

MMT

O [Monitor do Mercado de Trabalho do Distrito Federal - MMT](#) foi uma iniciativa da Secretaria do Trabalho e Emprego do Distrito Federal, por meio da Diretoria de Monitoramento e Articulação de Oportunidades de Trabalho, desenvolvida pelo IPEDF (antiga Codeplan) em 2017. Já em 2021, a remodelagem da plataforma digital integrou o Projeto Panorama da Capacitação do Distrito Federal como um produto da pesquisa, de forma a incluir as informações do mercado de trabalho relacionadas ao ensino profissional técnico.

O MMT tem como objetivo ser um instrumento de consulta instantâneas, dinâmicas e interativas sobre o mercado de trabalho do Distrito Federal, de forma eficiente, simples e acessível. Por isso, o monitor abrange um compilado de diversas variáveis sobre trabalho e emprego.

Diante disso, o MMT pode ser visto como uma ferramenta que oferece suporte tanto para indivíduos que desejam analisar o mercado de trabalho no qual pretendem se inserir quanto para o governo na formulação de políticas públicas embasadas em evidências na área do mercado de trabalho local. Além dos painéis interativos, a ferramenta disponibiliza o download dos dados para que os usuários possam realizar suas próprias análises.

Coube à *Coordenação de Estudos e Avaliação de Políticas Socioeconômicas - CEAPS* a elaboração, tratamento e atualização dos dados do site. A atualização dos dados do MMT é anual, e atualmente utiliza os dados da *Relação Anual de Informações Sociais - RAIS* e do *Cadastro Geral de Empregados e Desempregados - CAGED*.

A estrutura atual de painéis MMT se divide da seguinte maneira:

1. Página Inicial
2. Remuneração e Ocupações
 - Remuneração por CBOs;
 - Remuneração Média;
 - Proporção Vínculos;
 - Taxa de Rotatividade;
3. Ocupações Técnicas
 - Ocupações para Empresas;
 - Empresas para Ocupações;
 - Ocupações e Eixos Tecnológicos;
 - Remuneração e Vínculos;
4. Glossário de Termos
5. Links Externos

Roteiro de execução do MMT

MMT

Roteiro de execução do MMT

1. Extração dos dados

RAIS

Pacotes e funções necessários

Importação dos dados	
Tratamento dos dados	
Dummies	
Exportação	
Dados	
Dicionários	
2. Tratamento dos dados	
Painel 2 - Remunerações e ocupações	
2.1 - Ranking de CBOs	
Importação	
Tratamento	
Exportação	
2.2 - Remuneração mediana	
Importação	
Tratamento	
Exportação	
2.3 - Número de vínculos	
Importação	
Tratamento	
Exportação	
2.4 - Taxa de rotatividade	
Importação	
Tratamento	
Exportação	
Painel 3 - Ocupações técnicas	
3.1 - Ocupações para empresas	
Importação	
Tratamento	
Exportação	
3.2 - Empresas para ocupações	
Importação	
Tratamento	
Exportação	
3.3 - Ocupações e eixos tecnológicos	
Importação	
Tratamento	
Exportação	
3.4 - Remuneração e vínculos	
Importação	
Tratamento	
Exportação	

Este arquivo descreve detalhadamente todas as etapas para limpar o banco de dados usada para atualizar o Monitor do Mercado de Trabalho - MMT. Essas etapas incluem importar, limpar, organizar e transformar os dados. Tentamos cumprir as melhores práticas de processamento de dados dentro da estrutura do `tidyverse`.

1. Extração dos dados

RAIS

O arquivo `RAIS.R` se encontra em **1. Extração dos dados/RAIS**. Abra o projeto **RAIS.RProj** para executar o *script* corretamente.

Pacotes e funções necessários

Os pacotes, funções e configurações necessários para execução do *script* estão descritos abaixo:

```
library(odbc)
library(tidyverse)
library(sidrar)

`%notin%` <- Negate(`%in%`)
options(readr.show_col_types = FALSE)
```

O pacote `odbc` é necessário para realizar a conexão ao banco de dados `DB_CODEPLAN` e realizar a importação dos dados da RAIS para o ambiente `R`. O pacote `tidyverse` é responsável pela manipulação dos dados. O pacote `sidrar` é responsável por realizar o download das informações atualizadas do INPC-DF, utilizado para deflacionar os valores. A função `%notin%` funcionará como filtro para os dados.

Importação dos dados

As CBOs (protegidas e técnicas de nível médio e superior) são importadas para o ambiente `R` com o comando abaixo:

```
cbotecnica_nivel_medio <- as.character(read_csv("../Dados/CBO Técnica - Nível
médio.csv"))[[1]])
cbotecnica_nivel_superior <- as.character(read_csv("../Dados/CBO Técnica - Nível
superior.csv"))[[1]])
cbos_protegidas <- as.character(read_csv("../Dados/CBOs protegidas.csv"))[[1]])
```

As CBOs que compõe os eixos tecnológicos são importadas e realizamos o ajuste dos nomes de cada eixo.

```

### Leitura ----

eixos <- lapply(paste0("../Dados/Eixos - nível médio/", dir("../Dados/Eixos -
nível médio")), function(x) {read_delim(x, delim = ";", locale = locale(encoding =
"UTF-8"))})

eixos_superior <- lapply(paste0("../Dados/Eixos - nível superior/",
dir("../Dados/Eixos - nível superior")), function(x) {read_delim(x, delim = ";",
locale = locale(encoding = "ISO-8859-1"))})

### Padronização dos nomes ----

names(eixos) <- vapply(eixos, function(x){names(x)[[2]]}, "eixo")
names(eixos_superior) <- vapply(eixos_superior, function(x){names(x)[[2]]},
"eixo")

```

O INPC-DF em índice é importado diretamente do [Sistema IBGE de Recuperação Automática - SIDRA](#) utilizando o pacote `sidrar`. Fazemos o cálculo do INPC-DF acumulado por ano e posteriormente, calculamos o deflator com base nos preços do ano de 2022.

```

inpc <- get_sidra(api = "/t/2951/n6/5300108/v/44/p/all/c315/7169/d/v44%202") |>
# Até 2011
rbind(get_sidra(api = "/t/1100/n6/5300108/v/44/p/all/c315/7169/d/v44%202")) |>
# De 2012 até 2019
rbind(get_sidra(api = "/t/7063/n6/5300108/v/44/p/all/c315/7169/d/v44%202")) |>
# De 2020 em diante
select(data = "Mês (Código)",
inpc = "valor") |>
mutate(data = ym(data),
inpc = inpc / 100,
referencia = year(data)) |>
filter(referencia > 2009) |>
group_by(referencia) |>
summarise(inpc_anual = prod(1 + inpc)) |>
mutate(inpc_acumulado = cumprod(inpc_anual),
deflator = inpc_acumulado / nth(inpc_acumulado, -2))

```

Para realizar a importação da RAIS, realizamos a conexão ao banco de dados `DB_CODEPLAN` para verificar qual o ano mais recente disponível e, posteriormente, realizamos a importação, ano a ano, com as seguintes colunas selecionadas filtrando para dados apenas do DF:

- **referencia** - ano de referência dos dados;
- **vinculoativo3112** - se estava empregado;
- **tipovinculo** - o tipo de vínculo empregatício que o trabalhador possui;
- **escolaridade_2006_atual** - nível de escolaridade mais alto do trabalhador;
- **sexotrabalhador** - sexo do trabalhador;
- **racacor** - raça do trabalhador;
- **vlremdeznm** - valor da remuneração mensal do trabalhador;
- **vlremdezsm** - valor da remuneração mensal do trabalhador em salários mínimos;

- **tiposal** - tipo de salário do trabalhador (fixo, variável ou misto);
- **tempoemprego** - tempo de emprego do trabalhador no estabelecimento;
- **qtdhoracontr** - quantidade de horas efetivamente trabalhadas pelo trabalhador na semana;
- **vl_salario_contrato** - valor do salário contratado do trabalhador;
- **cbo2002** - CBO que identifica a ocupação do trabalhador;
- **cnae20subclasse** - subclasse da atividade econômica da empresa;
- **idade** - idade do trabalhador.

Os dados importados correspondem do período inicial de 2011 até o ano mais recente disponível da RAIS. Depois de extrair e empilhar cada ano, juntamos os dados da RAIS em um `data.frame` com o deflator do INPC-DF:

```
### Conexão ao Banco de Dados ----

db <- DBI::dbConnect(odbc(),
                     "db_codeplan",
                     uid = Sys.getenv("matricula"),
                     pwd = Sys.getenv("senha"))

ano <- as.numeric(substr(DBI::dbGetQuery(db, "SELECT TOP 2 TABLE_NAME
                                           FROM
DB_CODEPLAN.INFORMATION_SCHEMA.TABLES
                                           WHERE TABLE_TYPE = 'BASE TABLE' AND
TABLE_NAME LIKE 'vinc_202%'
                                           ORDER BY LEN(TABLE_NAME) DESC;")
[[1]][1], 6, 10))

### Objeto RAIS para o loop ----

rais <- NULL

### Loop que extrai e empilha os dados ----

for (i in 2011:ano) {

  temp <- DBI::dbGetQuery(db, paste0("SELECT referencia, vinculoativo3112,
tipovinculo, escolaridade_2006_atual as escolaridade, sexotrabalhador, racacor,
vlremdeznm as vlremundezebrnom, vlremdezsm as vlremundezebrosm, tiposal as
tiposalario, tempoemprego, qtdhoracontr, vl_salario_contrato as
vlsalariocontratual, cbo2002 as cboocupacao2002, cnae20subclasse, idade, null as
indtrabintermitente FROM DB_CODEPLAN.rais_id.vinc_", i, " WHERE municipio =
530010 and vinculoativo3112 = 1"))
  rais <- rbind(rais, temp)

  print(i)

}

### Fecha conexão com o Banco de Dados ----

dbDisconnect(db)
remove(temp, i)
```

```
dados <- left_join(raiz, inpc, by = "referencia")
```

Tratamento dos dados

Uma vez com todos os dados necessários importados, partimos para o tratamento, onde serão aplicados filtros e transformações no formato dos dados. O primeiro filtro é o de idade, deixando na base apenas trabalhadores maiores de 18 anos.

```
dados <- dados |>
  filter(idade >= 18)
```

O próximo filtro é de horas trabalhadas na semana, permanecendo na base apenas trabalhadores que trabalham mais de 10 horas por semana.

```
dados <- dados |>
  filter(qtdhoracontr > 10)
```

Após, retiramos da base as CBOs que são protegidas (~~adicionar definição de CBO protegida~~).

```
dados <- dados |>
  filter(cboocupacao2002 %notin% cbos_protegidas)
```

O último passo do tratamento dos dados é a criação de novas variáveis. Primeiramente, deflacionamos os salários a preços de 2022 na variável `vlremundezembrnom`. Após, criamos a variável `horas_mensais` a partir da variável `qtdhoracontr` e por último, calculamos o salário hora dos trabalhadores.

```
dados <- dados |>
  mutate(salario_dez_defl = vlremundezembrnom / deflator,
         horas_mensais = qtdhoracontr * 4,
         salario_hora = salario_dez_defl / horas_mensais)
```

Dummies

As dummies serão utilizadas posteriormente como filtro das variáveis. As primeiras dummies criadas são relacionadas às CBOs técnicas. A dummy `cbo_tec_em_complet = 1` indica os trabalhadores pertencentes às CBOs técnicas de **nível médio**, enquanto que `cbo_tec_sup = 1` indica os trabalhadores pertencentes às CBOs técnicas de **nível superior**. A dummy `cbo_tec = 1` indica que o trabalhador é de uma CBO técnica, seja de nível médio ou superior.

```
dados <- dados |>
  mutate(cbo_tec_em_complet = case_when(cboocupacao2002 %in%
    cbotecnica_nivel_medio & escolaridade %in% 7:8 & tipovinculo != 55 ~ 1, TRUE ~ 0),
         cbo_tec_sup = case_when(cboocupacao2002 %in% cbotecnica_nivel_superior &
    escolaridade %in% 9:11 & tipovinculo != 55 ~ 1, TRUE ~ 0),
         cbo_tec = case_when(cbo_tec_em_complet == 1 | cbo_tec_sup == 1 ~ 1, TRUE ~ 0))
```

A dummy de tipo de vínculo indica o tipo de vínculo empregatício que o trabalhador possui. Para os propósitos do MMT, apenas trabalhadores **celetistas** são considerados nas análises. São considerados trabalhadores celetistas os seguintes códigos da RAIS:

- **10** - trabalhador com contrato de trabalho por tempo indeterminado ou por prazo determinado, com uma pessoa jurídica que é empregadora individual;
- **15** - trabalhador com contrato de trabalho por tempo indeterminado ou por prazo determinado, com uma pessoa física que é empregadora individual;
- **20** - trabalhador com contrato de trabalho por tempo indeterminado ou por prazo determinado, com uma pessoa jurídica de direito privado;
- **25** - trabalhador com contrato de trabalho por tempo indeterminado ou por prazo determinado, com uma pessoa física de direito privado;
- **60** - trabalhador diretor de empresa ou sociedade vinculado ao empregador pessoa jurídica;
- **65** - trabalhador diretor de empresa ou sociedade vinculado ao empregador pessoa física;
- **70** - trabalhador rural vinculado a empregador pessoa jurídica por contrato de trabalho por tempo determinado ou por obra certa;
- **75** - trabalhador rural vinculado a empregador pessoa física por contrato de trabalho por tempo determinado ou por obra certa.

```
dados <- dados |>
  mutate(tipovinculo = case_when(tipovinculo %in% c(10, 15, 20, 25, 60, 65, 70,
75) ~ 1, # Celetista - 1
                                     tipovinculo %in% c(30, 31, 35) ~ 2,
                                     # Estatutário - 2
                                     tipovinculo == 55 ~ 3,
                                     # Aprendiz - 3
                                     tipovinculo %in% c(40, 50, 80, 90, 95, 96, 97,
-1) ~ 4)) # Outros - 4
```

A dummy de escolaridade agrupa os códigos de escolaridade em faixas. Os códigos são os seguintes:

- **1** - analfabeto;
- **2** - até o 5º ano incompleto do ensino fundamental (antiga 4ª série);
- **3** - 5º ano completo do ensino fundamental;
- **4** - do 6º ao 9º ano do ensino fundamental incompleto (antiga 5ª à 8ª série);
- **5** - ensino fundamental completo;
- **6** - ensino médio incompleto;
- **7** - ensino médio completo;
- **8** - educação superior incompleta;
- **9** - educação superior completa;
- **10** - mestrado completo;
- **11** - doutorado completo.

As faixas que foram definidas são:

- **1** - analfabeto;

- Entre **2 e 6** - ensino médio incompleto;
- Entre **7 e 8** - ensino médio completo;
- Entre **9 e 11** - superior completo (englobando pós-graduações).

```
dados <- dados |>
  mutate(escolaridade = case_when(escolaridade == 1 ~ 1,      # Analfabeto - 1
                                  escolaridade %in% 2:6 ~ 2,  # Médio incompleto
                                  escolaridade %in% 7:8 ~ 3,  # Médio completo -
                                  escolaridade %in% 9:11 ~ 4)) # Superior
completo - 4
```

As dummies dos eixos tecnológicos de nível médio e de nível superior informam se a CBO do trabalhador corresponde à alguma CBO técnica, seja de nível médio ou superior.

```
### Eixos - nível médio ----

dados <- dados |>
  mutate(eixo_amb_saude_em      = case_when(cboocupacao2002 %in% eixos$`Eixo de
Ambiente e Saúde`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_conteprocessos_em = case_when(cboocupacao2002 %in% eixos$`Eixo de
Controle e Processos Industriais`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_desedusoc_em      = case_when(cboocupacao2002 %in% eixos$`Eixo de
Desenvolvimento Educacional e Social`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~
0),
         eixo_negocios_em       = case_when(cboocupacao2002 %in% eixos$`Eixo de
Gestão e Negócios`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_infoecomunic_em   = case_when(cboocupacao2002 %in% eixos$`Eixo de
Informação e Comunicação`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_infraestrutura_em = case_when(cboocupacao2002 %in% eixos$`Eixo de
Infraestrutura`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_prodaliment_em    = case_when(cboocupacao2002 %in% eixos$`Eixo de
Produção Alimentícia`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_prodcult_em       = case_when(cboocupacao2002 %in% eixos$`Eixo de
Produção Cultural e Design`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_proindust_em      = case_when(cboocupacao2002 %in% eixos$`Eixo de
Produção Industrial`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_recnaturais_em    = case_when(cboocupacao2002 %in% eixos$`Eixo de
Recursos Naturais`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_seguranca_em     = case_when(cboocupacao2002 %in% eixos$`Eixo de
Segurança`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_hospelazer_em     = case_when(cboocupacao2002 %in% eixos$`Eixo de
Turismo, Hospitalidade e Lazer`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0),
         eixo_militar_em       = case_when(cboocupacao2002 %in% eixos$`Eixo
Militar`[[2]] & cbo_tec_em_complet == 1 ~ 1, TRUE ~ 0))

### Eixos - nível superior ----

dados <- dados |>
  mutate(eixo_amb_saude_sup      = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Ambiente e Saúde`[[2]] & cbo_tec_sup == 1 ~ 1, TRUE ~ 0),
```



```

eixo_conteprocessos_sup = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Controle e Processos Industriais`[[2]] & cbo_tec_sup == 1
~ 1, TRUE ~ 0),
eixo_desedusoc_sup      = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Desenvolvimento Educacional e Social`[[2]] & cbo_tec_sup
== 1 ~ 1, TRUE ~ 0),
eixo_negocios_sup       = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Gestão e Negócios`[[2]] & cbo_tec_sup == 1 ~ 1, TRUE ~
0),
eixo_infoecomunic_sup   = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Informação e Comunicação`[[2]] & cbo_tec_sup == 1 ~ 1,
TRUE ~ 0),
eixo_infraestrutura_sup = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Infraestrutura`[[2]] & cbo_tec_sup == 1 ~ 1, TRUE ~ 0),
eixo_prodaliment_sup    = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Produção Alimentícia`[[2]] & cbo_tec_sup == 1 ~ 1, TRUE ~
0),
eixo_prodcult_sup       = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Produção Cultural e Design`[[2]] & cbo_tec_sup == 1 ~ 1,
TRUE ~ 0),
eixo_proindust_sup      = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Produção Industrial`[[2]] & cbo_tec_sup == 1 ~ 1, TRUE ~
0),
eixo_recnaturais_sup    = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Recursos Naturais`[[2]] & cbo_tec_sup == 1 ~ 1, TRUE ~
0),
eixo_seguranca_sup     = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Segurança`[[2]] & cbo_tec_sup == 1 ~ 1, TRUE ~ 0),
eixo_hospelazer_sup     = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo de Turismo, Hospitalidade e Lazer`[[2]] & cbo_tec_sup == 1 ~
1, TRUE ~ 0),
eixo_militar_sup        = case_when(cboocupacao2002 %in%
eixos_superior$`Eixo Militar`[[2]] & cbo_tec_sup == 1 ~ 1, TRUE ~ 0))

```

Por último, convertamos a classe da coluna de CNAEs (`cnae20subclasse`) de `integer` para `character`.

```

dados <- dados |>
mutate(cnae20subclasse = as.character(cnae20subclasse))

```

Exportação

Após a importação, organização e tratamento dos dados, é exportado um arquivo `.RDS` para a pasta **1. Extração dos dados/Dados** com o nome de `RAIS.RDS`.

```

saveRDS(dados, file = "Dados/RAIS.RDS")

```

Dados

A pasta **/Dados** armazena todos os dados necessários para a execução do *script* de atualização da RAIS. Estão contidos:

- Os arquivos `.csv` dos 13 eixos tecnológicos de nível médio;
- Os arquivos `.csv` dos 12 eixos tecnológicos de nível superior;

- Os arquivos `.csv` de todas as CBOs técnicas de nível médio e nível superior;
- O arquivo `.csv` das CBOs protegidas;
- O arquivo final `.RDS` da RAIS.

Dicionários

A pasta **/Dicionários** armazena todos os dicionários utilizados para a conversão dos códigos de CBOs e CNAEs nos seus respectivos nomes, além dos dicionários do CAGED, CAGED novo e RAIS. Os dicionários são:

- CAGED;
- CAGED novo;
- CBO;
- CNAE;
- CNAE para setor CNAE;
- RAIS.

2. Tratamento dos dados

Painel 2 - Remunerações e ocupações

Para atualizar todos os painéis do painel 2, basta acessar **2. Tratamento dos dados** e abrir o projeto **Painéis.RProj** para executar o *script* corretamente. Após, execute o *script* `Painel 2.R`.

Os pacotes, funções e configurações necessários para execução do *script* estão descritos abaixo:

```
library(tidyverse)
library(imputeTS)
library(readxl)
library(odbc)

`%notin%` <- Negate(`%in%`)

foiAtualizado <- function(arquivo) {

  data <- Sys.Date()
  data_arquivo <- as.Date(file.info(paste0("Painel 2 - Remuneração e
Ocupações/Resultados/", arquivo, ".csv"))$mtime)

  if (data >= data_arquivo) {
    return(TRUE)
  } else {
    return(FALSE)
  }
}
```

O pacote `odbc` é necessário para realizar a conexão ao banco de dados `DB_CODEPLAN` e realizar a importação dos dados do CAGED para o ambiente `R`. O pacote `tidyverse` é responsável pela manipulação dos dados. O pacote `readxl` é responsável por ler arquivos `.xls` e `.xlsx`. O pacote `imputeTS` é utilizado para retirar `NAs` (números não disponíveis) da base. A função

`%notin%` funcionará como filtro para os dados. A função `foiAtualizado` retorna valores booleanos do tipo `TRUE` e `FALSE` e é utilizada para verificar se todos os painéis foram atualizados sem erros.

O objeto `RAIS.RDS` é importado nesse *script* e será fonte de dados de todos os *scripts* contidos na pasta **2. Tratamento dos dados/Painel 2 - Remuneração e Ocupações**.

2.1 - Ranking de CBOs

O ranking de CBOs é o *script* responsável por gerar os dados do painel 2.1. Aqui são gerados os rankings das famílias de CBOs com a maior e menor variação de vínculos e de salários, além das tabelas com dados de vínculos e mediana de salários.

O ranking é calculado utilizando como base dados do último ano disponível (período t) em comparação com o ano imediatamente anterior (período t_{-1}).

Importação

O dicionário de CBOs é importado e são selecionadas as colunas de interesse. Logo após, é criado um vetor com os códigos das famílias de CBOs que concentram 99% do total de vínculos. Esse vetor será utilizado para realizar um filtro na base.

```
estrutura_cbo <- readRDS("../1. Extração dos dados/Dicionários/Dicionário  
CBO.RDS") |>  
  select(cboocupacao2002, nome_cbo_ocupacao,  
         cbo_familia, nome_cbo_familia)
```

```
cbos_filtro <- c(4110, 5211, 5143, 4132, 5173, 3222, 5134, 5174, 4211,  
                4221, 5142, 4141, 2124, 2235, 7170, 1423, 5132, 7823,  
                2251, 2521, 5135, 1421, 4101, 7832, 7825, 5141, 4223,  
                7824, 7152, 3541, 5112, 4131, 5171, 7842, 2523, 3132,  
                8485, 3515, 2312, 2234, 8483, 9144, 5191, 1414, 2394,  
                3242, 4122, 2522, 3172, 2410, 3224, 4142, 7155, 3311,  
                2236, 6220, 7156, 5199, 9922, 3171, 3331, 5201, 3714,  
                2142, 7241, 2311, 3425, 2611, 7102, 4222, 7841, 2345,  
                4152, 3341, 3133, 3241, 6210, 3131, 2241, 7321, 5163,  
                5133, 1415, 5152, 2524, 9511, 4201, 2321, 5136, 2525,  
                4102, 7166, 2123, 2237, 7822, 7711, 3513, 2313, 3517,  
                7244, 1425, 7151, 3912, 2532, 3542, 7153, 3516, 2143,  
                3514, 4151, 2515, 3121, 5101, 4213, 6233, 8414, 5193,  
                5162, 7741, 2516, 3141, 8621, 9913, 3911, 1422, 2332,  
                2624, 5153, 2344, 2531, 9112, 7632, 9113, 3511, 3951,  
                7242, 2512, 4121, 2612, 6410, 7243, 1231, 3421, 5103,  
                2346, 7631, 7313, 1412, 5151, 9143, 9921, 8118, 3221,  
                7630, 3252, 9541, 7154, 3731, 7165, 9513, 7233, 1416,  
                2212, 3144, 5231, 1312, 3424, 3312, 3711, 5241, 2034,  
                1313, 2238, 7831, 7163, 3732, 3548, 2711, 7311, 3250,  
                2141, 7662, 3423, 4212, 2122, 1424, 2348, 1114, 2232,  
                2149, 6230, 8623, 7312, 5161, 7257, 3741, 9101, 5192,  
                9531, 1417, 5111, 8481, 7663, 7826, 7164, 3185, 2239,  
                8181, 8401, 3722, 2543, 8418, 1311, 2221, 3134, 7157,  
                3123, 3251, 3143, 7661, 2252, 9501, 1427, 7212, 7245,  
                7112, 3744, 1210, 3011, 9192, 6231, 9191, 2617, 6223,  
                3181, 3532, 7122, 3122, 2613, 1426, 2211, 2233, 2144,  
                3111, 6232, 3751, 8131, 5242, 9141, 7250, 2153, 8214,  
                2347, 3003, 7827, 9131, 7821, 3412, 1233, 6321, 3180,
```

```
7522, 2614, 5165, 9111, 5121, 7664, 3522, 2621, 2527,  
9102, 5131, 3721, 7652, 7213, 8625, 9193, 4241, 3188,  
7111, 3115, 4231, 7252, 5102, 7411)
```

Tratamento

O tratamento dos dados começa agrupando os dados da RAIS com o dicionário de CBOs e deixando apenas as CBOs que constituem 99% do total de vínculos na base.

```
rais_1 <- rais |>  
  left_join(estrutura_cbo |>  
    select(cboocupacao2002, cbo_familia)) |>  
  filter(cbo_familia %in% cbos_filtro)
```

Calculamos os dados do total de vínculos e mediana dos salários, tanto do último ano (período t), como do ano base (período $t-1$) com os códigos abaixo:

```
### Vínculos ----  
  
dados_recente_vinculos <- rais_1 |>  
  filter(tipovinculo == 1, # Celetista  
    referencia == max(referencia)) |> # Filtar pelo ano mais recente  
  group_by(cbo_familia) %>%  
  summarise(referencia = max(referencia),  
    vinculos_recente = n())  
  
### Salários ----  
  
dados_recente_salarios <- rais_1 |>  
  filter(tipovinculo == 1, # Celetista  
    referencia == max(referencia), # Filtar pelo ano mais recente  
    salario_hora > 0) |>  
  group_by(cbo_familia) %>%  
  summarise(referencia = max(referencia),  
    mediana_rendimento_recente = median(salario_dez_defl, na.rm = TRUE),  
    # Salário mediano  
    mediana_salario_hora_recente = median(salario_hora, na.rm = TRUE))  
  # Hora de trabalho mediana  
  
dados_recente <- dados_recente_salarios |>  
  left_join(dados_recente_vinculos)  
  
remove(dados_recente_salarios, dados_recente_vinculos)
```

Após realizar o cálculo para os dois períodos, agrupamos as duas bases, junto com o dicionário de CBOs e calculamos a variação de rendimentos e de vínculos com as seguintes fórmula para rendimentos e vínculos:

$$rendimento = \frac{\text{mediana do rendimento}_t}{\text{mediana do rendimento}_{t-1}} - 1$$

e

$$vinculos = \frac{\text{total de vinculos}_t}{\text{total de vinculos}_{t-1}} - 1$$

O código que aplicará essas fórmulas está descrito abaixo:

```
dados <- dados_base |>
  left_join(dados_recente, by = "cbo_familia") |>
  left_join(estrutura_cbo |>
    select(cbo_familia, nome_cbo_familia),
    by = "cbo_familia") |>
  mutate(variacao_rendimento = (mediana_rendimento_recente /
mediana_rendimento_base) - 1,
    variacao_vinculos = (vinculos_recente / vinculos_base) - 1) |>
  distinct_all()
```

Exportação

Após a importação, organização e tratamento dos dados, é exportado um arquivo `.csv` para a pasta **2. Tratamento dos dados/Painel 2 - Remuneração e Ocupações/Resultados** com o nome de `2.1 - Ranking de CBOs.csv`.

```
write_excel_csv2(dados, "Painel 2 - Remuneração e Ocupações/Resultados/2.1 -
Ranking de CBOs.csv")
```

2.2 - Remuneração mediana

A remuneração mediana é o *script* responsável por gerar os dados do painel 2.2. Aqui são gerados os valores da mediana de salário por hora trabalhada e salário mensal dividido por ano, faixas de escolaridade e agregado por trabalhadores técnicos e não técnicos.

Importação

A RAIS é importada com alguns filtros e os nomes de algumas colunas é substituído.

```
rais_2 <- rais |>
  filter(tipovinculo == 1,          # Celetista
    referencia != 2011,
    !is.na(cboocupacao2002)) |>
  select(ano = referencia,
    horas = horas_mensais,
    salario_hora,
    salario_dez_defl,
    tec = cbo_tec,
    tec_em = cbo_tec_em_complet,
    tec_sup = cbo_tec_sup,
    escolaridade)
```

O filtro `tipovinculo == 1` é para deixar apenas celetistas na base. O ano de 2011 e todas as linhas da coluna que contém os códigos das CBOs que são `NA` são retirados.

Tratamento

O tratamento dos dados começa separando a base em empregos técnicos e empregos não técnicos. Realizamos um filtro para deixar apenas trabalhadores técnicos e outro para retirar trabalhadores que estão listados com salários e horas trabalhadas iguais a 0.

Agrupamos os dados por ano e tipo de trabalhador técnico (de nível médio ou de nível superior) para calcular a mediana do salário hora e do salário mensal. Após, realizamos o cálculo novamente, mas apenas por ano, e com um `rbind()` juntamos todos os dados.

```
base_tec <- rais_2 |>
  filter(tec == 1,
         salario_hora > 0,
         horas > 0) |>
  mutate(tipo = case_when(tec_em == 1 ~ "Técnicos de nível médio",
                          tec_sup == 1 ~ "Técnicos de nível superior")) |>
  group_by(ano, tipo) |>
  summarise(salario_hora = median(salario_hora, na.rm = T),
            salario_mes = median(salario_dez_defl, na.rm = T),
            filtro = "Técnico") |>
rbind(rais_2 |>
      filter(tec == 1,
             salario_hora > 0,
             horas > 0) |>
      group_by(ano) |>
      summarise(salario_hora = median(salario_hora, na.rm = T),
                salario_mes = median(salario_dez_defl, na.rm = T),
                tipo = "Trabalhadores técnicos",
                filtro = "Técnico"))
```

Para a base de trabalhadores não técnicos, a coluna `tipo` passa a utilizar dados de escolaridade, porém, a estrutura do cálculo se mantém a mesma.

```
base_nao_tec <- rais_2 |>
  filter(tec == 0,
         salario_hora > 0,
         horas > 0) |>
  mutate(tipo = case_when(escolaridade == 1 ~ "Analfabeto",
                          escolaridade == 2 ~ "Até fundamental completo",
                          escolaridade == 3 ~ "Até médio completo",
                          escolaridade == 4 ~ "Superior completo")) |>
  group_by(ano, tipo) |>
  summarise(filtro = "Não técnico",
            salario_hora = median(salario_hora, na.rm = T),
            salario_mes = median(salario_dez_defl, na.rm = T)) |>
rbind(rais_2 |>
      filter(tec == 0,
             salario_hora > 0,
             horas > 0) |>
      group_by(ano) |>
      summarise(tipo = "Trabalhadores não técnicos",
                filtro = "Não técnico",
                salario_hora = median(salario_hora, na.rm = T),
                salario_mes = median(salario_dez_defl, na.rm = T)))
```

O último passo do tratamento dos dados é juntar as base de trabalhadores técnicos e trabalhadores não técnicos. Empilhamos as duas bases e criamos uma nova coluna de filtro.

```
base_tipo_emprego <- rbind(base_tec, base_nao_tec) |>
  mutate(geral = case_when(tipo %in% c("Trabalhadores técnicos", "Trabalhadores
  não técnicos") ~ "Geral",
                           TRUE ~ "Subgrupo")) |>
  arrange(tipo, ano)
```

Exportação

Após a importação, organização e tratamento dos dados, é exportado um arquivo `.csv` para a pasta **2. Tratamento dos dados/Painel 2 - Remuneração e Ocupações/Resultados** com o nome de `2.2 - Remuneração mediana.csv`.

```
write_excel_csv2(base_tipo_emprego, "Painel 2 - Remuneração e
  Ocupações/Resultados/2.2 - Remuneração mediana.csv")
```

2.3 - Número de vínculos

O número de vínculos é o *script* responsável por gerar os dados do painel 2.3. Aqui são gerados o número absoluto de vínculos por escolaridade e por nível técnico agrupados por ano.

Importação

A RAIS é importada com alguns filtros e os nomes de algumas colunas é substituído.

```
rais_3 <- rais |>
  filter(tipovinculo == 1,          # Celetista
         !is.na(cboocupacao2002)) |>
  select(ano = referencia,
         tec = cbo_tec,
         cbo_tec_em_complet,
         cbo_tec_sup,
         escolaridade)
```

O filtro `tipovinculo == 1` é para deixar apenas celetistas na base. Todas as linhas da coluna que contém os códigos das CBOs que são `NA` são retirados. As colunas `cbo_tec_em_complet` e `cbo_tec_sup` são variáveis do tipo *dummy* para identificar se o trabalhador é de uma CBO técnica de nível médio ou de nível superior, enquanto que a coluna `tec` apenas identifica se o trabalhador é de uma CBO técnica, sem distinção de nível.

Tratamento

O tratamento dos dados começa aplicando o filtro `tec == 1` para deixar apenas CBOs técnicas na base. Criamos a variável `tipo` para informar se o trabalhador é técnico de nível médio ou de nível superior.

Os dados são agrupados por ano e por tipo de trabalhador para após ser calculado o número de vínculos. Empilhamos os dados agrupando apenas por ano para gerar o número total de vínculos de trabalhadores técnicos, seja de nível médio ou superior.

```
base_tec <- rais_3 |>
  filter(tec == 1) |>
  mutate(tipo = case_when(cbo_tec_em_complet == 1 ~ "Técnico de nível médio",
                          cbo_tec_sup == 1 ~ "Técnico de nível superior")) |>
  group_by(ano, tipo) |>
  summarise(vinculos = n()) |>
  rbind(rais_3 |>
    filter(tec == 1) |>
    group_by(ano) |>
    summarise(vinculos = n(),
              tipo = "Trabalhadores técnicos"))
```

O próximo passo consiste na realização do mesmo cálculo, porém, agrupado por escolaridade. Criamos a variável `tipo` para informar o nível de escolaridade do trabalhador. Para essa operação, não é aplicado o filtro de trabalhadores técnicos.

```
base_escolaridade <- rais_3 |>
  mutate(tipo = case_when(escolaridade == 1 ~ "Analfabeto",
                          escolaridade == 2 ~ "Até fundamental completo",
                          escolaridade == 3 ~ "Médio completo",
                          escolaridade == 4 ~ "Superior completo")) |>
  group_by(ano, tipo) |>
  summarise(vinculos = n())
```

O penúltimo passo consiste no cálculo agrupado por ano aplicando o filtro para deixar apenas trabalhadores não técnicos.

```
base_nao_tec <- rais_3 |>
  filter(tec == 0) |>
  group_by(ano) |>
  summarise(vinculos = n(),
            tipo = "Trabalhadores não técnicos")
```

Com todas as variáveis devidamente calculadas, empilhamos as base em um único `data.frame`. Juntamos as base de trabalhadores técnicos e trabalhadores não técnicos para calcular o total de vínculos e após, empilhamos todas as bases (número de vínculos de trabalhadores técnicos, trabalhadores não técnicos e por escolaridade).

```
base_tipo_emprego <- rbind(base_tec, base_nao_tec) |>
  group_by(ano) |>
  summarise(vinculos = sum(vinculos),
            tipo = "Total") |>
  rbind(base_tec, base_nao_tec, base_escolaridade) |>
  arrange(tipo, ano)
```

Exportação

Após a importação, organização e tratamento dos dados, é exportado um arquivo `.csv` para a pasta **2. Tratamento dos dados/Painel 2 - Remuneração e Ocupações/Resultados** com o nome de `2.3 - Número de vínculos.csv`.


```
write_excel_csv2(base_tipo_emprego, "Painel 2 - Remuneração e  
Ocupações/Resultados/2.3 - Número de vínculos.csv")
```

2.4 - Taxa de rotatividade

A taxa de rotatividade é o *script* responsável por gerar os dados do painel 2.4. Aqui são gerados os números da taxa de rotatividade por faixas de escolaridade e por tipo de trabalhador (técnico ou não).

A taxa de rotatividade pode ser definida como o movimento recorrente de substituição de parte da força de trabalho utilizada em cada ciclo produtivo anual, através de demissões e admissões de trabalhadores que são realizadas dentro de um país e/ou unidade da federação. Tal movimento de substituição é medido por meio da taxa de rotatividade.

Os pacotes, funções e configurações necessários para execução do *script* estão descritos abaixo:

```
library(odbc)  
library(tidyverse)  
library(readxl)  
  
`%notin%` <- Negate(`%in%`)
```

O pacote `odbc` é necessário para realizar a conexão ao banco de dados `DB_CODEPLAN` e realizar a importação dos dados do CAGED para o ambiente `R`. O pacote `tidyverse` é responsável pela manipulação dos dados. O pacote `readxl` será necessário para ler arquivos `.xlsx` e `.xls`. E, por fim, a função `%notin%` funcionará como filtro para os dados.

Importação

Para realizar a importação do CAGED, realizamos a conexão ao banco de dados `DB_CODEPLAN` e, posteriormente, realizamos a importação, ano a ano, com as seguintes colunas selecionadas filtrando para dados apenas do DF:

- **anodeclarado** - ano declarado dos dados;
- **competencia** - competência (mês/ano) em que a movimentação foi declarada;
- **cbo2002ocupacao** - Classificação Brasileira de Ocupações;
- **grauinstrucao** - grau de instrução;
- **racacor** - raça do trabalhador;
- **sexo** - sexo do trabalhador;
- **idade** - idade do trabalhador;
- **indtrabintermitente** - contrato intermitente
- **indaprendiz** - referente a contrato de aprendizagem
- **salariomensal** - salário mensal do indivíduo;
- **tipomovdesagregado** - tipo de movimentação;
- **admitidosdesligados** - capta se o trabalhador foi admitido ou desligado;
- **saldomov** - saldo de movimentação.

```
### Novo CAGED ----
```

```

novo_caged <- DBI::dbGetQuery(db,
"SELECT
  CAST(SUBSTRING(CAST(competencia AS CHAR(6)), 1, 4) AS INT) AS anodeclarado,
  CAST(SUBSTRING(CAST(competencia AS CHAR(6)), 5, 2) AS INT) AS mesdeclarado,
  competencia,
  cbo2002ocupacao,
  graudeinstrucao AS grauinstrucao,
  racacor,
  sexo,
  idade,
  indtrabintermitente,
  indicadoraprendiz AS indaprendiz,
  CAST(REPLACE(valorsalariofixo, ',', '.') AS FLOAT) AS salariomensal,
  tipomovimentacao AS tipomovdesagregado,
  CASE WHEN saldo = 1 THEN 1 ELSE 2 END AS admitidosdesligados,
  saldo AS saldomov,
  'dp' AS tipo
FROM
  caged.caged_202001_atual_MOV
WHERE
  UF = 53
UNION ALL
SELECT
  CAST(SUBSTRING(CAST(competencia AS CHAR(6)), 1, 4) AS INT) AS anodeclarado,
  CAST(SUBSTRING(CAST(competencia AS CHAR(6)), 5, 2) AS INT) AS mesdeclarado,
  competencia,
  cbo2002ocupacao,
  graudeinstrucao AS grauinstrucao,
  racacor,
  sexo,
  idade,
  indtrabintermitente,
  indicadoraprendiz AS indaprendiz,
  CAST(REPLACE(valorsalariofixo, ',', '.') AS FLOAT) AS salariomensal,
  tipomovimentacao AS tipomovdesagregado,
  CASE WHEN saldo = 1 THEN 1 ELSE 2 END AS admitidosdesligados,
  saldo AS saldomov,
  'fp' AS tipo
FROM
  caged.caged_202002_atual_FOR
WHERE
  UF = 53") |>
  left_join(data.frame(anodeclarado = c(2011:2023),
                        vl_sm = c(545, 622, 678, 724, 788, 880, 937, 954, 998,
1045, 1100, 1212, 1320)))

### CAGED ----

caged <- DBI::dbGetQuery(db,
"SELECT
  CAST(SUBSTRING(CAST(competenciadeclarada AS CHAR(6)), 1, 4) AS INT) AS
anodeclarado,
  CAST(SUBSTRING(CAST(competenciadeclarada AS CHAR(6)), 5, 2) AS INT) AS
mesdeclarado,
  competenciadeclarada AS competencia,
  cbo2002ocupacao,

```

```

    grauinstrucao,
    racacor,
    sexo,
    idade,
    indtrabintermitente,
    indaprendiz,
    salariomensal,
    tipomovdesagregado,
    admitidosdesligados,
    saldomov,
    'dp' AS tipo
FROM
    caged.caged_200701_atual_DP
WHERE
    UF = 53 AND competenciadeclarada > 201001
UNION ALL
SELECT
    CAST(SUBSTRING(CAST(competenciamovimentacao AS CHAR(6)), 1, 4) AS INT) AS
anodeclarado,
    CAST(SUBSTRING(CAST(competenciamovimentacao AS CHAR(6)), 5, 2) AS INT) AS
mesdeclarado,
    competenciamovimentacao AS competencia,
    cbo2002ocupacao,
    grauinstrucao,
    CAST(NULL AS INT) AS racacor,
    sexo,
    idade,
    indtrabintermitente,
    indaprendiz,
    salariomensal,
    tipomovdesagregado,
    admitidosdesligados,
    saldomov,
    'fp' AS tipo
FROM
    caged.caged_200701_atual_fp
WHERE
    UF = 53 AND competenciamovimentacao >= 201001") |>
    left_join(data.frame(anodeclarado = c(2011:2023),
                        vl_sm = c(545, 622, 678, 724, 788, 880, 937, 954, 998,
1045, 1100, 1212, 1320)))

dbDisconnect(db)

```

A próxima etapa do *script* carrega a variável de grau de instrução da RAIS, aqui são considerados as seguintes faixas de estudo:

- 1 - Analfabeto;
- 2 - Ensino médio incompleto;
- 3 - Ensino médio completo;
- 4 - Superior completo.

```
rais_4 <- rais |>
  mutate(escolaridade = case_when(escolaridade == 1 ~ "Analfabeto",
                                   escolaridade == 2 ~ "Médio incompleto",
                                   escolaridade == 3 ~ "Médio completo",
                                   escolaridade == 4 ~ "Superior completo"))
```

As CBOs (técnicas de nível médio e superior) são importadas para o ambiente `.R` com o comando abaixo:

```
cbotec_em <- as.character(read_csv("../1. Extração dos dados/Dados/CBO Técnica -
Nível médio.csv"))[[1]])
cbotec_sup <- as.character(read_csv("../1. Extração dos dados/Dados/CBO Técnica -
Nível superior.csv"))[[1]])
```

Tratamento

Após a finalização de importação, passamos para a etapa de tratamento dos dados. O primeiro passo é o cálculo do número de trabalhadores da RAIS agrupado por ano e por CBO (a escolaridade é considerada como **geral**). Depois, realizamos o mesmo cálculo, porém, agrupado também por escolaridade.

```
rotatividade_rais_4 <- rbind(rais_4 |>
  group_by(referencia, cboocupacao2002) |>
  summarise(n_trabalhadores = n(),
            escolaridade = "Geral"),
rais_4 |>
  group_by(referencia, cboocupacao2002, escolaridade)
|>
  summarise(n_trabalhadores = n())) |>
mutate(cboocupacao2002 = as.character(cboocupacao2002)) |>
rename(anodeclarado = referencia,
       cbo2002ocupacao = cboocupacao2002)
```

Partindo para as bases do CAGED antigo, o código abaixo adiciona a etiqueta para a variável `admitidosdesligados`:

- **1** - admitidos
- **2** - desligados

Para a variável de escolaridade são considerados as seguintes faixas de estudo:

- **1** - analfabeto;
- Entre **2** e **6** - ensino médio incompleto;
- Entre **7** e **8** - ensino médio completo;
- Entre **9** e **11** - superior completo;
- **99** - `NA`.

```

rotatividade_caged <- caged |>
  mutate(cbo2002ocupacao = as.character(cbo2002ocupacao),
         admitidosdesligados = case_when(admitidosdesligados == 1 ~ "admitidos",
         admitidosdesligados == 2 ~
"desligados"),
         escolaridade = case_when(grauinstrucao == 1 ~ "Analfabeto",
         grauinstrucao %in% 2:6 ~ "Médio incompleto",
         grauinstrucao %in% 7:8 ~ "Médio completo",
         grauinstrucao %in% 9:11 ~ "superior completo",
         grauinstrucao == 99 ~ NA))

```

Para o cálculo da rotatividade da base do CAGED antigo, filtramos a coluna do tipo de movimentação para apenas os seguintes casos:

- 1 - Admissão por primeiro emprego;
- 2 - Admissão por reemprego;
- 4 - Desligamento por demissão sem justa causa;
- 5 - Desligamento por demissão com justa causa;
- 10 - Admissão por reintegração;
- 11 - Desligamento por término de contrato;
- 25 - Contrato trabalho prazo determinado;
- 43 - Término contrato trabalho prazo determinado;
- 90 - Desligamento por acordo empregado e empregador.

Da mesma forma que foi realizado na RAIS, ocorre o cálculo do número de trabalhadores agrupado por ano, CBO e o número de admitidos e desligados. Depois, repetimos o mesmo cálculo, porém, agrupado também por escolaridade. O último passo consiste na transformação do formato longo (*long*) para formato amplo (*wide*), com uma coluna para cada valor de `admitidosdesligados` e o valor correspondente de `rotacao`, que corresponde ao número de trabalhadores, resultando em um `data.frame` informando por ano, CBO e escolaridade, o número de trabalhadores admitidos e desligados.

```

rotatividade_caged_antigo <- rbind(rotatividade_caged |>
  filter(tipomovdesagregado %in% c(1, 2, 4, 5,
10, 11, 25, 43, 90)) |>
  group_by(anodeclarado, cbo2002ocupacao,
admitidosdesligados) |>
  summarise(rotacao = n(),
            escolaridade = "Gera1"),
rotatividade_caged |>
  filter(tipomovdesagregado %in% c(1, 2, 4, 5,
10, 11, 25, 43, 90)) |>
  group_by(anodeclarado, cbo2002ocupacao,
admitidosdesligados, escolaridade) |>
  summarise(rotacao = n())) |>
  pivot_wider(names_from = admitidosdesligados,
              values_from = rotacao) |>
  arrange(anodeclarado, cbo2002ocupacao)

```

O cálculo realizado acima é válido tanto para o CAGED antigo quanto para o CAGED novo, com uma diferença nos códigos do filtro do tipo de movimentação:

- **10** - Admissão por primeiro emprego;
- **20** - Admissão por reemprego;
- **25** - Admissão por contrato de trabalho com prazo determinado;
- **31** - Desligamento por demissão sem justa causa;
- **32** - Desligamento por demissão com justa causa;
- **35** - Admissão por reintegração;
- **43** - Término do contrato de trabalho com prazo determinado;
- **45** - Desligamento por término de contrato;
- **90** - Desligamento por acordo entre empregado e empregador;
- **97** - Admissão de tipo ignorado;
- **98** - Desligamento de tipo ignorado.

Após o tratamento em todas as três bases para deixá-las na mesma estrutura de colunas, é possível realizar o agrupamento dessas bases. Realizamos um tratamento para substituir os valores de **NA** por 0 nas colunas **admitidos** e **desligados** e criamos uma nova coluna para identificar se a CBO corresponde à uma CBO técnica de nível médio ou de nível superior.

A coluna **tipo** agrega as informações de escolaridade e trabalho técnico com as seguintes categorias:

- **Média dos trabalhadores técnicos** - engloba escolaridade **geral** e trabalhadores em CBOS técnicas;
- **Média dos trabalhadores não técnicos** - engloba escolaridade **geral** e trabalhadores em CBOS não técnicas;
- **Técnicos de nível médio** - engloba todas as escolaridades e trabalhadores em CBOS técnicas de nível médio;
- **Técnicos de nível superior** - engloba todas as escolaridades e trabalhadores em CBOS técnicas de nível superior;
- **Até fundamental completo** - engloba trabalhadores analfabetos (código **1**) e trabalhadores com até o ensino médio incompleto (código **2**);
- **Ensino médio completo** - engloba trabalhadores com ensino médio completo;
- **Ensino superior completo** - engloba trabalhadores com ensino superior completo, mestrado e doutorado.

A última variável de filtro criada serve para identificar se os trabalhadores são de CBOs técnicas ou não.

```
rotatividade <- merge(rotatividade_rais_4,
                      rbind(rotatividade_caged_novo,
                            rotatividade_caged_antigo), all.x = TRUE) |>
mutate(admitidos = case_when(is.na(admitidos) ~ 0,
                             TRUE ~ admitidos),
       desligados = case_when(is.na(desligados) ~ 0,
                              TRUE ~ desligados),
       cbo_tecnica = case_when(cbo2002ocupacao %in% cbotec_em ~ "tec_em",
```

```

cbo2002ocupacao %in% cbotec_sup ~ "tec_sup",
TRUE ~ "nao_tec"),
tipo = case_when(educacao == "Geral" & cbo_tecnica %in% c("tec_em",
"tec_sup") ~ "Média dos trabalhadores técnicos",
educacao == "Geral" & cbo_tecnica == "nao_tec" ~
"Média dos trabalhadores não técnicos",
educacao %in% c("Analfabeto", "Médio incompleto",
"Médio completo", "Superior completo") & cbo_tecnica == "tec_em" ~ "Técnicos de
nível médio",
educacao %in% c("Analfabeto", "Médio incompleto",
"Médio completo", "Superior completo") & cbo_tecnica == "tec_sup" ~ "Técnicos de
nível superior",
educacao %in% c("Analfabeto", "Médio incompleto") &
cbo_tecnica == "nao_tec" ~ "Até fundamental completo",
educacao == "Médio completo" & cbo_tecnica ==
"nao_tec" ~ "Médio completo",
educacao == "Superior completo" & cbo_tecnica ==
"nao_tec" ~ "Superior completo"),
filtro = case_when(cbo_tecnica %in% c("tec_em", "tec_sup") ~ "Técnico",
TRUE ~ "Não técnico"))

```

Finalizado a parte do tratamento, pode-se realizar o cálculo da taxa de rotatividade, conforme método proposto por Silva Filho e Silva (2014) ¹:

$$R = \frac{\sum(admitidos_t, desligados_t)}{total\ da\ força\ de\ trabalho_t}$$

O código que aplicará essa fórmula está descrito abaixo:

```

rotatividade <- rotatividade |>
group_by(tipo, filtro, anodeclarado) |>
summarise(n_trabalhadores = sum(n_trabalhadores),
admitidos = sum(admitidos),
desligados = sum(desligados)) |>
mutate(rotatividade = (admitidos + desligados) / n_trabalhadores)

```

Exportação

Após a importação, organização e tratamento dos dados, é exportado um arquivo `.csv` para a pasta **2. Tratamento dos dados/Painel 2 - Remuneração e Ocupações/Resultados** com o nome de `2.4 - Taxa de rotatividade.csv`.

```

write_excel_csv2(base_tipo_emprego, "Painel 2 - Remuneração e
Ocupações/Resultados/2.4 - Taxa de rotatividade.csv")

```

Painel 3 - Ocupações técnicas

Para atualizar todos os painéis do painel 3, basta acessar **2. Tratamento dos dados** e abrir o projeto **Painéis.RProj** para executar o *script* corretamente. Após, execute o *script* `Painel 3.R`.

Os pacotes, funções e configurações necessários para execução do *script* estão descritos abaixo:

```

library(tidyverse)
library(imputeTS)

```

```

library(readxl)
library(stringr)
library(treemapify)

`%notin%` <- Negate(`%in%`)

foiAtualizado <- function(arquivo) {

  data <- Sys.Date()
  data_arquivo <- as.Date(file.info(paste0("Painel 2 - Remuneração e
Ocupações/Resultados/"), arquivo, ".csv"))$mtime)

  if (data >= data_arquivo) {
    return(TRUE)
  } else {
    return(FALSE)
  }

}

```

O pacote `tidyverse` é responsável pela manipulação dos dados. O pacote `readxl` é responsável por ler arquivos `.xls` e `.xlsx`. O pacote `imputeTS` é utilizado para retirar `NAs` (números não disponíveis) da base. O pacote `stringr` é utilizado para a manipulação de vetores de texto. O pacote `treemapify` é utilizado para gerar os gráficos de árvore. A função `%notin%` funcionará como filtro para os dados. A função `foiAtualizado` retorna valores booleanos do tipo `TRUE` e `FALSE` e é utilizada para verificar se todos os painéis foram atualizados sem erros.

O objeto `RAIS.RDS` é importado nesse *script* e será fonte de dados de todos os *scripts* contidos na pasta **2. Tratamento dos dados/Painel 3 - Ocupações Técnicas**.

3.1 - Ocupações para empresas

Importação

Tratamento

Exportação

3.2 - Empresas para ocupações

Importação

Tratamento

Exportação

3.3 - Ocupações e eixos tecnológicos

Importação

Tratamento

Exportação

3.4 - Remuneração e vínculos

Importação

Tratamento

Exportação

1. da Silva Filho, L. A., & dos Santos, J. M. (2014). O que Determina a Rotatividade no Mercado de Trabalho Brasileiro?. RDE - Revista de Desenvolvimento Econômico, 15(28). [↵](#)