Big Data HPC

Handson utilizando PySpark

SENAI
CIMATEC

Tecnologia, Inovação
e Educação para a Indústria

Sistema FIEB | SENAI CIMATEC

PELO FUTURO DA INOVAÇÃO

APPROVED COMPANY
ISO 27001
Information Security Management Systems
QMS Certification Services

SENAI CIMATEC

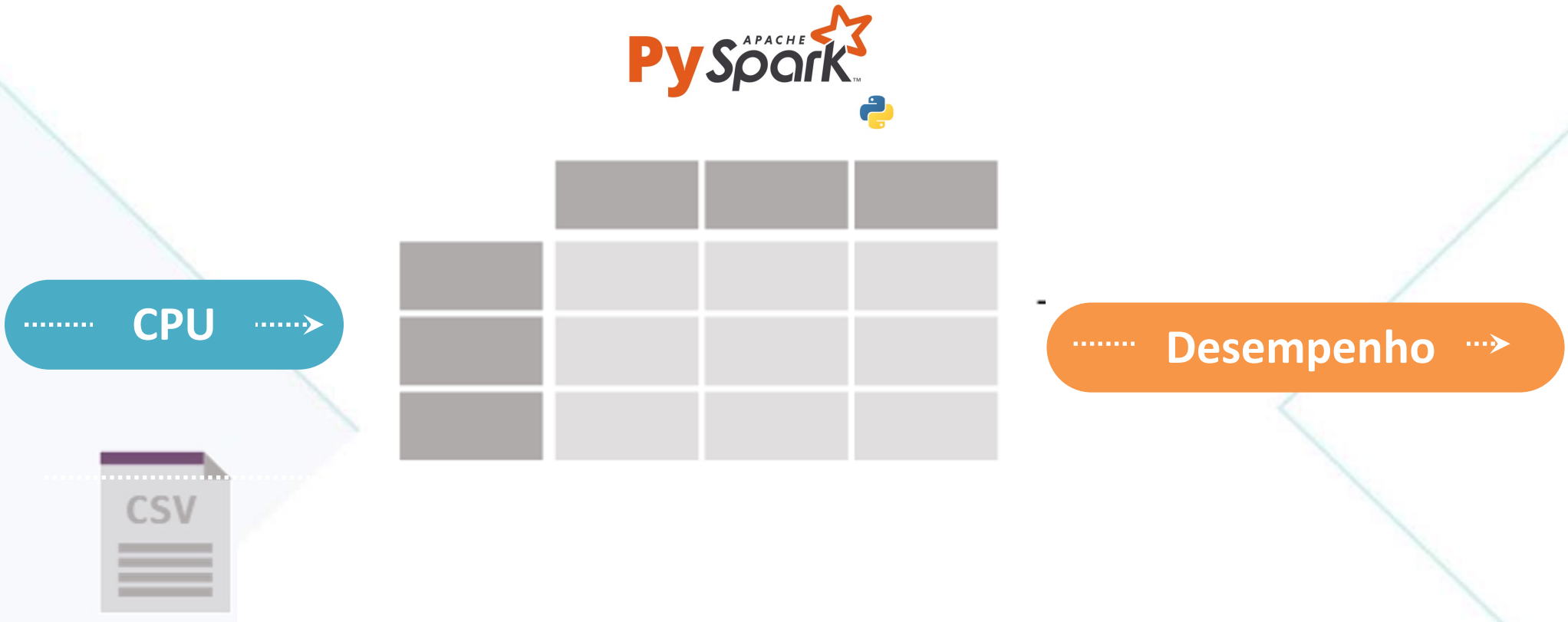
Big Data HPC

Handson utilizando PySpark

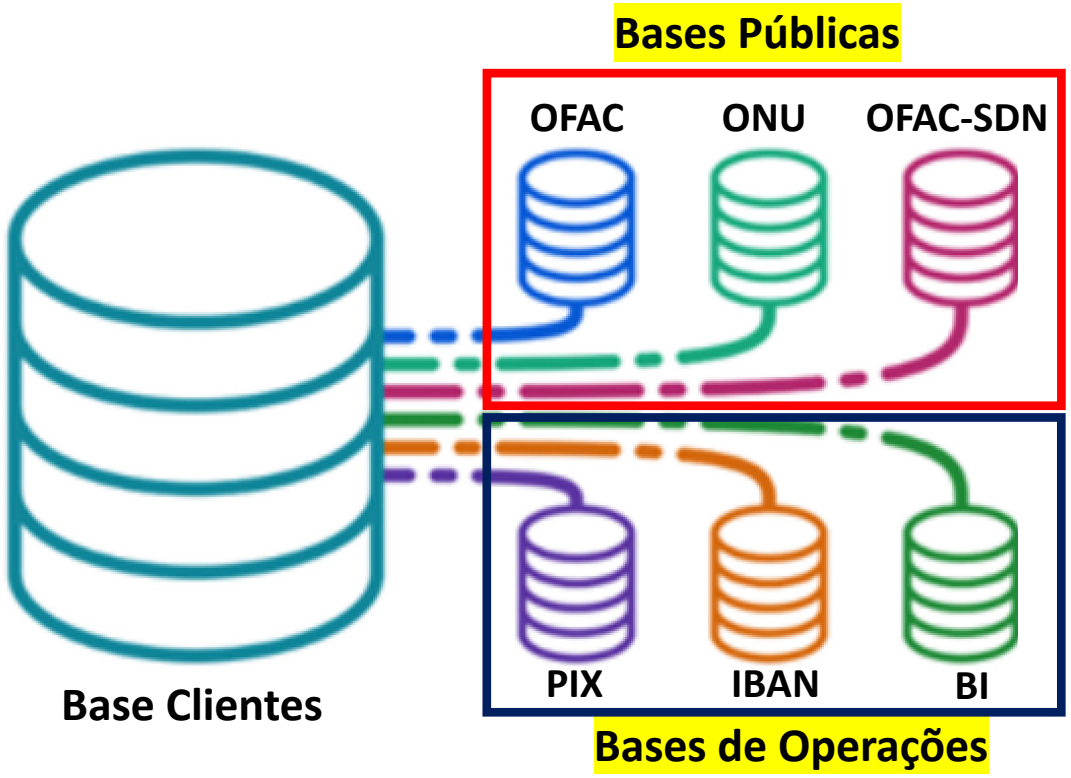
Murilo Boratto

https://github.com/muriloboratto/big-data-hpc

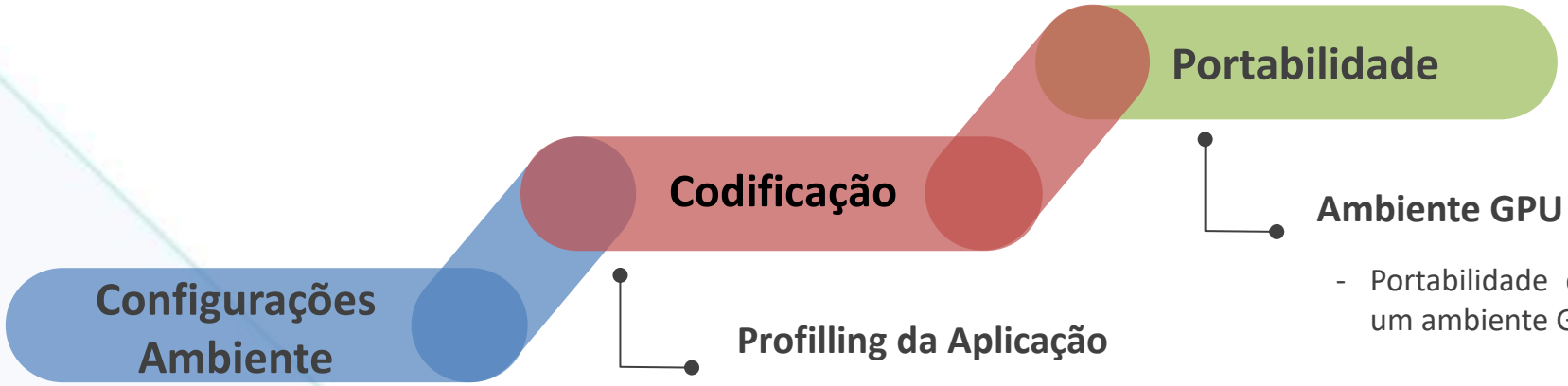
Estrutura do Estudo de Caso



Estudos de Caso



Níveis de Otimização



Busca pelos Parâmetros Ótimos de Execução

- Task Affinity no Supercomputador;
- Analogias com o ambiente Databricks do BV;

Profilling da Aplicação

- Mapeamento dos Custos Computacionais das Aplicações;

Otimização das Operações

- Operação de Join;
- Cálculo de Similaridade;

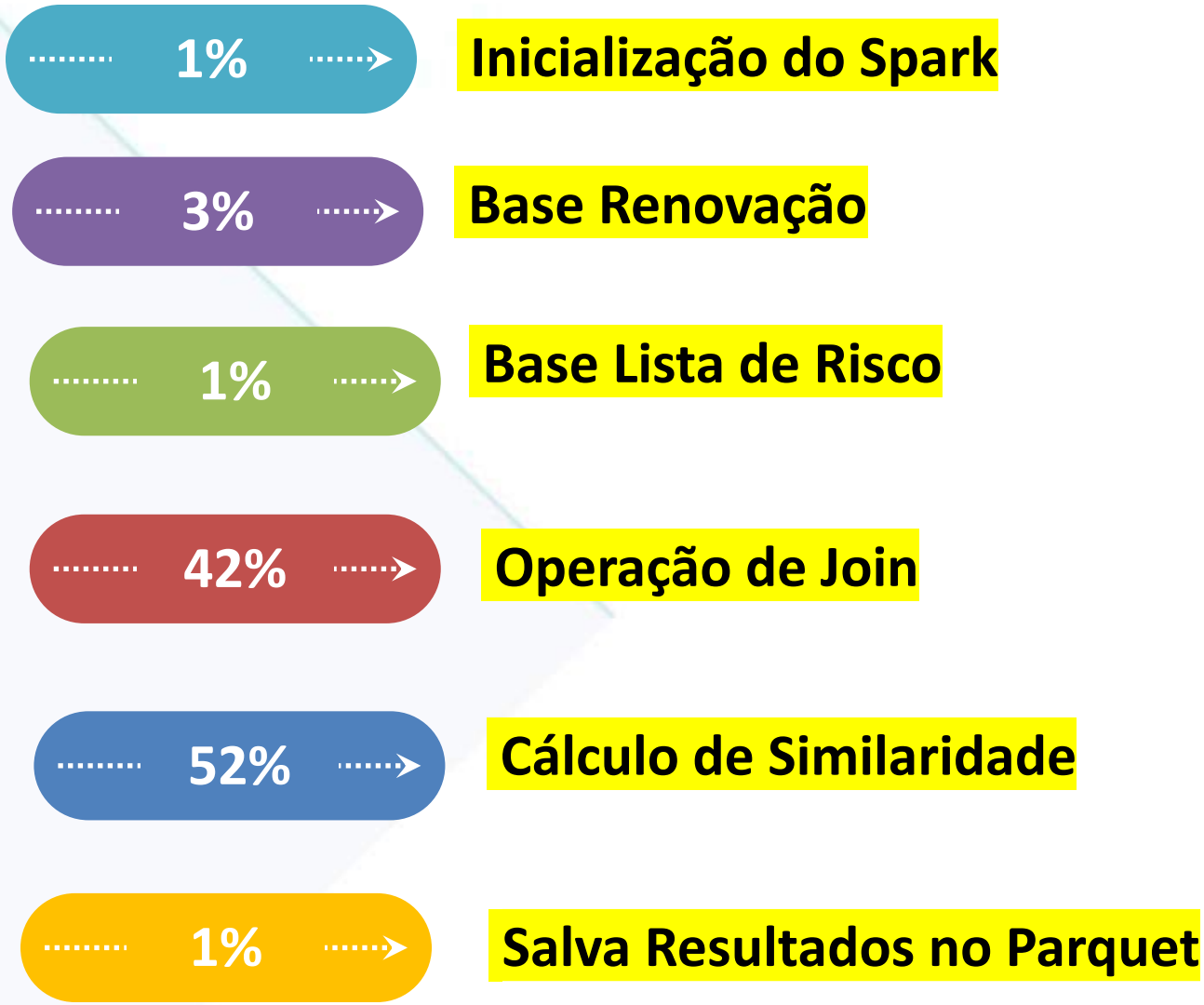
Ambiente GPU

- Portabilidade de Código para um ambiente GPU;

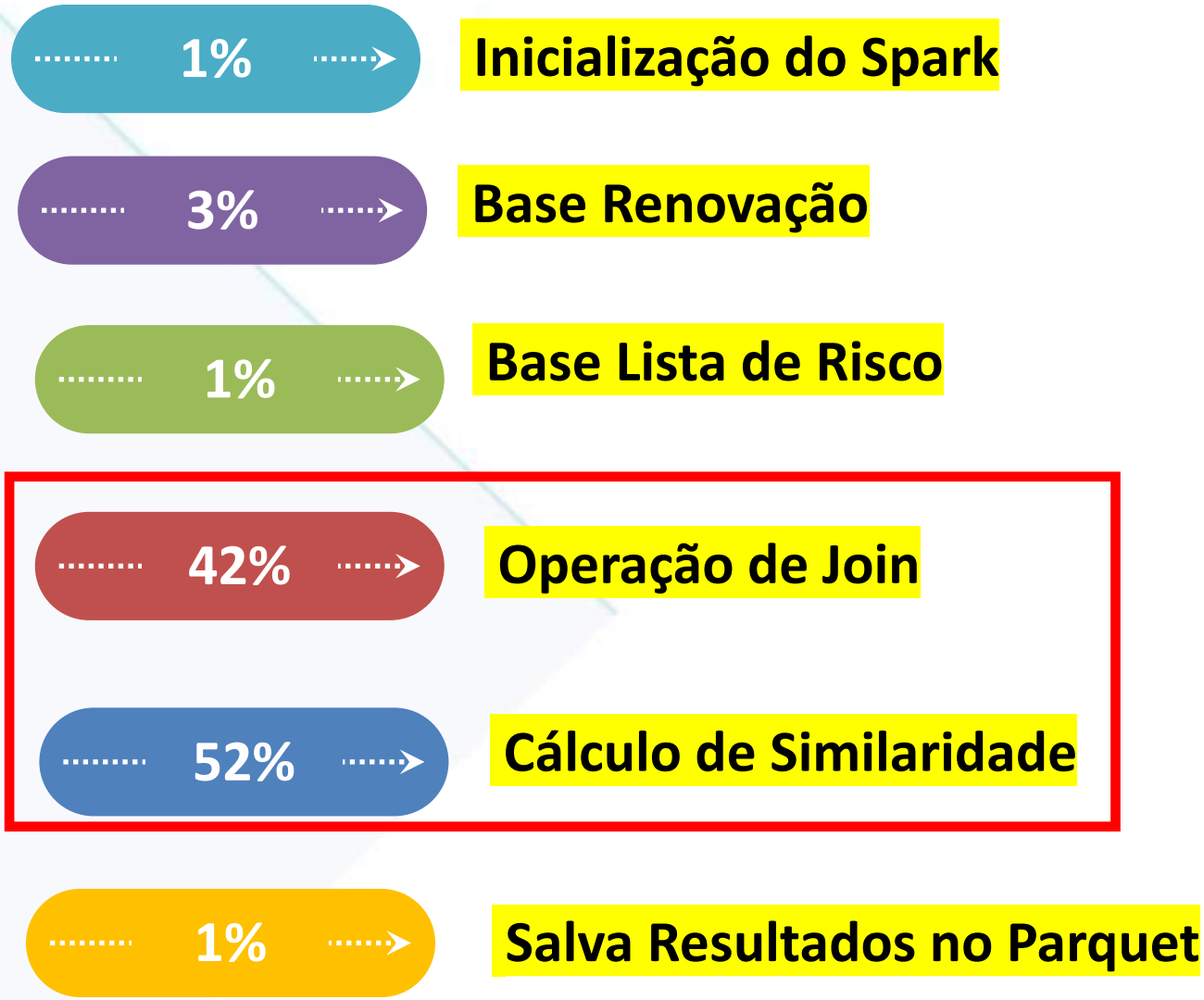
Base Pública OFAC-SDN – Profilling – Original

1. Inicialização do Spark
2. Base Renovação
3. Base Lista de Risco
4. Operação de Join
5. Cálculo de Similaridade
6. Salva Resultados no Parquet

Base Pública OFAC-SDN – Profilling – Original



Base Pública OFAC-SDN – Profilling – Original



Base Pública OFAC-SDN – Profilling – Original

 # Aplicação Original

```
df_match = df_renov.join(df_lista_risco,
           ((df_renov['TpPessoa'] == df_lista_risco['TIPO']) &
           (df_renov[‘m’] == df_lista_risco['m'])), how='inner')
```

 # Aplicação Otimizada

```
df_joined = df_renov.join(F.broadcast(df_lista_risco),["TIPO_RELACAO"])
```

Base Pública OFAC-SDN – Profilling – Original

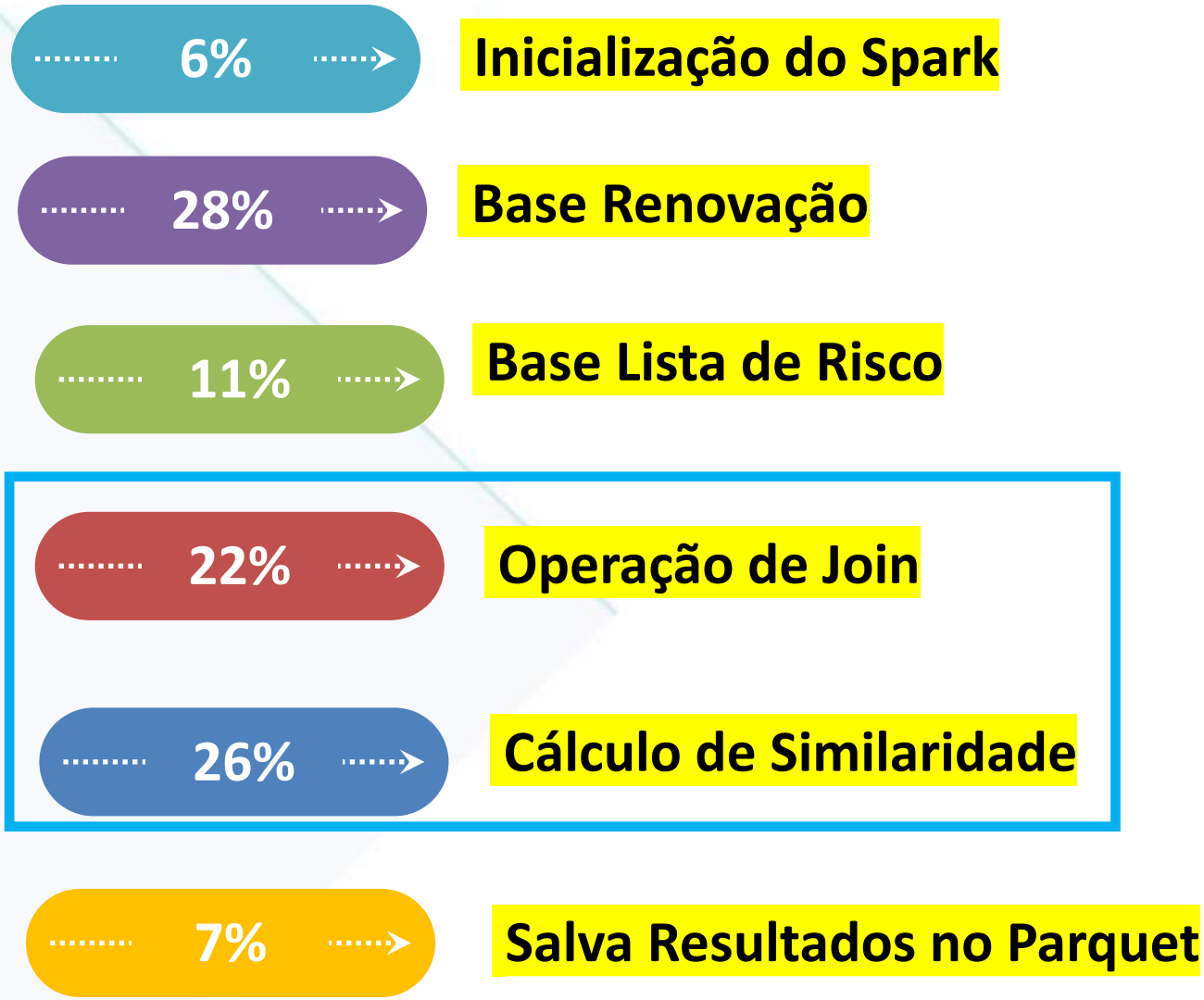
 # Aplicação Original

```
def lev(x, y):
    return fuzz.ratio(x,y)

similarity = udf(lev, IntegerType())

df = df.withColumn("SIMILARITY",
similarity(df.NOME,df.NOME_RISCO))

fil = df.filter(df.SIMILARITY) > 90)
```

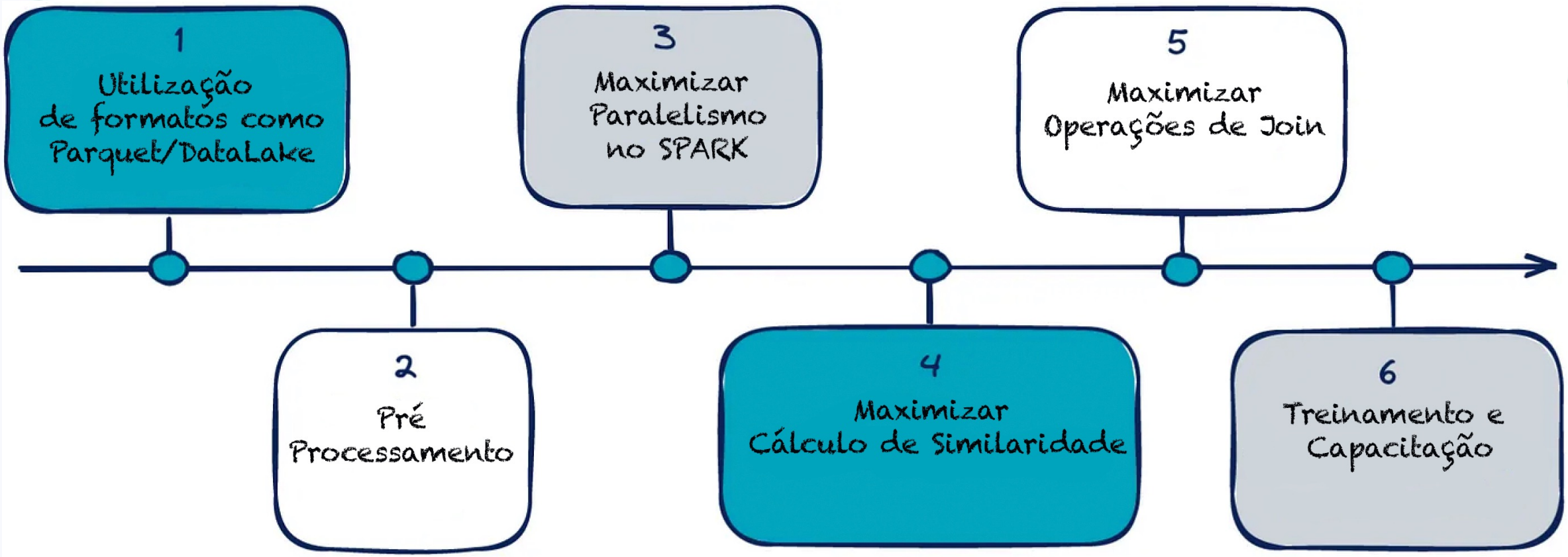
 # Aplicação Otimizada

```
lev    = F.levenshtein
str1   = F.col("NOME")
str2   = F.col("NOME_RISCO")
len_s1 = F.length(str1)
len_s2 = F.length(str2)

df_lev = df_joined.withColumn(
"SIMILARITY",(100 * (1 - (lev(str1, str2) /
(len_s1 + len_s2)))).cast("int"))

fil = df_lev.filter(F.col("SIMILARITY") > 90)
```

Base Pública OFAC-SDN – Profilling – Original



Recomendações para Otimizar a Aplicação PLD Spark do Banco BV


1
Utilização de formatos como Parquet/DataLake
2
Pré Processamento
3
Maximizar Paralelismo no SPARK
4
Maximizar Cálculo de Similaridade
5
Maximizar Operações de Join
6
Treinamento e Capacitação

Resultados Experimentais

PLD	OFAC	ONU	OFAC-SDN
ORIGINAL	135,20	860,92	9476,56
OTIMIZAÇÕES CPU	58,51	299,50	3157,90
SPEEDUP CPU	2,31X	2,87X	3X

Tabela: Tempos de Execução (seg.) comparando o desempenho das Bases Públicas em sistemas CPU.

Resultados Experimentais

PLD	OFAC	ONU	OFAC-SDN
ORIGINAL	135,20	860,92	9476,56
OTIMIZAÇÕES CPU	58,51	299,50	3157,90
OTIMIZAÇÕES GPU	22,10	143,00	1353,79
SPEEDUP CPU	2,31X	2,87X	3X
SPEEDUP GPU	6X	6X	7X

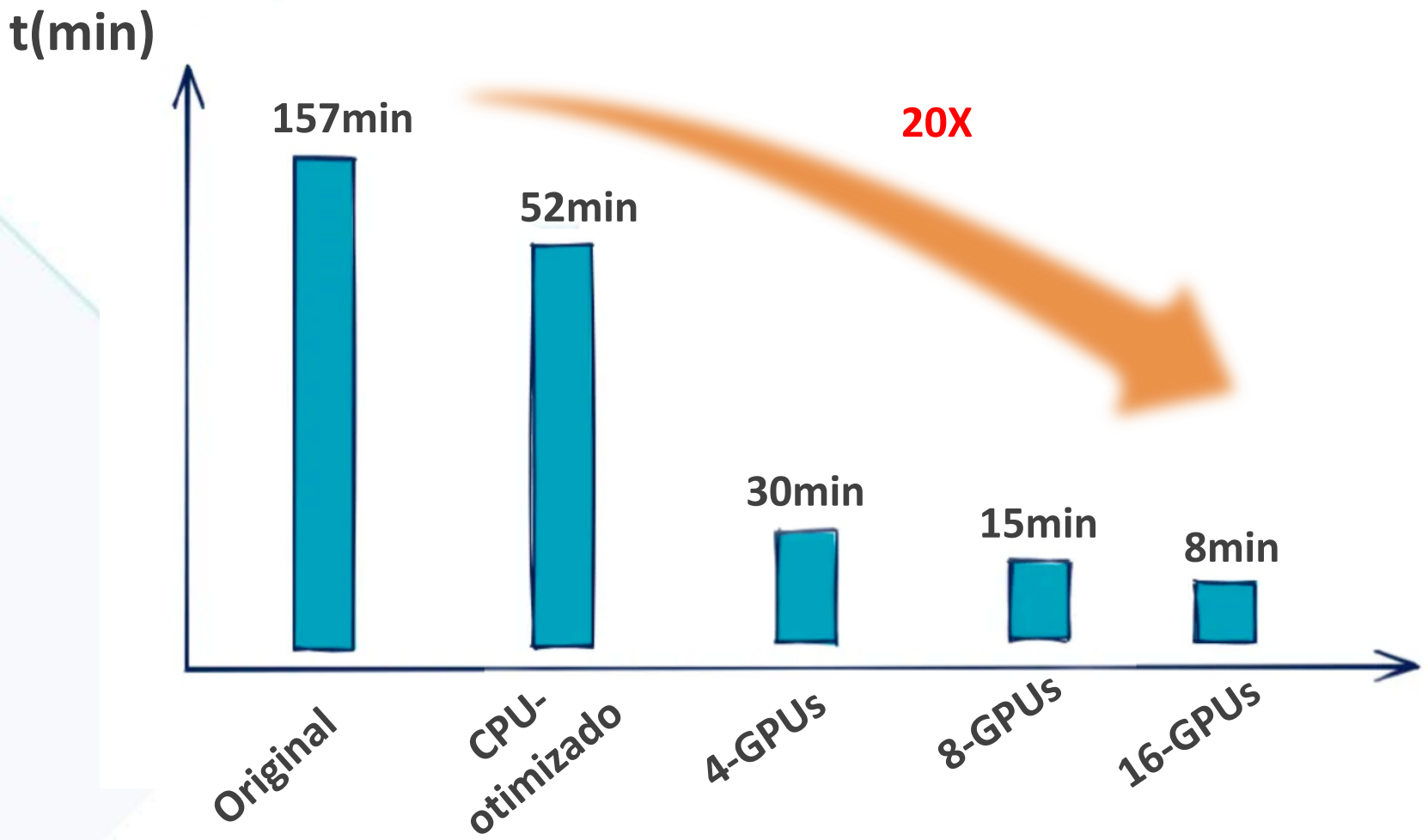
Tabela: Tempos de Execução (seg.) comparando o desempenho das Bases Públicas em sistemas CPU e 4-GPUs.

Resultados Experimentais

	4-GPUs	8-GPUs	16-GPUs
PLD	OFAC-SDN	OFAC-SDN	OFAC-SDN
ORIGINAL	9476,56	9476,56	9476,56
OTIMIZAÇÕES CPU	3157,90	3157,90	3157,90
OTIMIZAÇÕES GPU	1353,79	900,00	480,10
SPEEDUP CPU	3X	3X	3X
SPEEDUP GPU	7X	10X	20X

Tabela: Tempos de Execução (seg.) comparando o desempenho das Bases Públicas em sistemas CPU e multi-GPU.

Resultados Experimentais (OFAC-SDN)



Conclusões

- A utilização de Sistemas Multi-GPU torna-se praticamente indispensável para as bases com grande quantidade de dados.
- Uma série de otimizações tanto a nível de hardware/software podem ser aplicadas as aplicações melhorando ainda mais o desempenho.
- A idéia é simular com bases maiores, afim de comparar o desempenho computacional real obtido.

Big Data HPC

Handson utilizando PySpark

SENAI
CIMATEC

Tecnologia, Inovação
e Educação para a Indústria

Sistema FIEB | SENAI CIMATEC

PELO FUTURO DA INOVAÇÃO

APPROVED COMPANY
ISO 27001
Information Security Management Systems
QMS Certification Services