



**RCFD**

Rastreamento de Contatos no  
Enfrentamento a Doenças Infecciosas

## Aula 1 - Instalação do Go.Data



<b>Ficha Técnica</b>
<b>Supervisão</b> - Sarah Mendes <b>Coordenação Pedagógica</b> – Hirla Arruda <b>Conteudista</b> – Marcela <b>Revisão técnica</b> - Sarah Mendes <b>Revisão</b> – Yorrana Martins <b>Design Instrucional</b> – Guilherme Duarte <b>Ilustração</b> - Guilherme Duarte
<b>Supervisão</b> – Associação Brasileira de Profissionais de Epidemiologia de Campo – ProEpi Sara Ferraz <b>Supervisão</b> – Sala de Situação – Universidade de Brasília Jonas Brant
<b>Execução</b>
ProEpi SDS – Sala de situação – Universidade de Brasília
<b>Parceiros</b>
Project HOPE Project ECHO NIA TERO Brasil Foundation EXO plataforma
<b>Apoio</b>
SKOLL FOUNDATION

Copyright © 2021, Associação Brasileira de Profissionais de Epidemiologia de Campo.

Todos os direitos reservados.

A cópia total ou parcial, sem autorização expressa do(s) autor(es) ou com o intuito de lucro, constitui crime contra a propriedade intelectual, conforme estipulado na Lei nº 9.610/1998 (Lei de Direitos Autorais), com sanções previstas no Código Penal, artigo 184, parágrafos 1º ao 3º, sem prejuízo das sanções cabíveis à espécie.

## Sumário

Instalação Go.data .....	6
Instalação local .....	6
Instalação em servidor .....	6
Recursos necessários.....	6
Windows.....	7
Linux (32 ou 64 bits) .....	7
Instalação no Windows 10 .....	8
Instalação no Linux.....	13
Complementos de instalação .....	14
Node Version Manager .....	14
MongoDB.....	15
Execução do Go.Data no Linux.....	20
Configurações adicionais.....	28
Indicadores de avaliação .....	34
Gerenciamento da investigação de caso e rastreamento de contatos.....	34
Medições processuais programáticas e dos resultados.....	36
Conclusão .....	38

## Aula 1 - Instalação do Go.Data



Figura 1 – Logo Go.Data Por goarn

Olá!

Nesta aula você aprenderá a instalar o Go.Data no sistema operacional Windows e Linux, incluir configurações adicionais, interpretar indicadores de avaliação, gerenciar a investigação de casos e rastreamento de contatos e conhecer as medidas processuais programáticas e por resultados.

Ao final desta aula, você será capaz de:

- Instalar o Go.Data segundo o sistema operacional disponível;
- Compreender a importância dos indicadores de avaliação;
- Gerenciar investigação de casos e rastreamento de contatos com a ferramenta Go.Data.

## Instalação Go.data

Há dois tipos de instalação: a instalação local (autônoma) e a instalação em servidor.

### Instalação local

A instalação local é realizada em um computador onde armazena todos os dados do Go.Data nesse local. Recomenda-se essa instalação na investigação de surtos em que uma pessoa administra a entrada e saída de dados no sistema.

### Instalação em servidor

A instalação em servidor também é realizada em um computador, porém como um servidor após configurações de perfil de usuários (computadores e celulares) conectados com os dados desse servidor. Recomenda-se a instalação em servidor na investigação de surtos de grande extensão que exigem trabalho em equipe.



#### Fica a Dica!

É fundamental a leitura dos Termos de Uso e Licença do Software, encontrados na [comunidade oficial do Go.Data](#). Nessa comunidade, há documentos sobre o software e guias de apoio.

### Recursos necessários

Os recursos necessários em disco para cada tipo de instalação dependem do volume de dados que serão processados no sistema. Esses recursos variam conforme

o tamanho do surto, a quantidade de pessoas que terão acesso ao sistema, a região onde o surto será monitorado (Uma cidade? Um estado? Um país?) e a quantidade de variáveis que estarão presentes em cada caso, contato e monitoramento.

O espaço mínimo necessário em disco para funcionar o Go.Data é de 3GB (Gigabytes - unidade de medida de informação) em ambos os sistemas operacionais (Windows e Linux). Ressalta-se que esse espaço é considerado o mínimo necessário na instalação do Go.Data com os arquivos básicos de funcionamento do software, como usuário administrador e templates de surto.

O espaço livre reservado no desktop deve ser maior ao já ocupado na instalação do software (3 GB), como exemplo uma base de dados que ocupa memória de 150MB (Megabytes), equivalente à 0.15GB: **Cerca de 1.000 casos, 200 contatos, 2.000 monitoramentos diários, 30 usuários, 70 localizações e questionários com 20 variáveis.**

Recomenda-se reservar memória superior do desktop para aumentar a capacidade de uso do Go.Data.

Os requisitos mínimos de cada sistema operacional são essenciais no bom funcionamento do software..

## Windows

- Sistema operacional 64 bits.
- Windows 7 ou superior.
- Processador: 2GHz (frequência de clock do processador em GigaHertz).
- Memória RAM: 8GB.

## Linux (32 ou 64 bits)

- Sistema operacional 32 ou 64 bits.

- Sistemas operacionais: Ubuntu 12.04, Fedora 21, Debian 8 ou superiores.
- Processador: 2GHz (frequência de clock do processador em GigaHertz).
- Memória RAM: 4GB.

## Instalação no Windows 10

O primeiro passo de instalação do Go.Data no Windows 10 é acessar o link de download, fornecido pela Organização Mundial de Saúde (OMS) ou pela equipe de suporte do Go.Data, mediante contato.

Os arquivos para download são encontrados diretamente na opção “Go.Data Setup 2.35.0.exe”. O número **2.35.0** corresponde à versão do Go.Data. Selecione o arquivo com a extensão **.exe** para iniciar o processo de instalação.



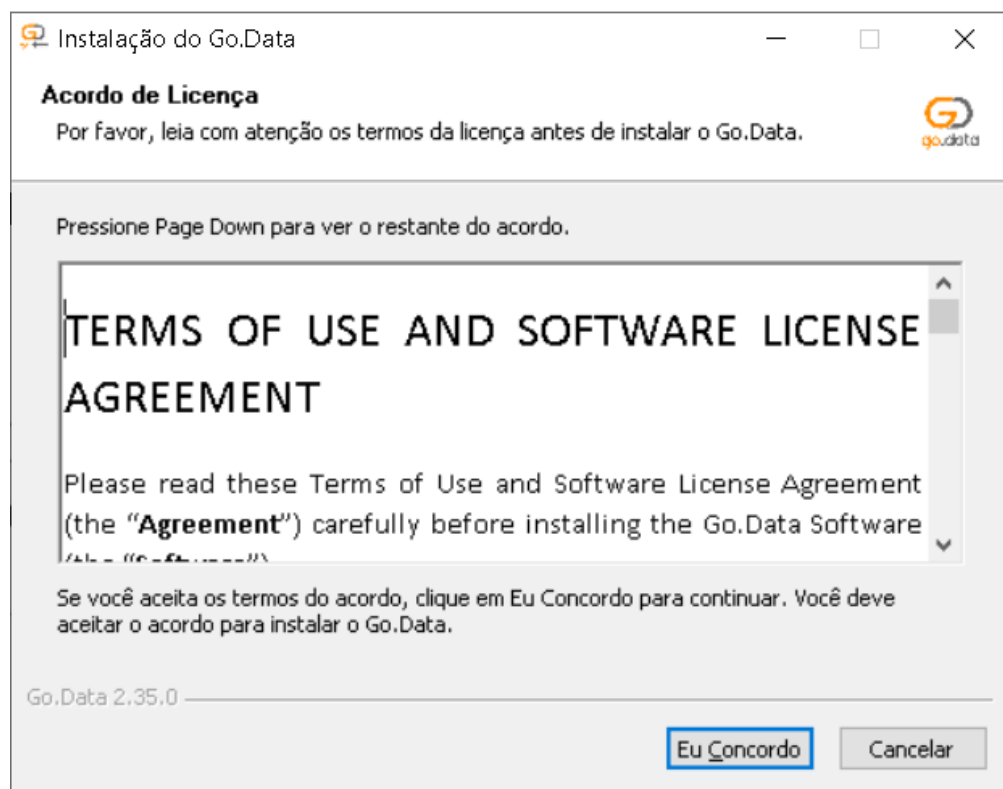
### Fica a Dica!

Uma das limitações do Go.Data na instalação é o idioma do sistema operacional diferente do idioma do teclado em inglês. Isso se deve ao problema com a aplicação do software, a ser corrigido nas futuras versões (WHO, 2020).

Após concluir o download do arquivo .exe, execute-o, com duplo clique do mouse no seu computador. Em seguida, abre uma janela de instalação do Go.Data. Leia os Termos de Uso e clique no botão “Eu Concordo” (Figura-1).

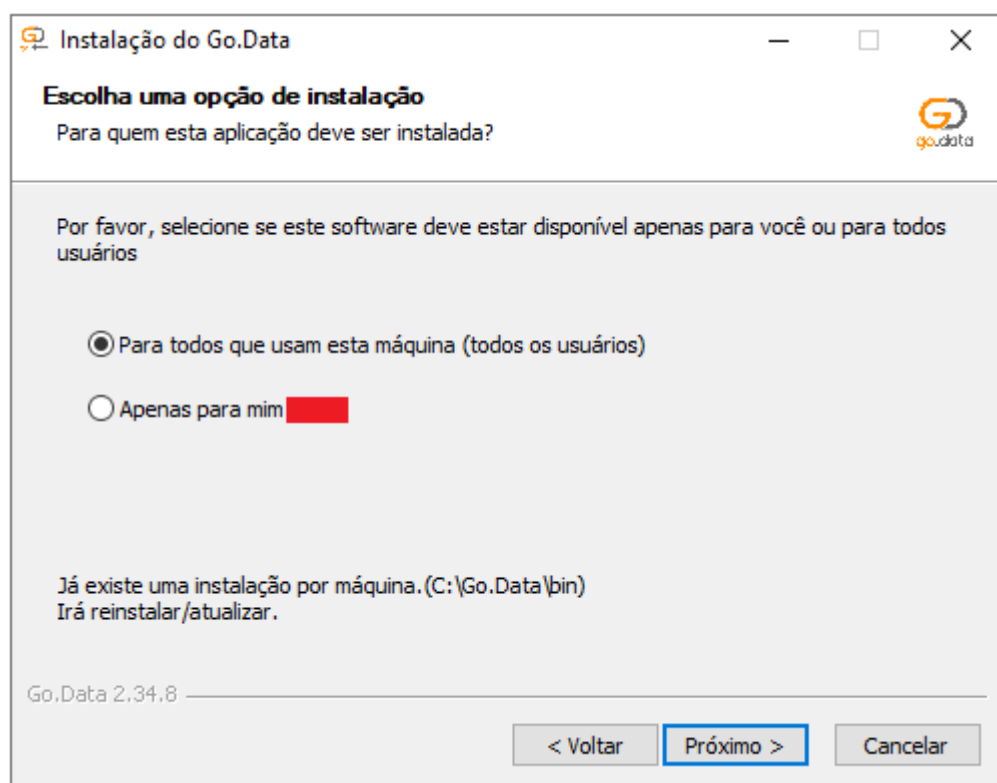
**Figura 1. Captura de tela sobre “Termos de Uso e Acordo de Licença” da janela Instalação do Go.Data.**





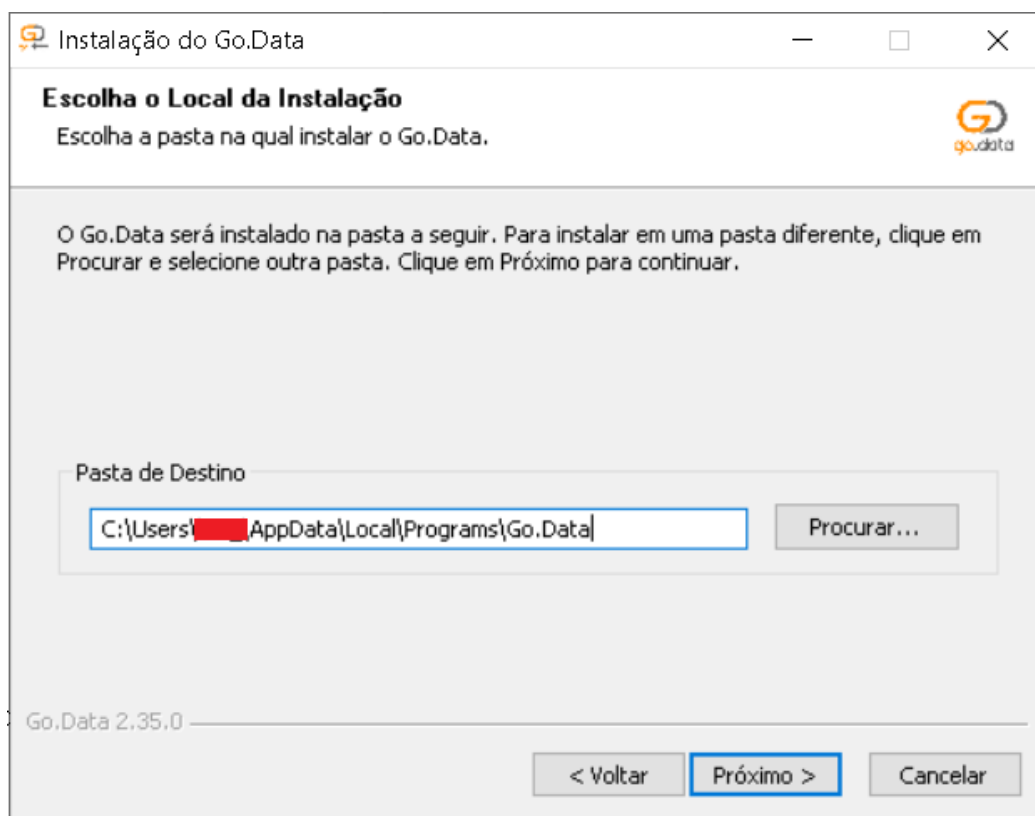
Selecione o tipo de acesso do usuário do Go.Data (Figura-2).

**Figura 2. Captura de tela sobre “Opção de acesso do usuário” da janela Instalação do Go.Data.**



Selecione o local de instalação onde os arquivos do Go.Data estarão armazenados no computador (Figura-3).

**Figura 3. Captura de tela sobre “Local da Instalação” da janela Instalação do Go.Data.**



#### Fica a Dica!

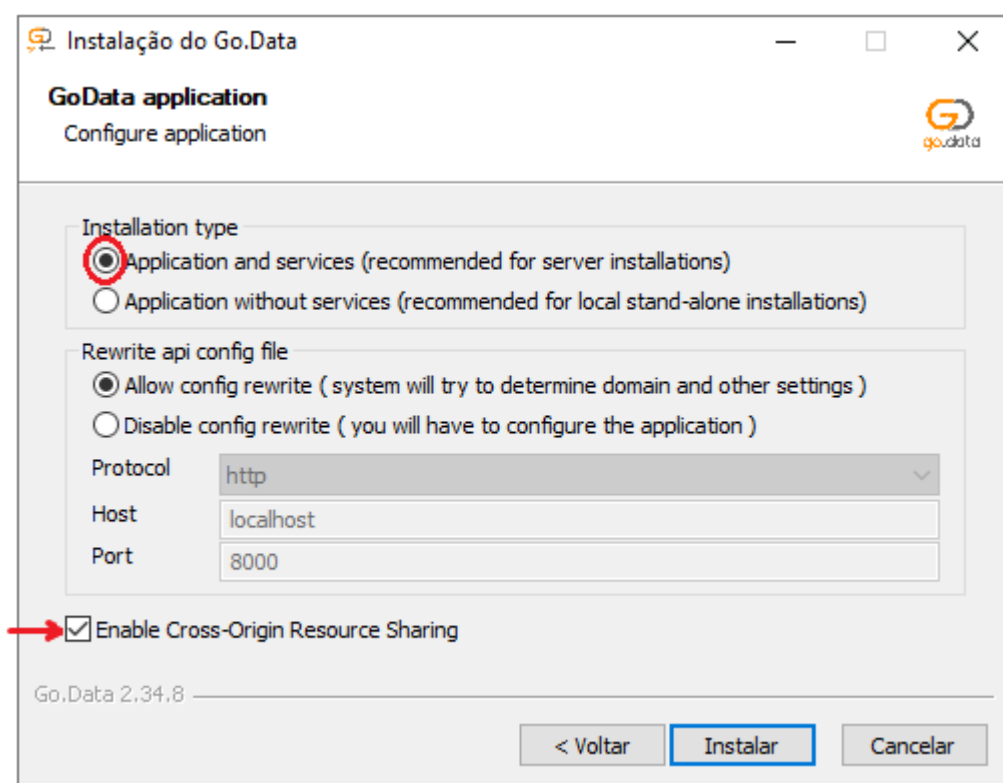
Caso deseje alterar o local para uma pasta diferente do padrão, clique em “Procurar”, navegue até o local desejado, depois clique em “Ok”.

Nas configurações de aplicação do software, marque a opções:

- “Application and services (recommended for server installations)”  
- Para instalação em servidor.
- “Application without services (recommended for local stand-alone installations)” - Para instalação local.
- “Allow config rewrite (system will try to determine domain and other settings)”.
- “Enable Cross-Origin Resource Sharing”.

Posteriormente, clique no botão “Instalar”. Ao finalizar a instalação do software, um ícone do Go.Data será criado na área de trabalho do desktop. Para executá-lo como administrador, clique na opção “Run as administrator” com o botão direito do mouse (Figura-4).

**Figura 4. Captura de tela sobre “Configurações de aplicação” da janela Instalação do Go.Data.**



Clique no botão “Launch Go.Data” para manter ou alterar as portas do MongoDB e do Go.Data (Figura-5).

**Figura 5. Captura de tela de configuração Launch Go.Data.**

Do you want to change the  
default Go.Data configuration?

Mongo Port

27017

Go.Data Port

3000

Encrypt data



Launch Go.Data

Go.Data 2.34.8  
Build 2006101800

O serviço de banco de dados e as configurações iniciais da aplicação do Go.Data estão disponíveis. Abra o Go.Data no endereço do localhost na porta selecionada, como “<http://localhost:3000/>” no navegador.

## Instalação no Linux

A sequência de passos é válida para instalação no Ubuntu 16.04. Recomenda-se instalar o Node Version Manager e o MongoDB para melhorar o desempenho do Go.Data no Linux.

## Complementos de instalação

### Node Version Manager

Inicia-se a instalação do Node Version Manager (NVM), versão 8.17.0 ,(NVM), gerenciador de versões do Node. Posteriormente, o pacote build-essential é instalado (Figura-6). Abra o terminal, digite os comandos “Update” e “Y”, depois pressione “Enter” para finalizar cada um dos complementos.

```
sudo apt-get update
```

**Figura 6. Captura de tela do Node Version Manager e Build-essential.**

```
sudo apt-get install build-essential -y
```

Comandos que se iniciam com “sudo”, solicitam a senha, pois permite privilégios de super usuários para usuários comuns da máquina. Então, digite a senha e pressione “Enter”.

A instalação do NVM ocorre a partir do comando digitado em v0.36.0 (Figura-7).

**Figura 7. Captura de tela do comando Node Version Manager.**

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.36.0/install.sh | bash
```

A versão do NVM pode ser alterada por outra mais recente, disponível no [repositório do NVM](#). Após instalar o NVM, reinicie o terminal para que as aplicações tenham efeito.

Para checar se a instalação foi concluída com sucesso, use o comando que retorna a versão do NVM (Figura-8). Caso esteja tudo ok, visualiza-se o modo remoto

do NVM (Figura-9). Esse comando retorna todas as versões disponíveis para instalação do node.

**Figura 8. Captura de tela do modo versão do Node Version Manager.**

```
nvm --version
```

Após instalado o NVM, iremos buscar as versões disponíveis do node:

**Figura 9. Captura de tela do modo remoto do Node Version Manager.**

```
nvm ls-remote node
```

Para instalar o NVM, certifique-se de utilizar a versão 8.x) (Figura-10):

**Figura 10. Captura de tela do comando instalação do Node Version Manager versão 8.x.**

```
nvm install v8.17.0
```

Para verificar se a instalação foi realizada corretamente, utilize o comando **version**, que retorna a versão instalada do node (Figura-11))

**Figura 11. Captura de tela do modo versão do Node Version Manager.**

```
node --version
```

## MongoDB

Inicialmente deve-se importar a chave pública GPC do MongoDB no comando (Figura-12).

**Figura 12. Captura de tela da chave pública GPC do MongoDB.**

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
```

Posteriormente, crie o arquivo `mongodb-org-3.4.list` (Figura-13).

**Figura 13. Captura de tela do arquivo de instalação do MongoDB.**

```
echo "deb http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4 multiverse" |  
sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list
```

Atualize o pacote do repositório local com o **apt-get update** (Figura-14).

**Figura 14. Captura de tela do comando de atualização do MongoDB.**

```
sudo apt-get update
```

A partir deste passo, instale a versão 3.4 do MongoDB. Caso deseje instalar uma versão diferente, substitua a versão desejada no lugar do “3.4” (Figura-15).

**Figura 15. Captura de tela do modo versão do MongoDB.**

```
sudo apt-get install -y mongodb-org=3.4 mongodb-org-server=3.4 mongodb-org-  
shell=3.4 mongodb-org-mongos=3.4 mongodb-org-tools=3.4
```

Para rodar o serviço MongoDB, necessita-se criar o arquivo de serviço no diretório “/etc/systemd/system/” (Figura-16). Utilize o editor, de preferência neste comando. Neste exemplo, utilizou-se o Nano:

**Figura 16. Captura de tela do arquivo de serviços no diretório MongoDB.**

```
sudo nano /etc/systemd/system/mongodb.service
```

Nesta etapa, o editor de texto está aberto. Adicione o código no arquivo **mongodb.service** (Figura-17).

**Figura 17. Captura de tela do arquivo `mongodb.service` do MongoDB.**

```
#Unit contains the dependencies to be satisfied before the service is started.
```



### [Unit]

Description=MongoDB

Database

After=network.target

Documentation=<https://docs.mongodb.org/manual>

```
# Service tells systemd, how the service should be started.  
# Key `User` specifies that the server will run under the mongodb user and  
# `ExecStart` defines the startup command for MongoDB server.
```

### [Service]

User=mongodb

Group=mongodb

ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

```
# Install tells systemd when the service should be automatically started.
```

```
# `multi-user.target` means the server will be automatically started during boot.
```

### [Install]

WantedBy=multi-user.target

Reitera-se que qualquer caractere copiado errado pode comprometer o funcionamento do serviço de banco de dados.

Após copiar o código do arquivo, pressione as teclas “CTRL + X” para sair. Em seguida, pressione a tecla “Y” para salvar o arquivo. Então, aperte “Enter” para confirmar.

De volta ao terminal comum, o serviço do systemd é atualizado para que o novo serviço do MongoDB criado seja efetivado (Figura-18).

**Figura 18. Captura de tela do arquivo systemd do MongoDB.**

```
systemctl daemon-reload
```

Reinicie o serviço do MongoDB com o systemctl, executando o comando “start” (Figura-19).

**Figura 19. Captura de tela do arquivo systemctl start do MongoDB.**

```
sudo systemctl start mongod
```

Verifique se o serviço do banco de dados foi iniciado corretamente na porta 27017(Figura-20).

**Figura 20. Captura de tela do arquivo netstat-plnt do MongoDB.**

```
netstat -plnt
```

Verifique também se o serviço foi iniciado com sucesso por meio do comando status (Figura-21).

**Figura 21. Captura de tela do arquivo status do MongoDB.**

```
sudo systemctl status mongod
```

Verifique se o retorno desse comando mostra o nome do serviço (mongod), “loaded” e “active”.

Ative o modo de início automático do serviço, com o “systemctl enable” e o nome do serviço (mongod), (Figura-22)

**Figura 22. Captura de tela do arquivo systemctl enable do MongoDB.**

```
sudo systemctl enable mongod
```

Utilize o Shell do MongoDB, que é a interface para acesso de serviços e funcionalidades do MongoDB. Para abrir o shell digite “mongo”:

**Figura 23. Captura de tela do comando mongo do MongoDB.**

```
mongo
```

Aplique o modo “use” para selecionar o database a ser usado, no caso o “admin” para usuário administrador (Figura-24).

**Figura 24. Captura de tela do modo use admin do MongoDB.**

```
use admin
```

O usuário root foi criado. Digite a senha desejada entre aspas no atributo “pwd”, substituindo a palavra “password”. Lembre-se que a senha será usada nos arquivos de configuração de banco de dados do Go.Data. O nome do usuário também pode ser alterado para o de sua preferência. (Figura-25).:

**Figura 25. Captura de tela do usuário criado no MongoDB.**

```
db.createUser({user:"admin", pwd:"password", roles:[{role:"root", db:"admin"}]})
```

Feche a janela Shell do MongoDB, pressionando as teclas “CTRL+C”. Reinicie o serviço, com o comando “restart” do “systemctl” (Figura-26).

**Figura 26. Captura de tela do modo restart no MongoDB.**

```
sudo systemctl restart mongod
```

Ao conectar o usuário criado, certifique-se de substituir o nome do seu usuário e senha definidos no comando de criação de usuário pelo “admin” e “password” (Figura-27)

**Figura 27. Captura de tela de autenticação no MongoDB.**

```
mongo -u admin -p password --authenticationDatabase admin
```

Após executar esse comando, verifique se o mongo está conectado. Ao finalizar a autenticação, a instalação do MongoDB está concluída. Os bancos de dados (databases) são abertos e executados no shell do mongo (Figura-28).

**Figura 28. Captura de tela do modo demonstração do banco de dados no MongoDB.**

```
show dbs
```

### **Execução do Go.Data no Linux**

Ao finalizar a instalação dos complementos NVM e MongoDB, inicia-se a instalação do Go.Data por meio da criação do diretório, como /godata/go-data-linux-x64 2.35.0.tar.gz. Recomenda-se criar a pasta raiz no diretório inicial do sistema operacional “/”. Para visitá-lo, utiliza-se “cd /godata”. O comando de super usuário “sudo” será aplicado sempre que executar qualquer comando na pasta raiz.

Denomine a pasta criada , por exemplo “godata”, “v2.35.0” na pasta raiz (Figura-29)

**Figura 29. Captura de tela do comando de criação do diretório do Go.Data.**

```
sudo mkdir /godata
```

Mova o arquivo de instalação para a pasta criada pelo usuário, com substituição do local onde o arquivo de instalação está salvo (formato .tar.gz) (Figura-30).

**Figura 30. Captura de tela do comando de localização do arquivo de instalação do Go.Data.**

```
sudo mv local-do-arquivo/de-instalação/go-data-linux-x64 2.35.0.tar.gz /godata/
```

Acesse a pasta raiz com o comando “cd” (Figura-31) e atribua permissões ao usuário no diretório go-data-linux-x64 2.35.0.tar.gz e substitua “seuusuário” pelo nome do usuário da máquina (Figura-32).

**Figura 31. Captura de tela do comando “cd” do Go.Data.**

```
cd /
```

**Figura 32. Captura de tela do comando de alteração do usuário do Go.Data.**

```
sudo chown seuusuário:seuusuário -R godata/
```

Acesse o diretório do arquivo de instalação por meio do comando “cd” (Figura-33) e extraia o arquivo go-data-linux-x64 2.35.0.tar.gz (Figura-34).

**Figura 33. Captura de tela do diretório do Go.Data.**

```
cd /godata
```

**Figura 34. Captura de tela do comando de extração dos arquivos Go.Data.**

```
tar -xpvf go-data-linux-x64\ 2.35.0.tar.gz
```

A estrutura da pasta “/godata” é visualizada (Figura-35)

**Figura 35. Captura da tela da estrutura da pasta “/godata”.**

```
:/godata/v2.34.8$ ls
app-management  go-data-linux-x64 2.34.8.tar  go-data-restore-backup-x64.sh  platforms
go-data         go-data-reset-password-x64.sh  go-data-x64.sh
:/godata/v2.34.8$ _
```

Acesse a pasta “build”, no diretório “go-data” para realizar a instalação do software (Figura-36)

**Figura 36. Captura de tela do comando build do Go.Data.**

```
cd go-data/build/
```

Na pasta “build/”, visualiza-se a estrutura da pasta (Figura-37).

**Figura 37 - Captura de tela da estrutura da pasta build do Go.Data.**

```
:/godata/v2.34.8$ cd go-data/build/
:/godata/v2.34.8/go-data/build$ ls
backups  common  favicon.ico  logs  npm-shrinkwrap.json  README.md  server
client  components  installer  node_modules  package.json  resources
:/godata/v2.34.8/go-data/build$
```

Aplique o usuário “root” nos comandos de instalação. O comando solicitará a inserção da senha. Posteriormente, pressione a tecla “Enter” (Figura-38).

**Figura 38. Captura de tela do comando de criação do diretório do Go.Data.**

```
sudo su
```

É necessário instalar a aplicação com o “npm install” (Figura-39) . Recomenda-se executar o NPM diretamente do diretório, com substituição da versão v8.17.0 pela versão do node (em ciano) e o nome do seu usuário da máquina por “seu-usuário” (Figura-40).

**Figura 39. Captura de tela do usuário do Go.Data.**

```
/home/seu-usuário/.npm/versions/node/v8.17.0/bin/npm install
```

**Figura 40 – Captura de tela de instalação do Go.Data.**

```
bin/npm install  
audited 668 packages in 10.903s  
  
4 packages are looking for funding  
  run 'npm fund' for details  
  
found 143 vulnerabilities (86 low, 10 moderate, 47 high)  
  run 'npm audit fix' to fix them, or 'npm audit' for details
```

Após essa etapa de instalação, configure o banco de dados no arquivo “datasources.json” na pasta “server/”, localizada em “/godata/go-data/build/server” (Figura-41), e configure o servidor no arquivo “config.json”, também na pasta “server/” (Figura-42).

**Figura 41. Captura de tela do comando executor do banco de dados do Go.Data.**

```
nano server/datasources.json
```

Altere o “password” (1) e “user” (2) para o usuário e senha do banco de dados do MongoDB, mantidas entre aspas (Figura-42).

**Figura 42 - Captura de tela do arquivo datasources.json.**

```
GNU nano 2.5.3 File: server/datasources.json
{
  "mongodb": {
    "host": "127.0.0.1",
    "port": 27017,
    "url": "",
    "database": "go-data",
    "password": " ", 1
    "name": "mongodb",
    "user": " ", 2
    "connector": "mongodb",
    "allowExtendedOperators": true,
    "enableGeoIndexing": true,
    "ignoreUndefined": true,
    "maxDepthOfQuery": 24,
    "lazyConnect": true,
    "authSource": "admin"
  },
  "email": {
    "name": "email",
    "connector": "mail",
    "transports": [
      {
```

Para salvar o arquivo e sair do Nano, pressione as teclas “CTRL + X”, clique na tecla “Y”, depois pressione “Enter” para confirmar.

As configurações de host público do Go.Data no arquivo “config.json” “condicionam” ao Go.Data em qual endereço ele deve rodar (Figura-43). Caso haja um domínio configurado, adicione a linha “host” no “public” (1) (Figura-44). Ao rodar o Go.Data localmente, não é necessário realizar esse procedimento, pois funcionará por meio do acesso “localhost:3000” na máquina.

**Figura 43 - Captura de tela do arquivo config.json.**

```
nano server/config.json
```

**Figura 44 - Captura de tela do arquivo config.json.**



```
GNU nano 2.5.3 File: server/config.json

    "limit": "100kb"
  },
  "cors": false
},
"public": {
  "host": "gd.sds.unb.br", 1
  "protocol": "https",
  "port": "80"
},
}
```

No diretório build, inicialize as configurações do banco de dados, para criar os dados básicos do Go.Data. Substitua a versão v8.17.0 pela versão do seu node (em ciano) e o nome do seu usuário da máquina por “seu-usuário” (Figura-45):

**Figura 45 - Captura de tela do comando de configuração da base de dados do Go.Data.**

```
/home/seu-usuário/.nvm/versions/node/v8.17.0/bin/npm run init-database
```

Atenção! Essa operação pode levar alguns minutos (Figura-46).

**Figura 46 - Captura de tela de configuração do banco de dados modo de inicialização.**

```
:/godata/v2.34.8/go-data/build# /home/.../.nvm/versions/node/v8.17.0/
bin/npm run init-database

> go-data-api@2.34.8 init-database /godata/v2.34.8/go-data/build
> node ./server/install/install.js init-database

Setting Up Database Initialisation...
```

Ao finalizar o processo de instalação e configuração, rode o Go.Data com o comando “start”. Substitua a versão v8.17.0 pela versão do seu node (em ciano) e o nome do seu usuário da máquina por “seu-usuário” (Figura-47).

**Figura 47 - Captura de tela do comando de usuário para iniciar o Go.Data.**

```
/home/seu-usuário/.npm/versions/node/v8.17.0/bin/npm start
```

**Figura 48 - Captura de tela do comando aplicação do Go.Data.**

```
:/godata/v2.34.8/go-data/build# /home/ / .npm/versions/node/v8.17.0/
bin/npm start
> go-data-api@2.34.8 start /godata/v2.34.8/go-data/build
> node .
```

Essa aplicação deve estar disponível no endereço local padrão (localhost:3000) e/ou no endereço público definido. Acesse esse endereço pelo navegador. Caso esteja usando o sistema operacional Server, sem interface gráfica, é necessário criar um serviço para o Go.Data rodar em segundo plano. Libere o terminal para outros fins. Ressalta-se que essa etapa é útil nos sistemas operacionais Desktop, com interface gráfica.

Para parar de rodar o Go.Data no terminal, pressione as teclas “CTRL + C”.

Crie o arquivo de serviço (service) no diretório “/etc/systemd/system/”, mesmo diretório para rodar o MongoDB (Figuras-49 e 50).

**Figura 49 - Captura de tela do comando do sistema operacional.**

```
cd /etc/systemd/system/
```

**Figura 50 - Captura de tela do arquivo service no diretório do sistema operacional.**

```
sudo nano godata.service
```

Ao abrir o editor de texto Nano, copie o código no arquivo godata.service (Figura-51).

**Figura 51 - Captura de tela do arquivo godata.service.**

```
[Unit]
```

```
Description=GoData Server
After=network.target

[Service]
User=root
Group=root
ExecStart=/usr/bin/node /godata/go-data/build/.

[Install]
WantedBy=multi-user.target
```

Substitua o código “godata” pelo nome do diretório criado no primeiro comando desta aula.

Para sair do Nano e salvar o arquivo, pressione as teclas “CTRL + X”, em seguida pressione a tecla “Y” para salvar o arquivo. Por fim, confirme com a tecla “ENTER”.

Retorne ao terminal comum e atualize o serviço do systemd para que o novo serviço do Go.Data criado seja efetivado no sistema operacional (Figura-52).

**Figura 52 - Captura de tela do arquivo systemctl daemon-reload.**

```
systemctl daemon-reload
```

Inicie o serviço do Go.Data com o systemctl, executando o “start” com o nome do serviço (Figura-53).

**Figura 53 - Captura de tela do comando start.**

```
sudo systemctl start godata
```

Verifique se o serviço foi iniciado com sucesso (Figura-54)

**Figura 54 - Captura de tela do status Go.Data.**

```
sudo systemctl status godata
```

Verifique se o retorno desse comando demonstra o nome do serviço (mongodb), “loaded” e “active”.

Ative o início automático do serviço quando o sistema operacional iniciar com o “systemctl enable” e, então, o nome do serviço (godata) (Figura-55).

**Figura 55 - Captura de tela do status systemctl enable godata.**

```
sudo systemctl enable godata
```

Se identificar erro, reinicie o serviço com o “restart” do “systemctl” (Figura-56).

**Figura 56 - Captura de tela do systemctl restart godata.**

```
sudo systemctl restart godata
```

Para parar o serviço do Go.Data, use o comando “stop” do “systemctl” (Figura-57).

**Figura 57 - Captura de tela do systemctl stop godata.**

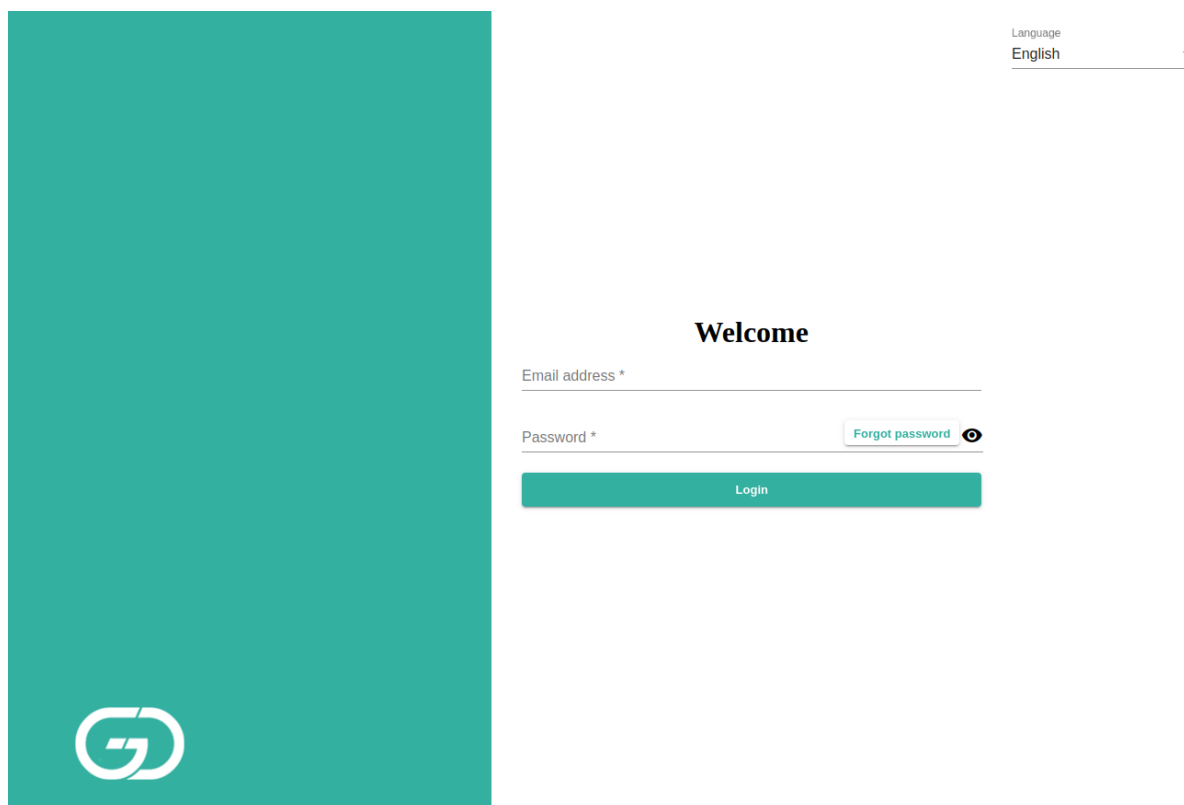
```
sudo systemctl stop godata
```

## Configurações adicionais

Com o Go.Data funcionando em ambiente Linux, seguem os passos das configurações adicionais e acesso a Backups, Logs e Documentação adicional tanto no Linux quanto no Windows.

Acesse o Go.Data com duplo clique do mouse no ícone do software no desktop (Figura-58).

**Figura 58 - Captura da tela inicial do Go.Data.**



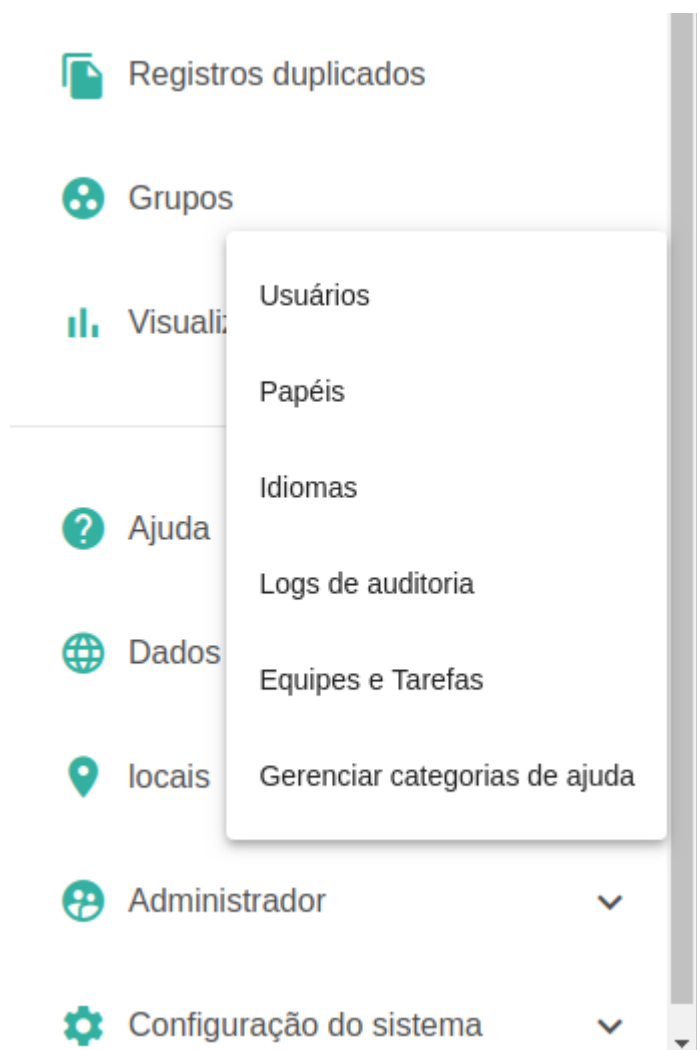
É possível alterar o idioma do Go.Data no canto superior direito da tela, no botão “Language”, para se familiarizar melhor com o software.

O primeiro acesso do Go.Data disponibiliza o usuário administrador. A partir desse usuário, é possível criar outros usuários com diferentes tipos de permissões no software. Realize o login com o usuário administrador. Em seguida, altere a senha do usuário administrador para uma senha de sua preferência.

- **E-mail:** admin@who.int
- **Senha:** admin

Acesse o menu lateral da tela e arraste o mouse em **Administrador**, depois clique em **Usuários** (Figura-59).

**Figura 59 - Captura de tela do Menu -Administrador - Usuários do Go.Data.**

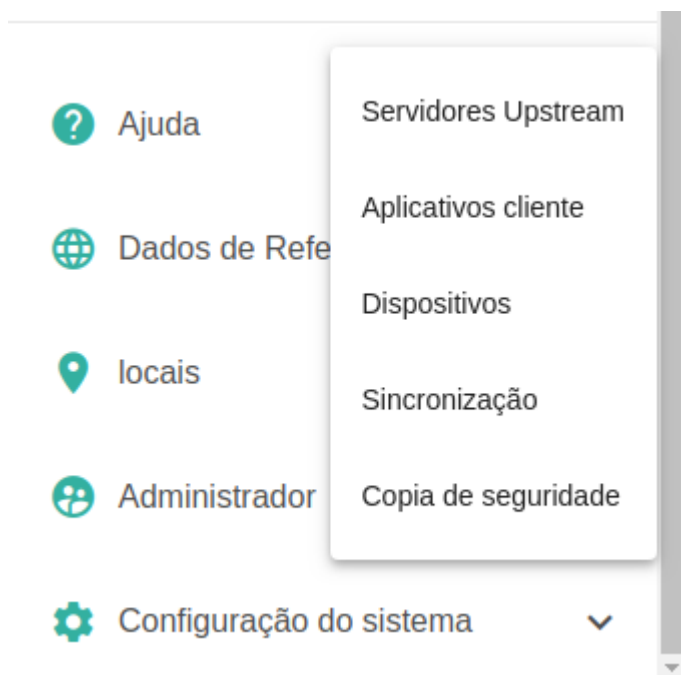


Na aba Usuários, clique no botão “Adicionar” e adicione os dados do usuário que deseja incluir no sistema. Caso o novo usuário também seja administrador, certifique-se de incluir todas as permissões de acesso ao software.

O Backup é um recurso de segurança, caso haja perdas inesperadas de dados no computador. Esse recurso consegue recuperar esses dados total ou parcialmente. Por isso, recomendamos que os Backups dos dados sejam realizados com frequência e inclusive salvos em outro computador ou HD externo ou nuvem.

Para configurar o Backup, acesse a aba **Configuração do sistema** e clique no botão **Copia de segurança** no menu lateral do Go.Data (Figura-60).

**Figura 60 - Captura de tela Menu de Configuração do Sistema do Go.Data.**



Nessa tela, o histórico dos Backups fica registrado no sistema. Para configurar o Backup Automático, clique no botão **Ações Rápidas**. Selecione a primeira opção **Cópia de seguridad restaurada** (Figura-61)

**Figura 61 - Captura de tela de Ações rápidas de Backup no Go.Data.**



Uma janela é aberta com informações do diretório onde serão salvos os backups, a frequência na qual eles foram realizados e o tempo que o Go.Data os mantém salvos no computador. Para salvar os backups automáticos, preencha os campos abaixo de acordo com suas necessidades. Não esqueça de manter ou alterar a pasta destino dos backups no campo **Local**. Em caso de dúvidas, acesse o diretório

de instalação do Go.Data no computador **/go-data/build/backups**, diretório padrão (Figura-62).

**Figura 62 - Captura de tela das Configurações de backup automático no Go.Data.**

### Configurações de backup automáticas

**Existing configuration:**  
 Description:  
 Local: /godata/v2.35.0/go-data/build/backups  
 Intervalo decopia de seguridade (horas): 24  
 Intervalo de retenção (dias): 30  
 Módulos: System Configuration, Data

---

Description

---

Local \*

/godata/v2.35.0/go-data/build/backups

---

Intervalo decopia de seguridade (horas) \*

24

---

Intervalo de retenção (dias) \*

30

---

Módulos \*

Sistema de configuração, Dados

---

CANCELAR    SALVAR

Ocasionalmente, quando o Go.Data demonstra algum erro relacionado à conexão, inconsistência do banco de dados ou do próprio sistema, uma mensagem de erro aparecerá na tela do computador. Porém, essa mensagem não decorre de informações completas. Por isso, é fundamental o acesso aos Logs do Administrador.

Em ambiente Windows, encontre os logs conforme cada módulo e tipo de instalação nas seguintes pastas:

Se você instalou apenas para seu usuário:

1.1. Logs da aplicação: {sua\pasta\de\instalação}\Go.Data\logs\app

Exemplo: C:\Users\YourUser\AppData\Roaming\Go.Data\logs\app



1.2. Logs de serviços: {sua\pasta\de\instalação}\Go.Data\resources\go-data\build\logs  
Exemplo: C:\Users\YourUser\AppData\Local\Programs\Go.Data\resources\go-data\build\logs

1.3. Logs de banco de dados: {sua\pasta\de\instalação}\Go.Data\db  
Exemplo: C:\Users\YourUser\AppData\Roaming\Go.Data\db

Se você selecionou a instalação para todos os usuários:

2.1. Logs da aplicação: {sua\pasta\de\instalação}\Go.Data\data\logs\app  
Exemplo: C:\Go.Data\data\logs\app

2.2. Logs de serviços: {sua\pasta\de\instalação}\Go.Data\bin\resources\go-data\build\logs  
Exemplo: C:\Go.Data\bin\resources\go-data\build\logs

2.3. Logs de banco de dados: {sua\pasta\de\instalação}\Go.Data\data\logs\db  
Exemplo: C:\Go.Data\data\logs\db

Em ambiente Linux, encontre os logs conforme cada módulo e tipo de instalação no seguinte diretório:

1. Logs gerais: {sua/pasta/de/instalação}/go-data/build/logs/

Nesse diretório há um arquivo denominado README e **application.log**. Para visualizar os Logs, abra um editor de texto. Os logs mais recentes se encontram no fim do arquivo.

Para finalizar esta aula, segue abaixo documentos oficiais (em inglês) e onde pode encontrá-los (Quadro-1).

**Quadro 1. Lista de documentos do Go.Data, segundo descrição e local de acesso.**

Nome do documento	Descrição	Localizado
User Guide	Guia de usuário do Go.Data, com instruções de uso.	Comunidade do Go.Data

Go.Data Implementation Guide	Guia de implementação de plano de uso, hospedamento e configurações do Go.Data.	
Go.Data IT Administrator Guide	Guia de Administrador de Infraestrutura, com informações sobre instalação, hospedagem, segurança, API, entre outros.	
Syncing the Go.Data Mobile app	Guia de sincronização do Go.Data versão Mobile com os dados de uma instância do Go.Data em computador.	
Terms of use and Software License Agreement	Termos de uso, licença e direitos autorais.	
Frequently Asked Questions (FAQ)	Documento com perguntas frequentes do Go.Data.	Referências
README	Informações do Go.Data, requisitos necessários e passo a passo de sua instalação.	Arquivo no formato .md (markdown) no diretório do Go.Data

## Indicadores de avaliação

Os indicadores são importantes ferramentas de avaliação e monitoramento de resultados e nos permitem priorizar esforços no alcance das metas. A avaliação periódica dos indicadores previne a falsa impressão de que os esforços estão ou não satisfatórios, já que o processo de entrevistas e identificação é demorado.

## Gerenciamento da investigação de caso e rastreamento de contatos

Os relatórios de avaliação podem ser produzidos a cada duas semanas e monitorados no Go.Data. Os indicadores podem ser incluídos nesses relatórios para registrar resultados da eficácia das operações de rastreamento de contatos no quadro de monitoramento e avaliação, inclusive nas atividades de resposta à Covid-19, como:

- Uso de software de rastreamento de contato especializado, como o Go.Data, para facilitar o processo de identificação, rastreamento e acompanhamento dos contatos (ECDC, 2020).
- Disponibilidade de aplicativos móveis na coleta de dados, como o Guardiões da Saúde, para apoiar no rastreamento de contatos, já que o rastreamento manual contribui em maior adesão e cobertura da população (ECDC, 2020).
- Número de investigações de casos atribuídos no período de transmissibilidade,- disponível na lista de casos.
- Número de indivíduos entrevistados no período de transmissibilidade.
- Número de investigações de casos encerradas no período ( período de tempo longo), ou seja, o desfecho e o panorama geral do surto, com a quantidade de recuperados e óbitos.
- Número e percentual de investigações de casos em que pelo menos um contato próximo foi obtido no período. A lista de contatos permite rastrear o caso de onde surgiu o contato.
- Número total de contatos provenientes do caso e investigações no período.



#### Fica a Dica!

É possível observar os contatos na tela **Contatos** no **Go.Data**, inclusive as cadeias de transmissão.

- Número de contatos que concluíram a quarentena / número total de contatos a realizar a quarentena no período. No Go.Data é possível obter esse número por meio da lista de contatos ao filtrar o campo “Situação final do acompanhamento” com o valor “Acompanhamento concluído”.
- Número de encaminhamentos para apoio social. Os contatos cadastrados devem ser orientados sobre o acesso ao apoio social.



#### Saiba Mais!

Conheça os indicadores associados à melhoria contínua da qualidade do rastreamento de contatos, baseado em porcentagem de casos clique aqui.

**Clique aqui!**

<https://www.cdc.gov/coronavirus/2019-ncov/global-covid-19/operational-considerations-contact-tracing.html>

## Medições processuais programáticas e dos resultados

Os dados analisados pela equipe de investigação de casos e rastreamento de contatos ajudarão a definir ações estratégicas de contenção do surto, seja na necessidade de capacitação técnica dos recursos humanos, seja na priorização de ações de controle nas áreas de risco onde há maior ocorrência da doença. Os relatórios podem ser produzidos regularmente e entregues a médio prazo (quinzenalmente ou mensalmente). Essas medidas processuais programáticas incluem:

- Número de investigações de casos atribuídos no período.
- Número de casos entrevistados no período.

- Número e percentual de casos entrevistados menor ou igual a 24 horas do relatório à autoridade de saúde no período.
- Número de investigações de casos encerradas no período.
- Número e percentual de casos que informaram pelo menos um contato próximo no período.
- Mediana em dias, desde o recebimento do relatório até a entrevista no período.
- Número total de contatos rastreados nas investigações de casos no período.
- Número total de contatos entrevistados/número total de contatos informado por casos no período.
- Média de contatos informados por entrevista dos casos no período.
- Número de casos que concluíram o período de isolamento/número total de casos indicados a se isolar no período.
- Número de contatos que concluíram a quarentena/número total de contatos indicados a realizar a quarentena no período.
- Número de encaminhamentos realizados ao suporte social.
- Número de encaminhamentos realizados para consulta clínica.

As medições programáticas dos resultados incluem:

- Número de casos entrevistados/número de investigações de casos.

- Número de contatos testados para SARS-COV-2/Número de contatos entrevistados.
- Número e percentual de novos casos confirmados decorrentes de contatos em quarentena.
- Número de contatos que estiveram em quarentena como resultado do rastreamento de contatos.
- Número e percentual de casos que concluíram o período de auto-isolamento.
- Número de contatos que concluíram 14 dias de quarentena/contatos notificados.

## Conclusão

Nesta aula você aprendeu a instalar o Go.Data tanto em nível local, como no servidor. Também aprendeu o passo a passo das configurações básicas para o melhor desempenho da ferramenta, assim como as configurações adicionais que devem ser executadas no programa e os indicadores que os softwares nos fornecem na investigação do surto.