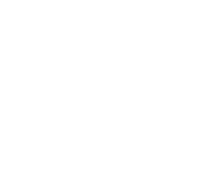
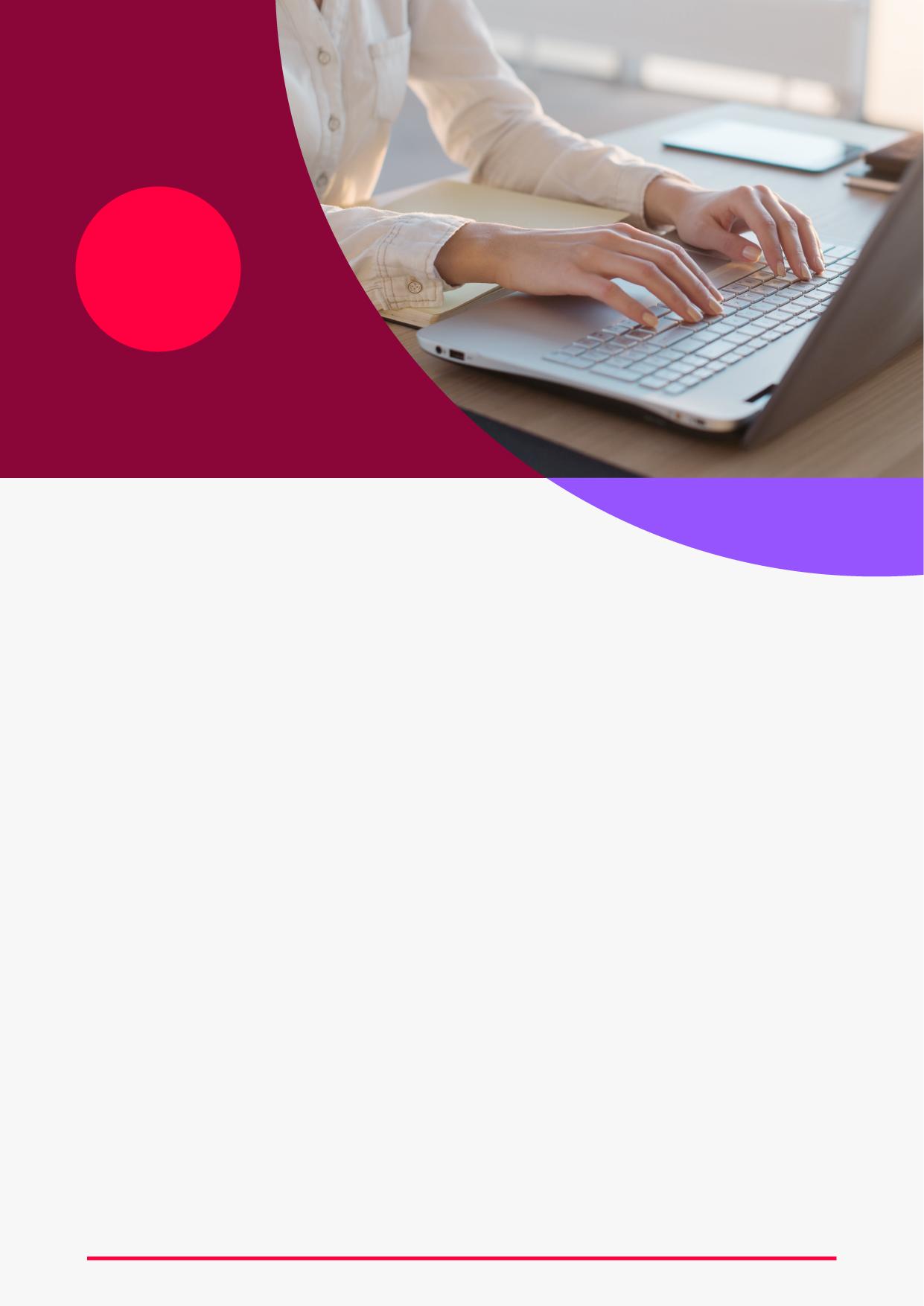
5 

P3TB2bA#y1P3TB3bA#y1P3TB4bA#y1P3#y1

**1. Introdução**

O objetivo desse trabalho é extrair informações de um dataset com a ferramenta Apache Spark. Os dados foram disponibilizados no Hadoop File System do servidor Hadoop da PUCPR.  
O Projeto foi dividido em 4 etapas.

**1.1 Reconhecimento do dataset.**

Análise do dicionário para entender o que cada coluna significa. Observação dos tipos de dados e da formatação dos dados.

**1.2 Compreensão dos fundamentos do Spark.**

Estudo de tópicos como sparkshell, sparkContext, RDD, actions e transformations.

**1.3 Processamento dos dados com Spark.**

Desenvolvimento de soluções que respondam às perguntas exigidas no projeto.

1. Quantidade de crimes por ano?
2. Quantidade de crimes por ano que sejam do tipo NARCOTICS?
3. Quantidade de crimes por ano que sejam do tipo NARCOTICS e tenham ocorrido em dias pares?
4. Mês com maior ocorrência de crimes?
5. Mês com a maior média de ocorrência de crimes?
6. Mês por ano com a maior ocorrência de crimes?
7. Mês com a maior ocorrência de crimes do tipo DECEPTIVE PRACTICE?
8. Dia do ano com a maior ocorrência de crimes?

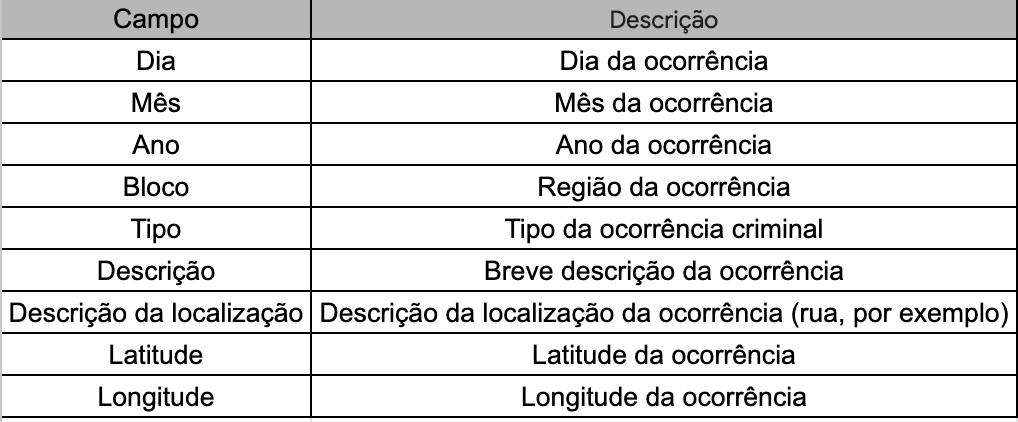
**1.4 Desenvolvimento do relatório técnico.**

Sintetização de todas as etapas, exposição de resultados e conclusão do projeto.

**2. Desenvolvimento**

**2.1 Reconhecimento do dataset.**

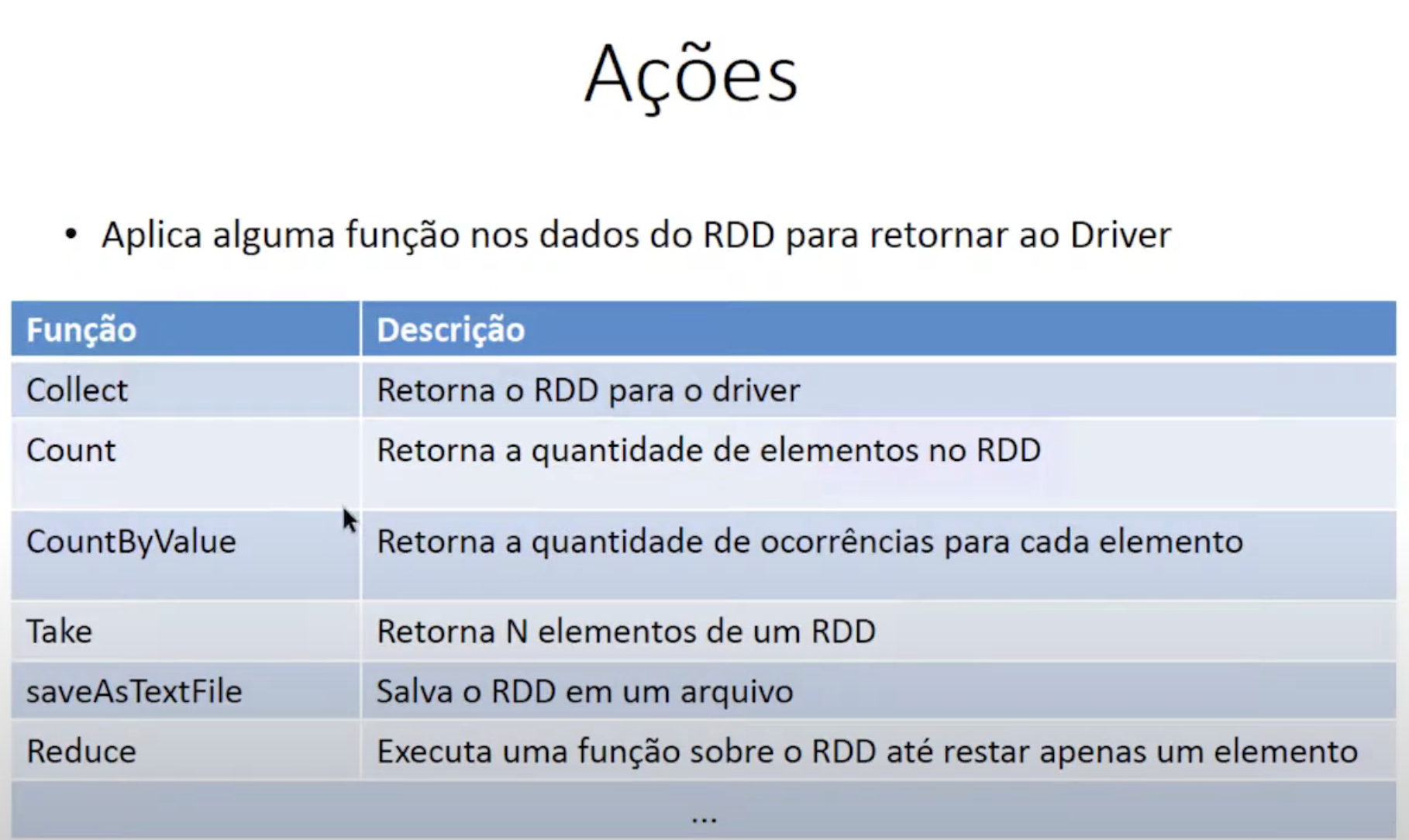
Análise do dicionário

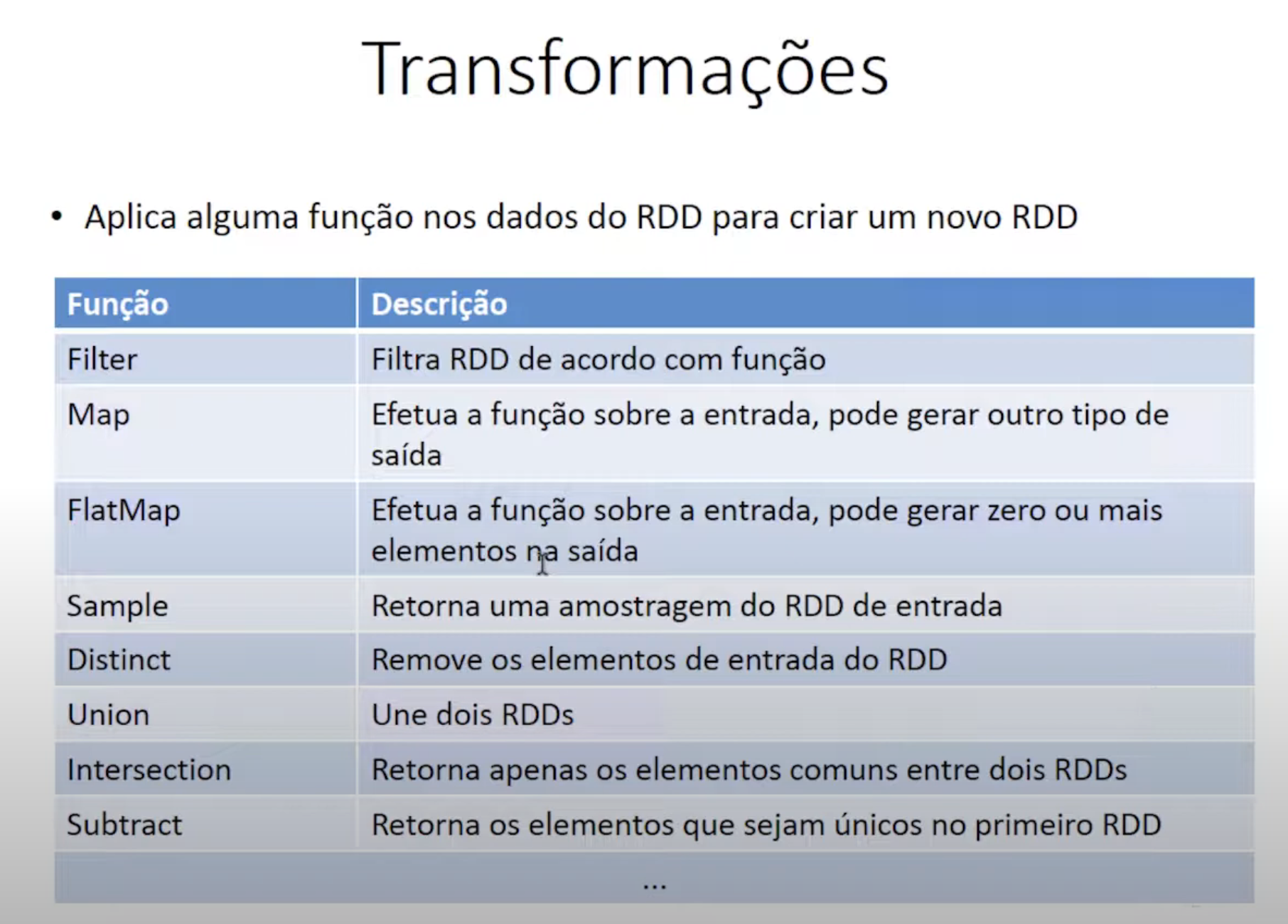


O arquivo vem de um csv. O que indica que todos os campos estão em texto.

**2.2 Estudo do Spark**

SparkContext = usado para criar uma sessão do Spark



****

**3 - RESULTADOS**

Antes de executar os códigos das soluções, é preciso criar um SparkContext e um RDD inicial. Esse código antecede todas as soluções que serão apresentadas em seguida.

****

Após rodar os programas Java, eram obtidas saídas nos consoles como a seguinte. Não colocarei todas as saídas para não poluir esse relatório. Como exemplo, a saída da questão 4. Em seguida apresentarei o código de cada solução e a última parte da saída que contém a resposta.

**cd /home2/ead2022/SEM1/henrique.guazzelli/NetBeansProjects/analise\_criminal\_spark; JAVA\_HOME=/usr/local/jdk1.8.0\_111 M2\_HOME=/usr/local/apache-maven-3.6.3 /usr/local/apache-maven-3.6.3/bin/mvn "-Dexec.args=-classpath %classpath com.pucpr.analise\_criminal\_spark.Questao4" -Dexec.executable=/usr/local/jdk1.8.0\_111/bin/java org.codehaus.mojo:exec-maven-plugin:1.2.1:exec**

**Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead of their jar artifacts.**

**Scanning for projects...**

**------------------< com.pucpr:analise\_criminal\_spark >------------------**

**Building analise\_criminal\_spark 1.0-SNAPSHOT**

**--------------------------------[ jar ]---------------------------------**

**Downloading from apache.snapshots: https://repository.apache.org/snapshots/commons-codec/commons-codec/maven-metadata.xml**

**Could not transfer metadata commons-codec:commons-codec/maven-metadata.xml from/to apache.snapshots (https://repository.apache.org/snapshots): Transfer failed for https://repository.apache.org/snapshots/commons-codec/commons-codec/maven-metadata.xml**

**--- exec-maven-plugin:1.2.1:exec (default-cli) @ analise\_criminal\_spark ---**

**Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties**

**23/11/28 00:00:37 INFO SparkContext: Running Spark version 2.3.3**

**23/11/28 00:00:37 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable**

**23/11/28 00:00:38 INFO SparkContext: Submitted application: questao2**

**23/11/28 00:00:38 INFO SecurityManager: Changing view acls to: henrique.guazzelli**

**23/11/28 00:00:38 INFO SecurityManager: Changing modify acls to: henrique.guazzelli**

**23/11/28 00:00:38 INFO SecurityManager: Changing view acls groups to:**

**23/11/28 00:00:38 INFO SecurityManager: Changing modify acls groups to:**

**23/11/28 00:00:38 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(henrique.guazzelli); groups with view permissions: Set(); users with modify permissions: Set(henrique.guazzelli); groups with modify permissions: Set()**

**23/11/28 00:00:38 INFO Utils: Successfully started service 'sparkDriver' on port 39972.**

**23/11/28 00:00:38 INFO SparkEnv: Registering MapOutputTracker**

**23/11/28 00:00:38 INFO SparkEnv: Registering BlockManagerMaster**

**23/11/28 00:00:38 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information**

**23/11/28 00:00:38 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up**

**23/11/28 00:00:38 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-1d4d454a-037c-4b9b-9f20-3eb6f8a6ad91**

**23/11/28 00:00:38 INFO MemoryStore: MemoryStore started with capacity 10.0 GB**

**23/11/28 00:00:38 INFO SparkEnv: Registering OutputCommitCoordinator**

**23/11/28 00:00:38 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.**

**23/11/28 00:00:38 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.**

**23/11/28 00:00:38 INFO Utils: Successfully started service 'SparkUI' on port 4042.**

**23/11/28 00:00:38 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://sandbox-host.hortonworks.com:4042**

**23/11/28 00:00:38 INFO Executor: Starting executor ID driver on host localhost**

**23/11/28 00:00:38 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 33282.**

**23/11/28 00:00:38 INFO NettyBlockTransferService: Server created on sandbox-host.hortonworks.com:33282**

**23/11/28 00:00:38 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy**

**23/11/28 00:00:38 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, sandbox-host.hortonworks.com, 33282, None)**

**23/11/28 00:00:38 INFO BlockManagerMasterEndpoint: Registering block manager sandbox-host.hortonworks.com:33282 with 10.0 GB RAM, BlockManagerId(driver, sandbox-host.hortonworks.com, 33282, None)**

**23/11/28 00:00:38 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, sandbox-host.hortonworks.com, 33282, None)**

**23/11/28 00:00:38 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, sandbox-host.hortonworks.com, 33282, None)**

**4 - Mes com maior ocorrencia de crimes**

**{12=69206, 08=85811, 09=80416, 11=70772, 05=109328, 04=98336, 10=79499, 03=88470, 06=99303, 02=64330, 07=86045, 01=68484}**

**------------------------------------------------------------------------**

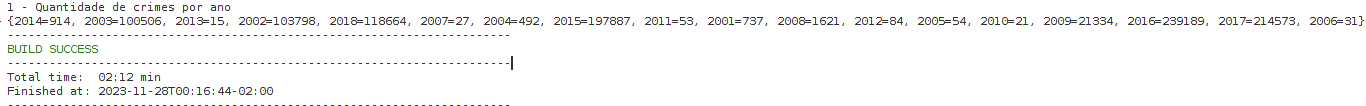
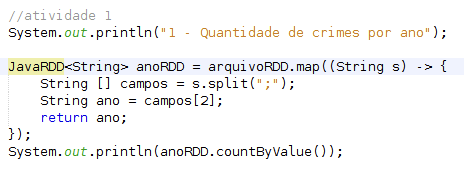
**BUILD SUCCESS**

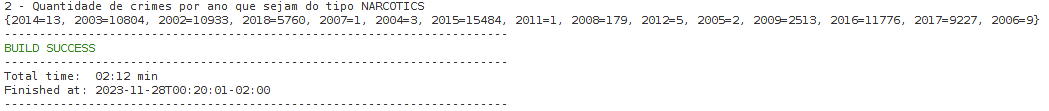
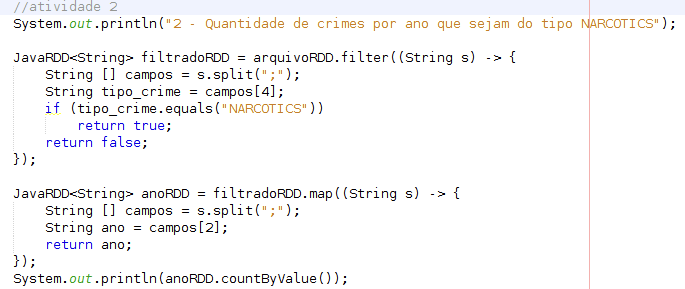
**------------------------------------------------------------------------**

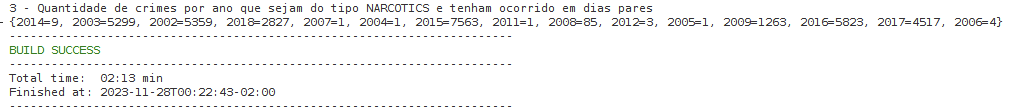
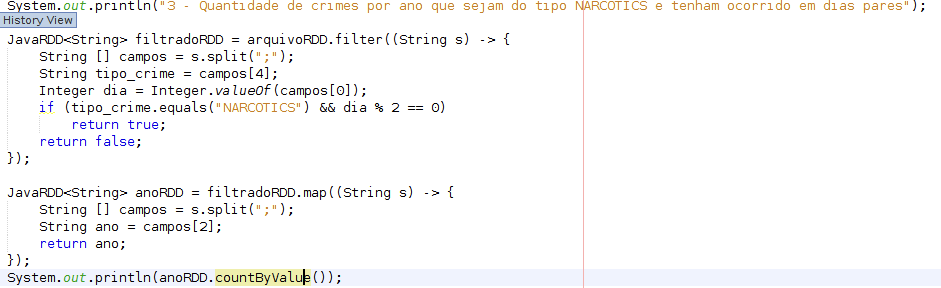
**Total time: 02:12 min**

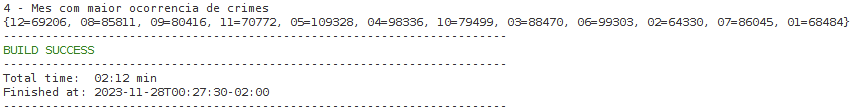
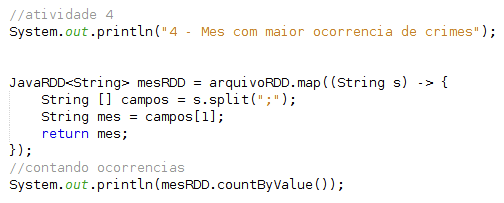
**Finished at: 2023-11-28T00:00:40-02:00**

**------------------------------------------------------------------------**

**3.1 Quantos crimes por ano?**

**3.2 Quantos crimes por ano do tipo narcotics?**

**3.3 Quantos crimes por ano do tipo narcotics em dias pares?**

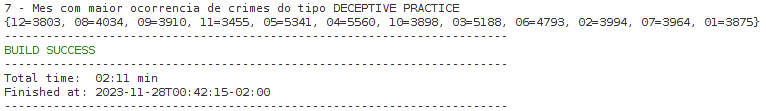
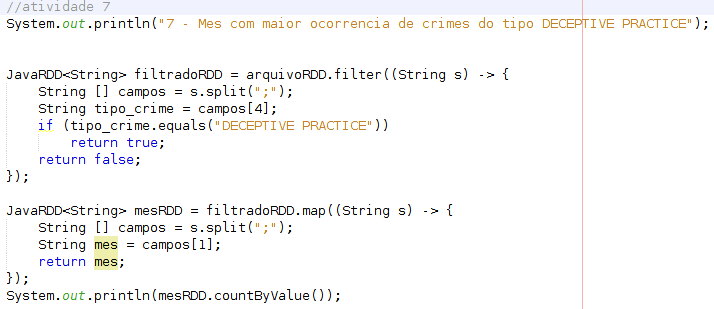
**3.4 Mês com maior ocorrência de crimes?**

Mês 5 com 109.328 ocorrências.

**3.5 Mês com a maior média de ocorrência de crimes?**

**3.6 Mês por ano com a maior ocorrência de crimes?**

**3.7 Mês com a maior ocorrência de crimes do tipo DECEPTIVE PRACTICE?**

****

Mes 4 com 5560 ocorrências.

**3.8 Dia do ano com a maior ocorrência de crimes?**

**4 - CONCLUSÕES**

O Apache Spark é uma ferramenta de processamento de dados muito utilizada pelas maiores empresas da atualidade. Seu sucesso é atribuído para o processamento em memória, que o torna muito mais rápido. Além disso, ele pode acessar dados de diversas fontes de Big Data, como o HDFS. Embora tenhamos usado Java, o Spark pode ser utilizado com as linguagens Python, Java, Scala e R.

**REFERÊNCIAS**

Spark – What is SparkSession Explained. Disponível em: <https://sparkbyexamples.com/spark/sparksession-explained-with-examples/>

. Acesso em: 23 de Novembro de 2023.

Create a Spark RDD using Parallelize. Disponível em: <https://sparkbyexamples.com/spark/how-to-create-an-rdd-using-parallelize/>

. Acesso em: 23 de Novembro de 2023.

Spark RDD Transformations with Examples - RDD Filter. Disponível em: <https://sparkbyexamples.com/apache-spark-rdd/spark-rdd-transformations/#rdd-filter>

. Acesso em: 23 de Novembro de 2023.

Spark – What is SparkSession Explained. Disponível em: <https://sparkbyexamples.com/spark/sparksession-explained-with-examples/>. Acesso em: 23 de Novembro de 2023.

Scala vs Python vs R vs Java Comparison. Disponível em: <https://www.knowledgehut.com/blog/programming/scala-vs-python-vs-r-vs-java>.Acesso em: 23 de Novembro de 2023.



.

