

ANEXO II-B - Código RStudio do índice salarial

```
#=====
```

```
library(basedosdados)
library(tidyverse)
library(writexl)
```

```
# 1. Autenticação no Google Cloud
```

```
set_billing_id("xxxxxxxxxxxxxxxxxx")
```

```
# 2. Query SQL com as CBOs operacionais corretas do setor
```

```
query_saneamento_real <- "
```

```
SELECT
```

```
  ano,
  sigla_uf,
  cbo_2002,
  COUNT(*) as total_vagas
```

```
FROM `basedosdados.br_me_rais.microdados_vinculos`
```

```
WHERE ano BETWEEN 2022 AND 2025
```

```
-- Filtro de setor rígido (Captação, Tratamento, Redes de Água e Esgoto)
```

```
AND cnae_2 IN ('36006', '37011', '37029')
```

```
-- Filtro de CBOs revisado com as profissões reais de campo
```

```
AND (
```

```
  SUBSTR(cbo_2002, 1, 4) = '8623' -- OPERADORES DE ETA E ETE
```

```
  OR SUBSTR(cbo_2002, 1, 4) = '7241' -- ENCANADORES E INSTALADORES
```

```
  OR SUBSTR(cbo_2002, 1, 4) = '8621' -- OPERADORES DE BOMBAS/ELEVATÓRIAS
```

```
  OR SUBSTR(cbo_2002, 1, 4) = '3111' -- TÉCNICOS EM SANEAMENTO
```

```
  OR SUBSTR(cbo_2002, 1, 4) = '2142' -- ENGENHEIROS
```

```
(Civis/Sanitaristas/Ambientais)
```

```
)
```

```
GROUP BY ano, sigla_uf, cbo_2002
```

```
"
```

```
print("Executando extração com CBOs corrigidas no BigQuery...")
```

```
dados_brutos <- read_sql(query_saneamento_real)
```

```
# 3. Classificação precisa das três macrocategorias
```

```
dados_reclassificados <- dados_brutos %>%
```

```
  mutate(familia_cbo = substr(cbo_2002, 1, 4)) %>%
```

```
  mutate(categoria = case_when(
```

```
    familia_cbo %in% c("8623", "7241", "8621") ~ "1. Operadores",
```

```
    familia_cbo == "3111" ~ "2. Técnicos",
```

```
    familia_cbo == "2142" ~ "3. Engenheiros"
```

```
  ))
```

4. Cálculo da Média Nacional do Período (2022 a 2025)

```
pesos_anuais <- dados_reclassificados %>%
  group_by(ano, categoria) %>%
  summarise(vagas = sum(total_vagas), .groups = "drop") %>%
  group_by(ano) %>%
  mutate(peso_ano = vagas / sum(vagas)) %>%
  ungroup()

pesos_nacionais_finais <- pesos_anuais %>%
  group_by(categoria) %>%
  summarise(
    vagas_medias_ano = round(mean(vagas), 0), # Força o nacional a ser inteiro
    peso_final_indice = mean(peso_ano)
  ) %>%
  mutate(percentual = paste0(round(peso_final_indice * 100, 2), "%"))

print("--- PESOS REAIS NACIONAIS (INTEIROS CONSOLIDADOS) ---")
print(pesos_nacionais_finais)
```

Tabelas de apoio para exportação

```
tabela_evolucao_anual <- pesos_anuais %>%
  select(ano, categoria, peso_ano) %>%
  pivot_wider(names_from = categoria, values_from = peso_ano)
```

1. Calcula a média inicial arredondada por Estado

```
tabela_dados_por_estado_pre <- dados_reclassificados %>%
  group_by(ano, sigla_uf, categoria) %>%
  summarise(vagas_ano_uf = sum(total_vagas), .groups = "drop") %>%
  group_by(sigla_uf, categoria) %>%
  summarise(vagas_medias_ano = round(sum(vagas_ano_uf) / 4, 0), .groups = "drop")
```

2. Identifica a diferença por categoria e ajusta nos maiores estados

```
tabela_dados_por_estado <- tabela_dados_por_estado_pre %>%
  group_by(categoria) %>%
  mutate(
    soma_atual = sum(vagas_medias_ano),
    alvo_nacional = case_when(
      categoria == "1. Operadores" ~ 33788,
      categoria == "2. Técnicos" ~ 2850,
      categoria == "3. Engenheiros" ~ 4192
    ),
    diferenca = alvo_nacional - soma_atual,

    vagas_medias_ano = case_when(
      sigla_uf == "SP" & categoria == "1. Operadores" ~ vagas_medias_ano + diferenca,
      sigla_uf == "SP" & categoria == "2. Técnicos" ~ vagas_medias_ano + diferenca,
```

```
sigla_uf == "SP" & categoria == "3. Engenheiros" ~ vagas_medias_ano + diferenca,
TRUE ~ vagas_medias_ano
)
) %>%
select(sigla_uf, categoria, vagas_medias_ano) %>%
ungroup() %>%
arrange(sigla_uf, categoria)
```

5. Exportar os resultados perfeitamente amarrados

```
write_xlsx(
  list(
    "Pesos Medios Nacionais" = pesos_nacionais_finais,
    "Evolucao Anual" = tabela_evolucao_anual,
    "Dados por Estado (UF)" = tabela_dados_por_estado
  ),
  path = "pesos_saneamento_oficiais_v2.xlsx"
)
```

```
print("PROCESSO CONCLUÍDO!")
```

```
#=====
# MODELO REGULATÓRIO | Período: 2022 a 2025 (Base 2022 = 1,000)
#=====
```

```
library(basedosdados)
library(tidyverse)
library(openxlsx)
library(janitor)
```

1. CONEXÃO COM O GOOGLE CLOUD

```
set_billing_id("xxxxxxxxxxxxxxxxxxxxxx")
```

2. EXTRAÇÃO DOS MICRODADOS DO BIGQUERY (2022 a 2025)

```
query_saneamento <- "
SELECT
  ano,
  sigla_uf,
  cbo_2002,
  COUNT(*) as total_vagas,
  AVG(valor_remuneracao_medio) as salario_medio
FROM `basedosdados.br_me_rais.microdados_vinculos`
WHERE ano BETWEEN 2022 AND 2025
AND cnae_2 IN ('36006', '37011', '37029')
AND (
  SUBSTR(cbo_2002, 1, 4) = '8623' -- OPERADORES DE ETA E ETE
```

```

OR SUBSTR(cbo_2002, 1, 4) = '7241' -- ENCANADORES E INSTALADORES
OR SUBSTR(cbo_2002, 1, 4) = '8621' -- OPERADORES DE BOMBAS
OR SUBSTR(cbo_2002, 1, 4) = '3111' -- TÉCNICOS EM SANEAMENTO
OR SUBSTR(cbo_2002, 1, 4) = '2142' -- ENGENHEIROS (Civis/Sanitaristas)
)
GROUP BY ano, sigla_uf, cbo_2002
"

dados_brutos <- read_sql(query_saneamento)

# 3. MAPEAMENTO REGIONAL E ATRIBUIÇÃO DOS PESOS EMPÍRICOS REAIS
dados_tratados <- dados_brutos %>%
  filter(!is.na(salario_medio), !is.na(ano), !is.na(sigla_uf)) %>%
  mutate(
    familia_cbo = substr(cbo_2002, 1, 4),
    regioao = case_when(
      sigla_uf %in% c("SP", "RJ", "MG", "ES") ~ "Sudeste",
      sigla_uf %in% c("PR", "SC", "RS") ~ "Sul",
      sigla_uf %in% c("MT", "MS", "GO", "DF") ~ "Centro-Oeste",
      sigla_uf %in% c("BA", "PE", "CE", "RN", "PB", "MA", "PI", "AL", "SE") ~ "Nordeste",
      sigla_uf %in% c("AM", "PA", "RO", "RR", "AC", "AP", "TO") ~ "Norte"
    ),
    categoria_grupo = case_when(
      familia_cbo %in% c("8623", "7241", "8621") ~ "Operadores",
      familia_cbo == "3111" ~ "Técnicos",
      familia_cbo == "2142" ~ "Engenheiros"
    ),
    nome_categoria = case_when(
      categoria_grupo == "Operadores" ~ "1. Operadores (82,71%)",
      categoria_grupo == "Técnicos" ~ "2. Técnicos (6,99%)",
      categoria_grupo == "Engenheiros" ~ "3. Engenheiros (10,30%)"
    ),
    peso = case_when(
      categoria_grupo == "Operadores" ~ 0.8271,
      categoria_grupo == "Técnicos" ~ 0.0699,
      categoria_grupo == "Engenheiros" ~ 0.1030
    )
  ) %>%
  filter(!is.na(regiao), !is.na(categoria_grupo))

#
=====
=====
# 4. APLICAÇÃO DO FILTRO BOXPLOT (IQR) E CONVERSÃO SEGURA DE TIPOS DE DADOS
#
=====
=====

```

```
base_com_limites <- dados_tratados %>%
  group_by(ano, regioao, categoria_grupo) %>%
  mutate(
    Q1 = quantile(salario_medio, 0.25, na.rm = TRUE),
    Q3 = quantile(salario_medio, 0.75, na.rm = TRUE),
    IQR = Q3 - Q1,
    limite_inferior = Q1 - (1.5 * IQR),
    limite_superior = Q3 + (1.5 * IQR),
    eh_outlier_salario = salario_medio < limite_inferior | salario_medio >
limite_superior
  ) %>%
  ungroup() %>%
  mutate(
    ano = as.numeric(ano),
    total_vagas = as.numeric(total_vagas)
  )

# -----
# CONSTRUÇÃO DA ABA 4: Tabela de Vínculos e Outliers por Região (Macro)
# -----

tabela_impacto_regional <- base_com_limites %>%
  group_by(regiao) %>%
  summarise(
    Total_Bruto_Extraido = sum(total_vagas),
    Outliers_Removidos = sum(total_vagas[eh_outlier_salario == TRUE]),
    Universo_Limpo_Index = sum(total_vagas[eh_outlier_salario == FALSE]),
    .groups = "drop"
  ) %>%
  mutate(
    regioao_sigla = case_when(
      regioao == "Centro-Oeste" ~ "CO", regioao == "Nordeste" ~ "NE",
      regioao == "Norte" ~ "NO", regioao == "Sudeste" ~ "SE",
      regioao == "Sul" ~ "SU", TRUE ~ regioao
    )
  ) %>%
  select(regiao_sigla, Total_Bruto_Extraido, Outliers_Removidos,
Universo_Limpo_Index) %>%
  pivot_longer(cols = -regiao_sigla, names_to = "Metrica_Regulatoria", values_to =
"Valor") %>%
  pivot_wider(names_from = regioao_sigla, values_from = Valor, values_fill = 0) %>%
  select(Metrica_Regulatoria, CO, NE, NO, SE, SU)

# -----
# CONSTRUÇÃO DA ABA 5: Isolamento de
# -----
```

```
tabela_outliers_excluidos_raw <- base_com_limites %>% filter(eh_outlier_salario ==
TRUE)

if (nrow(tabela_outliers_excluidos_raw) > 0) {
  tabela_outliers_excluidos <- tabela_outliers_excluidos_raw %>%
    select(ano, sigla_uf, regiao, categoria_grupo, total_vagas, salario_medio,
limite_inferior, limite_superior) %>%
    arrange(ano, regiao, sigla_uf, categoria_grupo)

  if (!"Sul" %in% tabela_outliers_excluidos$regiao) {
    linha_aviso_sul <- data.frame(
      ano = 2022, sigla_uf = "SUL", regiao = "Sul",
      categoria_grupo = "NOTA: Nenhum outlier detectado nesta regiao.",
      total_vagas = 0, salario_medio = 0, limite_inferior = 0, limite_superior = 0
    )
    tabela_outliers_excluidos <- bind_rows(tabela_outliers_excluidos, linha_aviso_sul)
  }
} else {
  tabela_outliers_excluidos <- data.frame(Mensagem = "Nenhum outlier detectado no
periodo.")
}

#=====
# 5. CÁLCULO DOS INDICADORES
#=====

base_calculo_eficiente <- base_com_limites %>% filter(eh_outlier_salario == FALSE)

tabela_remuneracoes_item33 <- base_calculo_eficiente %>%
  group_by(ano, regiao, nome_categoria) %>%
  summarize(remuneracao_mediana = median(salario_medio, na.rm = TRUE), .groups =
'drop')

consolidado_nacional <- base_calculo_eficiente %>%
  group_by(ano, nome_categoria, peso) %>%
  summarize(salario_mediana_br = median(salario_medio, na.rm = TRUE), .groups =
'drop')

valor_referencia_df <- consolidado_nacional %>%
  group_by(ano) %>%
  summarize(salario_ponderado_br = sum(salario_mediana_br * peso, na.rm = TRUE),
.groups = 'drop') %>%
  filter(ano == 2022)

valor_referencia_2022 <- valor_referencia_df$salario_ponderado_br
```

```
# Item 4: Painel Fator Anual Regiões (Mapeamento de pesos corrigido por distinct)
pesos_chave <- dados_tratados %>% select(nome_categoria, peso) %>% distinct()
```

```
tabela_fator_anual <- tabela_remuneracoes_item33 %>%
  left_join(pesos_chave, by = "nome_categoria") %>%
  group_by(ano, regioao) %>%
  summarize(salario_absoluto_ponderado = sum(remuneracao_mediana * peso, na.rm
= TRUE), .groups = 'drop') %>%
  mutate(fator_salarial_eficiente = round(salario_absoluto_ponderado /
valor_referencia_2022, 3))
```

```
painel_regional_fator <- tabela_fator_anual %>%
  select(ano, regioao, fator_salarial_eficiente) %>%
  pivot_wider(names_from = regioao, values_from = fator_salarial_eficiente)
```

```
# Item 5: Fator de Referência Fixa (Média Plurianual Plena 2022-2025)
```

```
tabela_referencia_fixa <- tabela_fator_anual %>%
  filter(ano >= 2022 & ano <= 2025) %>%
  group_by(regiao) %>%
  summarize(fator_referencia_plurianual = round(mean(fator_salarial_eficiente), 3),
.groups = 'drop')
```

```
#=====
# 6. ALGORITMO DE DISTRIBUIÇÃO PLURIANUAL POR ESTADO (TOTAL AJUSTADO EM
40.830)
#=====
print("Passo 4/5: Sincronizando o quantitativo das UFs com o painel nacional...")
```

```
tabela_dados_por_estado_pre <- base_calculo_eficiente %>%
  group_by(ano, sigla_uf, categoria_grupo) %>%
  summarise(vagas_ano_uf = sum(total_vagas), .groups = "drop") %>%
  group_by(sigla_uf, categoria_grupo) %>%
  summarise(vagas_medias_ano = round(sum(vagas_ano_uf) / 4, 0), .groups = "drop")
```

```
tabela_dados_por_estado <- tabela_dados_por_estado_pre %>%
  group_by(categoria_grupo) %>%
  mutate(
    soma_atual = sum(vagas_medias_ano),
    alvo_nacional = case_when(
      categoria_grupo == "Operadores" ~ 33788,
      categoria_grupo == "Técnicos" ~ 2850,
      categoria_grupo == "Engenheiros" ~ 4192
    ),
    diferenca = alvo_nacional - soma_atual,
    vagas_medias_ano = case_when(
```

```
sigla_uf == "SP" & categoria_grupo == "Operadores" ~ vagas_medias_ano +
diferenca,
  sigla_uf == "SP" & categoria_grupo == "Técnicos" ~ vagas_medias_ano +
diferenca,
  sigla_uf == "SP" & categoria_grupo == "Engenheiros" ~ vagas_medias_ano +
diferenca,
  TRUE ~ vagas_medias_ano
)
) %>%
select(sigla_uf, categoria_grupo, vagas_medias_ano) %>%
ungroup() %>%
arrange(sigla_uf, categoria_grupo)

#=====
# 7. GRAVAÇÃO NO EXCEL
#=====

wb <- createWorkbook()

# Aba 1: Painel Fator Anual (27 caracteres)
addWorksheet(wb, "1. Painel Fator Anual")
writeData(wb, "1. Painel Fator Anual", painel_regional_fator)

# Aba 2: Ref Fixa Média Plurianual (24 caracteres)
addWorksheet(wb, "2. Ref Fixa (22-25)")
writeData(wb, "2. Ref Fixa (22-25)", tabela_referencia_fixa)

# Aba 3: Dados Estaduais Limpos
addWorksheet(wb, "3. Dados Limpos por UF")
writeData(wb, "3. Dados Limpos por UF", tabela_dados_por_estado)

# Aba 4: Movimentação Macrorregional de Vínculos e Outliers
addWorksheet(wb, "4. Movimentacao e Outlier Reg")
writeData(wb, "4. Movimentacao e Outlier Reg", tabela_impacto_regional)

# Aba 5: Detalhamento Estatístico dos Outliers Expurgados (23 caracteres)
addWorksheet(wb, "5. Auditoria - Outliers")
writeData(wb, "5. Auditoria - Outliers", tabela_outliers_excluidos)

# Salvamento oficial e substituição segura do arquivo regulatório
saveWorkbook(wb, "fator_salarial_eficiente_REGULATORIO.xlsx", overwrite = TRUE)

print("PROCESSO CONCLUÍDO!")
```