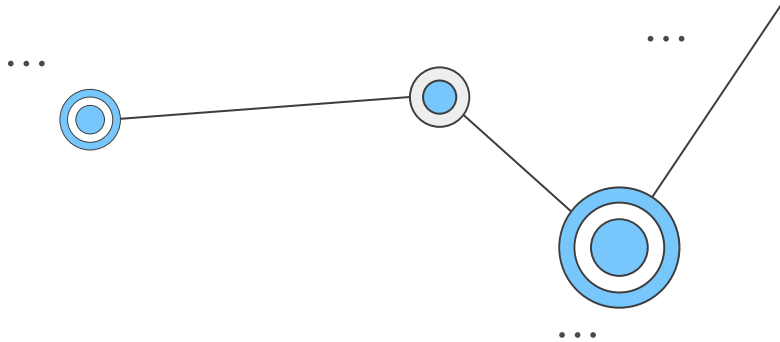




# Aplicação de Ciência de Dados em informações sobre gestão de pessoas do Poder Executivo Federal

Alef Farias  
Aldemir Melo

Tiago Paulino  
Jonas Amâncio



01

Motivação de Análise

02

Metodologia de  
tratamento dos dados

03

Metodologia de  
experimentação

04

Conclusões

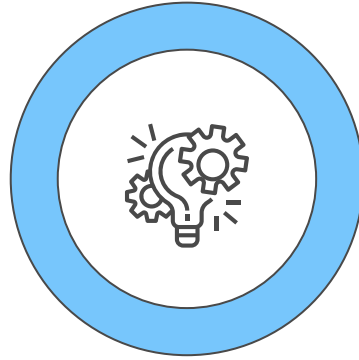




# 01

## Motivação de Análise





# Aplicação

O projeto tem como motivação à análise de dados a fim de apontar os locais onde ocorrerá um maior déficit de servidores públicos, ou seja, possíveis vagas de concurso público.

...



# Motivação



01

## Vacância

Surgimento de vagas de concurso público

02

## Órgãos

Órgãos do governo com potencial vacância

03

## Região

Região com maior índice de aposentados e abono

04

## Tempo de Aposentadoria

Tempo estimado para o servidor aposentar-se



02

# Metodologia de tratamento dos dados



# Metodologia de tratamento dos dados



## Limpeza dos dados

Nesta etapa, foram realizados procedimentos como: Remoção de dados ruidosos e inconsistências.



## Integração dos dados

Etapa onde houve integração e verificações entre ambas as bases de dados, ou seja, junção de múltiplas fontes de dados, no caso de abonos e aposentados.

# Metodologia de tratamento dos dados



## Redução dos dados

Onde ocorreu o processo de redução da dimensionalidade, com a exclusão de várias duplicatas e features que não seriam mais úteis na análise, o que levou também a redução da numerosidade desses dados.



## Transformação dos dados

A existência de dados despadronizados levou a necessidade de padronizá-los baseado no tipo com o qual fazia sentido, como por exemplo informações numéricas que em formato de Strings. Permitindo um processamento mais rápido



# Metodologia de tratamento dos dados

## Tratamento do tempo

```
#ABONOS

#ABONOS
#Separando tempo
ano_mes = list(df_abonos['Inicio_Abono'])
print(len(ano_mes))
matriz = []
for elemento in ano_mes:
    matriz.append(str(elemento))

meses = list()
anos = list()
for i in range(0, len(ano_mes)):
    meses.append(matriz[i][4]+matriz[i][5])
    anos.append(matriz[i][0]+matriz[i][1]+matriz[i][2]+matriz[i][3])

df_abonos['MÊS'] = meses
df_abonos['ANO'] = anos
```

# Metodologia de tratamento dos dados

## Tratamento de tipo

```
#Transformando valores de String para Float
df_abonos['Denominação do órgão de atuação'] = df_abonos['Denominação do órgão de atuação'].str.strip()
df_abonos['valor'] = df_abonos['valor'].str.replace(' ', '')
df_abonos['valor'] = df_abonos['valor'].str.replace(',', '.')
aux = list(df_abonos['valor'])
for i in range(0, len(aux)):
    aux[i] = float(aux[i])

df_abonos['valor'] = aux
#df_abonos
```

```
#Transformando Meses de String para Float
df_abonos['MÊS'] = df_abonos['MÊS'].str.replace(' ', '')
aux = list(df_abonos['MÊS'])
for i in range(0, len(aux)):
    aux[i] = int(aux[i])

df_abonos['MÊS'] = aux
#df_abonos
```

```
#Transformando Anos de String para Float
df_abonos['ANO'] = df_abonos['ANO'].str.replace(' ', '')
aux = list(df_abonos['ANO'])
for i in range(0, len(aux)):
    aux[i] = int(aux[i])

df_abonos['ANO'] = aux
#df_abonos
```

```
#Transformando anos_serviço_publico de String para Float
df_abonos['anos_serviço_publico'] = df_abonos['anos_serviço_publico']
aux = list(df_abonos['anos_serviço_publico'])
for i in range(0, len(aux)):
    aux[i] = int(aux[i])

df_abonos['anos_serviço_publico'] = aux
#df_abonos
```

```
#Transformando meses_serviço_publico de String para Float
df_abonos['meses_serviço_publico'] = df_abonos['meses_serviço_publico']
aux = list(df_abonos['meses_serviço_publico'])
for i in range(0, len(aux)):
    aux[i] = int(aux[i])

df_abonos['meses_serviço_publico'] = aux
df_abonos
```

# 03

## Metodologia de experimentação



04

Conclusões

