

Relatório ICE

Douglas Mangini

18/08/2021

Contents

Introdução	2
Ambiente regulatório	3
Tempo de processos	3
Tributação	3
Complexidade burocrática	4
Infraestrutura	6
Transporte interurbano	6
Condições urbanas	8
Mercados	10
Desenvolvimento econômico	10
Clientes potenciais	10
Apêndice	11

Introdução

O script *municode* (clique aqui) fabrica uma planilha em que estão os maiores municípios do Brasil, seus códigos ibge e população estimada¹. Esta planilha será carregada em todos os códigos que seguem.

Os dados de 2021 mostram que Santa Maria - RS foi substituída na lista dos 100 maiores municípios por Marabá - PA, então nesta edição trabalhamos com 101 municípios.

Abaixo, os nomes que demos aos subdeterminantes para facilitar a comunicação.

Table 1: nomenclatura

determinante	subdeterminante	nomenclatura
Ambiente Regulatório	Tempo de Processos	sd11
Ambiente Regulatório	Tributação	sd12
Ambiente Regulatório	Complexidade Tributária	sd13
Infraestrutura	Transporte Interurbano	sd21
Infraestrutura	Condições Urbanas	sd22
Mercado	Desenvolvimento Econômico	sd31
Mercado	Clientes Potenciais	sd32

¹<https://www.ibge.gov.br/estatisticas/sociais/populacao/9103-estimativas-de-populacao.html?edicao=28674&t=resultados>

Ambiente regulatório

Tempo de processos

clique aqui para ver o script

Tempo de viabilidade de locação

Fonte: <https://estatistica.redesim.gov.br/tempos-abertura>.

Período: 2020

Cidades faltantes: São José do Rio Preto (SP), Jundiaí (SP), Maringá (PR), Anápolis (GO)

Comentário: A coluna de interesse é *qtde. hh viabilidade end*. Calculamos o tempo médio por município e registramos como indicador o inverso desse número. Atribuímos nota zero às cidades faltantes. Detalhes estão comentados no script em apêndice.

Tempo de registro, cadastro, e viabilidade de nome

fonte: <https://estatistica.redesim.gov.br/tempos-abertura>.

Período: 2020

Cidades faltantes: São José do Rio Preto (SP), Jundiaí (SP), Maringá (PR), Anápolis (GO)

Comentário: As colunas de interesse são *qtde. hh viabilidade de nome*, *qtde. hh liberação db*, *qtde. hh horas deferimento*. Agrupamos por estado e somamos as 3 colunas. O indicador é o inverso da média simples por estado. Municípios do mesmo estado terão notas iguais.

Taxa de congestionamento em tribunais

Fonte: https://paineis.cnj.jus.br/QvAJAXZfc/opensoc.htm?document=qvw_l%2FPainelCNJ.qvw&host=QVS%40neodimio03&anonymous=true&sheet=shPDPrincipal

Período: 2020

Comentário: Usamos os parâmetros de pesquisa: justiça = Justiça Estadual, campos agrupados = Sede Município e tipo de variável = novos, pendentes e baixados. A taxa líquida de congestionamento é definida² como

$$1 - \frac{\text{baixados}}{\text{novos} + \text{pendentes}}$$

. O indicador é o inverso da taxa de congestionamento.

Tributação

clique aqui para ver o script

Alíquota interna do ICMS

Fonte: <https://siconfi.tesouro.gov.br/siconfi/index.jsf>

Período icms: 2020 Período pib: 2018

Comentário: Usamos a tabela Receitas Orçamentárias (Anexo I-C). Trabalhamos apenas com receitas brutas realizadas e a conta 1.1.1.8.02.0.0. O indicador é o inverso do icms por unidade de pib municipal. Detalhes estão comentados no script em apêndice.

²<https://www.cnj.jus.br/gestao-estrategica-e-planejamento/estrategia-nacional-do-poder-judiciario-2009-2014/indicadores/03-taxa-de-congestionamento/>

Alíquota interna do IPTU

Fonte: <https://siconfi.tesouro.gov.br/siconfi/index.jsf>

Período icms: 2020

Período pib = 2018

Cidades faltantes: Brasília (DF), Carapicuíba (SP)

Comentário: Usamos a tabela Receitas Orçamentárias (Anexo I-C). Trabalhamos apenas com receitas brutas realizadas e as contas 1.1.1.8.01.1.0 e 1.1.1.8.02.3.0. Detalhes estão comentados no script em apêndice.

Alíquota interna do ISS

Fonte: <https://siconfi.tesouro.gov.br/siconfi/index.jsf>

Período icms: 2020

Período pib = 2018

Cidades faltantes: Brasília (DF), Carapicuíba (SP), Uberaba (MG), São João de Meriti (RJ) Belford Roxo (RJ)

Comentário: Usamos a tabela Receitas Orçamentárias (Anexo I-C). Detalhes estão comentados no script em apêndice.

Qualidade da Gestão Fiscal

Fonte: <https://www.firjan.com.br/ifgf/>

Período: 2018

Comentário: Não há novos dados desde a última edição do ICE, então usamos os valores passados.

Complexidade burocrática

clique aqui para ver o script

Simplicidade tributária

Fonte: <https://siconfi.tesouro.gov.br/siconfi/index.jsf>

Período: 2020

Cidades faltantes: Brasília (DF)

Comentário: Para filtrar as contas orçamentárias corretas, lemos o relatório do ice passado³ no R. O indicador é o produto dos índices de Herfindahl-Hirshmann (ihh) e de visibilidade (iv). O ihh é a soma dos quadrados da participação relativa do tributo na arrecadação total. A visibilidade é a participação relativa de uma soma de de tributos⁴ na arrecadação total. Detalhes no script em apêndice.

CND's municipais

Fonte: <https://www.ibge.gov.br/estatisticas/sociais/saude/10586-pesquisa-de-informacoes-basicas-municipais.html?=&t=downloads>

Período: 2019

Comentário: variável binária igual a 1 se município emite certidão negativa de débitos. Utilizamos a quarta aba da planilha, coluna MTIC1211. Detalhes no script em apêndice.

³pdf_contas.pdf, pp.15-17

⁴IPTU, ITBI, ITR, IRRF

Atualização de zoneamento

fonte: <https://www.ibge.gov.br/estatisticas/sociais/saude/10586-pesquisa-de-informacoes-basicas-municipais.html?=&t=downloads>

Período: 2018

Comentário: quantidade de anos desde que o município mudou a lei de zoneamento. O IBGE excluiu essa pergunta na pesquisa MUNIC 2019, então verificamos os sítios eletrônicos das prefeituras.

Infraestrutura

Transporte interurbano

Para os indicadores de rodovias, portos e aeroportos, importamos os dados georeferenciados de cada estrutura e fizemos operações geométricas para encontrar os indicadores. Por exemplo, encontrar as rodovias que passam por um município reduz-se a encontrar o número de interseções entre o polígono do município e a linha que descreve a estrada. Esses cálculos estão no script em apêndice. ver script

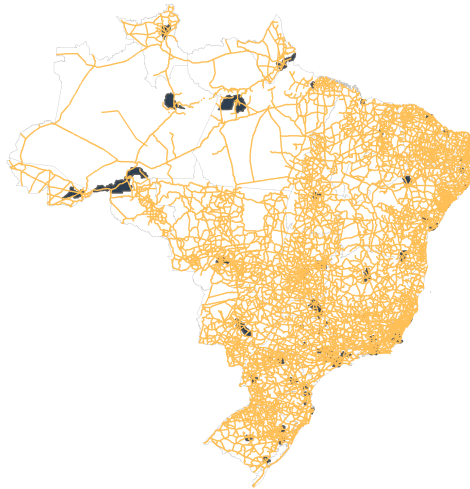


Figure 1: Exemplo de rodovia

Conectividade das rodovias

Fonte dos shape files das rodovias: <http://servicos.dnit.gov.br/vgeo/>

Período: 2021

Comentário: Consideramos apenas rodovias federais e estaduais. Construímos uma função que verifica se o município tem interseção com ao menos um trecho da rodovia. Nosso método implica incluir rodovias que estão próximas, mas não entram de fato no município. Consideramos este método melhor, pois reflete todas as opções de entrada e saída do território.

Número de decolagens por ano

fonte dos dados de voos: <https://www.anac.gov.br/acesso-a-informacao/dados-abertos/areas-de-atuacao/voos-e-operacoes-aereas/dados-estatisticos-do-transporte-aereo>

Fonte dos Shape files dos aeroportos: <https://www.gov.br/infraestrutura/pt-br/assuntos/dados-de-transportes/bit/bitmodosmapas#mapaero>

Período: 2020

Comentário: Decolagens de voos regulares com origem no município. Faltam dados para metade dos municípios. Para completar os faltantes, atribuímos um número total de decolagens com base aeroporto mais próximo (distância euclidiana).

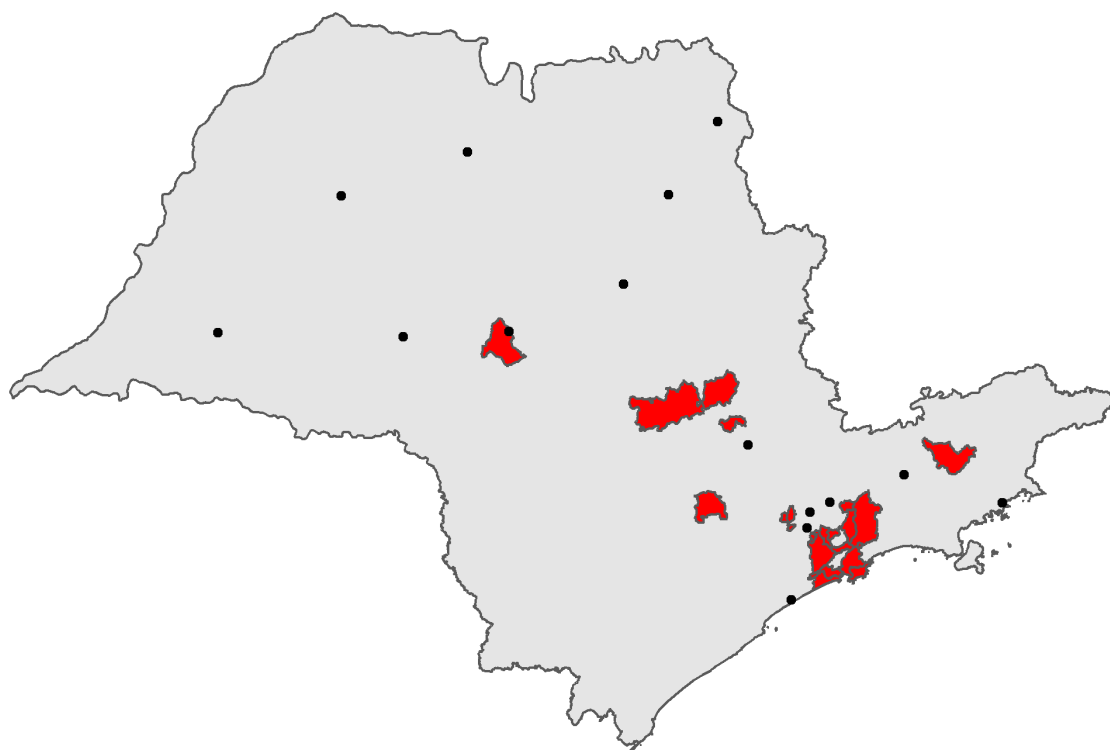


Figure 2: Exemplo: estado de SP e aeroportos mais próximos

Distância ao porto mais próximo

Fonte dos shape files dos portos: <https://www.gov.br/infraestrutura/pt-br/assuntos/dados-de-transportes/bit/bitmodosmapas>

Período: 2020

Comentário: Distância do porto mais próximo ao município. Consideramos apenas portos públicos ou os flúvias do Amazonas. Tomamos o centro do polígono que representa o município como referência (ver figura).

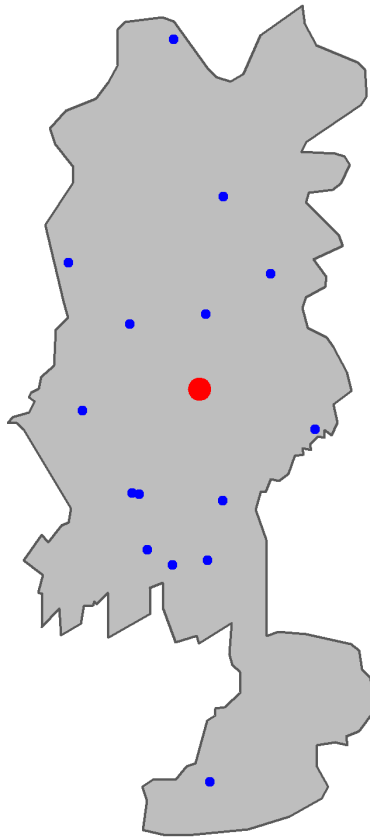


Figure 3: Exemplo: centro de Ananindeua-PA e portos mais próximos

Condições urbanas

[clique aqui para ver o script](#)

Acesso à internet rápida

Fonte: <https://dados.gov.br/dataset/dados-de-acessos-de-comunicacao-multimedia>

Período: 2020

Comentário: número de acessos à internet de alta velocidade (acima de 12Mbps) por habitante.

Preço médio do m²

fonte: <https://www.zapimoveis.com.br/>

Período: 2021

Comentário: raspamos o site a procura de imóveis a venda. Extraímos preço e área útil e fizemos a média para cada município. Excluimos do cálculo os anúncios que implicavam um preço de metro quadrado menor do que 100 reais e maior do que 10.000 reais⁵. Para cada município, somamos o preço de todos os anúncios e dividimos pela soma de todas as áreas. Procuramos por 350 anúncios de cada município.

Custo da energia elétrica

Fonte distribuidoras: [http://www2.aneel.gov.br/relatoriosrig/\(S\(fgy4psttnrfsam2x1s40fgib\)\)/relatorio.aspx?folder=sfe&report=DistribuidoradecadaMunicipio](http://www2.aneel.gov.br/relatoriosrig/(S(fgy4psttnrfsam2x1s40fgib))/relatorio.aspx?folder=sfe&report=DistribuidoradecadaMunicipio)

Fonte das tarifas: <https://www.aneel.gov.br/ranking-das-tarifas>

Período: 2021

Comentário: Montamos manualmente a base de dados que indica qual distribuidora atende cada município.

Taxa de Homicídios

fonte: <http://tabnet.datasus.gov.br/cgi/tabcgi.exe?sim/cnv/obt10br.def>

Período: 2019

Comentário: mortes causadas por agressão, ponderado pelo número estimado de habitantes em 2020. No site do datasus, selecionamos Conteúdo igual a *Óbitos p/ Ocorrência* e Grupo CID-10 igual a *agressões*. Os dados em formato excel estão na pasta de arquivos e o script em R para reproduzir o cálculo no apêndice deste documento.

⁵esses outliers aparecem em anúncios cujo preço refere-se a um prédio e a área a uma sala do prédio

Mercados

Desenvolvimento econômico

[clique aqui para ver o script](#)

índice de desenvolvimento humano

Fonte: <http://www.atlasbrasil.org.br/ranking>

Período: 2010

Comentário: Não há novos dados desde a última edição do índice. Apenas adicionamos as colunas código do município e sigla da UF.

Crescimento médio real do PIB

Fonte: sidra-IBGE

Período: 2014 a 2018

Comentário: crescimento médio do pib municipal. Usamos os pacotes `basedosdados` e `sidrar` para encontrar o pib municipal e calcular o deflator do pib.

Número de exportadoras sediadas no município

Fonte 1: Rais-IBGE

Fonte 2: <https://www.gov.br/produtividade-e-comercio-exterior/pt-br/assuntos/comercio-exterior/estatisticas/empresas-brasileiras-exportadoras-e-importadoras>

Período: 2019

Comentário: Usamos o pacote `basedosdados` para acessar a RAIS e obter a quantidade de funcionarios por empresa. Dividimos o número de empresas exportadoras pelo total de empresas sediadas no município com pelo menos um funcionário.

Clientes potenciais

[clique aqui para ver o script](#)

PIB per capita

Fonte: Sidra-IBGE

Período: 2018

Comentário: Acessamos os dados via base dos dados. Detalhes no script.

Proporção de grandes empresas

Fonte: Rais-IBGE

Período: 2019

Comentário: Razão de duas proporções: empresas grandes por empresas médias e médias por pequenas. Acessamos os dados via base dos dados. O tamanho da empresa é dado pela variável `qtde_vinculos_ativos`. Empresas pequenas têm entre 10 e 49 funcionários; médias entre 50 e 249; grandes, acima de 250.

Apêndice

maiores municípios

voltar para introdução

```
library(tidyverse)
library(readxl)
library(basedosdados)

# populacao estimada 2020

# mp20 <- read_xls("POP2020_20210331.xls", sheet = "Municípios", skip = 1) %>%
#   select(sigla_uf=1, nome = 4, pop =5 ) %>%
#   mutate(pop = pop %>% str_remove_all(padrao) %>% as.numeric) %>%
#   filter(rank(-pop) <= 100) %>%
#   arrange(-pop)

# estao no ranking atual mas nao no antigo
# setdiff(mp21$nome, mp20$nome)

# estao no ranking antigo mas nao no atual
# setdiff(mp20$nome, mp21$nome)

# expressao regular que procura ponto ou digitos entre parenteses
padrao <- "(\\.|\\\\(\\\\d+\\\\))"

mp21 <- read_xls("estimativa_dou_2021.xls", sheet = "Municípios", skip = 1)%>%
  select(sigla_uf=1, nome = 4, pop =5 ) %>%
  mutate(pop = pop %>% str_remove_all(padrao) %>% as.numeric)

mp21_top100 <- mp21 %>%
  filter(rank(-pop) <= 100 | (nome == "Santa Maria" & sigla_uf == "RS")) %>%
  arrange(-pop)

# codigo dos municipios -----
set_billing_id("ice2021")
diretorios <- read_sql(
  "SELECT * FROM `basedosdados.br_bd_diretorios_brasil.municipio`")

#juntar codigos e pop

municode <- diretorios %>%
  select(id_municipio,id_municipio_6,sigla_uf,nome) %>%
  right_join(mp21_top100)

# salvar no disco
write_csv(municode,"municode.csv")
```

sd11

voltar para Tempo de Processos

```
#
#
# SD 11 - AMBIENTE REGULATÓRIO - TEMPO DE PROCESSOS

library(tidyverse)
library(readxl)
library(abjutils)
library(janitor)

# maiores municipios
municode <- read_csv("municode.csv") %>%
  select(id_municipio, sigla_uf, nome) %>%
  mutate(nome = abjutils::rm_accent(nome) %>% str_to_title())

# pega colunas que interessam para os indicadores de endereco e nome
col_interesse <- read_excel(
  "ambiente_regulatorio/dados/sd11_localiza_nome/sd11abr20.xlsx",
  sheet = 1,
  skip = 1
) %>%
  names() %>%
  str_subset(pattern = "QTDE")

col_interesse <- col_interesse[c(1, 2, 5, 7)]

# Le os dados de abertura
aberturas <-
  list.files("ambiente_regulatorio/dados/sd11_localiza_nome/") %>%
  paste0("ambiente_regulatorio/dados/sd11_localiza_nome/", .) %>%
  map_dfr(~ read_excel(., sheet = 1, skip = 1))
# limpa os dados
aberturas_clean <- aberturas %>%
  select(sigla_uf = UF,
         nome = MUNICÍPIO,
         all_of(col_interesse)) %>%
  janitor::clean_names() %>%
  mutate(nome = str_to_title(nome))

# filtra os maiores municipios
aberturas_top <- aberturas_clean %>%
  right_join(municode)

# Tempo Viabilidade Localizacao -----
tempo_localiza <- aberturas_top %>%
  filter(!is.na(qtde_hh_viabilidade_end)) %>%
  group_by(id_municipio, sigla_uf, nome) %>%
  summarise(
    tempo_medio = mean(qtde_hh_viabilidade_end, na.rm = TRUE),
    sd11_localiza = 1 / tempo_medio
  )
```

```

# encontra cidades faltantes
id_faltantes <-
  setdiff(municode$id_municipio, tempo_localiza$id_municipio)

faltantes <- municode %>%
  filter(id_municipio %in% id_faltantes) %>%
  select(id_municipio, sigla_uf, nome)

# completa com 0 as faltantes
tempo_localiza <- tempo_localiza %>%
  ungroup() %>%
  add_row(
    id_municipio = as.double(faltantes$id_municipio),
    sigla_uf = faltantes$sigla_uf,
    nome = faltantes$nome,
    tempo_medio = 0,
    sd11_localiza = 0
  ) %>%
  arrange(-sd11_localiza)

# Tempo Viabilidade Nome -----
cols_sd11_nome <- c("nome", "dbe", "deferimento")

tempo_nome <- aberturas_clean %>%
  select(sigla_uf, nome, contains(cols_sd11_nome)) %>%
  group_by(sigla_uf) %>%
  summarise(across(contains("qtde"), ~ sum(.x, na.rm = TRUE)),
    ordens = n()) %>%
  rowwise(sigla_uf) %>%
  mutate(
    tempo_sum = sum(c_across(contains("qtde"))),
    tempo_nome_medio = tempo_sum / ordens,
    sd11_nome = 1 / tempo_nome_medio
  )

# monta numa planilha os indicadores endereco e nome
sd11_localiza_nome <- tempo_nome %>%
  select(sigla_uf, sd11_nome) %>%
  right_join(tempo_localiza) %>%
  select(id_municipio, sigla_uf, nome, i111 = sd11_localiza, i112 = sd11_nome)

# salva o arquivo
write_excel_csv(sd11_localiza_nome, "dados_finais/sd11_localiza_nome.xlsx")

# congestionamento em tribunais -----
# processos
novos <-
  read_excel("ambiente_regulatorio/dados/sd11_processos_novos.xlsx")

baixados <-
  read_excel("ambiente_regulatorio/dados/sd11_processos_baixados.xlsx")

pendentes <-

```

```

read_excel("ambiente_regulatorio/dados/sd11_processos_pendentes.xlsx")

sd11_congestionamento_tribunais <-
  list.files("ambiente_regulatorio/dados/", pattern = "sd11_processos_") %>%
  paste0("ambiente_regulatorio/dados/", .) %>%
  map(~ read_excel(., col_names = c("nome", .), skip = 1)) %>%
  reduce(full_join) %>%
  rename(
    nome = nome,
    baixados = 2,
    novos = 3,
    pendentes = 4
  ) %>%
  mutate(nome = nome %>% str_to_title() %>% abjutils::rm_accent()) %>%
  right_join(municode) %>%
  mutate(congest = 1 - (baixados / (novos + pendentes)),
         sd11_congestionamento = 1 / congest) %>%
  select(id_municipio, sigla_uf, nome, everything()) %>%
  arrange(-sd11_congestionamento)

write_excel_csv(
  sd11_congestionamento_tribunais,
  "ambiente_regulatorio/sd11_congestionamento_tribunais_completo.xlsx"
)

sd11_congestionamento_tribunais %>%
  select(1,2,3,8) %>%
  write_excel_csv("dados_finais/sd11_congestionamento_tribunais.xlsx")

```

sd12

voltar para Tributação

```
#
#
# sd 12 - AMBIENTE REGULATORIO - TRIBUTACAO

library(tidyverse)
library(basedosdados)

# maiores municipios
municode <-
  read_csv("municode.csv") %>% select(id_municipio, sigla_uf, nome)

# ICMS -----

icms <- read.delim(
  "ambiente_regulatorio/dados/finbra.csv",
  dec = ",",
  sep = ";",
  skip = 3
) %>%
  filter(Coluna == "Receitas Brutas Realizadas",
         str_detect(Conta, "1.1.1.8.02.0.0")) %>%
  select(sigla_uf = UF, icms = Valor)

# pib municipal
basedosdados::set_billing_id("ice2021")

pib_mun <- read_sql(
  "SELECT * FROM `basedosdados.br_ibge_pib.municipio` WHERE ano = 2018") %>%
  select(1:3)

# tabela auxiliar para relacionar municipio e estado
municode2 <- basedosdados::read_sql(
  "SELECT id_municipio,sigla_uf
  FROM `basedosdados.br_bd_diretorios_brasil.municipio`"
)

# pib estadual
pib_estadual <- pib_mun %>%
  left_join(municode2) %>%
  group_by(sigla_uf) %>%
  summarise(pib_estadual = sum(pib, na.rm = TRUE))

# unindo pib e icms
df <- full_join(icms, pib_estadual) %>%
  mutate(icms_pib = icms / pib_estadual,
         sd12_icms = 1 / icms_pib)

# tabela final
icms_final <- left_join(municode, select(df, sigla_uf, i121 = sd12_icms))

write_excel_csv(icms_final, "dados_finais/sd12_icms.xlsx")
```

```

# IPTU e ISS -----
# receitas dos municipios
# dados brutos
finbramun <- read.delim(
  "ambiente_regulatorio/dados/finbramun.csv",
  dec = ",",
  sep = ";",
  skip = 3
)
# dados limpos
finbramun_clean <- finbramun %>%
  select(
    id_municipio = Cod.IBGE,
    coluna = Coluna,
    conta = Conta,
    valor = Valor
  ) %>%
  filter(id_municipio %in% municode$id_municipio,
    coluna == "Receitas Brutas Realizadas") %>%
  select(-coluna) %>%
  as_tibble() %>%
  separate(conta,
    c("conta_num", "descricao"),
    sep = " ",
    extra = "merge") %>%
  filter(conta_num %>% str_detect("(\\d\\.){4}(\\d){2}(\\.\\d){2}"))

# somente iss e iptu
munitax <- finbramun_clean %>%
  filter(conta_num %in% c("1.1.1.8.01.1.0", "1.1.1.8.02.3.0")) %>%
  pivot_wider(names_from = descricao, values_from = valor) %>%
  select(id_municipio, iptu = 3, iss = 4) %>%
  left_join(municode)

munitax2 <- munitax %>%
  group_by(id_municipio, nome) %>%
  summarise(across(c("iptu", "iss"), ~ sum(.x, na.rm = TRUE))) %>%
  ungroup() %>%
  mutate(id_municipio = as.character(id_municipio)) %>%
  left_join(pib_mun) %>%
  mutate(
    iptu_pib = iptu / pib,
    iss_pib = iss / pib,
    sd12_iptu = 1 / iptu_pib,
    sd12_iss = 1 / iss_pib
  ) %>%
  select(id_municipio,
    nome,
    ano_tax = ano,
    iptu_pib,
    iss_pib,
    pib,
    sd12_iptu,

```



```

      sd12_iss) %>%
    arrange(-sd12_ipitu,-sd12_iss)

munitax2[10, 8] <- 0 # atribui valor zero para belford roxo
munitax2[22, 8] <- 0 # atribui valor zero para uberaba
munitax2[30, 8] <- 0 # atribui valor zero para s.j meriti

# econtra muni faltantes

# maiores munis que nao estao nos dados
m <- setdiff(municode$id_municipio, munitax2$id_municipio)

faltantes <- municode %>%
  filter(id_municipio %in% m) %>%
  mutate(id_municipio = as.character(id_municipio))

# e lhes da nota zero
munitax2 <- munitax2 %>%
  ungroup() %>%
  add_row(
    id_municipio = faltantes$id_municipio,
    nome = faltantes$nome,
    ano_tax = rep("2018", 2),
    iptu_pib = rep("0", 2),
    iss_pib = rep("0", 2),
    pib = rep("0", 2),
    sd12_ipitu = rep("0", 2),
    sd12_iss = rep("0", 2)
  ) %>%
  filter(!(is.na(nome) | is.na(sd12_iss))) %>%
  distinct(nome, .keep_all = TRUE)

sd12 <- munitax2 %>%
  select(1, 2, i122 = 7, i123 = 8)

write_excel_csv(sd12, "dados_finais/sd12_munitax.xlsx")

# indice de gestao fiscal -----

iqgf <-
  read_excel(
    "dados_finais/sd12_qualidade_gestao_fiscal.xlsx",
    sheet = "Indicador",
    col_names = c("sigla_uf", "nome", "i124"),
    skip = 1
  ) %>%
  full_join(municode, keep = FALSE) %>%
  select(id_municipio, everything()) %>%
  arrange(i124)

write_excel_csv(iqgf, "dados_finais/sd12_qualidade_gestao_fiscal.xlsx")

```

sd13

voltar para Complexidade Tributária

```
#
# sd13 - ambiente regulatorio - complexidade tributaria

library(tidyverse)
library(readxl)
library(pdftools)

# maiores municipios
municode <-
  read_csv("municode.csv") %>% select(id_municipio, sigla_uf, nome)

# conta das receitas para calcular os indices
contas_interessantes <- pdf_text("ambiente_regulatorio/pdf_contas.pdf") %>%
  str_split("\n") %>%
  unlist() %>%
  str_extract_all("(\\d\\.){4}(\\d){2}(\\.\\d){2}") %>%
  unlist()

# receitas dos municipios
df <- read.delim(
  "ambiente_regulatorio/dados/finbramun.csv",
  dec = ",",
  sep = ";",
  skip = 3
)

df_sep <- df %>%
  separate(Conta,
    c("conta_num", "descricao"),
    sep = " ",
    extra = "merge")

finbramun <- df_sep %>%
  select(
    id_municipio = Cod.IBGE,
    coluna = Coluna,
    conta = Conta,
    valor = Valor
  ) %>%
  filter(id_municipio %in% municode$id_municipio,
    coluna == "Receitas Brutas Realizadas") %>%
  select(-coluna) %>%
  as_tibble() %>%
  separate(conta,
    c("conta_num", "descricao"),
    sep = " ",
    extra = "merge") %>%
  filter(conta_num %>% str_detect("(\\d\\.){4}(\\d){2}(\\.\\d){2}"))

# HH index
ihh <- finbramun %>%
```

```

group_by(id_municipio) %>%
mutate(receita_total = sum(valor, na.rm = TRUE)) %>%
ungroup() %>%
filter(conta_num %in% contas_interessantes[1:52]) %>%
group_by(id_municipio, conta_num) %>%
summarise(conta_total_quad = (valor / receita_total) ^ 2) %>%
group_by(id_municipio) %>%
summarise(ihh = sum(conta_total_quad))

# iv index
iv <- finbramun %>%
  group_by(id_municipio) %>%
  mutate(receita_total = sum(valor)) %>%
  ungroup() %>%
  filter(conta_num %in% contas_interessantes[53:56]) %>%
  group_by(id_municipio, receita_total) %>%
  summarise(conta_total = sum(valor)) %>%
  mutate(iv = conta_total / receita_total) %>%
  select(id_municipio, iv)

# tem um 'municipio' faltando: brasilia
municode %>%
  filter(id_municipio == setdiff(municode$id_municipio, finbramun$id_municipio))

#tudo junto
df <- left_join(ihh, iv) %>%
  mutate(sd13_complexidade = ihh * iv) %>%
  left_join(municode) %>%
  select(id_municipio, sigla_uf, nome, everything()) %>%
  arrange(-sd13_complexidade) %>%
  add_row(
    id_municipio = 5300108,
    nome = "Brasília",
    ihh = 0,
    iv = 0,
    sd13_complexidade = 0
  )

# salvar
df %>% select(1:3, i131 = 6) %>%
  write_excel_csv(df, "dados_finais/sd13_simplicidade_tributaria.xlsx")

# CND -----
sd13_cnd <-
  read_excel("ambiente_regulatorio/dados/Base_MUNIC_2019_20210817.xlsx",
    sheet = 4) %>%
  select(id_municipio = 1, tem_cnd = MTIC1211) %>%
  mutate(tem_cnd = if_else(tem_cnd == "Sim", 1, 0)) %>%
  right_join(municode, keep = FALSE) %>%
  select(id_municipio, sigla_uf, nome, i132 = tem_cnd)

write_excel_csv(sd13_cnd, "dados_finais/sd13_cnd.xlsx")

```

```

# atualizacao de zoneamento -----

municode <- read_csv("municode.csv") %>%
  select(id_municipio, sigla_uf, nome)

sd13_zone <- read_excel(
  "ambiente_regulatorio/dados/Base_MUNIC_2018_xlsx_20201103.xlsx",
  sheet = 3,
  na = "-"
) %>%
  select(id_municipio = 1, update_zone = MLEG061) %>%
  mutate(dif = if_else(is.na(update_zone), Inf, 2019 - as.double(update_zone)),
         indice = 1 / dif) %>%
  right_join(municode, keep = FALSE) %>%
  select(id_municipio, sigla_uf, nome, i133 = indice)

write_excel_csv(sd13_zone, "dados_finais/sd13_zone.xlsx")

```

sd21

voltar para Transporte Interurbano

```
#
#
# SD21 - INFRAESTRUTURA - TRANSPORTE INTERURBANO

library(tidyverse)
library(sf)
library(geobr)

# municipios -----
mcoords <- read_municipality() %>% # coordenadas dos municipios
  select(id_municipio = 1,
         nome = 2,
         geom = 5)

municode <- read_csv("municode.csv") %>%
  select(id_municipio, sigla_uf, nome) %>%
  mutate(nome = str_to_title(nome)) %>%
  left_join(mcoords, keep = FALSE)

# aeroportos br -----

#unzip("infraestrutura/aeroportos/aerodromos-zip.zip",
#      exdir = "infraestrutura/aeroportos/voos_shapefiles",
#      junkpaths = TRUE, overwrite = TRUE)

aeroshp <-
  read_sf("infraestrutura/aeroportos/voos_shapefiles/Aerodromos.shp")

siglas_geo <- aeroshp %>%
  mutate(nm_municipip = str_to_title(nm_municipip)) %>%
  select(sigla_aero = cod_icao) # geometry is sticky

# voos 2020 -----
# unzip os dados apenas uma vez

#unzip("infraestrutura/aeroportos/Dados_Estatisticos.zip",
#      exdir = "infraestrutura/aeroportos/dados_voos",
#      junkpaths = TRUE, overwrite = TRUE)

# voos no mundo
voos <- read.delim("infraestrutura/aeroportos/dados_voos/Dados Estatísticos.csv",
                  sep = ";")

# no br
voos_br <- voos %>%
  as_tibble() %>%
  select(ANO,
         MÊS,
         contains("AEROPORTO.DE.ORIGEM"),
         GRUPO.DE.VOO,
```

```

    DECOLAGENS) %>%
filter(ANO == 2020,
      AEROPORTO.DE.ORIGEM..PAÍS. == "BRASIL",
      GRUPO.DE.VOO == "REGULAR") %>%
rename(sigla_aero = AEROPORTO.DE.ORIGEM..SIGLA.,
       nome = AEROPORTO.DE.ORIGEM..NOME.,
       sigla_uf = AEROPORTO.DE.ORIGEM..UF.) %>%
mutate(nome = str_to_title(nome),
       sigla_uf = if_else(nome == "Guaíra", "PR", sigla_uf)) %>% # dados errados
group_by(sigla_aero, sigla_uf, nome) %>%
summarise(decolagens = sum(DECOLAGENS, na.rm = TRUE)) %>%
ungroup()

# no br e nos 100 maiores munípios

decolagens <- voos_br %>%
  group_by(sigla_uf, nome) %>%
  summarise(decolagens = sum(decolagens, na.rm = TRUE)) %>%
  ungroup() %>%
  right_join(municode) %>%
  select(id_municipio, everything()) %>%
  arrange(id_municipio)

# nao estao na lista
naotem <- decolagens %>% filter(is.na(decolagens))

# geolocaliza os aeroportos br
voos_br <- left_join(voos_br, siglas_geo)

# data viz -----
# br_airports <- ggplot()+
#   geom_sf(data = geobr::read_state())+ # mapa base com shapes dos estados br
#   geom_sf(data = naotem$geometry, fill = "red")+
#   geom_sf(data = voos_br$geometry)+
#   theme(panel.background = element_blank(),
#         axis.text = element_blank(),
#         axis.ticks = element_blank())

# idsp <- naotem %>% filter(sigla_uf == "SP") %>% pull(id_municipio)
# sp <- filter(naotem, id_municipio %in% idsp)
# sp_voos <- filter(voos_br, sigla_uf == "SP") %>% pull(geometry)

# sp_airports <- ggplot()+
#   geom_sf(data = geobr::read_state(code_state = "SP"))+
#   geom_sf(data = sp, fill = "red")+
#   geom_sf(data = sp_voos)+
#   theme(panel.background = element_blank(),
#         axis.text = element_blank(),
#         axis.ticks = element_blank())

# ggsave("infraestrutura/aeroportos/br_airports.png", br_airports)
# ggsave("infraestrutura/aeroportos/sp_airports.png", sp_airports)

```

```

# achar um aeroporto para os muni faltantes -----

aeroprox <- function(id) {
  mun <- filter(naotem, id_municipio == id) %>%
    st_as_sf()

  r <- st_distance(mun, st_as_sf(voos_br), by_element = TRUE) %>%
    which.min()

  voos_br %>% filter(row_number() == r) %>%
    as.data.frame() %>%
    select(sigla_aero)
}

# decolagens com base em aeroporto mais perto
siglas_faltantes <- naotem %>%
  as.data.frame() %>%
  pull(id_municipio) %>%
  map(aeroprox) %>%
  unlist()

naotem <- naotem %>%
  as.data.frame() %>%
  select(1:3) %>%
  bind_cols(siglas_faltantes) %>%
  rename(sigla_aero = 4) %>%
  left_join(voos_br, by = "sigla_aero", suffix = c("", "_ref")) %>%
  select(!geometry)

# colunas _ref NA significam aeroporto na cidade
sd21_voos <- decolagens %>%
  as.data.frame() %>%
  select(!geom) %>%
  filter(!is.na(decolagens)) %>%
  bind_rows(naotem) %>%
  arrange(-decolagens) %>%
  rename(i212 = decolagens)

write_excel_csv(sd21_voos, "infraestrutura/aeroportos/sd21_voos_completo.xlsx")

sd21_voos %>% select(1,2,3,4) %>% write_excel_csv("dados_finais/sd21_voos.xlsx")

#
#
# sd22 - INFRAESTRUTURA - CONDICOES URBANAS - TARIFA ENERGIA ELETRICA

library(readxl)
library(tidyverse)

# dados brutos
df <- read_excel("infraestrutura/energia_eletrica/RankingB1-24-08-2021.xlsx")

# selecionamos tarifa convencional e limpamos dos dados

```

```

tarifa <- df %>%
  select(dist = Distribuidora,
         sigla_uf = UF,
         tarifa = `Tarifa Convencional`) %>%
  mutate(dist = dist %>% str_to_lower())

mudar_dist <- c(
  "boa vista" = "roraima energia",
  "equatorial pa" = "celpa",
  "equatorial al" = "ceal",
  "equatorial pi" = "cepisa",
  "eletropaulo" = "enel sp",
  "cebdis" = "ceb-dis",
  "celg-d" = "enel go",
  "celesc dis" = "celesc-dis",
  "copel dis" = "copel-dis",
  "rge" = "rge (agrupada)"
)

dist <-
  read_csv("infraestrutura/energia_eletrica/distribuidoras.csv") %>%
  pivot_longer(matches("^d\\d$"), names_to = "isdist", values_to = "dist") %>%
  filter(!is.na(dist)) %>%
  select(!c(id_municipio_6, isdist, pop, sigla_uf)) %>%
  mutate(dist = dist %>% str_replace_all(mudar_dist)) %>%
  left_join(tarifa, by = "dist") %>%
  group_by(id_municipio) %>%
  mutate(tarifa_media = mean(tarifa)) %>%
  distinct(id_municipio, .keep_all = TRUE) %>%
  arrange(nome)

write_excel_csv(dist,
  "infraestrutura/energia_eletrica/sd22_energia_completa.xlsx")

dist %>% select(1, 4, 2, i223 = 6) %>%
  write_excel_csv("dados_finais/sd22_energia_eletrica.xlsx")
#
#
# SD21 - INFRAESTRUTURA - TRANSPORTE INTERURBANO - PORTOS

library(tidyverse)
library(sf)

# portos -----
# unzip(zipfile = "infraestrutura/portos/portos-zip.zip",
#       exdir = "infraestrutura/portos/portos_shapefile", junkpaths = TRUE,
#       overwrite = TRUE)

portos <-
  read_sf("infraestrutura/portos/portos_shapefile/Portos.shp",

```



```

    as_tibble = TRUE)

# seleciona portos publicos e os fluviais do amazonas
portos_am <-
  c("Itacoatiara", "Manaus", "Tabatinga", "Parintins", "Eirunepé")

am <- portos %>% filter(MUNICIPIO %in% portos_am)

pub <- portos %>%
  filter(!str_detect(NOMEPORTO, "TUP"),
         SITUACAOPO == "Operando", !is.na(MUNICIPIO)) %>%
  bind_rows(am) %>%
  select(
    nome_porto = 3,
    sigla_uf = 21,
    nome = 20,
    geometry = 31
  ) %>%
  distinct(nome_porto, .keep_all = TRUE)

# municipios -----
municode <- read_csv("municode.csv") %>%
  select(id_municipio, sigla_uf, nome)

m <- geobr::read_municipality(year = 2018) %>%
  select(id_municipio = 1, geom) %>%
  right_join(municode) %>%
  select(id_municipio, sigla_uf, nome, geom) %>%
  st_as_sf()

# teste -----
# exemplo ananindeua - PA, amostra e centro

# anan <- m %>% filter(id_municipio == 1500800)
# s <- st_sample(anan$geom, size = 15)

# ananindeua_ports <- ggplot()+
#   geom_sf(data = anan, fill = "grey")+
#   geom_sf(data = s, color = "blue")+
#   geom_sf(data = st_centroid(anan), color = "red", size=4)+
#   theme(panel.background = element_blank(),
#         axis.text = element_blank(),
#         axis.ticks = element_blank())

# ggsave("infraestrutura/portos/ananindeua_ports.png", ananindeua_ports)

# funcao distancia portos ao centro do muni

dist <- function(id) {
  set.seed(1)
  m %>%
    filter(id_municipio == id) %>%
    st_centroid() %>%

```

```

    st_distance(pub$geometry, by_element = TRUE)
}

# distancia de cada muni a cada porto
munidist <- m$id_municipio %>%
  map(dist) %>%
  set_names(nm = m$id_municipio) %>%
  as_tibble()

# transpoe munidist e nomeia as colunas
# menor distancia em km e seu inverso (o indicador)

munidistt <- cbind(id_municipio = names(munidist), t(munidist)) %>%
  as_tibble %>%
  set_names(c("id_municipio", pub$nome_porto)) %>%
  mutate(across(-id_municipio, ~ round(as.numeric(.) / 1000)),
         id_municipio = as.numeric(id_municipio)) %>%
  rowwise(id_municipio) %>%
  mutate(menor_dist = min(c_across(-1)),
         sd21_portos = 1 / menor_dist) %>%
  select(id_municipio, menor_dist, sd21_portos, everything())

df <- left_join(municode, munidistt, keep = FALSE) %>%
  arrange(-sd21_portos)

write_excel_csv(df, "infraestrutura/portos/sd21_portos_completo.xlsx")

df2 <- df %>% select(1,2,3,5) %>% rename(i213 = sd21_portos)

write_excel_csv(df2, "dados_finais/sd21_portos.xlsx")
#
#
# SD 21 - INFRAESTRUTURA - TRANSPORTE INTERURBANO

library(tidyverse)
library(geobr)
library(sf)
# rodovias -----
# federais
unzip(
  "infraestrutura/rodovias/vw_snv_rod.zip",
  exdir = "infraestrutura/rodovias/fedroads_shapefile",
  junkpaths = TRUE,
  overwrite = TRUE
)

br <-
  sf::st_read("infraestrutura/rodovias/fedroads_shapefile/vw_snv_rod.shp",
             as_tibble = TRUE) %>%
  select(codigo =Codigo_BR, geometry) %>%
  mutate(codigo = paste0("br_", codigo))

```

```

# estaduais
unzip(
  "infraestrutura/rodovias/vw_cide_rod.zip",
  exdir = "infraestrutura/rodovias/estroads_shapefile",
  junkpaths = TRUE,
  overwrite = TRUE
)

er <-
  st_read("infraestrutura/rodovias/estroads_shapefile/vw_cide_rod.shp",
    as_tibble = T) %>%
  select(codigo = Codigo_Rod, geometry) %>%
  mutate(codigo = paste0("es_", codigo))
# rodovias federais e estaduais
roads <- bind_rows(br, er)

# localizacao dos municipios -----
municode <-
  read_csv("municode.csv") %>% select(id_municipio, sigla_uf, nome)

m <- geobr::read_municipality() %>%
  select(id_municipio = 1, geom) %>%
  right_join(municode) %>%
  select(id_municipio, sigla_uf, nome, geom) %>%
  st_as_sf()

p <- geobr::read_state() %>%
  ggplot() +
  geom_sf(
    fill = "white",
    color = "grey",
    size = .15,
    show.legend = F
  ) +
  geom_sf(data = m, fill = "#2D3E50") +
  geom_sf(data = roads %>% sample(1), color = "#FEBF57") +
  theme(
    panel.background = element_blank(),
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank()
  )

ggsave("infraestrutura/rodovias/exemplo_rodovia.png", p)

# funcoes auxiliares -----
roads_inteiras <- roads %>% split(.$codigo)

passam <- function(r) {
  i <- 1:100
  muninterr <- function(i) {
    # verifica se municipio m intersecta algum trecho da rodovia r
    st_intersects(m[i, ], r, sparse = FALSE) %>% any()
  }
}

```

```

}
i %>% map(possibly(muninterr, "erro"))
}

recebem <- roads_inteiras %>% map(passam)
# intersecao rodovias e municipios -----

df <- recebem %>% map_dfc(unlist) %>%
  filter(!br_010 == "erro") %>% # exclui Rio Branco
  mutate(across(where(is.character), as.logical)) %>%
  rowwise() %>%
  mutate(n_estradas = sum(c_across(everything())) %>%
  add_column(id_municipio = m$id_municipio[-2], .before = 1)

df_final <- df %>% select(1, ncol(df)) %>%
  left_join(municode) %>%
  select(id_municipio, sigla_uf, nome, i211 = n_estradas) %>%
  add_row(
    id_municipio = 1200401,
    sigla_uf = "AC",
    nome = "Rio Branco",
    i211 = 4
  )

write_excel_csv(df_final, "dados_finais/sd21_rodovias.xlsx")

```

sd22

voltar para Condições Urbanas

```
#
#
# sd22 INFRAESTRUTURA - CONDICÕES URBANAS - HOMICÍDIOS

library(tidyverse)

# dados brutos
municode <- read_csv("municode.csv") %>%
  select(id_municipio_6, id_municipio, sigla_uf, pop)

# le, separa e funde com os dados de municípios
h <-
  read_delim(
    "infraestrutura/homicidios/dados_homicidios",
    skip = 1,
    delim = ";",
    escape_double = FALSE,
    trim_ws = TRUE,
    col_names = c("municipio", "obitos")
  ) %>%
  separate(
    col = municipio,
    into = c("id_municipio_6", "municipio"),
    sep = " ",
    extra = "merge",
    convert = TRUE
  )

# calcula a taxa e o indicador
homicidios <- h %>%
  right_join(municode) %>%
  mutate(taxa_homicidios = obitos / pop,
         i224 = 1 / taxa_homicidios)

write_excel_csv(homicidios, "infraestrutura/homicidios/sd22_homicidios.xlsx")

homicidios %>%
  select(4,5,2,8) %>%
  write_excel_csv("dados_finais/sd22_homicidios.xlsx")
#
#
# SD22 - INFRAESTRUTURA - Acesso a internet rapida-----

library(tidyverse)

municode <- read_csv("municode.csv") %>% select(id_municipio, pop)

df <- read_delim(
  file = "infraestrutura/internet/Acessos_Banda_Larga_Fixa_2019-2020_Colunas.csv",
  delim = ";",
```

```

escape_double = FALSE,
trim_ws = TRUE
)

internet <- df %>%
  select(UF,
         id_municipio = `Código IBGE Município`,
         Município,
         `Faixa de Velocidade`,
         `2020-01`:`2020-12`) %>%
  filter(`Faixa de Velocidade` %in% c("> 34Mbps", "12Mbps a 34Mbps")) %>%
  gather(`2020-01`:`2020-12`, key = "mes", value = "acessos") %>%
  group_by(id_municipio, Município) %>%
  summarise(acessos = sum(acessos, na.rm = T)) %>%
  right_join(municode, keep = F) %>%
  mutate(acessos_hab = acessos / pop)

write_excel_csv(internet,
                "infraestrutura/internet/sd22_internet_completo.xlsx")

internet %>%
  select(1, 2, i221 = 5) %>%
  write_excel_csv("dados_finais/sd22_internet.xlsx")
#
#
# SD22 - PREÇO DO METRO QUADRADO -----
# bibliotecas -----
library(httr)
library(tidyverse)
library(basedosdados)
library(abjutils)

# funcao para baixar dados do site -----

get_muni <- function(uf, muni) {
  # import data-----
  u <- "https://glue-api.zapimoveis.com.br/v2/listings"
  query <- list(
    business = "SALE",
    categoryPage = "RESULT",
    includeFields = "search",
    listingType = "USED",
    addressState = uf,
    addressCity = muni,
    portal = "ZAP",
    addressType = "city",
    size = "350"
  )
  h <- httr::add_headers("X-domain" = "www.zapimoveis.com.br")

  resultado <- GET(u, query = query, h) %>%

```

```

content() %>%
  pluck("search", "result", "listings")

# explore -----
# substitui NULL (ausencia de dado) por NA (indica que nao ha dado)
nullToNA <- function(x) {
  x[unlist(map(x, is.null))] <- NA
  return(x)
}

area <- resultado %>%
  map( ~ pluck(., "listing", "usableAreas")) %>%
  nullToNA() %>%
  unlist() %>%
  as.numeric()

endereco <- resultado %>%
  map( ~ pluck(., "listing", "address", "locationId")) %>%
  nullToNA() %>%
  unlist()

# preços
get_price <- function(x) {
  unlist(price[[x]])['price']
}

price <- resultado %>%
  map( ~ pluck(., "listing", "pricingInfos")) %>%
  nullToNA()

pricef <- seq_along(price) %>% map(get_price) %>%
  unlist() %>%
  as.numeric()

# tudo junto
imoveis <-
  tibble(endereco = endereco,
          price = pricef,
          area = area)
imoveis

}

# importa dados dos municipios -----

set_billing_id("ice2021")

nome_uf <- read_sql(
  "SELECT id_municipio, nome_uf
  FROM `basedosdados.br_bd_diretorios_brasil.municipio`"
)

municode <- read_csv("municode.csv") %>%

```

```

mutate(id_municipio = as.character(id_municipio)) %>%
left_join(nome_uf) %>%
select(id_municipio, nome_uf, nome)

# raspa os dados para cada municipio -----

terrenos <- map2_dfr(municode$nome_uf, municode$nome, get_muni)

# organiza e exclui dados faltantes
terrenos_final <- terrenos %>%
  filter(area != 0 & !is.na(area)) %>%
  separate(
    endereco,
    sep = ">",
    into = c("país", "nome_uf", "NULL", "nome", "bairro"),
    extra = "merge"
  ) %>%
  select(nome_uf, nome, bairro, price, area)

gabarito <- terrenos_final %>%
  mutate(`m^2` = price / area) %>%
  arrange(-`m^2`)

df <- terrenos_final %>%
  mutate(m2 = price / area) %>%
  filter(between(m2, 100, 20000)) %>%
  group_by(nome_uf, nome) %>%
  summarise(
    price_total = sum(price, na.rm = TRUE),
    area_total = sum(area),
    amostra = n(),
    m2 = price_total / area_total
  )

df_final <- municode %>%
  mutate(across(everything(), rm_accent)) %>%
  left_join(df) %>%
  mutate(s22_m2 = 1 / m2)

write_excel_csv(df_final, "infraestrutura/terrenos/sd22_m2_completo.xlsx")

```


sd31

voltar para Desenvolvimento Econômico

```
#
#
# sd31 - MERCADOS - DESENVOLVIMENTO ECONÔMICO

library(tidyverse)
library(readxl)
library(basedosdados)
library(sidrar)
library(abjutils)

municode <- read_csv("municode.csv") %>%
  select(id_municipio, sigla_uf, nome) %>%
  mutate(id_municipio = as.character(id_municipio))

# idh -----
idh <- read_excel("dados_finais/sd31_idh.xlsx",
                  col_names = c("id_municipio", "nome", "i311")) %>%
  right_join(municode) %>%
  select(id_municipio, sigla_uf, nome, i311)

write_excel_csv(idh, "dados_finais/sd31_idh.xlsx")

# crescimento medio real do pib -----
# pib nominal municipal

basedosdados::set_billing_id("ice2021")

pib_mun <- basedosdados::read_sql(query = "SELECT *
FROM `basedosdados.br_ibge_pib.municipio`
WHERE ano >= 2014 AND ano <= 2018") %>%
  select(1:3) %>%
  pivot_wider(names_from = ano,
              values_from = pib,
              names_prefix = "pib_") %>%
  right_join(municode) %>%
  select(id_municipio, sigla_uf, nome, everything())

# pib_corrente e passado em milhoes de reais
d <-
  get_sidra(
    x = 6784,
    period = c("last" = 5),
    variable = c(9808, 9809)
  ) %>%
  as_tibble() %>%
  select(var = Variável,
         ano = Ano,
         valor = Valor) %>%
  pivot_wider(names_from = var, values_from = valor) %>%
  rename(ano = 1,
```

```

      pib_corrente = 2,
      pib_passado = 3) %>%
rowwise() %>%
mutate(relativo = pib_corrente / pib_passado) %>%
pull(relativo)

d <-
  as.list(c(1, d[2], d[2] * d[3], d[2] * d[3] * d[4], d[2] * d[3] * d[4] *
    d[5]) * 100)
d
# deflacionar
names(d) <- pib_mun %>% select(where(is.numeric)) %>% names(.)

pib_mun_real <- pib_mun %>%
  mutate(
    across(all_of(names(d)), ~ .x * 100 / d[[cur_column()]]),
    var_1415 = (pib_2015 / pib_2014) - 1,
    var_1516 = (pib_2016 / pib_2015) - 1,
    var_1617 = (pib_2017 / pib_2016) - 1,
    var_1718 = (pib_2018 / pib_2017) - 1
  ) %>%
  rowwise() %>%
  mutate(sd312 = mean(c_across(contains("var_")))) %>%
  arrange(-sd312)

write_excel_csv(pib_mun_real, "mercado/sd31_pib_var.xlsx")

pib_mun_real %>% select(1,2,3,13) %>%
  write_excel_csv(pib_mun_real, "dados_finais/sd31_pib_var.xlsx")

# exportadoras -----

vinculos <- basedosdados::read_sql(
  "SELECT ano, id_municipio, qtde_vinculos_ativos,tamanho_estabelecimento
  FROM `basedosdados.br_me_rais.microdados_estabelecimentos`
  WHERE ano = 2019",
  page_size = 300000
)

temvinculo <- vinculos %>%
  filter(qtde_vinculos_ativos != 0) %>%
  group_by(id_municipio) %>%
  summarise(n_empresas = n()) %>%
  right_join(municode) %>%
  mutate(nome = nome %>% str_to_title() %>% rm_accent())

exportadoras <-
  read_excel("mercado/EMPRESAS_CADASTRO_2019.xlsx", skip = 7) %>%
  select(sigla_uf = UF, nome = MUNICÍPIO) %>%
  count(sigla_uf, nome, name = "n_exp") %>%
  mutate(nome = nome %>% str_to_title() %>% rm_accent())

df <- left_join(temvinculo, exportadoras) %>%

```

```
select(id_municipio, sigla_uf, nome, n_empresas, n_exp) %>%  
mutate(i313 = n_exp / n_empresas)  
  
write_excel_csv(df, "mercado/sd31_exportadoras_completo.xlsx")  
  
df %>% select(1, 2, 3, 6) %>%  
write_excel_csv(df, "dados_finais/sd31_exportadoras.xlsx")
```

sd32

voltar para Clientes Potenciais

```
#
#
# SD32 MERCADOS - CLIENTES POTENCIAS

library(tidyverse)
library(basedosdados)

municode <- read_csv("municode.csv") %>%
  select(id_municipio, sigla_uf, nome) %>%
  mutate(id_municipio = as.character(id_municipio))

# pib per capita -----

basedosdados::set_billing_id("ice2021")

# puxa populacao e pib municipal de todos os municipios
df <- tibble(
  query = c(
    "SELECT *
    FROM `basedosdados.br_ibge_populacao.municipio`
    WHERE ano = 2018",

    "SELECT id_municipio, pib
    FROM `basedosdados.br_ibge_pib.municipio`
    WHERE ano = 2018"
  )
) %>%
  mutate(resultados = query %>% map(~ basedosdados::read_sql(.x))) %>%
  pull(resultados) %>%
  reduce(full_join)

# calcula pib per capita
df2 <- df %>%
  right_join(municode) %>%
  select(ano, id_municipio, sigla_uf, nome, populacao, pib) %>%
  mutate(i321 = pib / populacao)

write_excel_csv(df2, "mercado/sd32_pib_capita_completo.xlsx")

df2 %>% select(2:4,7) %>% write_excel_csv("dados_finais/sd32_pib_capita.xlsx")

# proporcao empresas grandes e medias sobre medias e pequenas -----

vinculos <- basedosdados::read_sql(
  "SELECT ano, id_municipio, qtde_vinculos_ativos,tamanho_estabelecimento
  FROM `basedosdados.br_me_rais.microdados_estabelecimentos`
  WHERE ano = 2019 AND qtde_vinculos_ativos != 0",
  page_size = 200000 # linhas por página na query
)
```

```

df <- vinculos %>%
  select(id_municipio, qtde_vinculos_ativos) %>%
  mutate(
    qtde_vinculos_ativos = as.integer(qtde_vinculos_ativos),
    pequena = between(qtde_vinculos_ativos, 10, 49),
    media = between(qtde_vinculos_ativos, 50, 249),
    grande = qtde_vinculos_ativos > 250
  ) %>%
  group_by(id_municipio) %>%
  summarise(across(where(is.logical), ~ sum(.x))) %>%
  right_join(municode) %>%
  select(id_municipio, sigla_uf, nome, everything()) %>%
  mutate(m_p = media / pequena,
         g_m = grande / media,
         i322 = m_p / g_m) %>%
  arrange(-i322)

write_excel_csv(df, "mercado/sd32_prop_empresas_completo.xlsx")

df %>%
  select(1:3,9) %>%
  write_excel_csv("dados_finais/sd32_prop_empresas.xlsx")

```