



ANEXO III

Código desenvolvido em linguagem R pelo Ceti-Saúde para geração automatizada no relatório em HTML.

```
*****
#Pacotes utilizados
```{r message=FALSE, warning=FALSE, echo=FALSE}
library(readxl)
library(dplyr)
library(lubridate)
library(tidyr)
library(stringr)
library(stringi)
library(openxlsx)
library(ggplot2)
library(lubridate)
library(dplyr)
library(readr)
```
*****
#Carregando a tabela

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela <- read.csv("ESAVI.csv",
 sep = ",",
 dec = ".",
 fileEncoding = "Latin1")
options(scipen = 999)
View(Tabela)
```
*****
# Questão 1

> Banco de dados

```{r}
View(Tabela)
```

> Quantas notificações graves entram no banco de dados no dia/trimestre/semestre/anual?

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
 select(Data.da.Notificação, Classificação.de.gravidade)

Tabela_reduzida <- Tabela_reduzida %>%
 separate_rows(
 Classificação.de.gravidade,
 sep = "\\|") %>%
 mutate(across(c(
```

---



```

 Classificação.de.gravidade),
 ~ str_trim(str_remove(x, "\\s*\\d+\\s*:\\s*"))
))
...

```

> Quantos casos graves entram por dia:

```

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  filter(Classificação.de.gravidade == "Grave") %>%
  count(Data.da.Notificação) %>%
  arrange(n)
...

```

> Quantos casos graves entram por trimestre:

```

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%
 filter(Classificação.de.gravidade == "Grave") %>%
 group_by(ano = year(Data.da.Notificação), trimestre = quarter(Data.da.Notificação)) %>%
 summarise(qtd = n()) %>%
 arrange(ano, trimestre)
...

```

> Quantos casos graves entram por semestre:

```

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%
  filter(Classificação.de.gravidade == "Grave") %>%
  mutate(semestre = ifelse(month(Data.da.Notificação) <= 6,1,2)) %>%
  group_by(ano = year(Data.da.Notificação), semestre) %>%
  summarise(qtd = n())
...

```

> Quantos casos graves entram por ano:

```

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%
 filter(Classificação.de.gravidade == "Grave") %>%
 group_by(ano = year(Data.da.Notificação)) %>%
 summarise(qtd = n())
...

```

\*\*\*\*\*

# Questão 2

> Quantas notificações graves: status em aberto/investigação/encerramento no dia/trimestre/semestre/anual?

```

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Data.da.Notificação, Classificação.de.gravidade, Situação.da.Notificação)
Tabela_reduzida <- Tabela_reduzida %>%
  separate_rows(
    Classificação.de.gravidade,
    sep = "\\|" ) %>%
  mutate(across(c(

```



```
Classificação.de.gravidade),
~ str_trim(str_remove(x, "^\\s*\\d+\\s*\\.\\s*"))
))
...

> Casos graves por dia com status encerrado:
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 filter(Classificação.de.gravidade == "Grave") %>%
 filter(Situação.da.Notificação == "Encerrado") %>%
 count(Data.da.Notificação) %>%
 arrange(n)
...

> Casos graves por trimestre com status encerrado:

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%
  filter(Classificação.de.gravidade == "Grave") %>%
  filter(Situação.da.Notificação == "Encerrado") %>%
  group_by(ano = year(Data.da.Notificação), trimestre = quarter(Data.da.Notificação)) %>%
  summarise(qtd = n())
...

> Casos graves por semestre com status encerrado:

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%
 filter(Classificação.de.gravidade == "Grave") %>%
 filter(Situação.da.Notificação == "Encerrado") %>%
 mutate(semestre = ifelse(month(Data.da.Notificação) <= 6,1,2)) %>%
 group_by(ano = year(Data.da.Notificação), semestre) %>%
 summarise(qtd = n())
...

> Casos graves por ano com status encerrado:

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%
  filter(Classificação.de.gravidade == "Grave") %>%
  filter(Situação.da.Notificação == "Encerrado") %>%
  group_by(ano = year(Data.da.Notificação)) %>%
  summarise(qtd = n())
...

> Casos graves por dia com status em aberto:

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 filter(Classificação.de.gravidade == "Grave") %>%
 filter(Situação.da.Notificação == "Aberto") %>%
 count(Data.da.Notificação) %>%
 arrange(n)
...

> Casos graves por trimestre com status em aberto:

```{r message=FALSE, warning=FALSE, echo=FALSE}
```



```
Tabela_reduzida %>%  
  mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%  
  filter(Classificação.de.gravidade == "Grave") %>%  
  filter(Situação.da.Notificação == "Aberto") %>%  
  group_by(ano = year(Data.da.Notificação), trimestre = quarter(Data.da.Notificação)) %>%  
  summarise(qtd = n())  
````
```

> Casos graves por semestre com status em aberto:

```
````{r message=FALSE, warning=FALSE, echo=FALSE}  
Tabela_reduzida %>%  
  mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%  
  filter(Classificação.de.gravidade == "Grave") %>%  
  filter(Situação.da.Notificação == "Aberto") %>%  
  mutate(semestre = ifelse(month(Data.da.Notificação) <= 6,1,2)) %>%  
  group_by(ano = year(Data.da.Notificação), semestre) %>%  
  summarise(qtd = n())  
````
```

> Casos graves por ano com status em aberto:

```
````{r message=FALSE, warning=FALSE, echo=FALSE}  
Tabela_reduzida %>%  
  mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%  
  filter(Classificação.de.gravidade == "Grave") %>%  
  filter(Situação.da.Notificação == "Aberto") %>%  
  group_by(ano = year(Data.da.Notificação)) %>%  
  summarise(qtd = n())  
````
```

> Casos graves por dia com status em avaliação:

```
````{r message=FALSE, warning=FALSE, echo=FALSE}  
Tabela_reduzida %>%  
  filter(Classificação.de.gravidade == "Grave") %>%  
  filter(Situação.da.Notificação == "Em Avaliação") %>%  
  count(Data.da.Notificação) %>%  
  arrange(n)  
````
```

> Casos graves por trimestre com status em avaliação:

```
````{r message=FALSE, warning=FALSE, echo=FALSE}  
Tabela_reduzida %>%  
  mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%  
  filter(Classificação.de.gravidade == "Grave") %>%  
  filter(Situação.da.Notificação == "Em Avaliação") %>%  
  group_by(ano = year(Data.da.Notificação), trimestre = quarter(Data.da.Notificação)) %>%  
  summarise(qtd = n())  
````
```

> Casos graves por semestre com status em avaliação:

```
````{r message=FALSE, warning=FALSE, echo=FALSE}  
Tabela_reduzida %>%  
  mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%  
  filter(Classificação.de.gravidade == "Grave") %>%  
  filter(Situação.da.Notificação == "Em Avaliação") %>%  
  mutate(semestre = ifelse(month(Data.da.Notificação) <= 6,1,2)) %>%
```



```
group_by(ano = year(Data.da.Notificação), semestre) %>%
summarise(qtd = n())
```

```

> Casos graves por ano com status em avaliação:

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida%>%
mutate(Data.da.Notificação = as.Date(Data.da.Notificação, format = "%d/%m/%Y")) %>%
filter(Classificação.de.gravidade == "Grave") %>%
filter(Situação.da.Notificação == "Em Avaliação") %>%
group_by(ano = year(Data.da.Notificação)) %>%
summarise(qtd = n())
```

```

\*\*\*\*\*

# Questão 3

> Quantas notificações graves encerramento Federal/CIFAVI? Se possível estratificar por período.

> Por dia

```
```{r}
Tabela %>%
filter(Classificação.de.gravidade == "Grave") %>%
filter(!is.na(`Encerramento.Cifavi...Data.Encerramento`)) %>%
mutate(data_notificacao = dmy(`Data.da.Notificação`)) %>%
count(data_notificacao)
```

```

> Por trimestre

```
```{r}
Tabela %>%
filter(Classificação.de.gravidade == "Grave") %>%
filter(!is.na(`Encerramento.Cifavi...Data.Encerramento`)) %>%
mutate(
data_notificacao = dmy(`Data.da.Notificação`),
ano = year(data_notificacao),
trimestre = quarter(data_notificacao)
) %>%
count(ano, trimestre)
```

```

> Por semestre

```
```{r}
Tabela %>%
filter(Classificação.de.gravidade == "Grave") %>%
filter(!is.na(`Encerramento.Cifavi...Data.Encerramento`)) %>%
mutate(
data_notificacao = dmy(`Data.da.Notificação`),
ano = year(data_notificacao),
semestre = if_else(month(data_notificacao) <= 6, 1, 2)
) %>%
count(ano, semestre)
```

```



> Por ano

```
```{r}
Tabela %>%
  filter(Classificação.de.gravidade == "Grave") %>%
  filter(!is.na(`Encerramento.Cifavi...Data.Encerramento`)) %>%
  mutate(ano = year(dmy(`Data.da.Notificação`))) %>%
  count(ano)
```

Questão 4

>Quantas notificações graves por óbito e seus status?

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Classificação.de.gravidade, Desfecho..evolução.do.caso., Situação.da.Notificação)
```

```
Tabela_reduzida <- Tabela_reduzida %>%
  separate_rows(
    Desfecho..evolução.do.caso.,
    Classificação.de.gravidade,
    sep = "\\|" %>%
  mutate(across(c(
    Desfecho..evolução.do.caso.,
    Classificação.de.gravidade),
    ~ str_trim(str_remove(x, "^\\s*\\d+\\s*\\.\\s*"))
  ))
```
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  filter(Classificação.de.gravidade == "Grave") %>%
  filter(Desfecho..evolução.do.caso. == "Óbito") %>%
  group_by(status = Situação.da.Notificação) %>%
  summarise(qtd = n())
```
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Número.da.Notificação, Data.da.Notificação, Classificação.de.gravidade,
Município.da.Notificação, Situação.da.Notificação)
```

```
Tabela_reduzida <- Tabela_reduzida %>%
  separate_rows(
    Classificação.de.gravidade,
    sep = "\\|" %>%
  mutate(across(c(
    Classificação.de.gravidade),
    ~ str_trim(str_remove(x, "^\\s*\\d+\\s*\\.\\s*"))
  ))
```
```

```

Questão 5
```



> Quantas notificações por Municípios anualmente total (Avaliação, aberto, encerrado)

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  mutate(ano = year(dmy(Data.da.Notificação))) %>% # extrai o ano da data
  group_by(ano, Município.da.Notificação, Número.da.Notificação) %>%
  summarise(qtd_notificacoes = n(), .groups = "drop") %>%
  arrange(ano, Município.da.Notificação)
```
```

> Quantas notificações por Municípios anualmente em avaliação

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  filter(Situação.da.Notificação == "Em Avaliação") %>%
  mutate(ano = year(dmy(Data.da.Notificação))) %>% # extrai o ano da data
  group_by(ano, Município.da.Notificação, Número.da.Notificação) %>%
  summarise(qtd_notificacoes = n(), .groups = "drop") %>%
  arrange(ano, Município.da.Notificação)
```
```

> Quantas notificações por Municípios anualmente encerrado

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  filter(Situação.da.Notificação == "Encerrado") %>%
  mutate(ano = year(dmy(Data.da.Notificação))) %>% # extrai o ano da data
  group_by(ano, Município.da.Notificação, Número.da.Notificação) %>%
  summarise(qtd_notificacoes = n(), .groups = "drop") %>%
  arrange(ano, Município.da.Notificação)
```
```

> Quantas notificações por Municípios anualmente em aberto

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  filter(Situação.da.Notificação == "Aberto") %>%
  mutate(ano = year(dmy(Data.da.Notificação))) %>% # extrai o ano da data
  group_by(ano, Município.da.Notificação, Número.da.Notificação) %>%
  summarise(qtd_notificacoes = n(), .groups = "drop") %>%
  arrange(ano, Município.da.Notificação)
```
```

\*\*\*\*\*

# Questão 6

> Quais os municípios que não registraram notificação e-SUS?

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
todos_municipios <- tibble::tibble(
  Municipio = c(
    "Abadia de Goiás", "Abadiânia", "Acreúna", "Adelândia", "Água Fria de Goiás",
    "Água Limpa", "Águas Lindas de Goiás", "Alexânia", "Aloândia", "Alto Horizonte",
    "Alto Paraíso de Goiás", "Alvorada do Norte", "Amaralina", "Americano do Brasil",
    "Amorinópolis", "Anápolis", "Ananguera", "Anicuns", "Aparecida de Goiânia",
    "Aparecida do Rio Doce", "Aporé", "Araçu", "Aragarças", "Aragoiânia", "Araguapaz",
    "Arenópolis", "Aruanã", "Aurilândia", "Avelinópolis", "Baliza", "Barro Alto",

```



```

"Bela Vista de Goiás", "Bom Jardim de Goiás", "Bom Jesus de Goiás", "Bonfinópolis",
"Bonópolis", "Brazabrantes", "Britânia", "Buriti Alegre", "Buriti de Goiás",
"Buritinópolis", "Cabeceiras", "Cachoeira Alta", "Cachoeira de Goiás",
"Cachoeira Dourada", "Caçu", "Caiapônia", "Caldas Novas", "Caldazinha",
"Campestre de Goiás", "Campinaçu", "Campinorte", "Campo Alegre de Goiás",
"Campo Limpo de Goiás", "Campos Belos", "Campos Verdes", "Carmo do Rio Verde",
"Castelândia", "Catalão", "Caturaí", "Cavalcante", "Ceres", "Cezarina",
"Chapadão do Céu", "Cidade Ocidental", "Cocalzinho de Goiás", "Colinas do Sul",
"Corumbá de Goiás", "Corumbaíba", "Cristalina", "Cristianópolis", "Crixás",
"Cumari", "Damianópolis", "Damolândia", "Davinópolis", "Diorama",
"Divinópolis de Goiás", "Doverlândia", "Edealina", "Edéia", "Estrela do Norte",
"Faina", "Fazenda Nova", "Firminópolis", "Flores de Goiás", "Formosa",
"Formoso", "Gameleira de Goiás", "Goianápolis", "Goiandira", "Goianésia",
"Goiania", "Goianira", "Goiás", "Goiatuba", "Gouvelândia", "Guapó",
"Guaraíta", "Guarani de Goiás", "Guarinos", "Heitorai", "Hidrolândia",
"Hidrolina", "Iaciara", "Inaciolândia", "Indiara", "Inhumas", "Ipameri",
"Ipiranga de Goiás", "Iporá", "Israelândia", "Itaberaí", "Itaguari", "Itaguaru",
"Itajá", "Itapaci", "Itapirapuã", "Itapuranga", "Itarumã", "Itauçu", "Itumbiara",
"Ivolândia", "Jandaia", "Jaraguá", "Jataí", "Jaupaci", "Jesópolis", "Joviânia",
"Jussara", "Lagoa Santa", "Leopoldo de Bulhões", "Luziânia", "Mairipotaba",
"Mambaí", "Mara Rosa", "Marzagão", "Matrinchã", "Maurilândia", "Mimoso de Goiás",
"Minaçu", "Mineiros", "Moiporá", "Monte Alegre de Goiás", "Montes Claros de Goiás",
"Montividiu", "Montividiu do Norte", "Morrinhos", "Morro Agudo de Goiás",
"Mossamedes", "Mozarlândia", "Mundo Novo", "Mutunópolis", "Nazário", "Nerópolis",
"Niquelândia", "Nova América", "Nova Aurora", "Nova Crixás", "Nova Glória",
"Nova Iguaçu de Goiás", "Nova Roma", "Nova Veneza", "Novo Brasil", "Novo Gama",
"Novo Planalto", "Orizona", "Ouro Verde de Goiás", "Ouvidor", "Padre Bernardo",
"Palestina de Goiás", "Palmeiras de Goiás", "Palmelo", "Palminópolis", "Panamá",
"Paranaiguara", "Paraúna", "Perolândia", "Petrópolis de Goiás", "Pilar de Goiás",
"Piracanjuba", "Piranhas", "Pirenópolis", "Pires do Rio", "Planaltina",
"Pontalina", "Porangatu", "Porteirão", "Portelândia", "Posse", "Professor Jamil",
"Quirinópolis", "Rialma", "Rianápolis", "Rio Quente", "Rio Verde", "Rubiataba",
"Sanclerlândia", "Santa Bárbara de Goiás", "Santa Cruz de Goiás",
"Santa Fé de Goiás", "Santa Helena de Goiás", "Santa Isabel", "Santa Rita do Araguaia",
"Santa Rita do Novo Destino", "Santa Rosa de Goiás", "Santa Tereza de Goiás",
"Santa Terezinha de Goiás", "Santo Antônio da Barra", "Santo Antônio de Goiás",
"Santo Antônio do Descoberto", "São Domingos", "São Francisco de Goiás",
"São João da Paraúna", "São João d'Aliança", "São Luis de Montes Belos",
"São Luiz do Norte", "São Miguel do Araguaia", "São Miguel do Passa Quatro",
"São Patrício", "São Simão", "Senador Canedo", "Serranópolis", "Silvânia",
"Simolândia", "Sítio d'Abadia", "Taquaral de Goiás", "Teresina de Goiás",
"Terezópolis de Goiás", "Três Ranchos", "Trindade", "Trombas", "Turvânia",
"Turvelândia", "Uirapuru", "Uruana", "Uruaçu", "Urutai", "Valparaíso de Goiás",
"Varão", "Vianópolis", "Vicentinópolis", "Vila Boa", "Vila Propício"
)
)

municipios_notificaram <- Tabela_reduzida %>%
  pull(Município.da.Notificação) %>%
  as.character()

# Extrai vetor de todos os municípios (sem sobrescrever o tibble!)
todos_municipios_vec <- todos_municipios$Município

# Diferença: quem não notificou

```



```
municipios_nao_notificaram <- setdiff(todos_municipios_vec, municipios_notificaram)

# Cria tibble final
df_nao_notificaram <- tibble::tibble(Municipio = municipios_nao_notificaram)

# Ordena em ordem alfabética (opcional)
df_nao_notificaram <- df_nao_notificaram %>% arrange(Municipio)

# Visualiza
df_nao_notificaram
````

Questão 7

> Qual o intervalo de tempo entre o ocorrido ESAVI e a data da notificação? (data de início - data da notificação)

````{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Data.da.Notificação, Data.de.início) %>%
  separate_rows(Data.de.início, sep = "\\|") %>%
  mutate(
    Data.de.início = str_trim(str_remove(Data.de.início, "^\\s*\\d+\\s*:\\s*")),
    # converter para Date
    Data.de.início = dmy(Data.de.início),
    Data.da.Notificação = dmy(Data.da.Notificação)
  )

Tabela_dias <- Tabela_reduzida %>%
  mutate(
    Diferença_dias = abs(as.numeric(difftime(Data.de.início, Data.da.Notificação, units = "days")))
  ) %>%
  arrange(Diferença_dias)

Tabela_dias
````

>Qual o intervalo de tempo entre a vacinação e o início dos sintomas? (aplicação da vacina - início dos sintomas)

````{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Data.da.aplicação, Data.de.início) %>%
  separate_rows(Data.de.início, sep = "\\|") %>%
  mutate(
    Data.de.início = str_trim(str_remove(Data.de.início, "^\\s*\\d+\\s*:\\s*")),
    Data.da.aplicação = str_trim(str_remove(Data.da.aplicação, "^\\s*\\d+\\s*:\\s*")),
    # converter para Date
    Data.de.início = dmy(Data.de.início),
    Data.da.aplicação = dmy(Data.da.aplicação)
  )

Tabela_dias <- Tabela_reduzida %>%
  mutate(
    Diferença_dias = abs(as.numeric(difftime(Data.de.início, Data.da.aplicação, units = "days")))
  ) %>%
```



```
arrange(Diferença_dias)
```

```
Tabela_dias
```

```
```
```

```
>Qual o intervalo de tempo entre a data de notificação e a data de encerramento ESTADUAL? (data da notificação - data de encerramento)
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
```

```
Tabela_reduzida <- Tabela %>%
```

```
  select(Encerramento.Estadual...Data.Encerramento, Data.da.Notificação) %>%
```

```
  separate_rows(Data.da.Notificação, sep = "\\|") %>%
```

```
  mutate(
```

```
    Data.da.Notificação = str_trim(str_remove(Data.da.Notificação, "^\\s*\\d+\\s*:\\s*")),
```

```
                                Encerramento.Estadual...Data.Encerramento =
```

```
str_trim(str_remove(Encerramento.Estadual...Data.Encerramento, "^\\s*\\d+\\s*:\\s*")),
```

```
    # converter para Date
```

```
    Data.da.Notificação = dmy(Data.da.Notificação),
```

```
    Encerramento.Estadual...Data.Encerramento = dmy(Encerramento.Estadual...Data.Encerramento)
```

```
)
```

```
Tabela_dias <- Tabela_reduzida %>%
```

```
  mutate(
```

```
    Diferença_dias = abs(as.numeric(difftime(Data.da.Notificação,
```

```
Encerramento.Estadual...Data.Encerramento, units = "days"))
```

```
  ) %>%
```

```
  arrange(Diferença_dias)
```

```
Tabela_dias
```

```
```
```

```
>Qual o intervalo de tempo entre a data de notificação e a data de encerramento MUNICIPAL? (data da notificação - data de encerramento)
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
```

```
Tabela_reduzida <- Tabela %>%
```

```
  select(Encerramento.Municipal...Data.Encerramento, Data.da.Notificação) %>%
```

```
  separate_rows(Data.da.Notificação, sep = "\\|") %>%
```

```
  mutate(
```

```
    Data.da.Notificação = str_trim(str_remove(Data.da.Notificação, "^\\s*\\d+\\s*:\\s*")),
```

```
                                Encerramento.Municipal...Data.Encerramento =
```

```
str_trim(str_remove(Encerramento.Municipal...Data.Encerramento, "^\\s*\\d+\\s*:\\s*")),
```

```
    # converter para Date
```

```
    Data.da.Notificação = dmy(Data.da.Notificação),
```

```
                                Encerramento.Municipal...Data.Encerramento =
```

```
dmy(Encerramento.Municipal...Data.Encerramento)
```

```
)
```

```
Tabela_dias <- Tabela_reduzida %>%
```

```
  mutate(
```

```
    Diferença_dias = abs(as.numeric(difftime(Data.da.Notificação,
```

```
Encerramento.Municipal...Data.Encerramento, units = "days"))
```

```
  ) %>%
```

```
  arrange(Diferença_dias)
```

```
Tabela_dias
```

```
```
```



>Qual o intervalo de tempo entre a data de notificação e a data de encerramento FEDERAL? (data da notificação - data de encerramento)

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Encerramento.Federal...Data.Encerramento, Data.da.Notificação) %>%
  separate_rows(Data.da.Notificação, sep = "\\|") %>%
  mutate(
    Data.da.Notificação = str_trim(str_remove(Data.da.Notificação, "^\\s*\\d+\\s*:\\s*")),
    Encerramento.Federal...Data.Encerramento =
str_trim(str_remove(Encerramento.Federal...Data.Encerramento, "^\\s*\\d+\\s*:\\s*")),
    # converter para Date
    Data.da.Notificação = dmy(Data.da.Notificação),
    Encerramento.Federal...Data.Encerramento = dmy(Encerramento.Federal...Data.Encerramento)
  )

Tabela_dias <- Tabela_reduzida %>%
  mutate(
    Diferença_dias = abs(as.numeric(difftime(Data.da.Notificação,
Encerramento.Federal...Data.Encerramento, units = "days")))
  ) %>%
  arrange(Diferença_dias)
```

Tabela_dias
```

>Qual o intervalo de tempo entre a data de notificação e a data de encerramento CIFAVI? (data da notificação - data de encerramento)

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Encerramento.Cifavi...Data.Encerramento, Data.da.Notificação) %>%
  separate_rows(Data.da.Notificação, sep = "\\|") %>%
  mutate(
    Data.da.Notificação = str_trim(str_remove(Data.da.Notificação, "^\\s*\\d+\\s*:\\s*")),
    Encerramento.Cifavi...Data.Encerramento =
str_trim(str_remove(Encerramento.Cifavi...Data.Encerramento, "^\\s*\\d+\\s*:\\s*")),
    # converter para Date
    Data.da.Notificação = dmy(Data.da.Notificação),
    Encerramento.Cifavi...Data.Encerramento = dmy(Encerramento.Cifavi...Data.Encerramento)
  )

Tabela_dias <- Tabela_reduzida %>%
  mutate(
    Diferença_dias = abs(as.numeric(difftime(Data.da.Notificação,
Encerramento.Cifavi...Data.Encerramento, units = "days")))
  ) %>%
  arrange(Diferença_dias)
```

Tabela_dias
```

\*\*\*\*\*

# Questão 8

> Quantas notificações classificadas por causalidade MUNICIPAL?



```
```{r message=FALSE, warning=FALSE, echo=FALSE}
```

```
Tabela_reduzida <- Tabela %>%  
  select(Encerramento.Municipal...Causalidade)
```

```
Tabela_reduzida <- Tabela_reduzida %>%  
  separate_rows(  
    Encerramento.Municipal...Causalidade,  
    sep = "\\|" ) %>%  
  mutate(across(c(  
    Encerramento.Municipal...Causalidade),  
    ~ str_trim(str_remove(x, "^\\s*\\d+\\s*:\\s*"))  
  ))
```

```
Tabela_reduzida %>%  
  count(Encerramento.Municipal...Causalidade)  
```
```

> Quantas notificações classificadas por causalidade \*ESTADUAL\*?

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
```

```
Tabela_reduzida <- Tabela %>%  
  select(Encerramento.Estadual...Causalidade)
```

```
Tabela_reduzida <- Tabela_reduzida %>%  
  separate_rows(  
    Encerramento.Estadual...Causalidade,  
    sep = "\\|" ) %>%  
  mutate(across(c(  
    Encerramento.Estadual...Causalidade),  
    ~ str_trim(str_remove(x, "^\\s*\\d+\\s*:\\s*"))  
  ))
```

```
Tabela_reduzida %>%  
  count(Encerramento.Estadual...Causalidade)  
```
```

> Quantas notificações classificadas por causalidade \*FEDERAL\*?

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
```

```
Tabela_reduzida <- Tabela %>%  
  select(Encerramento.Federal...Causalidade)
```

```
Tabela_reduzida <- Tabela_reduzida %>%  
  separate_rows(  
    Encerramento.Federal...Causalidade,  
    sep = "\\|" ) %>%  
  mutate(across(c(  
    Encerramento.Federal...Causalidade),  
    ~ str_trim(str_remove(x, "^\\s*\\d+\\s*:\\s*"))  
  ))
```

```
Tabela_reduzida %>%  
  count(Encerramento.Federal...Causalidade)  
```
```

---



> Quantas notificações classificadas por causalidade \*CIFAVI\*?

```
``` {r message=FALSE, warning=FALSE, echo=FALSE}
```

```
Tabela_reduzida <- Tabela %>%
  select(Encerramento.Cifavi...Causalidade)
```

```
Tabela_reduzida <- Tabela_reduzida %>%
  separate_rows(
    Encerramento.Cifavi...Causalidade,
    sep = "\\|" ) %>%
  mutate(across(c(
    Encerramento.Cifavi...Causalidade),
    ~ str_trim(str_remove(x, "^\\s*\\d+\\s*:\\s*"))
  ))
```

```
Tabela_reduzida %>%
  count(Encerramento.Cifavi...Causalidade)
```
```

```

```

# Questão 9

> É possível buscar por ESAVI listados de interesse especial? (coluna doenças CID 10)

```
``` {r message=FALSE, warning=FALSE, echo=FALSE}
```

```
todos_municipios <- tibble::tibble(
  Municipio = c(
    "T782","T783","T784","T886",
    "G40","G401","G402","G403","G404","G405","G406","G407","G408","G409",
    "R568","D683","R58","D69","G049","G048","G05",
    "G040","G369","R688","I829","I81","I82","I74",
    "K71","K716","K72","N17","N19","N178",
    "N25","N26","N27","N28","N29",
    "N10","N11","N12","N13","N14","N15","N16",
    "G030","G039","G373","G378","G379",
    "I409","I401","I408","I41","I42","I427","I514","I516",
    "K859","G510","G518","G519","I309",
    "D693","D69","G610","G618","G619","D694","I828",
    "M358","B342","U109","U071","M303","R651","D761","A483",
    "D696","D695","D699","Y58","Y580","Y581","Y582","Y583","Y584","Y585","Y586","Y587","Y588","Y589",
    "Y59","Y590","Y591","Y592","Y593","Y594","Y595","Y596","Y597","Y598","Y599",
    "T78"
  )
)
```

```
Tabela_reduzida <- Tabela %>%
  select(`Doenças.CID10`,`Município.da.Notificação`,`Data.da.Notificação`,
`Número.da.Notificação`) %>%
  separate_rows(`Doenças.CID10`, sep = "\\|") %>%
  mutate(
    `Doenças.CID10` = str_remove(`Doenças.CID10`,`^\\s*\\d+:\\s*`),
    `Doenças.CID10` = str_trim(`Doenças.CID10`),
    `Doenças.CID10` = str_extract(`Doenças.CID10`,`^[A-Z]\\d{2,3}(?:\\.\\.\\d+)?`)
  )
```
```



```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_filtrada <- Tabela_reduzida %>%
  filter(Doenças.CID10. %in% todos_municipios$Municipio)
Tabela_filtrada
```
```

> É possível buscar por ESAVI listados de interesse especial? (coluna diagnostico CID 10)

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(` Diagnóstico..CID.10.`,      ` Município.da.Notificação`,      ` Data.da.Notificação`,
` Número.da.Notificação`) %>%
  separate_rows(Diagnóstico..CID.10., sep = "\\") %>%
  mutate(
    # remove prefixos tipo "1:" ou "1047"
    Diagnóstico..CID.10. = str_remove(Diagnóstico..CID.10., "^\\s*\\d+?:?\\s*"),
    # tira espaços extras
    Diagnóstico..CID.10. = str_trim(Diagnóstico..CID.10.),
    # extrai só o CID no início da string
    Diagnóstico..CID.10. = str_extract(Diagnóstico..CID.10., "^[A-Z]\\d{2,3}(?:\\.\\d+)?")
  )
```
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_filtrada <- Tabela_reduzida %>%
  filter(Diagnóstico..CID.10. %in% todos_municipios$Municipio)
Tabela_filtrada
```
```

> É possível buscar por ESAVI listados de interesse especial? (Encerramento ESTADUAL diagnostico)

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Encerramento.Estadual...Diagnóstico, ` Município.da.Notificação`, ` Data.da.Notificação`,
` Número.da.Notificação`) %>%
  separate_rows(Encerramento.Estadual...Diagnóstico, sep = "\\") %>%
  mutate(
    # remove prefixos tipo "1:" ou "1047"
    Encerramento.Estadual...Diagnóstico = str_remove(Encerramento.Estadual...Diagnóstico,
    "^\\s*\\d+?:?\\s*"),
    # tira espaços extras
    Encerramento.Estadual...Diagnóstico = str_trim(Encerramento.Estadual...Diagnóstico),
    # extrai só o CID no início da string
    Encerramento.Estadual...Diagnóstico = str_extract(Encerramento.Estadual...Diagnóstico,
    "^[A-Z]\\d{2,3}(?:\\.\\d+)?")
  )
```
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_filtrada <- Tabela_reduzida %>%
  filter(Encerramento.Estadual...Diagnóstico %in% todos_municipios$Municipio)
Tabela_filtrada
```
```

> É possível buscar por ESAVI listados de interesse especial? (Encerramento MUNICIPAL diagnostico)

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Encerramento.Municipal...Diagnóstico, ` Município.da.Notificação`, ` Data.da.Notificação`,
` Número.da.Notificação`) %>%
```



```

separate_rows(Encerramento.Municipal...Diagnóstico, sep = "\\|") %>%
mutate(
  # remove prefixos tipo "1:" ou "1047"
  Encerramento.Municipal...Diagnóstico = str_remove(Encerramento.Municipal...Diagnóstico,
"^\s*\d+?:?\s*"),
  # tira espaços extras
  Encerramento.Municipal...Diagnóstico = str_trim(Encerramento.Municipal...Diagnóstico),
  # extrai só o CID no início da string
  Encerramento.Municipal...Diagnóstico = str_extract(Encerramento.Municipal...Diagnóstico,
"^[A-Z]\\d{2,3}(?:\\.\\d+)?")
)
...
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_filtrada <- Tabela_reduzida %>%
 filter(Encerramento.Municipal...Diagnóstico %in% todos_municipios$Municipio)
Tabela_filtrada
...

> É possível buscar por ESAVI listados de interesse especial? (Encerramento FEDERAL diagnostico)

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Encerramento.Federal...Diagnóstico, `Município.da.Notificação`, `Data.da.Notificação`,
`Número.da.Notificação`) %>%
  separate_rows(Encerramento.Federal...Diagnóstico, sep = "\\|") %>%
  mutate(
    # remove prefixos tipo "1:" ou "1047"
    Encerramento.Federal...Diagnóstico = str_remove(Encerramento.Federal...Diagnóstico,
"^\s*\d+?:?\s*"),
    # tira espaços extras
    Encerramento.Federal...Diagnóstico = str_trim(Encerramento.Federal...Diagnóstico),
    # extrai só o CID no início da string
    Encerramento.Federal...Diagnóstico = str_extract(Encerramento.Federal...Diagnóstico,
"^[A-Z]\\d{2,3}(?:\\.\\d+)?")
  )
...
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_filtrada <- Tabela_reduzida %>%
 filter(Encerramento.Federal...Diagnóstico %in% todos_municipios$Municipio)
Tabela_filtrada
...

> É possível buscar por ESAVI listados de interesse especial? (Encerramento CIFAVI diagnostico)

```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Encerramento.Cifavi...Diagnóstico, `Município.da.Notificação`, `Data.da.Notificação`,
`Número.da.Notificação`) %>%
  separate_rows(Encerramento.Cifavi...Diagnóstico, sep = "\\|") %>%
  mutate(
    # remove prefixos tipo "1:" ou "1047"
    Encerramento.Cifavi...Diagnóstico = str_remove(Encerramento.Cifavi...Diagnóstico, "^\s*\d+?:?\s*"),
    # tira espaços extras
    Encerramento.Cifavi...Diagnóstico = str_trim(Encerramento.Cifavi...Diagnóstico),
    # extrai só o CID no início da string

```



```
Encerramento.Cifavi...Diagnóstico = str_extract(Encerramento.Cifavi...Diagnóstico,
"^[A-Z]\\d{2,3}(?:\\.\\d+)?")
)
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_filtrada <- Tabela_reduzida %>%
 filter(Encerramento.Cifavi...Diagnóstico %in% todos_municipios$Municipio)
Tabela_filtrada
```
```

```
*****
```

```
# Questão 10
```

```
> É possível agrupar os dados por vacina, em ESAVI grave e não grave?
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
 select(Imunobiológico.vacina., Classificação.de.gravidade) %>%
 mutate(
 gravidade_split = str_split(Classificação.de.gravidade, "\\|"),
 vacina_split = str_split(Imunobiológico.vacina., "\\|"),
 max_len = pmax(lengths(gravidade_split), lengths(vacina_split)),
 gravidade_split = mapply(function(x, n) c(x, rep(NA, n - length(x))),
 gravidade_split, max_len, SIMPLIFY = FALSE),
 vacina_split = mapply(function(x, n) c(x, rep(NA, n - length(x))),
 vacina_split, max_len, SIMPLIFY = FALSE),
 combined = Map(function(g, v) tibble(gravidade = g, vacina = v),
 gravidade_split, vacina_split)
) %>%
 unnest(combined) %>%
 mutate(
 gravidade = str_trim(str_remove(gravidade, "\\s*\\d+?:\\s*")),
 vacina = str_trim(str_remove(vacina, "\\s*\\d+?:\\s*"))
)
View(Tabela_reduzida)
```
```

```
> Vacina com ESAVI grave
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 filter(gravidade== "Grave") %>%
 count(vacina) %>%
 arrange(n, vacina)
```
```

```
> Vacina com ESAVI não grave
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 filter(gravidade== "Não grave") %>%
 count(vacina)
```
```

```
*****
```

```
# Questão 11
```

```
> Sintomas separados por vacina recebida
```



```

` `` {r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
select(Imunobiológico..vacina., Reação...evento.adverso) %>%
mutate(
  gravidade_split = str_split(Reação...evento.adverso, "\\|"),
  vacina_split = str_split(Imunobiológico..vacina., "\\|"),
  max_len = pmax(lengths(gravidade_split), lengths(vacina_split)),
  gravidade_split = mapply(function(x, n) c(x, rep(NA, n - length(x))),
    gravidade_split, max_len, SIMPLIFY = FALSE),
  vacina_split = mapply(function(x, n) c(x, rep(NA, n - length(x))),
    vacina_split, max_len, SIMPLIFY = FALSE),
  combined = Map(function(g, v) tibble(gravidade = g, vacina = v),
    gravidade_split, vacina_split)
) %>%
unnest(combined) %>%
mutate(
  gravidade = str_trim(str_remove(gravidade, "^\\s*\\d+?:?\\s*")),
  vacina = str_trim(str_remove(vacina, "^\\s*\\d+?:?\\s*"))
)
View(Tabela_reduzida)
` ``
` `` {r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
  distinct(vacina, gravidade) %>%
  arrange(vacina, gravidade)
` ``

```

Questão 12

> É possível correlacionar lotes associados na notificação EX: cluster envolvendo o mesmo lote de vacina.

```

` `` {r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
  select(Lote)

Tabela_reduzida <- Tabela_reduzida %>%
  separate_rows(
    Lote,
    sep = "\\|") %>%
  mutate(across(c(
    Lote,
    ~ str_trim(str_remove(x, "^\\s*\\d+\\s*:\\s*"))
  ))

Tabela_reduzida %>%
  count(Lote) %>%
  arrange(n)
` ``

```

Questão 13

> Municípios que reportaram erro de imunização



```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida <- Tabela %>%
 select(Município.da.Notificação, Tipo.de.Evento, Data.da.Notificação, Situação.da.Notificação,
 Classificação.de.gravidade)
```

```
Tabela_reduzida <- Tabela_reduzida %>%
 separate_rows(
 Tipo.de.Evento,
 sep = "\\|" %>%
 mutate(across(c(
 Tipo.de.Evento,
 ~ str_trim(str_remove(.x, "^\\s*\\d+\\s*\\.\\s*"))
)))
```
```

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 filter(Tipo.de.Evento == "Erro de Imunização") %>%
 count(Município.da.Notificação)
```
```

Questão 14

> Tipo de evento por municípios (avaliacao, aberto e encerrado juntos)

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 mutate(ano = year(dmy(Data.da.Notificação))) %>% # extrai o ano da data
 group_by(ano, Município.da.Notificação, Tipo.de.Evento) %>%
 summarise(qtd_notificacoes = n(), .groups = "drop") %>%
 arrange(Tipo.de.Evento)
```
```

> Quantas notificações por Municípios anualmente em avaliação

```
```{r message=FALSE, warning=FALSE, echo=FALSE}

Tabela_reduzida %>%
 filter(Situação.da.Notificação == "Em Avaliação") %>%
 mutate(ano = year(dmy(Data.da.Notificação))) %>% # extrai o ano da data
 group_by(ano, Município.da.Notificação, Tipo.de.Evento) %>%
 summarise(qtd_notificacoes = n(), .groups = "drop") %>%
 arrange(Tipo.de.Evento)
```
```

> Quantas notificações por Municípios anualmente encerrado

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 filter(Situação.da.Notificação == "Encerrado") %>%
 mutate(ano = year(dmy(Data.da.Notificação))) %>% # extrai o ano da data
 group_by(ano, Município.da.Notificação, Tipo.de.Evento) %>%
 summarise(qtd_notificacoes = n(), .groups = "drop") %>%
 arrange(Tipo.de.Evento)
```
```



> Quantas notificações por Municípios anualmente em aberto

```
```{r message=FALSE, warning=FALSE, echo=FALSE}
Tabela_reduzida %>%
 filter(Situação.da.Notificação == "Aberto") %>%
 mutate(ano = year(dmy(Data.da.Notificação))) %>% # extrai o ano da data
 group_by(ano, Município.da.Notificação, Tipo.de.Evento) %>%
 summarise(qtd_notificacoes = n(), .groups = "drop") %>%
 arrange(Tipo.de.Evento)
```
```

Questão 15

> Municípios que nunca registraram erro de imunização

```
```{r message=FALSE, warning=FALSE, echo=FALSE}

todos os municípios que aparecem na base
todos <- Tabela_reduzida %>%
 distinct(Município.da.Notificação)

municípios que reportaram erro de imunização
com_erro <- Tabela_reduzida %>%
 filter(Tipo.de.Evento == "Erro de Imunização") %>%
 distinct(Município.da.Notificação)

municípios que não reportaram erro
sem_erro <- todos %>%
 anti_join(com_erro, by = "Município.da.Notificação")

View(sem_erro)
```
```
