



# Fundamentos de Big Data

## Demo 1 Hello BigData

Centro Universitário 7  
Setembro - Uni7

**Especialização em  
Ciência de Dados com Big  
Data, BI e Data Analytics**

Prof. Manoel Ribeiro

# BigData no Windows

- ★ Estrutura de pastas (sem acentos ou espaços, padrão linux)

c:\bigdata

- haddop
  - bin
    - winutils.exe
- spark
- java
- data-integration
  - lib
    - mysql-connector-java-5.1.46.jar

- ★ Copiar toda pasta \bigdata do pendriver do professor

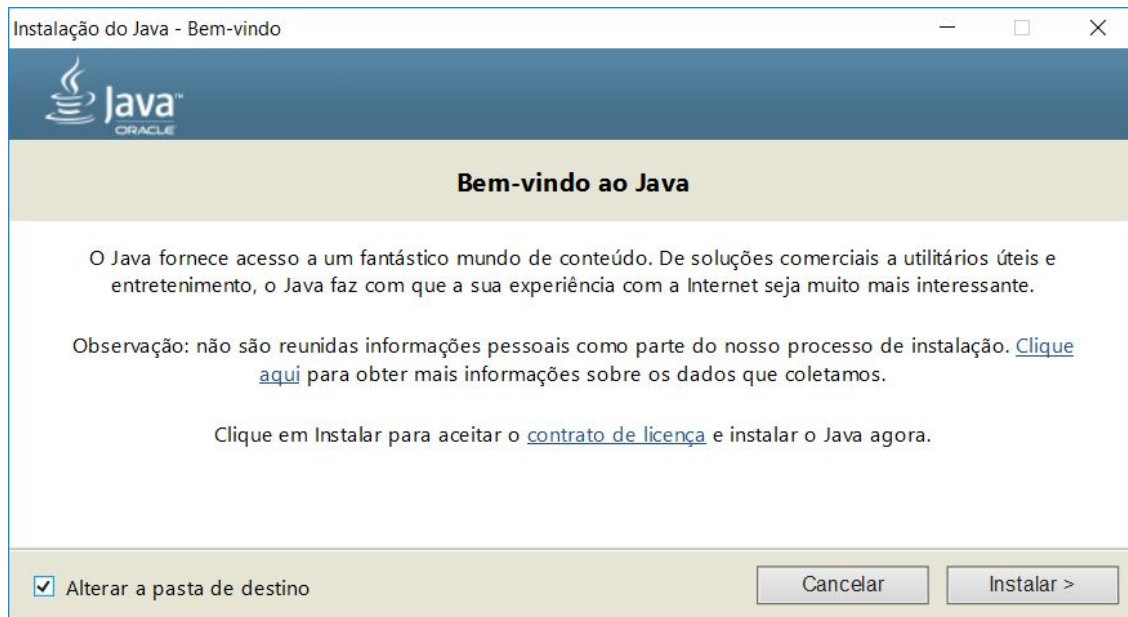
# Instalando Hadoop e Spark



# Pré Requisitos para Windows

## ★ JRE - Java

- Instalar JRE em diretório customizado (padrão linux)
- c:\bigdata\java



# Pré Requisitos para Windows

- ★ HADOOP 3.1.1
  - <http://hadoop.apache.org/releases.html>
  - binary 3.1.1
  - download `hadoop-3.1.1.tar.gz`
  - Extrair a pasta `hadoop-3.1.1` em em **c:\bigdata\**
  - **Renomear pasta `hadoop-3.1.1` para `hadoop`**
- ★ copiar `setup\WINUTILS.exe` para `c:\bigdata\hadoop\bin`
- ★ copiar `setup\hadoop.dll` para `c:\bigdata\hadoop\bin`
- ★

# Pré Requisitos para Windows

## ★ PYTHON

- Instalar python em diretório customizado (padrão linux)
- c:\bigdata\python

# Pré Requisitos para Windows

- ★ SPARK 2.3.0
  - <https://spark.apache.org/downloads.html>
  - Pre-built for Apache Hadoop 2.7 and later
  - Direct Download
  - Abrir spark-2.3.0-bin-hadoop2.7.tgz
  - Abrir pasta spark-2.3.0-bin-hadoop2.7
  - Extrair conteúdo desta pasta para **c:\bigdata\spark**



# Setup

```
>set HADOOP_HOME=C:\bigdata\hadoop
>set SPARK_HOME=c:\bigdata\spark
>cd %HADOOP_HOME%\bin
>hdfs namenode -format
c:\bigdata\hadoop\bin>winutils ls c:\tmp\
drwxrwxrwx 1 LSB\manoe1.ribeiro LSB\Domain Users 0 Oct
3 2017 c:\tmp\hive

c:\bigdata\hadoop\bin>winutils chmod 777 c:\tmp\
```



# Iniciando

```
>cd %SPARK_HOME%\bin  
c:\bigdata\spark\bin>pyspark  
Welcome to
```



version 2.2.0

```
Using Python version 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014  
22:15:05)  
SparkSession available as 'spark'.  
>>>
```

# Hello World - Spark Hive!

```
>>> data=[('Iphone8', 5000), ('Pixel2', 4000),  
('GalaxyS8', 3000), ('MotoZ', 2500)]
```

```
>>> df=spark.createDataFrame(data,  
('smartphone', 'valor'))
```

```
>>> df.printSchema()
```

```
root
```

```
 |-- smartphone: string (nullable = true)
```

```
 |-- valor: long (nullable = true)
```

Fim