Objetivo do teste:

Avaliar o seu conhecimento em Pyspark e boas práticas de programação.

Carregue a base abaixo, aplique os casos e grave a saída em um arquivo .parquet. Base:

001;José;Anápolis;São Paulo;01-09-1900
02;Igor;Anápolis;São Paulo;11-09-1977
3;Leonardo;Anápolis;São Paulo;21-12-2000
04;Humberto;Pato Branco;Rio Grande do Sul;13-11-1964
005;Isaias;Pato Branco;Rio Grande do Sul;07-07-2002
6;Lucas;Taua;Ceará;05-09-1984

Schema da base: cod_cliente, Nome, Município, Estado, data de nascimento.

- **Caso 1 – Adicionar 1 coluna com um contador sequencial por Município e ordenar por Estado.**
- **Caso 2 - Adicionar 1 coluna com a Idade em anos e na coluna cod_cliente formatar o campo com 3 posições a esquerda completando com "0".**
- **Caso 3 - Adicionar 1 coluna com a data de atualização, preenchendo com a data do dia da execução e retirar os caracteres especiais do campo Estado.**

OBS.: Criei um csv da base

Preparando o dataset

In [1]:

```
from pyspark.sql.functions import  row_number, to_date, datediff, current_date, flo
from pyspark import SparkContext
import pandas as pd
from pyspark.sql import SQLContext
from pyspark.sql.window import *
```

In [2]:

```
sc = SparkContext()
sqlContext=SQLContext(sc)
df=pd.read_csv('base.csv', sep = ';', encoding='utf-8')
sdf=sqlContext.createDataFrame(df)
```

In [3]:

```
df
```

Out[3]:

	cod_cliente	Nome	Município	Estado	data de nascimento
0	1	José	Anápolis	São Paulo	01-09-1900
1	2	Igor	Anápolis	São Paulo	11-09-1977
2	3	Leonardo	Anápolis	São Paulo	21-12-2000
3	4	Humberto	Pato Branco	Rio Grande do Sul	13-11-1964
4	5	Isaias	Pato Branco	Rio Grande do Sul	07-07-2002
5	6	Lucas	Taua	Ceará	05-09-1984

In [4]:

```
sdf
```

Out[4]:

```
DataFrame[cod_cliente: bigint, Nome: string, Município: string, Estad
o: string, data de nascimento: string]
```

Caso 1 – Adicionar 1 coluna com um contador sequencial por Município e ordenar por Estado.

In [5]:

```
sdf = sdf.withColumn("row_num_municipio", row_number().over(Window.partitionBy("Mun
sdf.show()
```

```
+-----------+--------+-----------+----------------+-----------------
+-----------------+
|cod_cliente|    Nome|  Município|          Estado|data de nascimento
|row_num_municipio|
+-----------+--------+-----------+----------------+-----------------
+-----------------+
|          6|   Lucas|       Taua|           Ceará|       05-09-1984
|                1|
|          4|Humberto|Pato Branco|Rio Grande do Sul|       13-11-1964
|                1|
|          5|  Isaias|Pato Branco|Rio Grande do Sul|       07-07-2002
|                2|
|          1|    José|   Anápolis|       São Paulo|       01-09-1900
|                1|
|          3|Leonardo|   Anápolis|       São Paulo|       21-12-2000
|                3|
|          2|    Igor|   Anápolis|       São Paulo|       11-09-1977
|                2|
+-----------+--------+-----------+----------------+-----------------
+-----------------+
```

Caso 2 - Adicionar 1 coluna com a Idade em anos e na coluna cod_cliente formatar o campo com 3 posições a esquerda completando com “0”.

In [6]:

```
from pyspark.sql.functions import to_date, datediff, current_date, floor, lpad, dat

# criando coluna com idade
sdf = sdf.withColumn("idade",floor(datediff(current_date(), to_date("data de nascim

# formatando cod_cliente para três dígitos
sdf = sdf.withColumn("cod_cliente", lpad("cod_cliente", 3, '0'))


sdf.show()
```

```
+----------+--------+----------+----------------+-----------------
+-----------------+-----+
|cod_cliente|    Nome|  Município|          Estado|data de nascimento
|row_num_municipio|idade|
+-----------+--------+----------+----------------+-----------------
+-----------------+-----+
|        006|   Lucas|       Taua|           Ceará|        05-09-1984
|                1|   36|
|        004|Humberto|Pato Branco|Rio Grande do Sul|        13-11-1964
|                1|   56|
|        005|  Isaias|Pato Branco|Rio Grande do Sul|        07-07-2002
|                2|   19|
|        001|    José|   Anápolis|        São Paulo|        01-09-1900
|                1|  120|
|        002|    Igor|   Anápolis|        São Paulo|        11-09-1977
|                2|   43|
|        003|Leonardo|   Anápolis|        São Paulo|        21-12-2000
|                3|   20|
+-----------+--------+----------+----------------+-----------------
+-----------------+-----+
```

Caso 3 - Adicionar 1 coluna com a data de atualização, preenchendo com a data do dia da execução e retirar os caracteres especiais do campo Estado.

In [7]:

```
# criando coluna com a data de atualização

sdf = sdf.withColumn("data_de_atualizacao",date_format(current_date(), 'dd-MM-yyyy'
sdf.show()
```

```
+-----------+--------+-----------+-----------------+------------------
+-----------------+-----+-------------------+
|cod_cliente|    Nome|  Município|           Estado|data de nascimento
|row_num_municipio|idade|data_de_atualizacao|
+-----------+--------+-----------+-----------------+------------------
+-----------------+-----+-------------------+
|        006|   Lucas|       Taua|            Ceará|        05-09-1984
|                1|   36|         19-07-2021|
|        005|  Isaias|Pato Branco|Rio Grande do Sul|        07-07-2002
|                2|   19|         19-07-2021|
|        004|Humberto|Pato Branco|Rio Grande do Sul|        13-11-1964
|                1|   56|         19-07-2021|
|        001|    José|   Anápolis|        São Paulo|        01-09-1900
|                1|  120|         19-07-2021|
|        002|    Igor|   Anápolis|        São Paulo|        11-09-1977
|                2|   43|         19-07-2021|
|        003|Leonardo|   Anápolis|        São Paulo|        21-12-2000
|                3|   20|         19-07-2021|
+-----------+--------+-----------+-----------------+------------------
+-----------------+-----+-------------------+
```

In [8]:

```
# removendo caracteres especiais
sdf = sdf.withColumn("Estado", regexp_replace("Estado", "[áàâã]", "a"))
sdf = sdf.withColumn("Estado", regexp_replace("Estado", "[éèê]", "e"))
sdf = sdf.withColumn("Estado", regexp_replace("Estado", "[íï]", "i"))
sdf = sdf.withColumn("Estado", regexp_replace("Estado", "[óôõö]", "o"))
sdf = sdf.withColumn("Estado", regexp_replace("Estado", "[úü]", "u"))

sdf.show()
```

```
+-----------+--------+-----------+-----------------+------------------
+------------------+-----+-------------------+
|cod_cliente|    Nome|  Município|           Estado|data de nascimento
|row_num_municipio|idade|data_de_atualizacao|
+-----------+--------+-----------+-----------------+------------------
+------------------+-----+-------------------+
|        006|   Lucas|       Taua|            Ceara|        05-09-1984
|                1|   36|         19-07-2021|
|        004|Humberto|Pato Branco|Rio Grande do Sul|        13-11-1964
|                1|   56|         19-07-2021|
|        005|  Isaias|Pato Branco|Rio Grande do Sul|        07-07-2002
|                2|   19|         19-07-2021|
|        001|    José|   Anápolis|        Sao Paulo|        01-09-1900
|                1|  120|         19-07-2021|
|        003|Leonardo|   Anápolis|        Sao Paulo|        21-12-2000
|                3|   20|         19-07-2021|
|        002|    Igor|   Anápolis|        Sao Paulo|        11-09-1977
|                2|   43|         19-07-2021|
+-----------+--------+-----------+-----------------+------------------
+------------------+-----+-------------------+
```

salvando o arquivo em formato parquet

In [9]:

```
sdf = sdf.withColumnRenamed("data de nascimento", "data_de_nascimento")
sdf.write.parquet("base.parquet")
```