

Orientações para o uso da *Data Quality Indicators (DQI) tool* para avaliação de dados de anomalias congênicas no Sinasc

A ferramenta proposta por Groisman e colaboradores (<https://doi.org/10.1002/bdr2.1474>) avalia uma lista de indicadores de qualidade de dados para programas de vigilância de anomalias congênicas. Esses indicadores abordam aspectos de detecção, descrição, codificação e classificação, e estão sistematizados em uma planilha do Excel (versão 3; 2022), disponível gratuitamente em: <http://www.icbdsr.org/data-quality-indicators-tool/>. Para o uso da ferramenta na realidade brasileira a partir de dados do Sinasc, são necessárias algumas adaptações, o que envolve a exclusão de cinco indicadores e a adaptação de outros dois (Anexo I).

A aplicação da ferramenta consiste em inserir os dados em quatro tabelas distintas da aba “*Input Dashboard*” da planilha. A Tabela 1 recebe informações sobre o número de nascidos vivos, distribuídos por faixas etárias de idade materna. A Tabela 2 recebe dados de contagem total e estratificada de anomalias congênicas selecionadas (ou grupos de anomalias), segundo a classificação clínica (anomalia congênita isolada, múltiplas anomalias congênicas ou síndromes), sendo utilizada para o cálculo dos indicadores de detecção e classificação. A Tabela 3 recebe dados de contagem de nascidos vivos com síndrome de Down, também organizados por faixas etárias de idade materna, que, juntamente com os dados da Tabela 1, permitem calcular um indicador específico de detecção: a razão entre casos observados e esperados de síndrome de Down. Por fim, a Tabela 4 recebe dados de contagens de anomalias congênicas selecionadas com descrição detalhada, servindo para o cálculo de indicadores de descrição e codificação.

Como mencionado acima, o preenchimento da Tabela 2 da ferramenta requer a classificação clínica de cada caso. Contudo, considerando a natureza da vigilância passiva de anomalias congênicas no Sinasc, faz-se necessária uma abordagem adaptada para a classificação adequada dos casos em anomalia isolada, múltiplas anomalias ou síndrome. Para isso, é necessário aplicar o algoritmo de classificação clínica (Anexo II).

O código completo em linguagem R para a extração de contagens a partir de um conjunto de dados do Sinasc (por exemplo, os dados do Distrito Federal de 2023), utilizado para o preenchimento das tabelas do “*Input Dashboard*” da ferramenta, está apresentado no Anexo III. O algoritmo de classificação clínica está incluído nesse código; portanto, para utilizar a ferramenta, é necessário apenas dispor da base de dados que se deseja avaliar e executar o script no R. A partir disso, serão gerados quatro arquivos em formato Excel, correspondentes a cada uma das tabelas do “*Input Dashboard*”. Em seguida, basta inserir os dados gerados nas respectivas tabelas da planilha da ferramenta. Os indicadores serão calculados automaticamente e poderão ser visualizados na aba “*DQI Report*”. A Tabela 1 do “*DQI Report*” apresenta a síntese dos indicadores, enquanto a Tabela 2 exibe os resultados detalhados de cada um deles. Em complemento à adaptação da ferramenta de indicadores de qualidade de dados acima descritos, a Unidade Técnica de Vigilância de Anomalias Congênitas (UT-VAC) desenvolveu outros indicadores, considerando o conjunto de variáveis disponíveis no Sinasc. Esses indicadores foram inspirados nas experiências internacionais de avaliação da qualidade dos dados em programas de vigilância de anomalias congênitas, bem como no Programa de Qualificação das Ações de Vigilância em Saúde (PQA-VS) do Ministério da Saúde. Os indicadores propostos contemplam aspectos de completude, consistência, oportunidade e especificidade. O anexo IV apresenta o código em R para extrair os indicadores propostos específicos para as variáveis disponíveis no Sinasc.

A UT-VAC se coloca à disposição, caso haja necessidade de informações adicionais, por meio do e-mail: anomaliascongenitas@saude.gov.br e/ou dos telefones (61) 3315-7704/7716.

ANEXO I

Adaptações da ferramenta para avaliação de indicadores de qualidade de dados de vigilância de anomalias congênicas

Quadro I – Indicadores do *Data Quality Indicators (DQI)* desenvolvido por Groisman e colaboradores (<https://doi.org/10.1002/bdr2.1474>) que foram adaptados.

Categoria	Definição do indicador	Parâmetro para entrada de dados	Adaptação do parâmetro	Observação	Conduta
Especificação	Proporção (%) de casos com cardiopatias congênicas específicas	Anomalias cardíacas congênicas (Q20.x-Q25.x,Q26.2)	Anomalias cardíacas congênicas (Q20.x-Q26.x)	O Sinasc tem um conjunto maior de códigos habilitados sob Q26	Adaptado
Detecção	Prevalência de defeitos transversais > defeitos pré-axiais > prevalência de defeitos pós-axiais	Deficiências transversais de membros (Q71.1,Q71.2,Q71.3,Q71.30,Q71.80,Q72.1,Q72.2,Q72.3,Q72.30,Q72.4,Q72.80,Q73.1)		Especificação não aplicável	Excluído
		Deficiências pré-axiais de membros (Q71.31,Q71.4,Q72.31,Q72.5)		Especificação não aplicável	
		Deficiências pós-axiais de membros (Q71.5,Q72.6)		Aplicável, contudo, o indicador relaciona esse parâmetro aos outros dois não aplicáveis	
Detecção	Prevalência de polidactilia pós-axial > prevalência de polidactilia pré-axial	Polidactilia pré-axial (Q69.00, Q69.1, Q69.20)		Especificação não aplicável	Excluído
		Polidactilia pós-axial (Q69.02, Q69.22)		Especificação não aplicável	

Descrição	Proporção (%) de casos de espinha bífida com especificação da cobertura cutânea (aberta ou fechada)	Número de casos de espinha bífida com cobertura cutânea especificada – aberta ou fechada (Q05.x1, Q05.x2)		Especificação não aplicável	Excluído
Descrição	Proporção (%) de casos de microtia com grau especificado	Número de casos de microtia com grau especificado (Q16.0, Q17.22, Q17.23)		Especificação não aplicável	Excluído
Descrição	Proporção (%) de casos com tipo (pré-axial, meso-axial e pós-axial) e localização (mãos, pés, lateralidade) especificados	Polidactilia (Q69)		Aplicável, contudo, o indicador relaciona esse parâmetro a outros não aplicável	Excluído
		Número de casos com tipo (pré-axial, meso-axial e pós-axial) e localização (mãos, pés, lateralidade) especificados (Q69.0x–Q69.2x)		Especificação não aplicável	
Descrição	Proporção (%) de nascidos vivos com Síndrome de Down e anomalia cardíaca congênita associada deve ser 40% ou maior	Casos com anomalias cardíacas congênitas entre nascidos vivos com síndrome de Down (Q20.x-Q25.x,Q26.2)	Casos com anomalias cardíacas congênitas entre nascidos vivos com síndrome de Down (Q20.x-Q26.x)	O Sinasc tem um conjunto maior de códigos habilitados sob Q26	Adaptado

ANEXO II

Algoritmo para classificação de casos com anomalias congênitas

Quadro 1 – Descrição das regras para classificação de casos com anomalias congênitas a partir de códigos CID-10 habilitados no Sinasc (adaptado de *Updated EUROCAT multiple congenital anomaly algorithm*, v. 17 November 2022 - <https://doi.org/10.1002/bdr2.2314>)

Passo	Regra	Classificação
1	Excluir códigos de anomalias menores (tabela 1) sem anomalia maior associada	-
2	Classificar todos os casos com código de síndromes (tabela 2) como síndrome	Síndrome
3	Classificar casos com apenas códigos do mesmo órgão/sistema (tabela 3) como anomalia isolada	Isolada
4	Classificar casos com apenas um código como anomalia isolada	Isolada
5	Classificar combinações de códigos de sequências ou combinações conhecidas (tabela 4) como anomalia isolada (anomalia primária)	Isolada
6	Classificar casos com mais de um código como anomalias múltiplas	Múltipla

Quadro 2 – Anomalias menores passíveis de notificação no SINASC (adaptado do 3.2 *Minor anomalies and other conditions for exclusion, EUROCAT Guide 1.5* - https://eu-rd-platform.jrc.ec.europa.eu/system/files/public/eurocat/Guide_1.5_Chapter_3.2.pdf)

Código	Título
D18.0	Hemangioma de qualquer localização
Q10.1	Ectrópio congênito
Q13.5	Esclerótica azul
Q17.0	Pavilhão supranumerário
Q17.1	Macrotia
Q17.2	Microtia
Q17.3	Outras deformidades da orelha
Q17.4	Anomalia de posição da orelha
Q17.5	Orelhas proeminentes
Q17.9	Malformação congênita não especificada da orelha
Q18.0	Seio, fístula e cisto de origem branquial
Q18.1	Seio, fístula e cisto pré-auricular
Q18.2	Outras malformações da fenda branquial
Q18.4	Macrostomia
Q18.5	Microstomia
Q18.6	Macroqueilia
Q18.7	Microqueilia
Q18.9	Malformação congênita não especificada da face e do pescoço
Q26.1	Persistência da veia cava superior esquerda
Q27.0	Ausência congênita e hipoplasia da artéria umbilical
Q31.5	Laringomalácia congênita
Q32.0	Traqueomalácia congênita
Q32.2	Broncomalácia congênita
Q33.1	Lobo pulmonar supranumerário
Q35.7	Fenda da úvula
Q38.1	Anquiloglossia
Q38.2	Macroglossia
Q40.0	Estenose hipertrófica congênita do piloro
Q40.1	Hérnia congênita de hiato
Q43.0	Divertículo de Meckel
Q44.4	Cisto do colédoco
Q50.1	Cisto ovariano de desenvolvimento
Q50.2	Torsão congênita do ovário
Q50.5	Cisto embrionário do ligamento largo
Q52.3	Imperfuração do hímen
Q52.5	Fusão dos lábios vulvares
Q52.7	Outras malformações congênicas da vulva
Q53	Testículos não descidos
Q54.4	Corda venérea congênita
Q61.0	Cisto congênito único do rim
Q62.7	Refluxo vésico-uretero-renal congênito

Q63.3	Rim hiperplásico e gigante
Q65.3	Subluxação congênita unilateral do quadril
Q65.4	Subluxação congênita bilateral do quadril
Q65.5	Subluxação congênita não especificada do quadril
Q65.6	Quadril instável
Q65.8	Outras deformidades congênitas do quadril
Q65.9	Deformidade congênita não especificada do quadril
Q67.0	Assimetria facial
Q67.1	Deformidade facial por compressão
Q67.2	Dolicocefalia
Q67.3	Plagiocefalia
Q67.4	Outras deformidades congênitas do crânio, da face e da mandíbula
Q67.5	Deformidades congênitas da coluna vertebral
Q67.6	Tórax escavado
Q67.7	Tórax carinado
Q67.8	Outras deformidades congênitas do tórax
Q68.0	Deformidade congênita do músculo esternocleidomastoideu
Q68.3	Encurvamento congênito do fêmur
Q68.4	Encurvamento congênito da tíbia e da perônio [fíbula]
Q68.5	Encurvamento congênito de ossos longos não especificados do membro inferior
Q75.2	Hipertelorismo
Q75.3	Macrocefalia
Q76.0	Espinha bífida oculta
Q76.5	Costela cervical
Q82.5	Nevo não neoplásico congênito
Q83.3	Mamilo acessório
Q84.5	Hipertrofia e alargamento das unhas
Q84.6	Outras malformações congênitas das unhas
Q89.9	Malformações congênitas não especificadas
Q95	Rearranjos cromossômicos balanceados

Quadro 3 – Síndromes (adaptado do *EUROCAT Syndrome Guide, definition and coding of Syndromes*, v. July 2017 - <https://eu-rd-platform.jrc.ec.europa.eu/sites/default/files/EUROCAT%20Syndrome%20Guide%20Revision%20Final%20version%20September%202017.pdf>)

Código	Título
Q761	Síndrome de Klippel-Feil
Q770	Acondrogenesia
Q771	Nanismo tanatofórico
Q772	Síndrome das costelas curtas
Q773	Condrodisplasia puntacta
Q774	Acondroplasia
Q775	Displasia diastrófica
Q776	Displasia condroectodérmica
Q777	Displasia espondiloepifisária
Q778	Outras osteocondrodisplasias com anomalias do crescimento dos ossos longos e da coluna vertebral
Q779	Osteocondrodisplasia não especificada com anomalias do crescimento dos ossos longos e da coluna vertebral
Q780	Osteogênese imperfeita
Q781	Displasia poliostótica fibrosa
Q782	Osteopetrose
Q783	Displasia diafisária progressiva
Q784	Encondromatose
Q785	Displasia metafisária
Q786	Exostoses congênicas múltiplas
Q788	Outras osteocondrodisplasias especificadas
Q789	Osteocondrodisplasia não especificada
Q796	Síndrome de Ehlers-Danlos
Q800	Ictiose vulgar
Q801	Ictiose ligada ao cromossomo X
Q802	Ictiose lamelar
Q803	Eritrodermia ictiosiforme bulhosa congênita
Q804	Feto arlequin
Q808	Outras ictioses congênicas
Q809	Ictiose congênita não especificada
Q810	Epidermólise bolhosa simples
Q811	Epidermólise bolhosa letal
Q812	Epidermólise bolhosa distrófica
Q818	Outras epidermólises bolhosas
Q819	Epidermólise bolhosa não especificada
Q820	Linfedema hereditário
Q821	Xeroderma pigmentoso
Q822	Mastocitose
Q823	Incontinentia pigmenti
Q824	Displasia ectodérmica (anidrótica)

Q850	Neurofibromatose (não maligna)
Q851	Esclerose tuberosa
Q858	Outras facomatoses não classificadas em outra parte
Q859	Facomatose não especificada
Q860	Síndrome fetal alcoólico (dismórfico)
Q861	Síndrome fetal devida à hidantoína
Q862	Dismorfismo devido ao Warfarin
Q868	Outras síndromes com malformações congênicas devidas a causas exógenas conhecidas
Q870	Síndromes com malformações congênicas afetando predominantemente o aspecto da face
Q871	Síndromes com malformações congênicas associadas predominantemente com nanismo
Q872	Síndromes com malformações congênicas afetando predominantemente os membros
Q873	Síndromes com malformações congênicas com hipercrecimento precoce
Q874	Síndrome de Marfan
Q875	Outras síndromes com malformações congênicas com outras alterações do esqueleto
Q878	Outras síndromes com malformações congênicas especificadas, não classificadas em outra parte
Q90 – Q94 e Q96 – Q99	Malformações congênicas, deformidades e anomalias cromossômicas

Quadro 4 – Conjunto de códigos pertencentes a um mesmo órgão ou sistema (adaptado do 3.4 *Multiple congenital anomaly algorithm*, EUROCAT Guide 1.5 - https://eu-rd-platform.jrc.ec.europa.eu/system/files/public/eurocat/Guide_1.5_Chapter_3.4.pdf) a ser classificado como anomalia isolada

Órgão/Sistema	Códigos
Defeitos de tubo neural	Q00 Q01 Q05
Cardiopatias	Q20 Q21 Q22 Q23 Q24 Q25 Q26
Anomalias congênicas dos rins e do trato urinário	Q60 Q61 Q62 Q63 Q64 Q794
casos com códigos apenas para anomalias do sistema nervoso, não defeitos de tubo neural	Q02 Q03 Q04 Q06 Q07
Anomalias oculares	Q10 Q11 Q12 Q13 Q14 Q15
Anomalias de orelha/ouvido	Q16 Q17
Anomalias respiratórias	Q300 Q32 Q33 Q34
Fissuras orofaciais	Q35 Q36 Q37
Atresia do intestino delgado	Q411 Q412 Q413 Q414

	Q415 Q416 Q417 Q418
Anomalias genitais	Q50 Q51 Q52 Q54 Q55 Q56
Defeitos de membros	Q65 Q66 Q67 Q68 Q69 Q70 Q71 Q72 Q73 Q74

Quadro 5 – Sequências e combinações conhecidas (adaptado do 3.4 *Multiple congenital anomaly algorithm*, EUROCAT Guide 1.5 - https://eu-rd-platform.jrc.ec.europa.eu/system/files/public/eurocat/Guide_1.5_Chapter_3.4.pdf) a ser classificadas como anomalias isoladas

Sequência	Código principal	Códigos associados
Síndrome do abdome em ameixa seca (prune belly syndrome)	Q794	Q60 Q61 Q62 Q63 Q64
Espinha bífida - sistema nervoso central anomalias - talipes - luxação do quadril - hidronefrose congênita	Q05	Q03 Q04 Q06 Q07 Q65 Q66 Q620
Anencefalia – anomalia adrenal	Q00	Q891
Agenesia/displasia renal bilateral - pulmão hipoplasia - talipes	Q601 Q606	Q336 Q66
Gastrosquise/omfalocele - anomalias de fixação intestinal - atresia do intestino delgado	Q793 Q792	Q433 Q41
Atresia e estenose anorretal - fístula retovaginal	Q42	Q522
Hérnia diafragmática - hipoplasia pulmonar - Anomalias de fixação intestinal	Q790	Q336 Q433
NTD - Malformação de Arnold Chiari - cordão umbilical preso	Q01 Q05	Q070 Q068
Anomalia da Polônia - simbraquidactilia - aplasia do músculo peitoral	Q798	Q70 Q748
Sequência de Pierre Robin	Q870	Q35
Holoprosencefalia/arinencefalia - fendas orofaciais - anomalias cerebrais	Q042 Q041	Q35 Q36 Q37 Q02 Q03 Q04
Pâncreas anular - atresia/estenose do duodeno	Q451	Q410

ANEXO III

Código para extração de contagens para implementação na ferramenta de Groisman e colaboradores

```
# dqi
library(readr)
library(read.dbc)
library(dplyr)
library(stringr)
library(purrr)
library(tibble)
library(tidyr)
library(writexl)
library(readr)
library(foreign)

# importar base de dados para avaliacao
sinasc

# table 1
sinasc$IDADEMAE <- as.numeric(sinasc$IDADEMAE)

sinasc$faixa_etaria <- cut(
  sinasc$IDADEMAE,
  breaks = c(-Inf, 19, 24, 29, 34, 39, 44, 45, Inf),
  labels = c("<20", "20-24", "25-29", "30-34", "35-39", "40-44", ">45", "Age unknown"),
  right = TRUE
)

# Substituir "Age unknown" apenas para valores NA ou inválidos
sinasc$faixa_etaria[is.na(sinasc$IDADEMAE)] <- "Age unknown"

# Ver frequência
nvtable1 <- sinasc %>%
  group_by(faixa_etaria) %>%
  summarise(
    total_nascidos = n(),
    .groups = "drop"
  )

write_xlsx(nvtable1, "C:/.../nvtable1.xlsx")

#####
# processamento de codigos CID

# Função para processar cada string de códigos
# definir AC menores a ser desconsideradas
```

```

menores <- c(
  "D180", "Q101", "Q135",
  "Q170", "Q171", "Q172", "Q173", "Q174", "Q175", "Q179",
  "Q180", "Q181", "Q182", "Q184", "Q185", "Q186", "Q187", "Q189",
  "Q261", "Q270",
  "Q315", "Q320", "Q322", "Q331", "Q357",
  "Q381", "Q382", "Q400", "Q401", "Q430",
  "Q444", "Q501", "Q502", "Q505",
  "Q523", "Q525", "Q527",
  "Q530", "Q531", "Q532", "Q539",
  "Q544", "Q610", "Q627", "Q633",
  "Q653", "Q654", "Q655", "Q656", "Q658", "Q659",
  "Q670", "Q671", "Q672", "Q673", "Q674", "Q675", "Q676", "Q677", "Q678",
  "Q680", "Q683", "Q684", "Q685",
  "Q752", "Q753", "Q760", "Q765", "Q825", "Q833", "Q845", "Q846", "Q899", "Q95"
)

```

```

# Função para processar cada string de códigos
processa_codigos <- function(codigos, menores) {
  if (is.na(codigos) || codigos == "") return(NA)

```

```

  # Extrai todos os códigos (letras Q ou D seguidas de 2 ou 3 dígitos)
  codigos_divididos <- str_extract_all(codigos, "[QD][0-9]{2,3}")[[1]]

```

```

  # Identifica códigos menores e não menores
  e_menor <- codigos_divididos %in% menores
  qtd_menores <- sum(e_menor)
  qtd_nao_menores <- sum(!e_menor)

```

```

  # Nova lógica: se houver ao menos 1 não-menor e 2 ou mais menores → manter todos
  if (qtd_nao_menores >= 1 && qtd_menores >= 2) {
    return(paste(codigos_divididos, collapse = ""))
  }

```

```

  # Caso contrário, remove os menores
  codigos_filtrados <- codigos_divididos[!e_menor]

```

```

  # Se não sobrar nenhum código, retorna NA
  if (length(codigos_filtrados) == 0) return(NA)

```

```

  # Retorna os códigos restantes
  return(paste(codigos_filtrados, collapse = ""))
}

```

```

# Aplica ao dataframe
sinasc <- sinasc %>%

```

```
mutate(CODANOMAL1 = supply(CODANOMAL, processa_codigos, menores =  
menores))
```

```
# definir as síndromes
```

```
sindromes <- c(  
  "Q761",  
  "Q770", "Q771", "Q772", "Q773", "Q774", "Q775", "Q776", "Q777", "Q778", "Q779",  
  "Q780", "Q781", "Q782", "Q783", "Q784", "Q785", "Q786", "Q788", "Q789",  
  "Q796",  
  "Q800", "Q801", "Q802", "Q803", "Q804", "Q808", "Q809",  
  "Q810", "Q811", "Q812", "Q818", "Q819",  
  "Q820", "Q821", "Q822", "Q823", "Q824",  
  "Q850", "Q851", "Q858", "Q859",  
  "Q860", "Q861", "Q862", "Q868",  
  "Q870", "Q871", "Q872", "Q873", "Q874", "Q875", "Q878",  
  "Q90", "Q91", "Q92", "Q93", "Q94",  
  "Q96", "Q97", "Q98", "Q99"  
)
```

```
# Criar padrão regex com os códigos das síndromes
```

```
padrao_sindromes <- paste0(" ", paste(sindromes, collapse = "|"), ")")
```

```
# regra 2 atribui classificacao de síndrome
```

```
sinasc <- sinasc %>%
```

```
  mutate(classificacao = case_when(  
    is.na(CODANOMAL1) ~ NA_character_,  
    str_detect(CODANOMAL1, padrao_sindromes) ~ "sind",  
    TRUE ~ NA_character_  
  ))
```

```
# regra 3 classifica anomalias de mesmo órgão como isolada
```

```
# Lista de códigos por grupo
```

```
grupos <- list(  
  defeito_tubo_neural = c(  
    "Q00", "Q01", "Q05",  
    paste0("Q00", 0:9),  
    paste0("Q01", 0:9),  
    paste0("Q05", 0:9)  
  ),  
  cardiopatias = c(  
    "Q20", "Q21", "Q22", "Q23", "Q24", "Q25", "Q26",  
    paste0("Q20", 0:9),  
    paste0("Q21", 0:9),  
    paste0("Q22", 0:9),  
    paste0("Q23", 0:9),  
    paste0("Q24", 0:9),  
    paste0("Q25", 0:9),  
  )  
)
```

```
paste0("Q26", 0:9)
),
rins_urinario = c(
  "Q60", "Q61", "Q62", "Q63", "Q64", "Q794",
  paste0("Q60", 0:9),
  paste0("Q61", 0:9),
  paste0("Q62", 0:9),
  paste0("Q63", 0:9),
  paste0("Q64", 0:9)
),
sn_outros = c(
  "Q02", "Q02X", "Q03", "Q04", "Q06", "Q07",
  paste0("Q02", 0:9),
  paste0("Q03", 0:9),
  paste0("Q04", 0:9),
  paste0("Q06", 0:9),
  paste0("Q07", 0:9)
),
oculares = c(
  "Q10", "Q11", "Q12", "Q13", "Q14", "Q15",
  paste0("Q10", 0:9),
  paste0("Q11", 0:9),
  paste0("Q12", 0:9),
  paste0("Q13", 0:9),
  paste0("Q14", 0:9),
  paste0("Q15", 0:9)
),
ouvido = c(
  "Q16", "Q17",
  paste0("Q16", 0:9),
  paste0("Q17", 0:9)
),
respiratorias = c(
  "Q300",
  "Q32", "Q33", "Q34",
  paste0("Q32", 0:9),
  paste0("Q33", 0:9),
  paste0("Q34", 0:9)
),
fissuras = c(
  "Q35", "Q36", "Q37",
  paste0("Q35", 0:9),
  paste0("Q36", 0:9),
  paste0("Q37", 0:9)
),
atresia_intestino = c(
  "Q411", "Q412", "Q413", "Q414", "Q415", "Q416", "Q417", "Q418"
```



```

),
genitais = c(
  "Q50", "Q51", "Q52", "Q54", "Q55", "Q56",
  paste0("Q50", 0:9),
  paste0("Q51", 0:9),
  paste0("Q52", 0:9),
  paste0("Q54", 0:9),
  paste0("Q55", 0:9),
  paste0("Q56", 0:9)
),
membros = c(
  "Q65", "Q66", "Q67", "Q68", "Q69", "Q70", "Q71", "Q72", "Q73", "Q74",
  paste0("Q65", 0:9),
  paste0("Q66", 0:9),
  paste0("Q67", 0:9),
  paste0("Q68", 0:9),
  paste0("Q69", 0:9),
  paste0("Q70", 0:9),
  paste0("Q71", 0:9),
  paste0("Q72", 0:9),
  paste0("Q73", 0:9),
  paste0("Q74", 0:9)
)
)
)

```

```

# classifica codigos com regra 3 e 4 (apenas 1 codigo é isolada)
for (i in seq_along(sinasc$classificacao)){
  if (is.na(sinasc$classificacao[i])){
    cids <- str_extract_all(sinasc$CODANOMAL1[i], "[QD][0-9]{2,3}")
    for (j in seq_along(grupos)){
      if(all(cids[[1]] %in% grupos[[j]])){
        sinasc$classificacao[i] <- "isol"
      }
    }
  }
}
}

```

```

# regra de sequencias
for (i in seq_along(sinasc$classificacao)){
  if (is.na(sinasc$classificacao[i])){
    cids <- str_extract_all(sinasc$CODANOMAL1[i], "[QD][0-9]{2,3}")
    if(any(cids[[1]] %in% c("Q794")) & any(cids[[1]] %in% c(
      "Q60", paste0("Q60", 0:9),
      "Q61", paste0("Q61", 0:9),
      "Q62", paste0("Q62", 0:9),
      "Q63", paste0("Q63", 0:9),
      "Q64", paste0("Q64", 0:9)))){

```

```

sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q605")) & any(cids[[1]] %in% c(
  "Q03", paste0("Q03", 0:9),
  "Q04", paste0("Q04", 0:9),
  "Q06", paste0("Q06", 0:9),
  "Q07", paste0("Q07", 0:9),
  "Q65", paste0("Q65", 0:9),
  "Q66", paste0("Q66", 0:9),
  "Q620"))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q00", paste0("Q00", 0:9))) & any(cids[[1]] %in% c("Q891"))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q601", "Q606")) & any(cids[[1]] %in% c("Q336", "Q66",
paste0("Q66", 0:9)))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q793", "Q792")) & any(cids[[1]] %in% c("Q433", "Q41",
paste0("Q41", 0:9)))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q42", paste0("Q42", 0:9))) & any(cids[[1]] %in% c("Q522"))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q790")) & any(cids[[1]] %in% c("Q336", "Q433"))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q01", paste0("Q01", 0:9), "Q05", paste0("Q05", 0:9))) &
any(cids[[1]] %in% c("Q070", "Q068"))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q870")) & any(cids[[1]] %in% c("Q35", paste0("Q35", 0:9)))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q042", "Q041")) & any(cids[[1]] %in% c(
  "Q35", paste0("Q35", 0:9),
  "Q36", paste0("Q36", 0:9),
  "Q37", paste0("Q37", 0:9),
  "Q02", paste0("Q02", 0:9),
  "Q03", paste0("Q03", 0:9),
  "Q04", paste0("Q04", 0:9)
))){
  sinasc$classificacao[i] <- "isol"
}
if(any(cids[[1]] %in% c("Q451")) & any(cids[[1]] %in% c("Q410"))){

```

```

    sinasc$classificacao[i] <- "isol"
  }
}
}

```

regra 6 atribui classificacao de multiplas

```

sinasc <- sinasc %>%
  mutate(classificacao = case_when(
    !is.na(classificacao) ~ classificacao,
    nchar(str_remove_all(CODANOMAL1, "[^:alnum:]")) >= 5 ~ "mult",
    nchar(str_remove_all(CODANOMAL1, "[^:alnum:]")) < 5 ~ "isol",
    TRUE ~ classificacao
  ))

```

Função para contar casos por padrão

```

contar_anomalias_sinasc <- function(sinasc, padrao, nome_classificacao) {
  sinasc %>%
    group_by(classificacao) %>%
    summarise(
      n_casos = sum(str_detect(CODANOMAL1, padrao), na.rm = TRUE),
      .groups = "drop"
    ) %>%
    bind_rows(
      summarise(sinasc,
        classificacao = "Total",
        n_casos = sum(str_detect(CODANOMAL1, padrao), na.rm = TRUE))
    ) %>%
    arrange(desc(n_casos)) %>%
    mutate(tipo_anomalia = nome_classificacao, .before = 1)
}

```

Tabela com padrões e nomes descritivos

```

padroes <- tibble::tibble(
  padrao = c(
    "Q20|Q21|Q22|Q23|Q24|Q25|Q26",
    "Q200|Q201|Q203|Q204|Q213|Q220|Q224|Q230|Q234|Q251|Q252|Q262",
    "Q00|Q01|Q05",
    "Q00",
    "Q05",
    "Q01",
    "Q35|Q36|Q37",
    "Q36",
    "Q37",
    "Q35",
    "Q540|Q541|Q542|Q543|Q548|Q549",
    "Q540",

```

```

"Q541",
"Q542|Q543",
"Q71|Q72|Q73",
"Q71",
"Q72",
"QQQ",
"QQQ",
"Q715|Q726",
"Q66",
"Q660",
"Q664",
"Q16|Q172",
"Q69",
"QQQ",
"QQQ",
"Q792|Q793|Q795",
"Q792",
"Q793"
),
nome = c(
  "Congenital heart defects (Q20.x-Q26.x)",
  "Selected critical congenital heart defects (3)",
  "Neural tube defects (Q00, Q01, Q05)",
  "Anencephaly (Q00)",
  "Spina bifida (Q05)",
  "Encephalocele (Q01)",
  "Oral clefts (Q35-Q37)",
  "Cleft lip only (Q36)",
  "Cleft lip and palate (Q37)",
  "Cleft palate only (Q35)",
  "Hypospadias (Q54.0-Q54.3, Q54.8, Q54.9)",
  "First degree hypospadias (Q54.0)",
  "Second degree hypospadias (Q54.1)",
  "Third degree hypospadias (Q54.2-Q54.3)",
  "Limb deficiencies (Q71-Q73)",
  "Upper limb deficiencies (Q71)",
  "Lower limb deficiencies (Q72)",
  "Transverse limb deficiencies",
  "Preaxial limb deficiencies",
  "Postaxial limb deficiencies",
  "Talipes (Q66)",
  "Equinovarus (Q66.0)",
  "Calcaneovalgus (Q66.4)",
  "Microtia/anotia (Q16; Q17.2x)",
  "Polydactyly (Q69)",
  "Preaxial (Q69.00,Q69.1,Q69.20)",
  "Postaxial (Q69.02,Q69.22)",

```

```

    "Ventral Abdominal wall defects (Q79.2, Q79.3, Q79.5)",
    "Omphalocele (Q79.2)",
    "Gastroschisis (Q79.3)"
  )
)

# Aplicar a função a todos os padrões e unir os resultados
nvtable2 <- purrr::map2_dfr(
  .x = padroes$padrao,
  .y = padroes$nome,
  .f = ~ contar_anomalias_sinasc(sinasc, .x, .y)
)

nvtable2 <- nvtable2 %>%
  filter(!is.na(classificacao)) %>%
  pivot_wider(
    names_from = classificacao,
    values_from = n_casos,
    values_fill = 0 # preenche valores ausentes com 0
  )

nvtable2 <- as.data.frame(nvtable2)

# somar manualmente fissura labial e palatina (conta tbm Q35 e Q36)
# Define os índices de interesse
idx <- unique(c(
  intersect(which(str_detect(sinasc$CODANOMAL1, "Q35")),
    which(str_detect(sinasc$CODANOMAL1, "Q36"))),
  which(str_detect(sinasc$CODANOMAL1, "Q37"))
))

# Cria a tabela de frequência com os níveis desejados
tab <- table(factor(sinasc$classificacao[idx], levels = c("isol", "mult", "sind")))

# Calcula o total de casos (soma das categorias)
total <- sum(tab)

# Cria vetor com os valores na ordem correta
valores <- c(total, as.numeric(tab))

# Atribui os valores às colunas correspondentes
nvtable2[9, c("Total", "isol", "mult", "sind")] <- valores

# Gastroschisis (Q79.3) Estratificados por classificacao e faixa etária
estratificado <- sinasc %>%

```

```

mutate(tipo_anomalia = if_else(IDADEMAE < 20, "q793_<20 anos", "q793_≥20
anos")) %>%
group_by(classificacao, tipo_anomalia) %>%
summarise(
  n_Q793 = sum(str_detect(CODANOMAL1, "Q793"), na.rm = TRUE),
  .groups = "drop"
)

```

Gastroschisis (Q79.3) Totais por faixa etária (sem estratificação)

```

totais <- sinasc %>%
mutate(tipo_anomalia = if_else(IDADEMAE < 20, "q793_<20 anos", "q793_≥20
anos")) %>%
group_by(tipo_anomalia) %>%
summarise(
  n_Q793 = sum(str_detect(CODANOMAL1, "Q793"), na.rm = TRUE),
  .groups = "drop"
) %>%
mutate(classificacao = "Total") %>%
select(classificacao, everything())

```

Gastroschisis (Q79.3) Combina os dois

```

gastrosquise_estrat2 <- bind_rows(estratificado, totais) %>%
arrange(tipo_anomalia, desc(n_Q793))

```

```

gastrosquise_estrat2 <- gastrosquise_estrat2 %>%
filter(!is.na(classificacao)) %>%
filter(!is.na(tipo_anomalia)) %>%
pivot_wider(
  names_from = classificacao,
  values_from = n_Q793,
  values_fill = 0
)

```

```

nvtable2 <- rbind(nvtable2, gastrosquise_estrat2)

```

Down syndrome (Q90) Totais por faixa etária (sem estratificação)

```

down_estrat2 <- sinasc %>%
mutate(tipo_anomalia = if_else(IDADEMAE < 35, "q90_<35 anos", "q90_≥35 anos"))
%>%
group_by(tipo_anomalia) %>%
summarise(
  n_Q90 = sum(str_detect(CODANOMAL1, "Q90"), na.rm = TRUE),
  .groups = "drop"
) %>%
mutate(classificacao = "Total") %>%
select(classificacao, everything())

```

```

down_estrat2 <- down_estrat2 %>%
  filter(!is.na(tipo_anomalia)) %>%
  pivot_wider(
    names_from = classificacao,
    values_from = n_Q90,
    values_fill = 0
  )

down_estrat2 <- down_estrat2 %>%
  bind_rows(
    down_estrat2 %>%
      summarise(
        tipo_anomalia = "Down_syndrome_Q90",
        across(where(is.numeric), sum, na.rm = TRUE)
      )
  ) %>%
  # Reordena as linhas conforme a ordem desejada
  mutate(
    tipo_anomalia = factor(tipo_anomalia,
                          levels = c("Down_syndrome_Q90", "q90_<35 anos", "q90_≥35 anos"))
  ) %>%
  arrange(tipo_anomalia)

down_estrat2$isol <- NA
down_estrat2$mult <- NA
down_estrat2$sind <- NA

nvtable2 <- rbind(nvtable2, down_estrat2)
write_xlsx(nvtable2, "C:/.../brasil_nvtable2.xlsx")

# nvtable3
nvtable3 <- sinasc %>%
  group_by(faixa_etaria) %>%
  summarise(
    n_Q90 = sum(str_detect(CODANOMAL1, "Q90"), na.rm = TRUE),
    .groups = "drop"
  )

write_xlsx(nvtable3, "C:/.../brasil_nvtable3.xlsx")

# nvtable4
contar_anomalias2 <- function(simof, padrao, nome_classificacao) {
  simof %>%
    summarise(
      n_casos = sum(str_detect(CODANOMAL1, padrao), na.rm = TRUE)
    ) %>%
    mutate(

```

```

    classificacao = "Total",
    tipo_anomalia = nome_classificacao,
    .before = 1
  )
}

```

Tabela com padrões e nomes descritivos

```

padroes2 <- tibble::tibble(
  padrao = c(
    "Q360|Q361|Q369",
    "Q361",
    "Q351|Q353|Q357",
    "Q37",
    "QQQ",
    "Q050|Q051|Q052|Q053|Q055|Q056|Q057|Q058",
    "Q010|Q011|Q012|Q018",
    "QQQ", # codigo abaixo (Q540|Q541|Q542|Q543|Q548) & Q564

    "Q710|Q711|Q712|Q713|Q714|Q715|Q716|Q720|Q721|Q722|Q723|Q724|Q725|
    Q726|Q727",
    "Q660|Q661|Q662|Q663|Q664|Q665|Q666|Q667",

    "Q200|Q201|Q202|Q203|Q204|Q205|Q206|Q210|Q211|Q212|Q213|Q214|Q215|
    Q216|Q217|

    218|Q220|Q221|Q222|Q223|Q224|Q225|Q226|Q230|Q231|Q232|Q233|Q234|Q2
    40|Q241|
    Q242|Q243|Q244|Q245|Q246|Q250|Q251|Q252|Q253|Q254|Q255|Q256|Q257|
    Q260|Q261|Q262|Q263|Q264|Q265|Q266|Q267|Q268|Q269",
    "QQQ",
    "QQQ",
    "QQQ"
  ),
  nome = c(
    "cleft lip only with laterality specified (Q36.0,Q36.1,Q36.90)",
    "median cleft lip (Q36.1)",
    "cleft palate, with specified extension - hard and/or soft palate
    (Q35.1,Q35.3,Q35.5,Q35.7)",
    "cleft lip and palate coded under the heading Q37",
    "spina bifida cases with skin coverage specified - open or closed (Q05.x1,Q05.x2) ",
    "spina bifida cases with specified level
    (Q05.0x,Q05.1x,Q05.2x,Q05.3x,Q05.5x,Q05.6x,Q05.7x,Q05.8x)",
    "encephalocele - occipital, temporal, frontal, etc
    (Q01.0,Q01.1,Q01.2,Q01.8,Q01.80,Q01.81 ,Q01.82,Q01.83)",
    "ambiguous genitalia if hypospadias is the only diagnosis (Q54.0-Q54.3,Q54.8,Q54.9
    & Q56.4)",
    "limb deficiencies type and localization specified (Q71.0-Q71.6,Q72.0-72.7)",

```



```

"rotation and position specified (Q66.0-Q66.7)",
"congenital heart defects (Q20.0-Q26)",
"microtia with specified degree (Q16.0,Q17.22,Q17.23)",
"type and localization specified (Q69.0x-Q69.2x)",
"Down syndrome (Q20.x-Q26.x)"
)
)

# Aplicar a função a todos os padrões e unir os resultados
nvtable4 <- purrr::map2_dfr(
  .x = padroes2$padrao,
  .y = padroes2$nome,
  .f = ~ contar_anomalias2(sinasc, .x, .y)
)

nvtable4[8,3] <- length(intersect(which(str_detect(sinasc$CODANOMAL1,
"Q540|Q541|Q542|Q543|Q548")),
  which(str_detect(sinasc$CODANOMAL1, "Q564"))))

nvtable4[14,3] <- length(intersect(which(str_detect(sinasc$CODANOMAL1,
"Q20|Q21|Q22|Q23|Q24|Q25|Q26")),
  which(str_detect(sinasc$CODANOMAL1, "Q90"))))

write_xlsx(nvtable4, "C:/.../brasil_nvtable4.xlsx")

#####
# loop para extrair contagens por UF

# criar coluna para estratificar por UF
cod_uf <- tibble::tibble(
  uf_code = c("12", "27", "13", "16", "29", "23", "53", "32", "52", "21", "31",
    "50", "51", "15", "25", "26", "22", "41", "33", "24", "43", "11",
    "14", "42", "28", "35", "17"),
  uf_sigla = c("AC", "AL", "AM", "AP", "BA", "CE", "DF", "ES", "GO", "MA", "MG",
    "MS", "MT", "PA", "PB", "PE", "PI", "PR", "RJ", "RN", "RS", "RO",
    "RR", "SC", "SE", "SP", "TO")
)

sinasc$CODMUNNASC <- as.character(sinasc$CODMUNNASC)

sinasc <- sinasc %>%
  mutate(uf_code = substr(CODMUNNASC, 1, 2)) %>%
  left_join(cod_uf, by = "uf_code") %>%
  rename(uf = uf_sigla)

ufs <- unique(sinasc$uf)

```

```

for (uf_atual in ufs) {
  # Define o caminho da pasta da UF
  pasta_uf <- paste0("C:/.../", uf_atual)

  # Cria a pasta, se ainda não existir
  if (!dir.exists(pasta_uf)) {
    dir.create(pasta_uf)
  }

  sinasc_uf <- sinasc %>% filter(uf == uf_atual)

  nvtable1 <- sinasc_uf %>%
    group_by(faixa_etaria) %>%
    summarise(total_nascidos = n(), .groups = "drop")

  write_xlsx(nvtable1, paste0(pasta_uf, "/", uf_atual, "_nvtable1.xlsx"))

  nvtable2 <- purrr::map2_dfr(
    .x = padroes$padrao,
    .y = padroes$nome,
    .f = ~ contar_anomalias_sinasc(sinasc_uf, .x, .y)
  ) %>%
    filter(!is.na(classificacao)) %>%
    pivot_wider(names_from = classificacao, values_from = n_casos, values_fill = 0) %>%
    as.data.frame()

  idx <- unique(c(
    intersect(which(str_detect(sinasc_uf$CODANOMAL, "Q35")),
    which(str_detect(sinasc_uf$CODANOMAL, "Q36"))),
    which(str_detect(sinasc_uf$CODANOMAL, "Q37"))
  ))

  tab <- table(factor(sinasc_uf$classificacao[idx], levels = c("isol", "mult", "sind")))
  total <- sum(tab)
  valores <- c(total, as.numeric(tab))
  nvtable2[9, c("Total", "isol", "mult", "sind")] <- valores

  # Gastrosquise
  estratificado <- sinasc_uf %>%
    mutate(tipo_anomalia = if_else(IDADEMAE < 20, "q793_<20 anos", "q793_≥20
anos")) %>%
    group_by(classificacao, tipo_anomalia) %>%
    summarise(n_Q793 = sum(str_detect(CODANOMAL, "Q793"), na.rm = TRUE),
    .groups = "drop")

  totais <- sinasc_uf %>%

```

```

mutate(tipo_anomalia = if_else(IDADEMAE < 20, "q793_<20 anos", "q793_≥20
anos")) %>%
  group_by(tipo_anomalia) %>%
  summarise(n_Q793 = sum(str_detect(CODANOMAL, "Q793"), na.rm = TRUE),
.groups = "drop") %>%
  mutate(classificacao = "Total") %>%
  select(classificacao, everything())

gastrosquise_estrat2 <- bind_rows(estratificado, totais) %>%
  filter(!is.na(classificacao), !is.na(tipo_anomalia)) %>%
  pivot_wider(names_from = classificacao, values_from = n_Q793, values_fill = 0)

nvtable2 <- rbind(nvtable2, gastrosquise_estrat2)

# Down syndrome
down_estrat2 <- sinasc_uf %>%
  mutate(tipo_anomalia = if_else(IDADEMAE < 35, "q90_<35 anos", "q90_≥35 anos"))
%>%
  group_by(tipo_anomalia) %>%
  summarise(
    n_Q90 = sum(str_detect(CODANOMAL, "Q90"), na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(classificacao = "Total") %>%
  select(classificacao, everything())

down_estrat2 <- down_estrat2 %>%
  filter(!is.na(tipo_anomalia)) %>%
  pivot_wider(
    names_from = classificacao,
    values_from = n_Q90,
    values_fill = 0
  )

down_estrat2 <- down_estrat2 %>%
  bind_rows(
    down_estrat2 %>%
      summarise(
        tipo_anomalia = "Down_syndrome_Q90",
        across(where(is.numeric), sum, na.rm = TRUE)
      )
  ) %>%
  # Reordena as linhas conforme a ordem desejada
  mutate(
    tipo_anomalia = factor(tipo_anomalia,
      levels = c("Down_syndrome_Q90", "q90_<35 anos", "q90_≥35 anos"))
  ) %>%

```

```

  arrange(tipo_anomalia)

down_estrat2$isol <- NA
down_estrat2$mult <- NA
down_estrat2$sind <- NA

nvtable2 <- rbind(nvtable2, down_estrat2)
nvtable2 <- nvtable2 %>%
  select(tipo_anomalia, Total, isol, mult, sind)

write_xlsx(nvtable2, paste0(pasta_uf, "/", uf_atual, "_nvtable2.xlsx"))

# nvtable3
nvtable3 <- sinasc_uf %>%
  group_by(faixa_etaria) %>%
  summarise(n_Q90 = sum(str_detect(CODANOMAL, "Q90"), na.rm = TRUE), .groups =
"drop")

write_xlsx(nvtable3, paste0(pasta_uf, "/", uf_atual, "_nvtable3.xlsx"))

# nvtable4
nvtable4 <- purrr::map2_dfr(
  .x = padroes2$padrao,
  .y = padroes2$nome,
  .f = ~ contar_anomalias2(sinasc_uf, .x, .y)
)

nvtable4[8,3] <- length(intersect(
  which(str_detect(sinasc_uf$CODANOMAL, "Q540|Q541|Q542|Q543|Q548")),
  which(str_detect(sinasc_uf$CODANOMAL, "Q564"))
))

nvtable4[14,3] <- length(intersect(
  which(str_detect(sinasc_uf$CODANOMAL, "Q20|Q21|Q22|Q23|Q24|Q25|Q262")),
  which(str_detect(sinasc_uf$CODANOMAL, "Q90"))
))

write_xlsx(nvtable4, paste0(pasta_uf, "/", uf_atual, "_nvtable4.xlsx"))
}

```

ANEXO IV

Código para extração de outros indicadores específicos

```
library(dplyr)
library(stringr)

# Vetor com códigos não especificados
inespecifico <- c(
  "Q019", "Q039", "Q049", "Q054", "Q059", "Q069", "Q079", "Q129", "Q139", "Q149",
  "Q159", "Q169",
  "Q179", "Q189", "Q209", "Q219", "Q229", "Q239", "Q249", "Q259", "Q264", "Q269",
  "Q279", "Q289",
  "Q309", "Q319", "Q339", "Q349", "Q359", "Q378", "Q379", "Q399", "Q403", "Q409",
  "Q429", "Q439",
  "Q459", "Q519", "Q529", "Q539", "Q549", "Q559", "Q563", "Q564", "Q602", "Q605",
  "Q613", "Q619",
  "Q639", "Q649", "Q652", "Q655", "Q659", "Q669", "Q685", "Q699", "Q709", "Q719",
  "Q729", "Q730",
  "Q731", "Q738", "Q749", "Q759", "Q769", "Q779", "Q789", "Q799", "Q809", "Q819",
  "Q829", "Q839",
  "Q849", "Q859", "Q909", "Q913", "Q917", "Q929", "Q939", "Q959", "Q969", "Q979",
  "Q984", "Q989", "Q999"
)

# Cálculo dos indicadores
indicadores <- tibble::tibble(
  indicador = c(
    "Sexo ignorado sem DDS",
    "IDANOMAL não preenchido",
    "IDANOMAL = 9 (ignorado)",
    "IDANOMAL = 1 e CODANOMAL vazio",
    "Prevalência Q897 por 10 mil",
    "Prevalência Q898 por 10 mil",
    "Prevalência Q899 por 10 mil",
    "Prevalência de códigos inespecíficos por 10 mil",
    "DN recebida após 60 dias"
  ),
  valor = c(
    100 - (length(which(sinasc$SEXO == "I" & str_detect(sinasc$CODANOMAL, "Q56")))/
length(which(sinasc$SEXO == "I"))) * 100,
    length(which(is.na(sinasc$IDANOMAL)))/length(sinasc$IDANOMAL) * 100,
    length(which(sinasc$IDANOMAL == 9))/length(sinasc$IDANOMAL) * 100,
    length(which(sinasc$IDANOMAL == 1 & is.na(sinasc$CODANOMAL)))/
length(which(sinasc$IDANOMAL == 1)) * 100,
    length(which(str_detect(sinasc$CODANOMAL, "Q897")))/length(sinasc$ID_SINASC)
* 10000,
```

```

length(which(str_detect(sinasc$CODANOMAL, "Q898")))) / length(sinasc$ID_SINASC)
* 10000,
length(which(str_detect(sinasc$CODANOMAL, "Q899")))) / length(sinasc$ID_SINASC)
* 10000,
length(which(str_detect(sinasc$CODANOMAL, paste0(inespecifico, collapse = "|"))))
/ length(sinasc$ID_SINASC) * 10000,
length(which(as.numeric(sinasc$DIFDATA) > 60)) / length(sinasc$ID_SINASC) * 100 ,
length(which(as.numeric(sinasc$DIFDATA) > 60 & sinasc$IDANOMAL==1)) /
length(which(sinasc$IDANOMAL==1)) * 100
)
)

```

```

write_xlsx(indicadores, "C:/.../brasil_indicadores.xlsx")

```

```

#####

```

```

#####

```

```

# loop por uf

```

```

# Inicializa uma lista para armazenar os dados por UF

```

```

resultados_lista <- list()

```

```

for (uf_atual in ufs) {

```

```

  sinasc_uf <- sinasc %>% filter(uf == uf_atual)

```

```

  # Calcula os indicadores

```

```

  valores <- c(

```

```

    "Sexo ignorado sem DDS" =

```

```

    100 - (length(which(sinasc_uf$SEXO == "I" & str_detect(sinasc_uf$CODANOMAL,
"Q56")))) /

```

```

    length(which(sinasc_uf$SEXO == "I")) * 100),

```

```

    "IDANOMAL não preenchido" =

```

```

    length(which(is.na(sinasc_uf$IDANOMAL))) / length(sinasc_uf$IDANOMAL) * 100,

```

```

    "IDANOMAL = 9 (ignorado)" =

```

```

    length(which(sinasc_uf$IDANOMAL == 9)) / length(sinasc_uf$IDANOMAL) * 100,

```

```

    "IDANOMAL = 1 e CODANOMAL vazio" =

```

```

    length(which(sinasc_uf$IDANOMAL == 1 & is.na(sinasc_uf$CODANOMAL))) /

```

```

    length(which(sinasc_uf$IDANOMAL == 1)) * 100,

```

```

    "Prevalência Q897 por 10 mil" =

```

```

    length(which(str_detect(sinasc_uf$CODANOMAL, "Q897")))) /

```

```

    length(sinasc_uf$ID_SINASC) * 10000,

```

```

"Prevalência Q898 por 10 mil" =
  length(which(str_detect(sinasc_uf$CODANOMAL, "Q898")))/
length(sinasc_uf$ID_SINASC) * 10000,

"Prevalência Q899 por 10 mil" =
  length(which(str_detect(sinasc_uf$CODANOMAL, "Q899")))/
length(sinasc_uf$ID_SINASC) * 10000,

"Prevalência de códigos inespecíficos por 10 mil" =
  length(which(str_detect(sinasc_uf$CODANOMAL, paste0(inespecifico, collapse =
"|")))/
length(sinasc_uf$ID_SINASC) * 10000,

"DN recebida após 60 dias" =
  length(which(as.numeric(sinasc_uf$DIFDATA) > 60)) / length(sinasc_uf$ID_SINASC)
* 100 ,

"DN recebida após 60 dias entre ACs" =
  length(which(as.numeric(sinasc_uf$DIFDATA) > 60 & sinasc_uf$IDANOMAL==1)) /
length(which(sinasc_uf$IDANOMAL==1)) * 100
)

# Cria um data.frame com UF e indicadores como colunas
df_uf <- as.data.frame(t(valores))
df_uf$UF <- uf_atual

# Adiciona à lista
resultados_lista[[uf_atual]] <- df_uf
}

# Une todos os resultados em um único data.frame
df_final <- bind_rows(resultados_lista) %>%
  relocate(UF) # Coloca a coluna UF na frente

write_xlsx(df_final, "C:/.../ufs_indicadores.xlsx")

```