

Pré-requisitos

Este documento descreve os pré-requisitos para a instalação do MD2 Quality Manager, Portal do Titular.

- MD2 Quality Manager
- Portal do Titular
- MD2 Repo PostgreSQL
- Pós-instalação dos requisitos
 - Servidor MD2 Quality Manager
 - MD2 Repo PostgreSQL
 - Servidor Portal do Titular

MD2 Quality Manager

Pré-requisitos para configuração do servidor que irá receber a aplicação Quality Manager:

- Sistema Operacional:
 - Red Hat Enterprise Linux 7.9
 - Suse Linux Enterprise Linux 12 SP 4
 - Suse Linux Enterprise Linux 15 SP 3
 - Ubuntu 20.04
- CPU:
 - 2 Cores
- Memória:
 - 8Gb
- Disco:
 - 100 Gb
- Softwares:
 - Openjdk 11
 - Apache Tomcat 9.0.43
 - PostgreSQL 12*

*O PostgreSQL pode ser instalado em um servidor apartado desde que seja configurado para receber conexões remotas.

Portal do Titular

Pré-requisitos para configuração do servidor que irá receber a aplicação Portal do Titular:

Aplicação deve ser hospedada em um Web Server que esteja disponível para acesso por usuários externos à rede (DMZ), como os clientes da empresa.

- Sistema Operacional:
 - Red Hat Enterprise Linux 7.9
 - Suse Linux Enterprise Linux 12 SP 4
 - Suse Linux Enterprise Linux 15 SP 3
 - Ubuntu 20.04
 - Oracle Linux 7.9
- CPU:
 - 2 Cores
- Memória:
 - 1Gb
- Disco:
 - 50 Gb
- Softwares:
 - Openjdk 11
 - Apache Tomcat 9.0.43

MD2 Repo PostgreSQL

Em caso do repositório PostgreSQL for ficar em um servidor apartado do servidor em que o MD2 Quality Manager será instalado, deve-se seguir os requisitos abaixo:

- CPU:
 - 2 Cores
- Memória:
 - 8Gb
- Disco:
 - 100 Gb
- Softwares:
 - PostgreSQL 12

Pós-instalação dos requisitos

Após a instalação dos pré-requisitos, será necessário fazer algumas configurações nos softwares instalados para o MD2 Quality Manager e Portal do Titular funcionarem perfeitamente.

Servidor MD2 Quality Manager

Após a instalação dos pré-requisitos, será necessário efetuar alguns procedimentos para o MD2 Quality Manager funcionar perfeitamente:

A MD2 disponibilizará um pacote com os artefatos para a configuração da aplicação do MD2 Quality Manager e Portal do Titular. Neste pacote estarão presentes os seguintes arquivos:

- qm_static.tar.gz
- qualityManager-prj.war
- qualitymanager_DB.psql.gz
- Mensageria_backend
- Mensageria_frontend
- portal-titular-ws.war
- configPortal.properties
- configPortal.js

Neste documento, iremos passar pelos os procedimentos que deverão ser feitos com os arquivos relacionados ao **MD2 Quality Manager**.

- **qm_static.tar.gz**

O qm_static se trata do diretório onde ficarão armazenados todos os anexos que serão feitos dentro da aplicação do Md2 Quality Manager.

O arquivo qm_static.tar.gz deverá ser descompactado dentro do diretório **/opt** conforme está sendo ilustrado nos prints abaixo:

```
sudo tar -xvzf qm_static.tar.gz
```

```
sudo mv /tmp/qm_static /opt
```

```
sudo chown -R tomcat:tomcat /opt/qm_static/
```

```
root@qm-hom:/opt# pwd
/opt
root@qm-hom:/opt# ls -ll
total 8
```

Após descompactar e mover a pasta qm_static para o diretório /opt, deve-se editar o arquivo de configuração do Tomcat inserindo a linha "ReadWritePaths=/opt/qm_static/" conforme o print abaixo:

```
sudo vim /lib/systemd/system/tomcat9.service
```

```
systemctl daemon-reload
```

```
Group=tomcat
```

```
systemctl stop tomcat9.service
```

```
ReadOnlyDirectories=/etc/tomcat9/Catalina/  
CacheDirectory=/etc/tomcat9  
CacheDirectory=/var/lib/tomcat9  
QualityManagerPrj=prj.war  
ProtectSystem=strict  
ReadWritePaths=/etc/tomcat9/Catalina/  
ReadWritePaths=/var/lib/tomcat9/webapps/  
ReadWritePaths=/var/log/tomcat9/  
ReadWritePaths=/opt/qm_static
```

Após executar os passos anteriores, deve-se mover o arquivo qualityManager-prj.war para o diretório de Tomcat9 do serviço.

```
sudo cp qualityManager-prj.war /var/lib/tomcat9/webapps/
```

```
systemctl start tomcat9.service
```

• Mensageria_backend

Deve-se acessar o diretório para onde os artefatos da instalação foram copiados e transferir os arquivos do backend para uma outra pasta. No caso deste documento, vamos utilizar a pasta /var/lib/Mensageria como path do Mensageria Backend.

```
sudo cp -r Mensageria_backend/ /var/lib/Mensageria
```

Acessando o diretório do Mensageria Backend e manipulando os artefatos:

```
cd /var/lib/Mensageria
```

```
sudo mv rewrite.config /var/lib/tomcat9/conf/Catalina/localhost/
```

```
chown -R tomcat:tomcat /var/lib/tomcat9/conf/Catalina/localhost/rewrite.config
```

No arquivo **mensageria.service** é indicado o path do arquivo de configuração do programa, o mesmo deve apontar para o local onde foi colocado o arquivo “application.properties” e “servico-mensageria.jar”.

```
sudo vim mensageria.service
```

```
[Unit]
Description=Serviço de mensageria
Requires=network.target
After=syslog.target network.target

[Service]
Type=simple
WorkingDirectory=/var/lib/Mensageria/
ExecStart=/usr/bin/java -jar /var/lib/Mensageria/servico.mensageria.jar --spring.config.location=/var/lib/Mensageria/application.properties
ExecStop=/usr/bin/java -jar /var/lib/Mensageria/servico.mensageria.jar stop
User=root

[Install]
WantedBy=default.target
```

```
sudo mv mensageria.service /etc/systemd/system/
```

Será necessário editar o arquivo **server.xml** e inserir o trecho abaixo:

```
<Valve className="org.apache.catalina.valves.rewrite.RewriteValve" />
```

```
systemctl restart tomcat9.service
```

```
<Valve className="org.apache.catalina.valves.rewrite.RewriteValve" />
```

```
systemctl start mensageria.service
```

```
systemctl status mensageria.service
```

```
root@qm-hom:/opt# systemctl status mensageria.service
● mensageria.service - Serviço de mensageria
   Loaded: loaded (/etc/systemd/system/mensageria.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-08-03 15:40:32 -03; 4s ago
     Main PID: 538724 (java)
        Tasks: 18 (limit: 9442)
       Memory: 238.3M
      CGroup: /system.slice/mensageria.service
              └─538724 /usr/bin/java -jar /var/lib/Mensageria/mensageria.jar --spring.config.location=application.properties
```

• Mensageria_frontend

Deve-se acessar o diretório para onde os artefatos da instalação foram copiados e transferir os arquivos do Frontend para o diretório **/var/lib/tomcat9/webapps** seguindo os seguintes comandos:

```
sudo mkdir /var/lib/tomcat9/webapps/conf
```

```
sudo mv configPortal.json /var/lib/tomcat9/webapps/conf
```

```
sudo mv mensageria/ /var/lib/tomcat9/webapps
```

```
chown -R tomcat:tomcat /var/lib/tomcat9/webapps/conf/
```



```
chown -R tomcat:tomcat /var/lib/tomcat9/webapps/mensageria/
```

Edição do arquivo configPortal.json inserindo o endereço do serviço mensageria backend:

```
vim /var/lib/tomcat9/webapps/conf/configPortal.json
```

```
{
  "production": true,
  "config": {
    "portal": {
      "apiURLBase": "http://<IP ou endereço DNS do servidor>/servico-
mensageria",
      "clientId": "angular-app",
      "clientSecret": "@321"
    }
  }
}
```

```
systemctl restart tomcat9.service
```

```
"config": {
  "portal": {
```

MD2 Repo PostgreSQL

Após a instalação do PostgreSQL 12 será necessário fazer algumas configurações para que o servidor do MD2 Quality Manager consiga conectar o banco de dados:

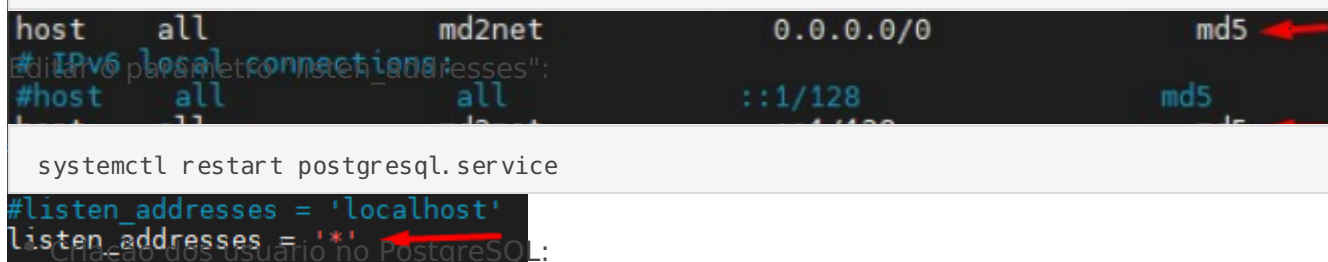
- Permitindo conexões remotas editando o arquivo pg_hba.conf e postgresql.conf:

```
vim /etc/postgresql/12/main/pg_hba.conf
```

Inserir:

```
host all md2net 0.0.0.0/0 md5
host all md2net ::1/128 md5
```

```
vim /etc/postgresql/12/main/postgresql.conf
```



The screenshot shows the postgresql.conf file with the following configurations:
listen_addresses = 'localhost' (commented out)
listen_addresses = '*' (uncommented, with a red arrow pointing to it)
host all md2net 0.0.0.0/0 md5 (with a red arrow pointing to md5)
host all md2net ::1/128 md5 (with a red arrow pointing to md5)

```
systemctl restart postgresql.service
```

```
sudo -i -u postgres psql -c "create user md2net with encrypted password 'md2net2018';"
```

```
sudo -i -u postgres psql -c "alter user md2net with superuser;"
```

```
sudo -i -u postgres psql -c "create user mdm with encrypted password 'md2net2018';"
```

- Criação do database “qualitymanager”:

```
sudo -i -u postgres psql -c "CREATE DATABASE "qualitymanager" WITH OWNER md2net TEMPLATE =  
template0  
ENCODING = 'UTF8' LC_COLLATE = 'pt_BR.UTF-8' LC_CTYPE = 'pt_BR.UTF-8';"
```

- Criação do database “mensageria_qm”:

```
sudo -i -u postgres psql -c "CREATE DATABASE 'mensageria_qm' WITH OWNER md2net TEMPLATE =  
template0  
ENCODING = 'UTF8' LC_COLLATE = 'pt_BR.UTF-8' LC_CTYPE = 'pt_BR.UTF-8';"
```

- Criação do arquivo .pgpass:

```
cd ~
```

```
vim .pgpass
```

Inserir:

```
127.0.0.1:5432:qualitymanager:md2net:md2net2018
```

```
chmod 600 .pgpass
```

- Execução do script baseline:

```
gunzip -c qualitymanager_DB.gz | psql -h 127.0.0.1 -U md2net qualitymanager
```

Servidor Portal do Titular

Após a instalação dos pré-requisitos, será necessário efetuar alguns procedimentos para o Portal do Titular funcionar perfeitamente.

Neste documento, iremos passar pelos os procedimentos que deverão ser feitos com os arquivos relacionados ao **Portal do Titular**:

- **configPortal.properties**

Este arquivo deverá estar configurado da seguinte forma:

```
sudo mv configPortal.properties /var/lib/tomcat9/conf
config.portal.port=(porta que o Quality Manager utiliza)
config.portal.admin.enabled=true

chown -R tomcat:tomcat /var/lib/tomcat9/conf/configPortal.properties

config.portal.passFullAccess=(Senha criptografada do usuário autenticado de sistema)
config.portal.passPartialAccess=(Usuário não autenticado de sistema)
config.portal.passPartialAccess=(Senha criptografada do usuário não autenticado de sistema)
config.portal.admin.enabled=true

sudo mv configPortal.js /var/lib/tomcat9/conf

config.portal.enabledConsultaBasica=true

chown -R tomcat:tomcat /var/lib/tomcat9/conf/configPortal.js

config.portal.labelConsultaOcorrencia=(Label de consulta de ocorrencia)
config.portal.enabledConsultaOcorrencia=true
config.portal.reCaptcha.publicKey=(Chave pública do reCaptcha, caso utilize)
config.portal.reCaptcha.secretKey=(Chave privada do reCaptcha, caso utilize)

systemctl stop tomcat9.service

config.portal.secure.authenticationHost=(host da API do cliente que retorna o CPF)
config.portal.secure.authenticationEndpoint=FrontbackClient

sudo cp portal-titular-ws.war /var/lib/tomcat9/webapps

systemctl start tomcat9.service
```