--r--. 1 hadmin hadoop  141420 nov 26 18:16 metamorfosis.txt
-rw-r--r--. 1 hadmin hadoop  724725 nov 26 18:16 orgullo.txt
```

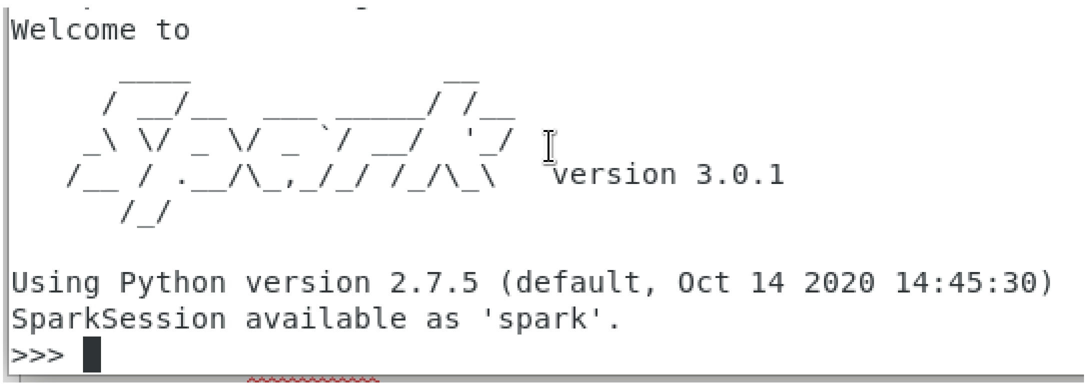
Nota: es importante comprobar que los libros se hayan descargado correctamente. Hay veces que el Proyecto Gutenberg da fallos al descargar (ej. URLs que dejan de existir, libros que se descargan como un html, en binario, etc).

Arrancar pyspark

Ejecutamos pyspark

```
[moranjesus@localhost libros]$ pyspark
```

Tendremos



```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.0.1
      /_/

Using Python version 2.7.5 (default, Oct 14 2020 14:45:30)
SparkSession available as 'spark'.
>>> 
```

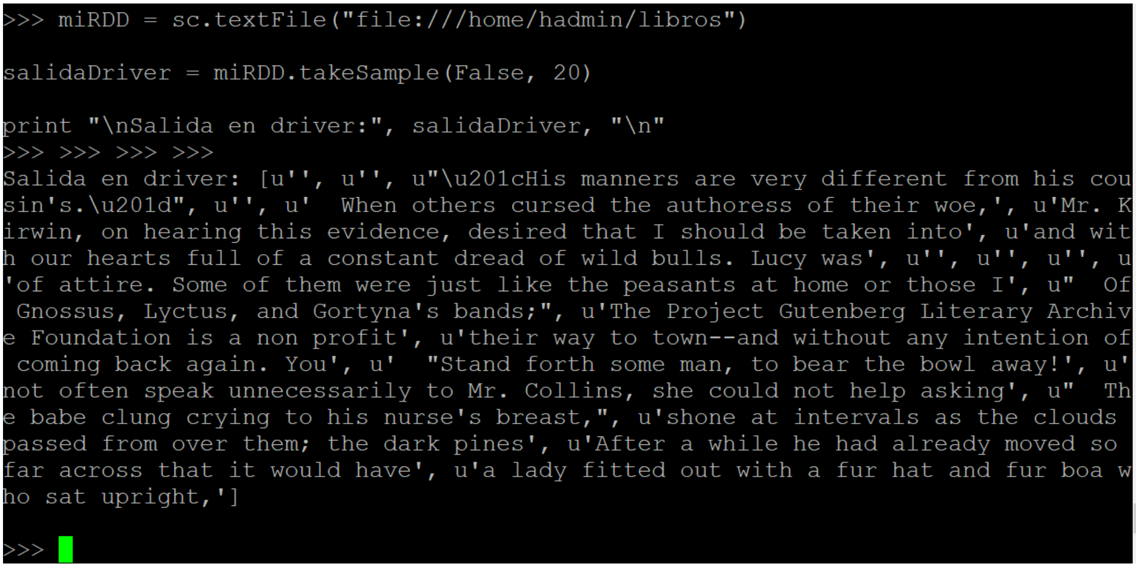
Lectura de archivos

Tendremos que utilizar el objeto SparkContext (sc en pyspark)

sc.textfile

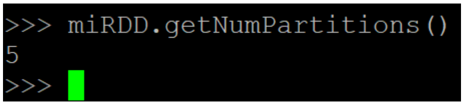
Leemos línea a línea todos los libros

```
miRDD = sc.textFile("file:///home/hadmin/libros")

salidaDriver = miRDD.takeSample(False, 20)

print "\nSalida en driver:", salidaDriver, "\n"
```



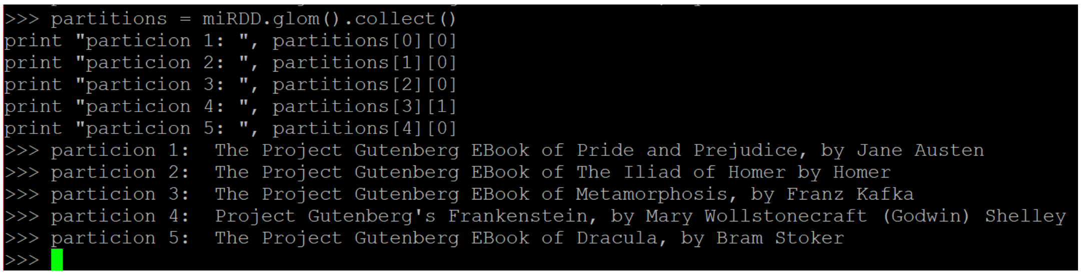
```
>>> miRDD = sc.textFile("file:///home/hadmin/libros")

salidaDriver = miRDD.takeSample(False, 20)

print "\nSalida en driver:", salidaDriver, "\n"
>>> >>> >>> >>>
Salida en driver: [u'', u'', u"\u201cHis manners are very different from his cou
sin's.\u201d", u'', u'  When others cursed the authoress of their woe,', u'Mr. K
irwin, on hearing this evidence, desired that I should be taken into', u'and wit
h our hearts full of a constant dread of wild bulls. Lucy was', u'', u'', u'', u
'of attire. Some of them were just like the peasants at home or those I', u"  Of
 Gnossus, Lyctus, and Gortyna's bands;", u'The Project Gutenberg Literary Archiv
e Foundation is a non profit', u'their way to town--and without any intention of
 coming back again. You', u'  "Stand forth some man, to bear the bowl away!', u'
not often speak unnecessarily to Mr. Collins, she could not help asking', u"  Th
e babe clung crying to his nurse's breast,", u'shone at intervals as the clouds
passed from over them; the dark pines', u'After a while he had already moved so
far across that it would have', u'a lady fitted out with a fur hat and fur boa w
ho sat upright,']

>>> 
```

```
miRDD.getNumPartitions()
```



```
>>> miRDD.getNumPartitions()
5
>>> 
```

```
partitions = miRDD.glom().collect()
print "particion 1: ", partitions[0][0]
print "particion 2: ", partitions[1][0]
print "particion 3: ", partitions[2][0]
print "particion 4: ", partitions[3][1]
print "particion 5: ", partitions[4][0]
```

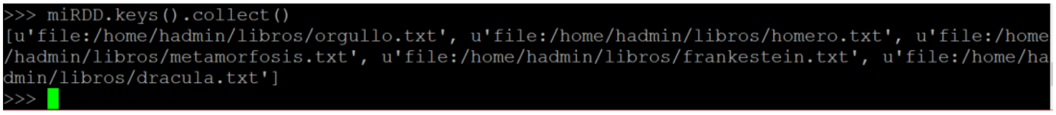


```
>>> partitions = miRDD.glom().collect()
print "particion 1: ", partitions[0][0]
print "particion 2: ", partitions[1][0]
print "particion 3: ", partitions[2][0]
print "particion 4: ", partitions[3][1]
print "particion 5: ", partitions[4][0]
>>> particion 1:  The Project Gutenberg EBook of Pride and Prejudice, by Jane Austen
>>> particion 2:  The Project Gutenberg EBook of The Iliad of Homer by Homer
>>> particion 3:  The Project Gutenberg EBook of Metamorphosis, by Franz Kafka
>>> particion 4:  Project Gutenberg's Frankenstein, by Mary Wollstonecraft (Godwin) Shelley
>>> particion 5:  The Project Gutenberg EBook of Dracula, by Bram Stoker
>>>
```

sc.wholeTextfile

Leemos todos los libros teniendo como clave el libro (archivo) y como valor todos los datos. Ojo, no es <nombre libro, línea libro>. Lo que hace es <nombre libro, libro>

```
miRDD = sc.wholeTextFiles("file:///home/hadmin/libros")

salidaDriver = miRDD.takeSample(False, 20)

print "\nSalida en driver:", salidaDriver, "\n"

miRDD.keys().collect()
```

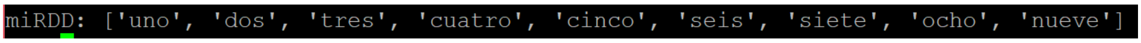


```
>>> miRDD.keys().collect()
[u'file:/home/hadmin/libros/orgullo.txt', u'file:/home/hadmin/libros/homero.txt', u'file:/home
/hadmin/libros/metamorfosis.txt', u'file:/home/hadmin/libros/frankestein.txt', u'file:/home/ha
dmin/libros/dracula.txt']
>>>
```

sc.parallelize

Cargamos unos datos de prueba

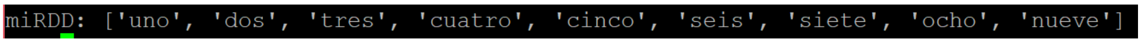
```
miRDD = sc.parallelize(["uno", "dos", "tres",
                        "cuatro", "cinco", "seis",
                        "siete", "ocho", "nueve"])
print "\nmiRDD:", miRDD.collect()
```



```
miRDD: ['uno', 'dos', 'tres', 'cuatro', 'cinco', 'seis', 'siete', 'ocho', 'nueve']
```

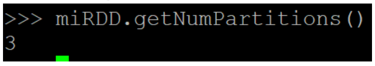
sc.parallelize con número de particiones

```
miRDD = sc.parallelize(["uno", "dos", "tres",
                        "cuatro", "cinco", "seis",
                        "siete", "ocho", "nueve"], 3)
print "\nmiRDD:", miRDD.collect()
```



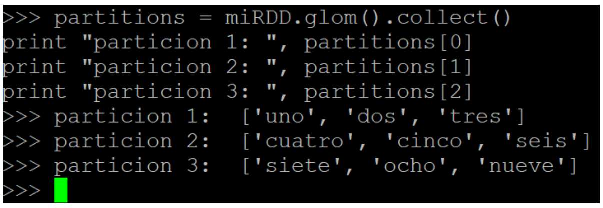
```
miRDD: ['uno', 'dos', 'tres', 'cuatro', 'cinco', 'seis', 'siete', 'ocho', 'nueve']
```

```
miRDD.getNumPartitions()
```



```
>>> miRDD.getNumPartitions()
3
```

```
partitions = miRDD.glom().collect()
print "particion 1: ", partitions[0]
print "particion 2: ", partitions[1]
print "particion 3: ", partitions[2]
```

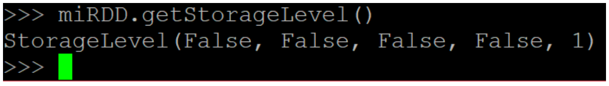


```
>>> partitions = miRDD.glom().collect()
print "particion 1: ", partitions[0]
print "particion 2: ", partitions[1]
print "particion 3: ", partitions[2]
>>> particion 1:  ['uno', 'dos', 'tres']
>>> particion 2:  ['cuatro', 'cinco', 'seis']
>>> particion 3:  ['siete', 'ocho', 'nueve']
>>>
```

Cache

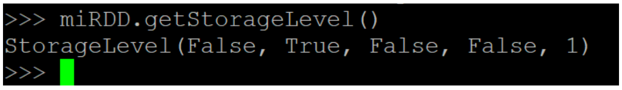
Cacheamos el RDD en memoria cuando se ejecute la primera acción

```
miRDD = sc.parallelize(["uno", "dos", "tres",
                        "cuatro", "cinco", "seis",
                        "siete", "ocho", "nueve"], 3)
miRDD.getStorageLevel()
```



```
>>> miRDD.getStorageLevel()
StorageLevel(False, False, False, False, 1)
>>>
```

```
miRDD.cache()

miRDD.getStorageLevel()
```

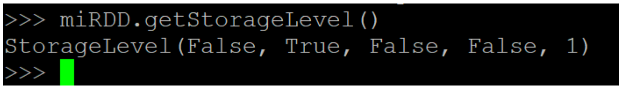


```
>>> miRDD.getStorageLevel()
StorageLevel(False, True, False, False, 1)
>>>
```

unpersist

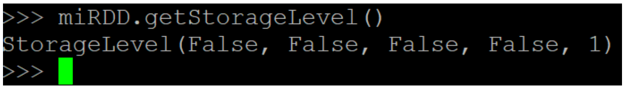
Quitamos el dataset del nivel de cache/persistencia que tenga

```
miRDD = sc.parallelize(["uno", "dos", "tres",
                        "cuatro", "cinco", "seis",
                        "siete", "ocho", "nueve"], 3)
miRDD.cache()

miRDD.getStorageLevel()
```



```
>>> miRDD.getStorageLevel()
StorageLevel(False, True, False, False, 1)
>>>
```

```
miRDD.unpersist()

miRDD.getStorageLevel()
```

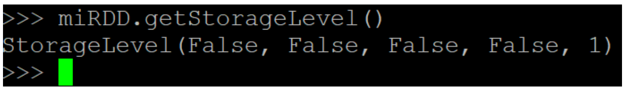


```
>>> miRDD.getStorageLevel()
StorageLevel(False, False, False, False, 1)
>>>
```

persist

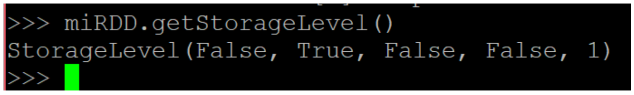
Cacheamos el RDD en memoria cuando se ejecute la primera acción

```
miRDD = sc.parallelize(["uno", "dos", "tres",
                        "cuatro", "cinco", "seis",
                        "siete", "ocho", "nueve"], 3)
miRDD.getStorageLevel()
```



```
>>> miRDD.getStorageLevel()
StorageLevel(False, False, False, False, 1)
>>> 
```

```
from pyspark.storagelevel import StorageLevel
miRDD.persist(StorageLevel.MEMORY_ONLY)

miRDD.getStorageLevel()
```



```
>>> miRDD.getStorageLevel()
StorageLevel(False, True, False, False, 1)
>>> 
```