

# Projeto Integrador

Alunos:

Alvaro Bastos Goto

André Ribeiro de Brito

Lucas José Strujak

Vicenzo Freeze Agustini

## Introdução a Análise e Ciência de Dados

1) O trabalho prático deve inserir dados no banco de dados implantado no PostgreSQL usando a linguagem de programação R. Para isso, pode-se gerar dados artificiais ou fazer a carga através de arquivos externos (.csv, por exemplo).

#carregar pacotes utilizados

```
library(readxl)
```

```
library(tidyverse)
```

#abrir as planilhas

```
alvara2017<-read_excel("C:/Users/jaque/OneDrive/Pós Ciência de Dados UTFPR/Mod 1 - Projeto Integrador/RelatorioMensal_ALV_JAN_DEZ2017.xlsx", sheet="RelatorioMensal_ALV_JAN_DEZ2017")
```

```
alvara2018<-read_excel("C:/Users/jaque/OneDrive/Pós Ciência de Dados UTFPR/Mod 1 - Projeto Integrador/RelatorioMensal_ALV_JAN_DEZ2018.xlsx", sheet="RelatorioMensal_ALV_JAN_DEZ2018")
```

```
alvara2019<-read_excel("C:/Users/jaque/OneDrive/Pós Ciência de Dados UTFPR/Mod 1 - Projeto Integrador/RelatorioMensal_ALV_JAN_DEZ2019.xlsx", sheet="RelatorioMensal_ALV_JAN_DEZ2019")
```

```
alvara2020<-read_excel("C:/Users/jaque/OneDrive/Pós Ciência de Dados UTFPR/Mod 1 - Projeto Integrador/RelatorioMensal_ALV_JAN_DEZ2020.xlsx", sheet="RelatorioMensal_ALV_JAN_DEZ2020")
```

```
alvara2021<-read_excel("C:/Users/jaque/OneDrive/Pós Ciência de Dados UTFPR/Mod 1 - Projeto Integrador/RelatorioMensal_ALV_JAN_DEZ2021.xlsx", sheet="RelatorioMensal_ALV_JAN_DEZ2021")
```

#união dos arquivos em uma tabela única

```
alvara2017_2021_0<-rbind(alvara2017,alvara2018,alvara2019,alvara2020,alvara2021)
```

#transforma os atributos que contem registro de data e estão no tipo caracter para tipo data

```
alvara2017_2021_1<-mutate(alvara2017_2021_0,
```

```
  `Data Criação Alvará`=str_sub(`Data Criação Alvará`, start = 2, end=11),
```

```
  `Data Início Obra`=str_sub(`Data Início Obra`, start = 2, end=11),
```

```
  `Data Conclusão Obra`=str_sub(`Data Conclusão Obra`, start = 2, end=11),
```

```
  `Data Vistoria`=str_sub(`Data Vistoria`, start = 2, end=11)
```

```
)
```

```
alvara2017_2021<-mutate(alvara2017_2021_1,
```

```
  `Data Criação Alvará`=strptime(`Data Criação Alvará`,format="%d/%m/%Y"),
```

```
  `Data Início Obra`=strptime(`Data Início Obra`,format="%d/%m/%Y"),
```

```
  `Data Conclusão Obra`=strptime(`Data Conclusão Obra`,format="%d/%m/%Y"),
```

```
  `Data Vistoria`=strptime(`Data Vistoria`,format="%d/%m/%Y"),
```

```
  `Número CVCO`=as.double(`Número CVCO`),
```

```
  `Tipo Vistoria`=as.character(`Tipo Vistoria`)
```

```
)
```

```
#cria tabela imovel
```

```
imovel<-select(alvara2017_2021,'Indicação Fiscal','Inscrição Imobiliária','Logradouro',  
  'Número','Bairro','Grupo Zoneamento','Abrangência',  
  'Quantidade Pavimentos','Quantidade de Unidades Residenciais',  
  'Quantidade Unidades Não Residenciais','Número Alvará')
```

```
#cria tabela alvara
```

```
alvara<-select(alvara2017_2021,'Data Criação Alvará','Data Início Obra','Data Conclusão Obra',  
  'Número Alvará','Uso(s) Alvará','Sub-Uso(s) Alvará','Finalidade','Material(is)',  
  'Metragem Área Remanescente','Metragem Construída Lote','Número de CAPACs Utilizadas',  
  'ACA-Área Adicional de Construção','Área Liberada','Metragem Área Reforma Alvará',  
  'Quantidade Blocos Alvará','Quantidade Sub-Solo Alvará','Número Registro Crea/Cau AU',  
  'Número Registro Crea/Cau RT','Firma Construtora','Número CVCO','Tipo Vistoria',  
  'Data Vistoria')
```

```
#cria tabela autor do projeto
```

```
autor_projeto0<-distinct(na.omit(select(alvara2017_2021,  
  'Número Registro Crea/Cau AU',  
  'Autor do Projeto',  
  'Data Criação Alvará'))))  
autor_projeto1<-mutate(author_projeto0,  
  Ano = as.numeric(format(`Data Criação Alvará`, "%Y"))*10000,  
  Mes = as.numeric(format(`Data Criação Alvará`, "%m"))*100,  
  Dia = as.numeric(format(`Data Criação Alvará`, "%d"))  
  )  
autor_projeto2<-mutate(author_projeto1, Data_Numero=Ano+Mes+Dia)  
autor_projeto3<-group_by(author_projeto2,`Número Registro Crea/Cau AU`)  
autor_projeto4<-summarise(author_projeto3, Maximo=max(Data_Numero))  
autor_projeto5<-merge(author_projeto3,autor_projeto4)  
autor_projeto6<-mutate(author_projeto5, Diferenca= Data_Numero - Maximo)  
autor_projeto7<-filter(author_projeto6, Diferenca==0)  
autor_projeto<-select(author_projeto7,'Número Registro Crea/Cau AU',  
  'Autor do Projeto')
```

```
#cria tabela responsavel tecnico
```

```
resp_tecnico0<-distinct(na.omit(select(alvara2017_2021,  
  'Número Registro Crea/Cau RT',  
  'Responsável Técnico',  
  'Data Criação Alvará'))))  
resp_tecnico1<-mutate(resp_tecnico0,  
  Ano = as.numeric(format(`Data Criação Alvará`, "%Y"))*10000,  
  Mes = as.numeric(format(`Data Criação Alvará`, "%m"))*100,  
  Dia = as.numeric(format(`Data Criação Alvará`, "%d"))  
  )  
resp_tecnico2<-mutate(resp_tecnico1, Data_Numero=Ano+Mes+Dia)  
resp_tecnico3<-group_by(resp_tecnico2,`Número Registro Crea/Cau RT`)  
resp_tecnico4<-summarise(resp_tecnico3, Maximo=max(Data_Numero))  
resp_tecnico5<-merge(resp_tecnico3,resp_tecnico4)  
resp_tecnico6<-mutate(resp_tecnico5, Diferenca= Data_Numero - Maximo)  
resp_tecnico7<-filter(resp_tecnico6, Diferenca==0)  
resp_tecnico<-select(resp_tecnico7,'Número Registro Crea/Cau RT',  
  'Responsável Técnico')
```

```
# Conectar e povoar Banco de dados ----
```

```
# Carregar pacotes
```

```
library(RPostgres)
```

```
con <- dbConnect(Postgres(),  
  user = "postgres",  
  password = "chonga",  
  host = 'localhost',  
  port = 5432,  
  dbname = "ProjetoIntegrador")
```

```
dbListTables(con)
```

```
#inserir dados na tabela autor_projeto
```

```
dbReadTable(con, "autor_projeto")
```

```
dbExecute(con,  
  "INSERT INTO autor_projeto(numero_crea_cau_au, nome)  
  VALUES ($1, $2)", params = unname(as.list(autor_projeto)))  
dbReadTable(con, "autor_projeto")
```

```
#inserir dados na tabela responsavel_tecnico
```

```
dbReadTable(con, "responsavel_tecnico")
```

```
dbExecute(con,  
  "INSERT INTO responsavel_tecnico(numero_crea_cau_rt, nome)  
  VALUES ($1, $2)", params = unname(as.list(resp_tecnico)))  
dbReadTable(con, "responsavel_tecnico")
```

```
#inserir dados na tabela alvara
```

```
dbReadTable(con, "alvara")
```

```
dbExecute(con,  
  "INSERT INTO alvara(data_criacao_alvara, data_inicio_obra, data_conclusao_obra,  
  numero_alvara, usos_alvara, sub_usos_alvara, finalidade, material,  
  metragem_area_remanescente, metragem_construida_lote, numero_de_capacs_utilizadas,  
  aca_area_adicional_de_construcao, area_liberada, metragem_area_reforma_alvara,  
  quantidade_blocos_alvara, quantidade_sub_solo_alvara, numero_crea_cau_au,  
  numero_crea_cau_rt, firma_construtora, numero_cvco, tipo_vistoria ,  
  data_vistoria)  
  VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10,  
  $11, $12, $13, $14, $15, $16, $17, $18, $19, $20, $21, $22)",  
  params = unname(as.list(alvara)))  
dbReadTable(con, "alvara")
```

```
#inserir dados na tabela imovel
```

```
dbReadTable(con, "imovel")
```

```
dbExecute(con,  
  "INSERT INTO imovel(indicacao_fiscal, inscricao_imobiliaria,  
  logradouro, numero, bairro, grupo_zoneamento, abrangencia,  
  quantidade_pavimento, quantidade_de_unidades_residenciais,  
  quantidade_unidades_ao_residenciais, numero_alvara)  
  VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10,  
  $11)",  
  params = unname(as.list(imovel)))  
dbReadTable(con, "imovel")
```

2) sua aplicação em R deve executar as 5 consultas definidas anteriormente

# 1 consulta envolvendo projeção e seleção

#projeção

```
resposta1<-as_tibble(dbGetQuery(con,
                                "SELECT indicacao_fiscal, logradouro, numero
                                FROM imovel
                                WHERE bairro ILIKE 'BOQUEIR%';"))
```

# 1 consulta que envolva junção externa (LEFT OUTER JOIN ou RIGHT OUTER JOIN ou FULL OUTER JOIN)

```
resposta2<-as_tibble(dbGetQuery(con,
                                "SELECT a.numero_alvara, rt.nome
                                FROM alvara a LEFT JOIN responsavel_tecnico rt
                                ON a.numero_crea_cau_rt=rt.numero_crea_cau_rt;"))
```

#1 consulta que envolva pelo menos uma operação de conjunto (UNION ou INTERSECT ou EXCEPT)

```
resposta3<-as_tibble(dbGetQuery(con,
                                "SELECT rt.nome FROM responsavel_tecnico rt
                                UNION
                                SELECT au.nome FROM autor_projeto au;"))
```

#1 consulta que envolva divisão relacional

```
resposta4<-as_tibble(dbGetQuery(con,
                                "SELECT au.nome
                                FROM autor_projeto au
                                WHERE EXISTS ( (SELECT DISTINCT numero_alvara
                                FROM alvara)
                                EXCEPT
                                (SELECT numero_alvara FROM alvara
                                WHERE au.numero_crea_cau_au= alvara.numero_crea_cau_au));"))
```

#1 consulta com operação de agregação e agrupamento (Função de agregação + GROUP BY)

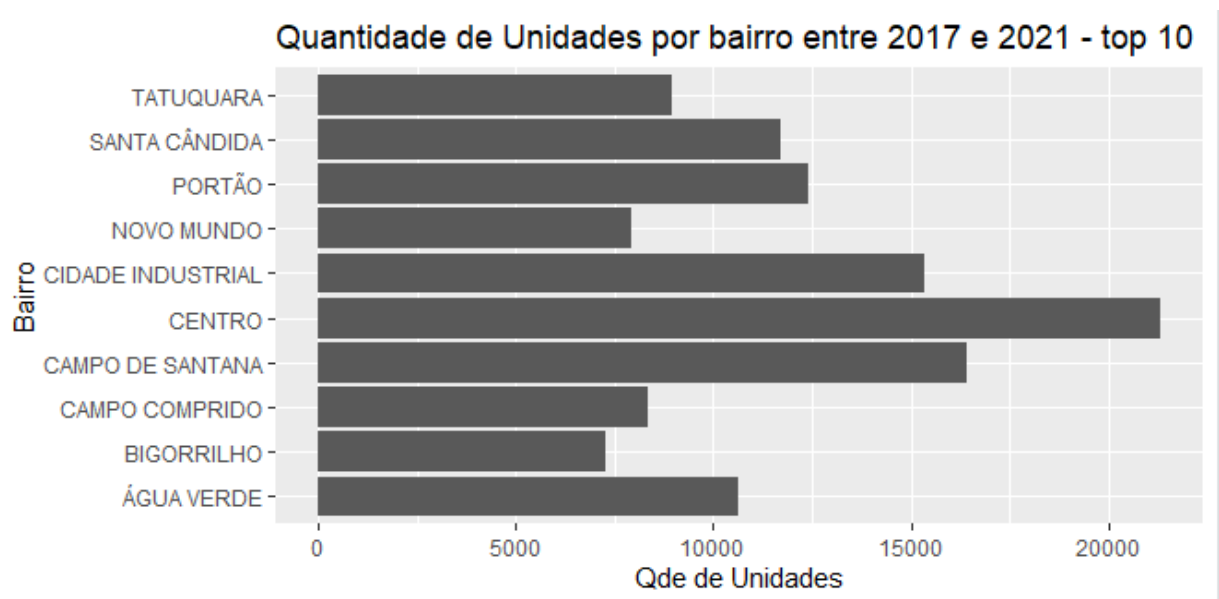
```
resposta5<-as_tibble(dbGetQuery(con,
                                "SELECT finalidade, count(alvara), avg(metragem_area_reforma_alvara)
                                FROM alvara
                                WHERE metragem_area_reforma_alvara>0 AND FINALIDADE IS NOT NULL
                                GROUP BY finalidade;"))
```

3) apresentar os resultados em vários formatos diferentes: (i) gráficos de barras, linhas e colunas, (ii) tabelas

# gráfico de barras: Top 10 regiões com mais alvarás de unidades residenciais e não residências entre 2017-2021

```
grafico1<-as_tibble(dbGetQuery(con,
  "SELECT bairro,
sum(quantidade_de_unidades_residenciais)+sum(quantidade_de_unidades_residenciais) as
Qde_Unidades
FROM alvara al, imovel im
WHERE al.numero_alvara=im.numero_alvara
GROUP BY BAIRRO
ORDER BY Qde_Unidades DESC
LIMIT 10;"))
```

```
grafico1 %>%
  ggplot(aes(x = qde_unidades, y = bairro)) +
  geom_col() +
  theme(legend.position="none") +
  labs(title = "Quantidade de Unidades por bairro entre 2017 e 2021 - top 10",
    y = 'Bairro',
    x = 'Qde de Unidades')
```

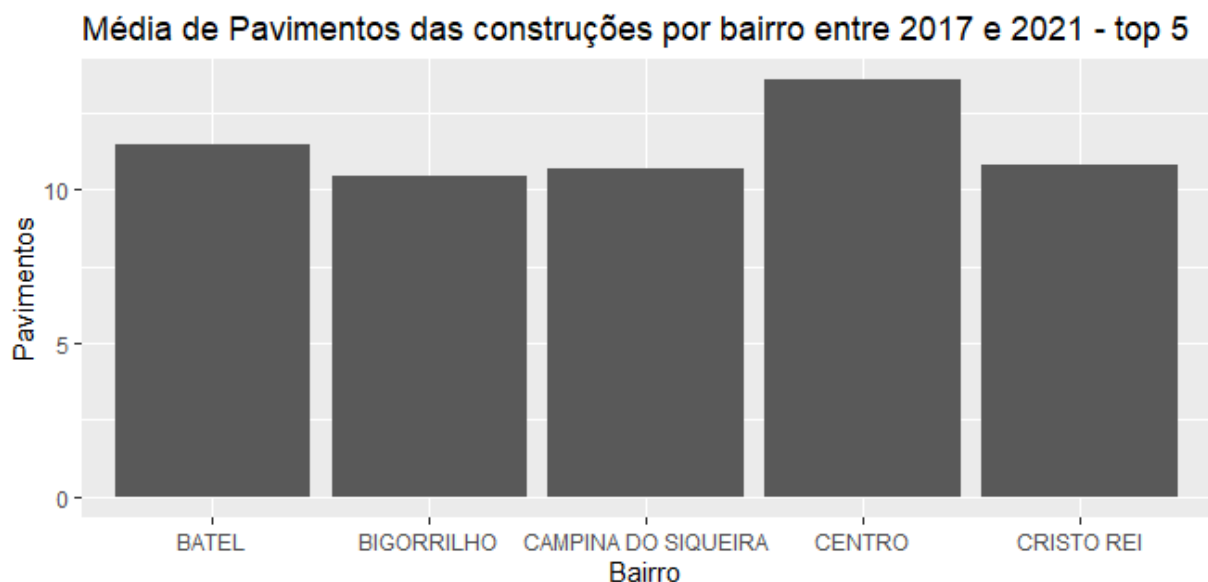


	bairro	qde_unidades
1	CENTRO	21316
2	CAMPO DE SANTANA	16396
3	CIDADE INDUSTRIAL	15340
4	PORTÃO	12400
5	SANTA CÂNDIDA	11700
6	ÁGUA VERDE	10628
7	TATUQUARA	8948
8	CAMPO COMPRIDO	8348
9	NOVO MUNDO	7904
10	BIGORRILHO	7256

# gráfico de colunas: Top 5 regiões com maiores quantidades médias de pavimentos por construção entre 2017 e 2021

```
grafico2<-as_tibble(dbGetQuery(con,
  "SELECT bairro, avg(quantidade_pavimento) as pavimentos
  FROM alvara al, imovel im
  WHERE al.numero_alvara=im.numero_alvara and quantidade_pavimento>=4
  GROUP BY BAIRRO
  ORDER BY pavimentos DESC
  limit 5;"))
```

```
grafico2 %>%
  ggplot(aes(x = bairro, y = pavimentos)) +
  geom_col() +
  theme(legend.position="none") +
  labs(title = "Média de Pavimentos das construções por bairro entre 2017 e 2021- top 5",
    y = 'Pavimentos',
    x = 'Bairro')
```

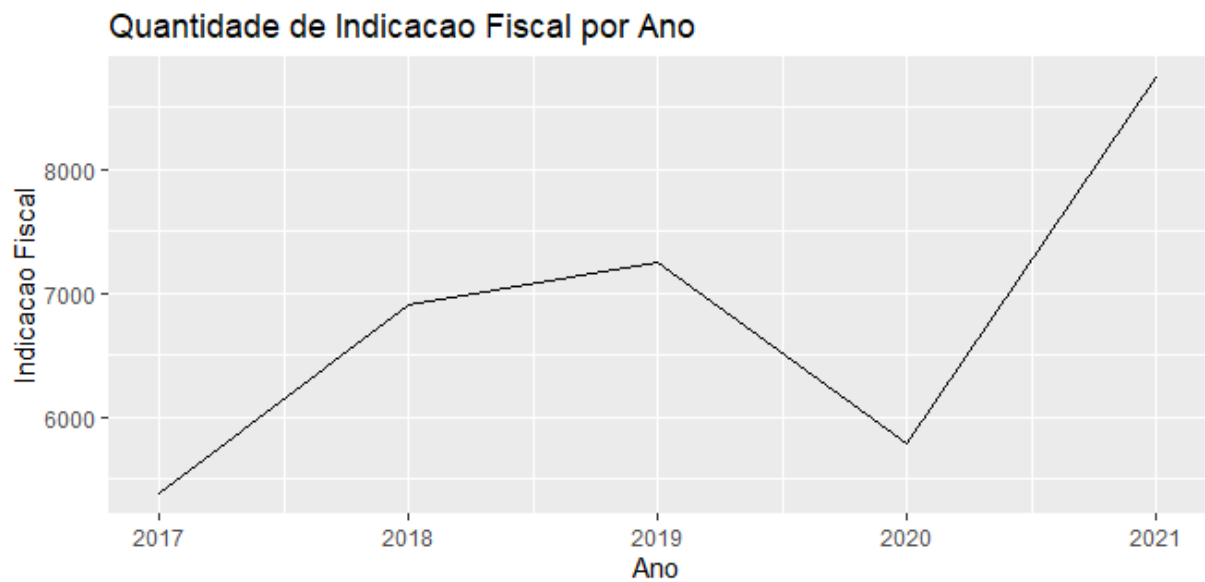


	bairro	pavimentos
1	CENTRO	13.57377
2	BATEL	11.48387
3	CRISTO REI	10.78261
4	CAMPINA DO SIQUEIRA	10.66667
5	BIGORRILHO	10.46341

# gráfico de linha: quantidade de indicação fiscal por ano

```
grafico3<-as_tibble(dbGetQuery(con,
  "SELECT extract(year from data_criacao_alvara) as Ano, count(indicacao_fiscal) as
  Indicacao_Fiscal
  FROM alvara al, imovel im
  WHERE al.numero_alvara=im.numero_alvara
  GROUP BY Ano
```

```
ORDER BY Ano;"))
grafico3 %>%
  ggplot(aes(x = as.integer(ano), y = as.integer(indicacao_fiscal))) +
  geom_line()+
  theme(legend.position="none") +
  labs(title = "Quantidade de Indicacao Fiscal por Ano",
        y = 'Indicacao Fiscal',
        x = 'Ano')
```



	ano	indicacao_fiscal
1	2017	5388
2	2018	6904
3	2019	7252
4	2020	5794
5	2021	8738