

processo

August 16, 2023

```
[2]: from app.src.sor import job_ingestao_bancos_sor
from app.src.sor import job_ingestao_reclamacoes_sor
from app.src.sor import job_ingestao_empregados_sor
from app.src.sot import job_ingestao_bancos_sot
from app.src.sot import job_ingestao_reclamacoes_sot
from app.src.sot import job_ingestao_empregados_sot
from app.src.spec import job_ingestao_01_spec
from app.utils.utils import read_csv_files_recursively
from plotly import express as px
from datetime import date
```

```
[3]: DATA_REFERENCIA: date = date(2023,8,14)
```

```
[4]: %%time
# Processa dados raw Bancos para gerar camada Source of Record (SoR)
job: job_ingestao_bancos_sor.ETLJob = job_ingestao_bancos_sor.
    ↳ ETLJob(input_path="./app/data/input/Bancos/",
                                                    ↳
    ↳ output_path="./app/data/output/sor/tb_bancos/",
                                                    ↳
    ↳ dat_ref=DATA_REFERENCIA)
job.run()
read_csv_files_recursively("./app/data/output/sor/tb_bancos/", has_header=True).
    ↳ head()
```

Listing files with size > 0 bytes in ./app/data/input/Bancos/:

CPU times: user 12.5 ms, sys: 13.8 ms, total: 26.3 ms

Wall time: 49.3 ms

```
[4]: shape: (5, 3)
```

Segmento	CNPJ	Nome
---	---	---
str	i64	str
S1	0	BANCO DO BRASIL - PRUDENCIAL
S1	60746948	BRABESCO - PRUDENCIAL

S1	30306294	BTG PACTUAL - PRUDENCIAL
S1	360305	CAIXA ECONOMICA FEDERAL - PRUDEN...
S1	60872504	ITAU - PRUDENCIAL

```
[5]: %%time
# Processa dados da SoR, realiza tratamento e salva na camada SoT dos bancos
job: job_ingestao_bancos_sot.ETLJob = job_ingestao_bancos_sot.
↳ ETLJob(input_path=f"./app/data/output/sor/tb_bancos/"
↳ anomesdia={DATA_REFERENCIA.strftime('%Y%m%d')}"/",
↳ output_path="./app/data/output/sot/tb_bancos/",
↳ dat_ref=DATA_REFERENCIA)
job.run()
read_csv_files_recursively("./app/data/output/sot/tb_bancos/", has_header=True).
↳ head()
```

Listing files with size > 0 bytes in
./app/data/output/sor/tb_bancos/anomesdia=20230814/:
CPU times: user 21.3 ms, sys: 1.02 ms, total: 22.3 ms
Wall time: 21.8 ms

[5]: shape: (5, 4)

nom_segto_instituicao	num_cnpj_instituicao	nom_instituicao	
dat_hor_psst			
---	---	---	---
str	i64	str	f64
S1	0	BANCO DO BRASIL	
1.6922e9			
S1	60746948	BRABESCO	
1.6922e9			
S1	30306294	BTG PACTUAL	
1.6922e9			
S1	360305	CAIXA ECONOMICA FEDERAL	
1.6922e9			
S1	60872504	ITAU	
1.6922e9			

```
[6]: %%time
# Processa dados raw Reclamacoes para gerar camada Source of Record (SoR)
job: job_ingestao_reclamacoes_sor.ETLJob = job_ingestao_reclamacoes_sor.
    ↪ETLJob(input_path="./app/data/input/Reclamacoes/",
    ↪output_path="./app/data/output/sor/tb_reclamacoes/")
job.run()
read_csv_files_recursively("./app/data/output/sor/tb_reclamacoes/",
    ↪has_header=True, options={"dtypes": {"Quantidade total de clientes": "float64",
    ↪SCR": "str"}}).head()
```

Listing files with size > 0 bytes in ./app/data/input/Reclamacoes/:
CPU times: user 14.9 ms, sys: 13.4 ms, total: 28.3 ms
Wall time: 33.4 ms

[6]: shape: (5, 15)

Ano	Trimestre	Categoria	Tipo	...	Quantidade	Quantidade
Quantidade						
---	---	---	---		total de	de clientes
de	---					
i64	str	str	str		clientes	CCS
clientes	str					
					C...	---
SCR					---	str

					str	
str						
2021	3º	Demais	Conglomerad	...	29707	28217
4009	null	bancos e	o			
		financeiras				
2021	3º	Demais	Conglomerad	...	1320447	999951
873341	null	bancos e	o			
		financeiras				
2021	3º	Demais	Banco/finan	...	453405	130
453275	null	bancos e	ceira			

			financeiras			
2021	3º		Demais	Banco/finan	...	18217
11683		null				9981
			bancos e	ceira		
			financeiras			
2021	3º		Demais	Conglomerad	...	420486
153423		null				268221
			bancos e	o		
			financeiras			

```
[7]: %%time
# Processa dados da SoR, realiza tratamento e salva na camada SoT das
↳reclamações
job: job_ingestao_reclamacoes_sot.ETLJob = job_ingestao_reclamacoes_sot.
↳ETLJob(input_path="./app/data/output/sor/tb_reclamacoes/",
↳output_path="./app/data/output/sot/tb_reclamacoes/")
job.run()
read_csv_files_recursively("./app/data/output/sot/tb_reclamacoes/",
↳has_header=True).head()
```

Listing files with size > 0 bytes in ./app/data/output/sor/tb_reclamacoes/:
CPU times: user 50.8 ms, sys: 11.6 ms, total: 62.4 ms
Wall time: 94 ms

[7]: shape: (5, 15)

ano	num_trimes	nom_catego	nom_tipo_i	...	qtd_totl_c	qtd_clie_
qtd_clie_	dat_hor_p					
---	tre	ria	nstituicao		lie_css_sc	ccs
sst						scr
i64	---	---	---		r	---
---						---
	i64	str	str		---	i64
f64						i64

2021 4248	4	1.6922e9	Demais Bancos, Financeira s e Ins...	Conglomera do	...	33095	31565
2021 null	4	1.6922e9	Demais Bancos, Financeira s e Ins...	Banco/fina nceira	...	2011869	2011869
2021 996638	4	1.6922e9	Demais Bancos, Financeira s e Ins...	Conglomera do	...	1577152	1224075
2021 473693	4	1.6922e9	Demais Bancos, Financeira s e Ins...	Banco/fina nceira	...	473823	130
2021 14355	4	1.6922e9	Demais Bancos, Financeira s e Ins...	Banco/fina nceira	...	23421	13123

```
[8]: %%time
# Processa dados raw Empregados para gerar camada Source of Record (SoR)
```

```

job: job_ingestao_empregados_sor.ETLJob = job_ingestao_empregados_sor.
↳ ETLJob(input_path="./app/data/input/Empregados/",
↳ output_path="./app/data/output/sor/tb_empregados/",
↳ dat_ref=DATA_REFERENCIA)
job.run()
read_csv_files_recursively("./app/data/output/sor/tb_empregados/",
↳ has_header=True).head()

```

Listing files with size > 0 bytes in ./app/data/input/Empregados/:
CPU times: user 6.48 ms, sys: 1.48 ms, total: 7.96 ms
Wall time: 5.99 ms

[8]: shape: (5, 23)

employer_	reviews_c	culture_c	salaries_	...	Perspecti	Segmento	
Nome	match_per						
name	ount	ount	count		va	---	---
cent							
---	---	---	---		positiva	str	str

str	str	str	str		da		
str							
					empresa(...		

					str		
BNP	13000	4100	20000	...	63.0	S3	BNP
100							
Paribas							
PARIBAS							
BTG	1600	683	2800	...	73.0	S1	BTG
100							
Pactual							
PACTUAL							
Banco	175	74	271	...	47.0	S3	
ALFA	100						
Alfa							
Banco BMG	445	232	704	...	63.0	S3	BMG
100							

Banco	11000	3300	19000	...	66.0	S1
BRADESCO	100					
Bradesco						

```
[9]: %%time
# Processa dados da SoR, realiza tratamento e salva na camada SoT dos Empregados
job: job_ingestao_empregados_sot.ETLJob = job_ingestao_empregados_sot.
    ↳ ETLJob(input_path=f"./app/data/output/sor/tb_empregados/"
    ↳ anomesdia={DATA_REFERENCIA.strftime('%Y%m%d')}",
    ↳ output_path="./app/data/output/sot/tb_empregados/",
    ↳ dat_ref=DATA_REFERENCIA)
job.run()
read_csv_files_recursively("./app/data/output/sot/tb_empregados/",
    ↳ has_header=True).head()
```

Listing files with size > 0 bytes in
./app/data/output/sor/tb_empregados/anomesdia=20230814:
CPU times: user 8.87 ms, sys: 569 µs, total: 9.43 ms
Wall time: 7.7 ms

[9]: shape: (5, 22)

qtd_aval	qtd_aval_	qtd_aval_	qtd_aval_	...	vlr_pct_r	vlr_pct_p	
nom_insti	"dat_hor_						
---	cult	salr	benef		ecom_outr	erspec_po	
tuicao	psst"						
i64	---	---	---		_pess	s_institu	---

	i64	i64	i64		---	icao	str
f64					f64		
						f64	
13000	4100	20000	3600	...	77.0	63.0	BNP
1.6922e9							
PARIBAS							
1600	683	2800	635	...	78.0	73.0	BTG

```

1.6922e9

PACTUAL
 175      74      271      105      ...  68.0      47.0
ALFA      1.6922e9
 445      232      704      277      ...  79.0      63.0      BMG
1.6922e9
11000     3300     19000     3200     ...  81.0      66.0
BRADESCO  1.6922e9

```

```

[10]: %%time
# Gera relatório final cruzando todas as SoTs
job: job_ingestao_01_spec.ETLJob = job_ingestao_01_spec.
    ↳ ETLJob(dat_ref=DATA_REFERENCIA)
job.run()
relatorio = read_csv_files_recursively(f"./app/data/output/spec/
    ↳ report_atividade/anomesdia={DATA_REFERENCIA.strftime('%Y%m%d')}"/",
    ↳ has_header=True)
relatorio.head()

```

CPU times: user 56.2 ms, sys: 31.1 ms, total: 87.3 ms
Wall time: 86.6 ms

[10]: shape: (5, 10)

```

  nom_insti  num_cnpj_  nom_segto  max_qtd_t  ...  avg_ind_g  avg_ind_r
dat_hor_p  anomesdi
  tuicao      instituic  _institui  otl_clie_      eral_inst  emu_bene_
sst        a
  ---        ao        cao        ccs_scr        ituicao  instituic
---        ---
  str        ---        ---        ---        ---        ao
str        i64
        i64        str        i64        f64        ---
                                f64

  BRADESCO  60746948  S1        102814534  ...  4.2        4.3
2023-08-1  20230814

```

00:54:0

6.656496	BOFA	62073200	S3	4928	...	3.8	3.8	
2023-08-1	20230814							
	MERRILL							6
00:54:0								
	LYNCH							
6.656496	SICREDI	1181521	S3	7410928	...	4.4	4.1	
2023-08-1	20230814							6
00:54:0								
6.656496	CAIXA	360305	S1	148308904	...	4.4	4.3	
2023-08-1	20230814							6
	ECONOMICA							
00:54:0								
	FEDERAL							
6.656496	SANTANDER	90400888	S1	62028657	...	3.9	4.0	
2023-08-1	20230814							6
00:54:0								
6.656496								

```
[11]: import pandas as pd
relatorio = pd.read_csv(f"./app/data/output/spec/report_atividade/
↳anomesdia={DATA_REFERENCIA.strftime('%Y%m%d')}tb_relatorio_final.csv", sep=";
↳")
```

```
[12]: df = pd.DataFrame(relatorio,
↳columns=["nom_segto_instituicao", "nom_instituicao"]).
↳groupby("nom_segto_instituicao").count()
px.bar(df,
x=list(df.index),
y=list(df["nom_instituicao"]),
labels={"y": "Quantidade de Instituições",
"x": "Segmento Instituição"},
title="Quantidade de Instituições por Segmento")
```

```
[13]: df = relatorio.sort_values("max_qtd_totl_clie_ccs_scr", ascending=False)
px.bar(df,
x=list(df["nom_instituicao"]),
y=list(df["max_qtd_totl_clie_ccs_scr"]),
```

```

labels={"y": "Quantidade de Clientes",
        "x": "Instituições Financeiras"},
title="Quantidade de Clientes por Instituição Financeira")

```

```

[14]: df = relatorio.sort_values("avg_ind_recl",ascending=False)
px.bar(df,
        x=list(df["nom_instituicao"]),
        y=list(df["avg_ind_recl"]),
        labels={"y": "Índice Médio Reclamações",
                "x": "Instituições Financeiras"},
        title="Índice Médio Reclamações por Instituição Financeira")

```

```

[15]: df = relatorio.sort_values("sum_qtd_totl_recl",ascending=False)
px.bar(df,
        x=list(df["nom_instituicao"]),
        y=list(df["sum_qtd_totl_recl"]),
        labels={"y": "Quantidade Total Reclamações",
                "x": "Instituições Financeiras"},
        title="Quantidade Reclamações por Instituição Financeira")

```

```

[16]: df = relatorio.sort_values("avg_ind_geral_instituicao",ascending=False)
px.bar(df,
        x=list(df["nom_instituicao"]),
        y=list(df["avg_ind_geral_instituicao"]),
        labels={"y": "Índice Satisfação Geral Funcionários",
                "x": "Instituições Financeiras"},
        title="Índice Médio Geral Satisfação Funcionários por Instituição_
↪Financeira")

```

```

[17]: df = relatorio.sort_values("avg_ind_remu_bene_instituicao",ascending=False)
px.bar(df,
        x=list(df["nom_instituicao"]),
        y=list(df["avg_ind_remu_bene_instituicao"]),
        labels={"y": "Índice Satisfação Remuneração",
                "x": "Instituições Financeiras"},
        title="Índice Médio Satisfação Remuneração por Instituição Financeira")

```