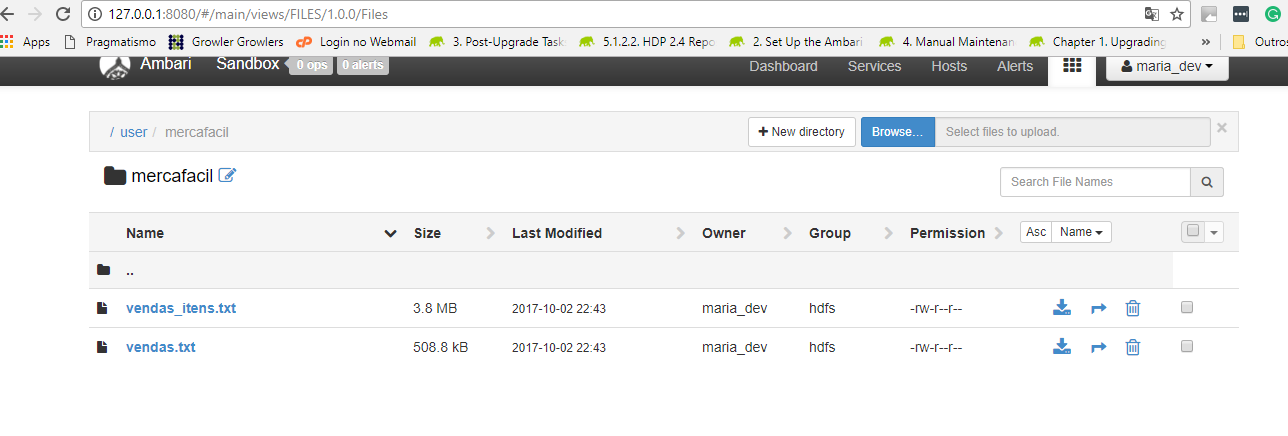
# Ambiente

* CentOS 6.7
* Horton Works HDP 2.4
* Scala 2.10
* Spark 1.6
* MongoDB 3.4
* Spark-mongo 1.1.0

# 1 Carregamento dos arquivos CSV para o HDFS

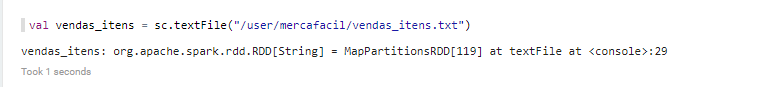
1. Usando o Ambari para carregar



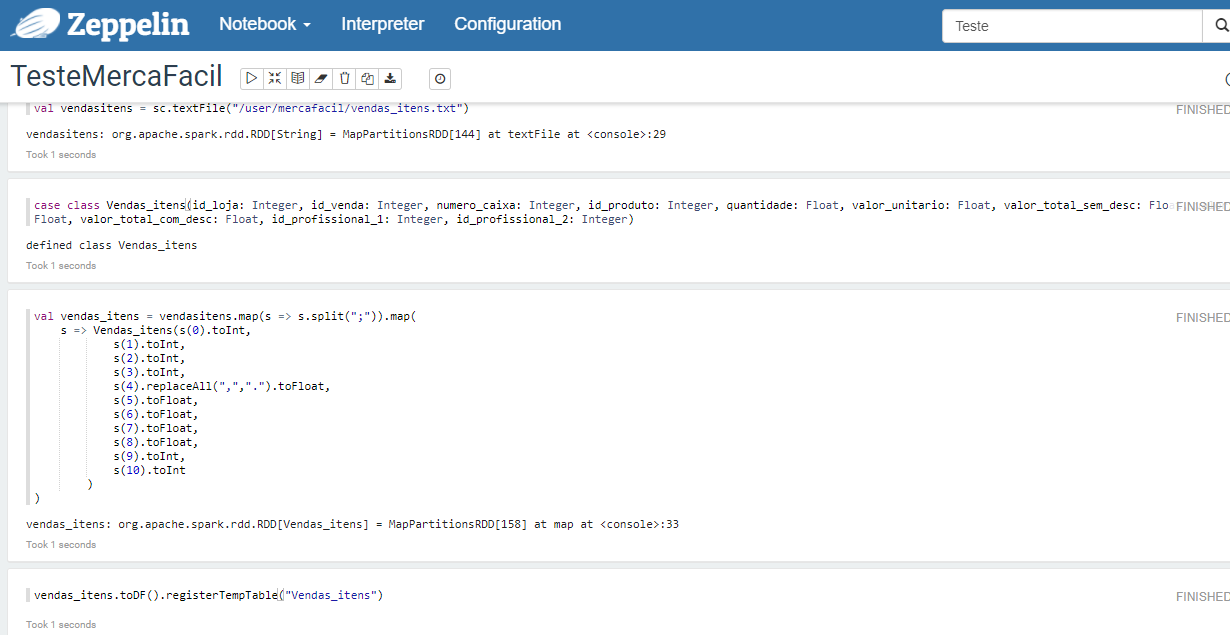
1. Via linha de comando
   1. hadoop fs -mkdir /user/mercafacil
   2. hadoop fs -put /tmp/vendas.txt /user/mercafacil
   3. hadoop fs -put /tmp/vendas\_itens.txt /user/mercafacil
   4. hadoop fs -ls /user

2 Leitura dos arquivos CSV do HDFS para o Spark usando Zeppelin

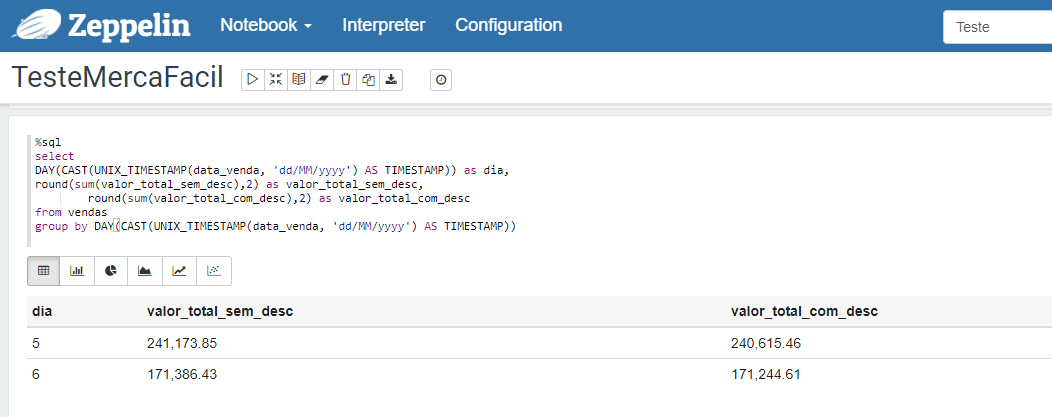
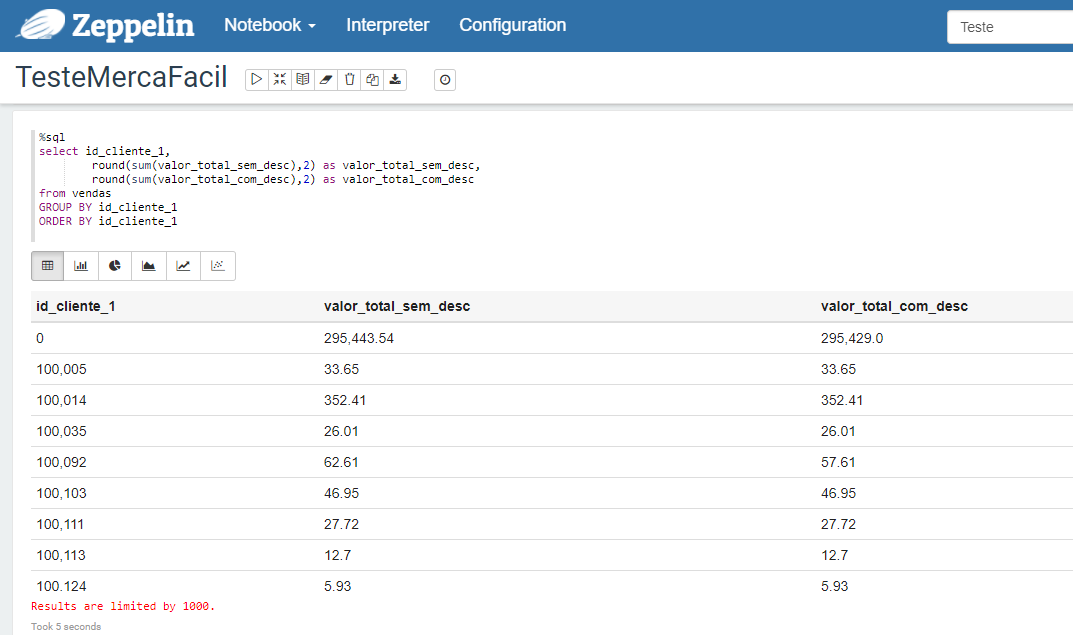
1. Criado os SQLContext para carregar os arquivos para o Spark

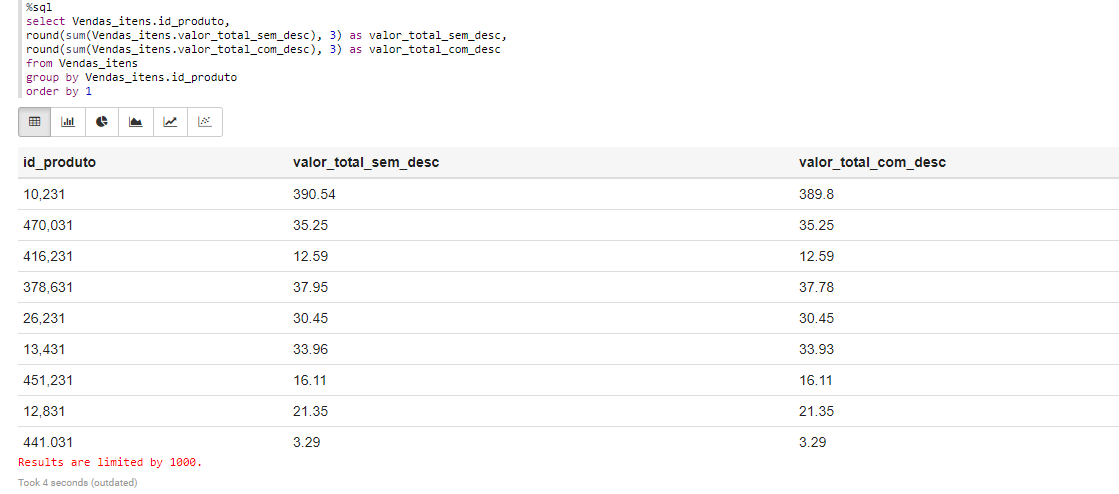


1. Construído a estrutura dos DataFrames e armazenado em tabelas temporárias para futura consulta



1. Criado as consultas em SQL para validar os resultados.





# 3 Instalado o MongoDB

1. Instalação do MongoDB

vi /etc/yum.repos.d/mongodb-org-3.4.repo

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/6/mongodb-org/3.4/x86\_64/

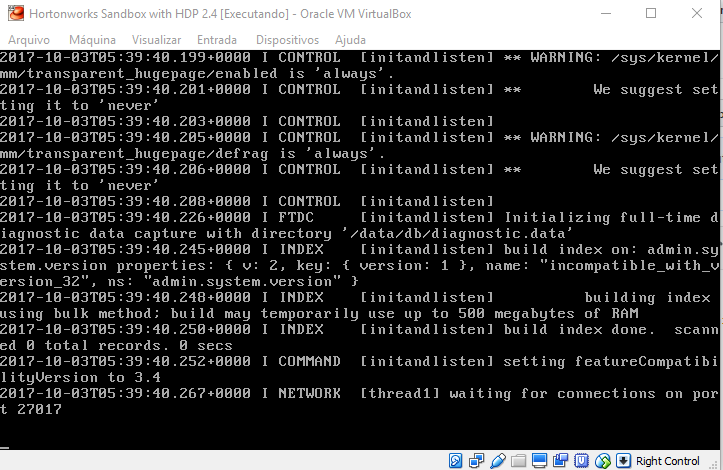
enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

Mkdir /data/db

service mongod start

1. Instancia do mongod escutando na porta 27017



# 4 Scala

1. Habilitar o Spark acessar o MongoDB usando o Scala

./bin/spark-shell –conf "spark.mongodb.output.uri=mongodb://127.0.0.1/results.collection" --packages org.mongodb.spark:mongo-spark-connector\_2.10:1.1.0

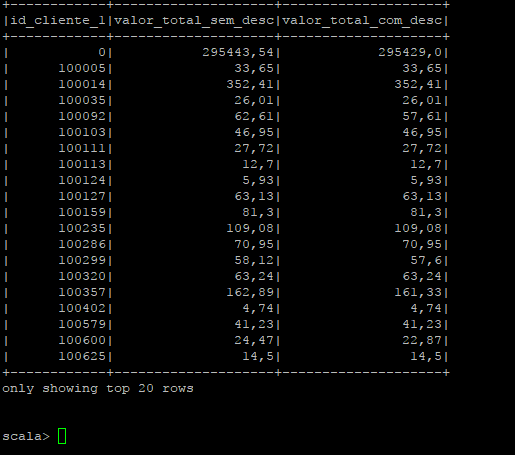
1. Executando código Scala para verificar o conteúdo do DataFrame antes de gravar no MongoDB

Figura - Total de vendas por cliente

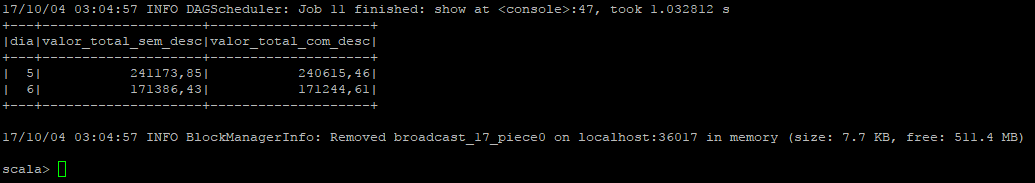


Figura - Total de vendas por dia

1. Gravação dos dados no MongoDB
   1. vendas\_por\_cliente.saveToMongoDB(WriteConfig(Map("uri" -> "mongodb://127.0.0.1/mercafacil.vendasporcliente")))
   2. vendas\_por\_dia.saveToMongoDB(WriteConfig(Map("uri" -> "mongodb://127.0.0.1/mercafacil.vendaspordia")))
   3. vendas\_por\_produto.saveToMongoDB(WriteConfig(Map("uri" -> "mongodb://127.0.0.1/mercafacil.vendasporproduto")))

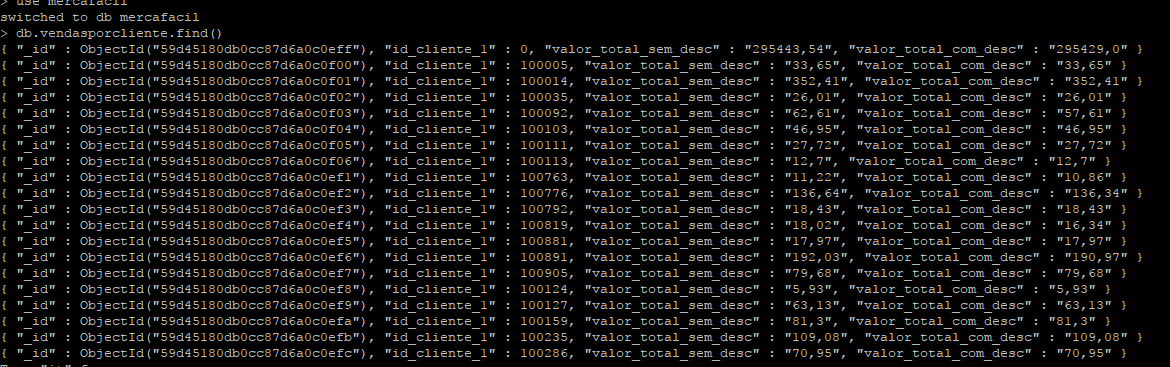


Figura - Dados de vendas por cliente gravados no Mongo



Figura - Dados de vendas por dia gravados no Mongo

# 5 Scripts Scala

1. Tabela Vendas

import sqlContext.implicits.\_

import com.mongodb.spark.config.\_

import com.mongodb.spark.\_

import com.mongodb.spark.sql.\_

val sqlContext = new org.apache.spark.sql.SQLContext(sc)

val bankText = sc.textFile("/user/mercafacil/vendas.txt")

case class Vendas(id\_loja: Integer, id\_venda: Integer, numero\_caixa: Integer, data\_venda: String, hora\_venda: String, valor\_total\_sem\_desc: Float, valor\_desconto: Float, valor\_total\_com\_desc: Float, id\_cliente\_1: Integer, id\_cliente\_2: Integer)

val vendas = bankText.map(s => s.split(";")).map(

s => Vendas(s(0).toInt,

s(1).toInt,

s(2).toInt,

s(3).toString,

s(4).toString,

s(5).toFloat,

s(6).toFloat,

s(7).toFloat,

s(8).toInt,

s(9).toInt

)

)

vendas.toDF().registerTempTable("vendas")

--Total de vendas cliente

--%sql

--select id\_cliente\_1, round(sum(valor\_total\_sem\_desc),2) as valor\_total\_sem\_desc, round(sum(valor\_total\_com\_desc),2) as --valor\_total\_com\_desc from vendas GROUP BY id\_cliente\_1 ORDER BY id\_cliente\_1

val vendas\_por\_cliente = sqlContext.sql("select id\_cliente\_1, translate(round(sum(valor\_total\_sem\_desc),2),'.',',') as valor\_total\_sem\_desc, translate(round(sum(valor\_total\_com\_desc),2),'.',',') as valor\_total\_com\_desc from vendas GROUP BY id\_cliente\_1 ORDER BY id\_cliente\_1")

--Total de vendas por dia

--%sql

--select DAY(CAST(UNIX\_TIMESTAMP(data\_venda, 'dd/MM/yyyy') AS TIMESTAMP)) as dia, round(sum(valor\_total\_sem\_desc),2) as --valor\_total\_sem\_desc, round(sum(valor\_total\_com\_desc),2) as valor\_total\_com\_desc from vendas group by DAY(CAST(UNIX\_TIMESTAMP(data\_venda, --'dd/MM/yyyy') AS TIMESTAMP))

val vendas\_por\_dia = sqlContext.sql("select DAY(CAST(UNIX\_TIMESTAMP(data\_venda, 'dd/MM/yyyy') AS TIMESTAMP)) as dia, translate(round(sum(valor\_total\_sem\_desc),2),'.',',') as valor\_total\_sem\_desc, translate(round(sum(valor\_total\_com\_desc),2),'.',',') as valor\_total\_com\_desc from vendas group by DAY(CAST(UNIX\_TIMESTAMP(data\_venda, 'dd/MM/yyyy') AS TIMESTAMP))")

vendas\_por\_cliente.saveToMongoDB(WriteConfig(Map("uri" -> "mongodb://127.0.0.1/mercafacil.vendasporcliente")))

vendas\_por\_dia.saveToMongoDB(WriteConfig(Map("uri" -> "mongodb://127.0.0.1/mercafacil.vendaspordia")))

1. Tabela vendas\_itens

import sqlContext.implicits.\_

import com.mongodb.spark.config.\_

import com.mongodb.spark.\_

import com.mongodb.spark.sql.\_

val vendasitens = sc.textFile("/user/mercafacil/vendas\_itens.txt")

case class Vendas\_itens(id\_loja: Integer, id\_venda: Integer, numero\_caixa: Integer, id\_produto: Integer, quantidade: Float, valor\_unitario: Float, valor\_total\_sem\_desc: Float, valor\_desconto: Float, valor\_total\_com\_desc: Float, id\_profissional\_1: Integer, id\_profissional\_2: Integer)

val vendas\_itens = vendasitens.map(s => s.split(";")).map(

s => Vendas\_itens(s(0).toInt,

s(1).toInt,

s(2).toInt,

s(3).toInt,

s(4).replaceAll(",",".").toFloat,

s(5).toFloat,

s(6).toFloat,

s(7).toFloat,

s(8).toFloat,

s(9).toInt,

s(10).toInt

)

)

vendas\_itens.toDF().registerTempTable("Vendas\_itens")

--Total de vendas por produto

--%sql

--select Vendas\_itens.id\_produto, translate(round(sum(Vendas\_itens.valor\_total\_sem\_desc), 3), '.', ',') as valor\_total\_sem\_desc, --translate(round(sum(Vendas\_itens.valor\_total\_com\_desc), 3), '.', ',') as valor\_total\_com\_desc from Vendas\_itens group by --Vendas\_itens.id\_produto order by 1

val vendas\_por\_produto = sqlContext.sql("select Vendas\_itens.id\_produto, translate(round(sum(Vendas\_itens.valor\_total\_sem\_desc), 3), '.', ',') as valor\_total\_sem\_desc, translate(round(sum(Vendas\_itens.valor\_total\_com\_desc), 3), '.', ',') as valor\_total\_com\_desc from Vendas\_itens group by Vendas\_itens.id\_produto order by 1")

vendas\_por\_produto.saveToMongoDB(WriteConfig(Map("uri" -> "mongodb://127.0.0.1/mercafacil.vendasporproduto")))