VM Quality Manager

A VM MD2 Quality Manager hospeda o Web Server Tomcat, permitindo a publicação e disponibilização da aplicação e o repositório, no SGBD Postgres.

- Introdução
- Acessos
- Configurando sistema operacional, firewall e repositório
- Configurando Tomcat para utilizar porta 80, 443
- Configurando para que aplicação QM fique no contexto raiz
- Conectando remotamente (cliente externo) ao repositório
- Configurando a aplicação QM para conectar em um banco de dados PostgresSQL remoto
- Configurando o Web Server Tomcat para não apresentar detalhes do servidor (fingerprint)
- Setup inicial de usuários e hierarquia
- Segurança - autenticação recaptcha
- Segurança – Integração LDAP
- Configurando Conexão de Banco de dados para módulos MDM Governance
- Configurando Conexão de Banco de dados para acesso ao ELK
- Back-up
- Backup de Virtual Machine (VM) para Oracle VirtualBox
- Gerenciar WebServer Tomcat
- Gerenciar o banco de de dados Postgres
- Acompanhamento de logs via sistema
- Acompanhamento de logs via aplicação
- Migração Legado GED

Introdução

A VM MD2 Quality Manager hospeda o Web Server Tomcat, permitindo a publicação e disponibilização da aplicação e o repositório, no SGBD Postgres.

O repositório da aplicação fica no database chamado “**qualitymanager**”. Este armazena todos os dados referentes a aplicação, como configurações, acessos, treinamentos, processos, ocorrências, tarefas e etc. A exceção são para os “documentos anexos”, funcionalidade disponível em alguns módulos. Estes anexos ficam armazenados no diretório “**/opt/qm_static/**”.

Por default, o serviço do banco de dados Postgres está configurado para receber conexões remotas. A recomendação é que fique liberada somente para facilitar a validação da instalação.

O Web application ARchive (.war), aplicação MD2 Quality Manager, é armazenado no diretório do Web Server (Catalina Home) “**/var/lib/tomcat9/webapps/**”.

Acessos

A VM do QualityManager, por padrão, contém 2 usuários:

Usuário	Senha	Obs.
root	NTzXpzZNg7Pv5BB6sCbK	Não tem permissão para conectar via SSH.
md2net	MwbURVeWAvrwGg6xncKc4h4P6n	i. Restrição para executar somente comandos de administração dos serviços Tomcat e Postgres. ii. Não tem permissão para desligar ou reiniciar a VM.

Recomenda-se a troca das senhas default da instalação.

A VM do ElasticSearch, por padrão, contém os usuários:

Usuário	Senha	Obs.
root	zGrHjh9qkCA9guzbg	Não tem permissão para conectar via SSH.
md2net	fuYFvMuyz2xHY4Uaap6g	
elastic	md2@qm@md2	Usuário para autenticar na aplicação, não do S.O.
kibana	md2@qm@md2	Usuário para autenticar na aplicação, não do S.O.

Configurando sistema operacional, firewall e repositório

Configurando

Sistema Operacional, firewall e repositório

Configurando a rede:

A VM MD2 Quality Manager, por padrão, está configurada com IP fixo e considerando a interface de rede “ens33”. Procure o administrador de rede para fazer a configuração inicial adequada.

Desabilitando conexão remota ao repositório:

A VM MD2 Quality Manager tem o Uncomplicated Firewall (UFW) habilitado. Após a instalação, recomenda-se desabilitar a possibilidade de conectar remotamente ao banco de dados “qualitymanager”. Para isso, deve-se executar os comandos:

```
$ su -
$ vi /etc/ufw/before.rules
```

Comentar a linha marcada abaixo:

*nat
:PREROUTING ACCEPT [0:0]
-I PREROUTING -i ens33 -p tcp --dport 49159 -j DNAT --to-destination 127.0.0.1:5432
COMMIT

reboot

Configurando SSL/HTTPS:

Algumas instalações podem requerer a habilitação de conexão segura (SSL/HTTPS). Para isso, necessita-se de um certificado (.cer) e este ser importado em um javakeystore (.jks). Comandos para configuração:

```
$ su -
$ vi /etc/tomcat9/server.xml
```

E insira o trecho abaixo:

```
 <Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keystoreFile="path/javakeystore.jks"
           keystorePass="SenhaDoKeystore"
           keyPass="SenhaDaChave"
 />
```

Atenção que o path do keystore é baseado a partir da raiz do Tomcat (Catalina Home). As senhas são as utilizadas no momento da criação do javakeystore.

Logo após do trecho já existente:

```
<Connector port="8090" protocol="HTTP/1.1"
               connectionTimeout="20000"
               URIEncoding="UTF-8"
               redirectPort="8443" />
```

Configure a constraint de segurança no web.xml. Antes da tag “</webapp>”, bem no final do arquivo, insira o trecho abaixo:

```
    # sudo vi /var/lib/tomcat9/conf/web.xml


    <security-constraint>
         <web-resource-collection>
             <web-resource-name>Entire Application</web-resource-name>
             <url-pattern>/*</url-pattern>
         </web-resource-collection>
    <!-- auth-constraint goes here if you require authentication -->
          <user-data-constraint>
              <transport-guarantee>CONFIDENTIAL</transport-guarantee>
          </user-data-constraint>
```

```
</security-constraint>
```

Reinicie o servidor:

```
$ reboot
```

Configurando Tomcat para utilizar porta 80, 443

Por padrão a plataforma Unix permite que apenas o root possa utilizar portas inferiores a 1024. Para fazer a configuração do Tomcat na porta 80 e/ou 443, precisa-se instalar o “Authbind”.

Instalar o Authbind:

```
$ sudo apt-get install authbind
```

Configurar o bind:

```
$ sudo touch /etc/authbind/byport/80
$ sudo chmod 500 /etc/authbind/byport/80
$ sudo chown tomcat9 /etc/authbind/byport/80
```

Confirme se o usuário que executa o serviço é tomcat9 ou tomcat para executar o comando acima corretamente.

Vá no arquivo “tomcat9” e descomente o parâmetro authbind e configure como “yes”.

```
$ sudo vi /etc/default/tomcat9
```

Configurando para que aplicação QM fique no contexto raiz

Para que aplicação MD2 QualityManager fique exposta sem contexto, por exemplo:

```
“http: //nome_maquina: 8080/”
```

Ao invés de ter que digitar a URL completa, por exemplo:

```
“http: //nome_maquina: 8080/qualityManager-prj/”
```

Precisa-se configurar o arquivo “server.xml”:

```
$ sudo vi /var/lib/tomcat9/conf/server. xml
```

Adicionando a linha abaixo:

```
<Context path="" docBase="qualityManager-prj" reloadable="true" />
```

Dentro da tag “Host”, ficando algo como:

> <Host name="localhost" appBase="webapps"
> unpackWARs="true" autoDeploy="true" deployOnStartup="true">
> <Context path="" docBase="qualityManager-prj" reloadable="true"/>
> <!-- <Context path="ROOT" docBase="ROOT" reloadable="true">
> <WatchedResource>WEB-INF/web.xml</WatchedResource>
> </Context>-->
> <!-- SingleSignOn valve, share authentication between web applications
> Documentation at: /docs/config/valve.html -->
> <!--
> <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
> -->

<!-- Access log processes all example.
Documentation at: /docs/config/valve.html
Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
prefix="localhost_access_log." suffix=".txt"
pattern="%h %l %u %t "%r" %s %b" />
</Host>

Conectando remotamente (cliente externo) ao repositório

Algumas instalações e cenários de solução podem precisar conectar com o banco de dados Postgres, repositório do QualityManager.

Por padrão de instalação, o acesso é impedido. Para fazer a liberação precisa-se ajustar 3 parâmetros:

1. Firewall. Liberar o acesso externo através dos comandos:

```
ufw status
ufw allow <porta postgres>
```

Configuração do Postgresql:

```
/etc/postgresql/9.5/main/postgresql.conf
linha "listen_address='*'"
```

Configuração de autenticação de cliente no Postgresql:

```
/etc/postgresql/9.5/main/pg_hba.conf
linha "host all md2net 0.0.0.0/0 md5"
```

Configurando a aplicação QM para conectar em um banco de dados PostgresSQL remoto

O QualityManager depende de uma conexão de banco de dados PostgresSQL instalado junto com a solução. Mas arquitetura do sistema permite conectar o aplicativo a um banco PostresSQL instalado em outro servidor e com parâmetros de segurança diferentes ao default entregues com a solução. Para isto poderá deberá ser adicionado um arquivo de propriedades de configuração na pasta de **lib** da instalação tomcat (padrão é /usr/share/tomcat9/lib/). Este arquivo so pode ser manipulado por especialistas com conhecimento da arquitetura do sistema. O arquivo de configuração deverá possuir o nome **qm.app.properties**. O arquivo pode ser criado e editado por qualquer editor de texto de deve seguir o padrão de arquivo de propriedades java, contendo em cada linha uma propriedade no formato [nome do parametro]=[vallor]. Para o caso das propriedades de conexão de banco de dados, a seguir um exemplo de propriedades que redirecionam a conexão do banco de dados a um endereço ip diferente do localhost e propriedades de banco, usuário e esquema diferente do default.

```
qm.app.db.hibernate.connection.url=jdbc:postgresql://192.168.0.187:5432/dbcorp
qm.app.db.hibernate.connection.username=qmuser
qm.app.db.hibernate.connection.password=8OrGW5Z1U5fajb7lubrgvA==
qm.app.db.hibernate.default_schema=qualitymanger
```

A informação de senha deve ser criptografa mediante a utilização da ferramenta PortalPass. O arquivo pode ser adicionado em qualquer fase da instalação ou atualização do aplicavo, sendo necessário a renicialização do servidor Tomcat para as informações serem efetivadas.

Configurando o Web Server Tomcat para não apresentar detalhes do servidor (fingerprint)

O fato do Web Server Tomcat apresentar e disponibilizar detalhes do servidor, como versão do sistema operacional e versão do Tomcat, pode ser entendido como uma vulnerabilidade.
Para impedir que os usuários, ao acessarem uma página que não exista, e tenha acesso as essas informações, deve-se modificar o arquivo "server.xml" do Tomcat, localizado em "/var/lib/tomcat9/conf/server.xml" e incluir o seguinte trecho abaixo dentro da tag "Host":

```
            # trecho acima
            <Valve className="org.apache.catalina.valves.ErrorReportValve"
                   showReport="false"
                  showServerInfo="false"/>


          </Host>
        </Engine>
      </Service>
    </Server>
```

Setup inicial de usuários e hierarquia

Toda instância do Quality Manager será formado um usuário “admin full” e um usuário “admin” por unidade. O “admin full” será exclusivamente responsável pela criação de unidades, por toda estrutura inicial, departamento (admin), setor (admin) e usuário (admin) e liberação de permissão inicial.

Abaixo estão dispostos todos os passos detalhados necessários por toda a configuração inicial de uma unidade no Quality Manager.

(Etapa 1) - Usuário: Admin Full

1. Logar na solução MD2 Quality Manager;
2. Acessar configurações;
3. Acessar estrutura organizacional;
4. Acessar unidade;
5. Criar unidade;
6. Preencher todas as informações necessárias;
7. Salvar alterações;
8. Acessar departamento;
9. Criar departamento (Nome sugerido: Departamento Administrador - “Nome da Unidade”);
10. Salvar alterações;
11. Acessar setor;
12. Criar setor (Nome sugerido: Setor Administrador - “Nome da Unidade”);
13. Salvar alterações;
14. Realizar logoff e logon no sistema (Passo temporário necessário para atualização de setores no sistema);
15. Acessar configurações;
16. Acessar estrutura organizacional;
17. Acessar usuários;
18. Criar novo usuário;
19. Atribuir grupo de acesso “Master”;
20. Atribuir o setor criado no passo 12;
21. Preencher informações obrigatórias, colocando o e-mail do responsável por este usuário pois todas as definições de senha serão enviados para o de referência (Nome sugerido: Admin - “Nome da Unidade”);
22. Salvar alterações;
23. Realizar logoff no sistema;

(Etapa 2) - Usuário: Admin (criado na etapa 18)

1. Logar na solução MD2 Quality Manager;
2. Acessar configurações;
3. Acessar estrutura organizacional;
4. Acessar grupo;
5. Criar novo grupo;
6. Preencher as informações necessárias (Nome sugerido: Master - “Nome da Unidade”);
7. Salvar alterações;
8. Acessar associação de módulos aos grupos de acesso;
9. Selecionar o grupo criado no passo 5;
10. Ativar todas as opções para todos os módulos (Permissão de acesso e gravação) e definir o nível de acesso registros como “Todos os registros”;
11. Salvar alterações;
12. Realizar logoff no sistema;

Usuário: Admin Full

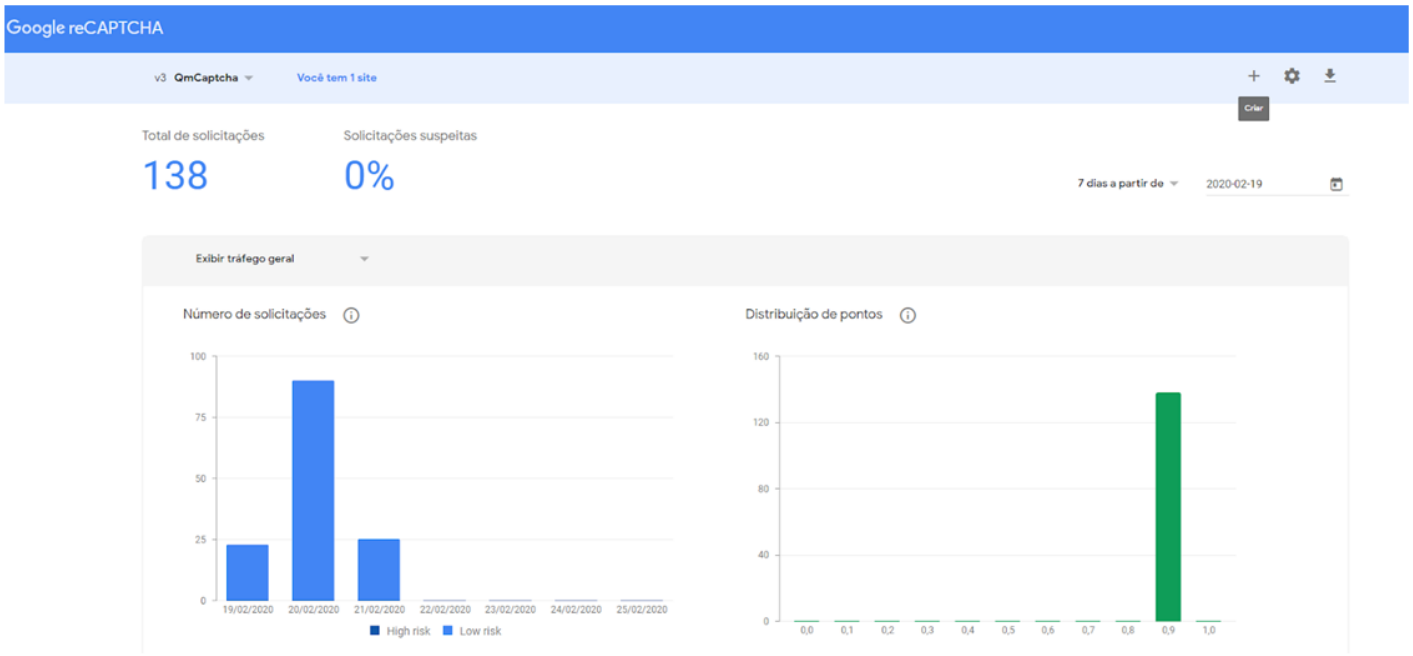
1. Logar na solução MD2 Quality Manager;
2. Acessar configurações;
3. Acessar estrutura organizacional;
4. Acessar grupo;
5. Editar o grupo “Master” (Referente ao grupo de acesso do admin full);
6. Colocar o grupo criado no passo 5 da Etapa 2 como grupo subordinado;
7. Salvar alterações;
8. Acessar usuários;
9. Editar usuário admin criado no passo 18 da etapa 1, e atribuir o grupo de acesso criado no passo 5 da etapa 2;
10. Salvar alterações
11. Realizar logoff no sistema;

Realizados todos os passos descritos, a partir do usuário admin da unidade, o sistema estará pronto para todas as configurações e cadastros necessários.

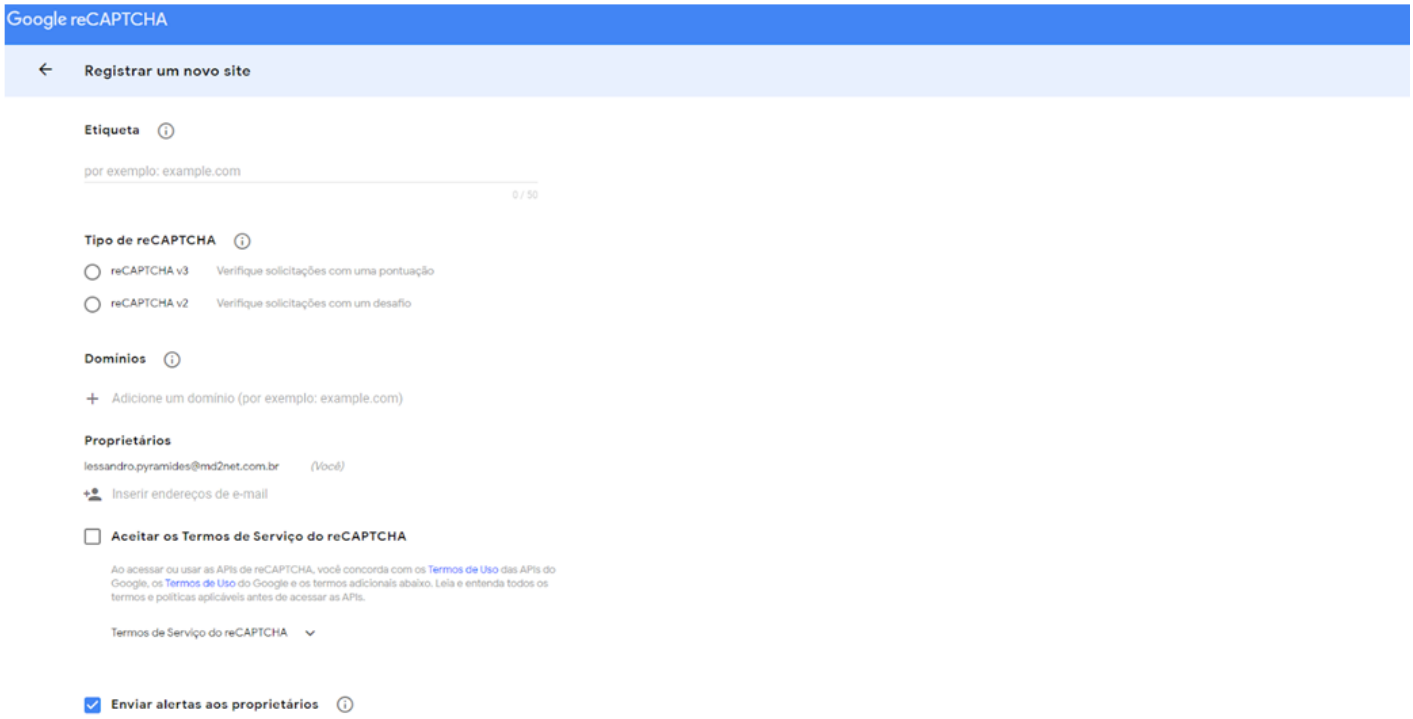
Segurança - autenticação recaptcha

Uma vez que o Quality Manager está disponível para acesso, é possível configurar um método de segurança adicional para a autenticação dos usuários. Essa segurança é realizada através do serviço reCaptcha do Google, onde o usuário informa um par de chaves (pública e privada), e o sistema realiza a autenticação utilizando o serviço.

As chaves são geradas diretamente no site do *Google, em https://www.google.com/recaptcha/admin.

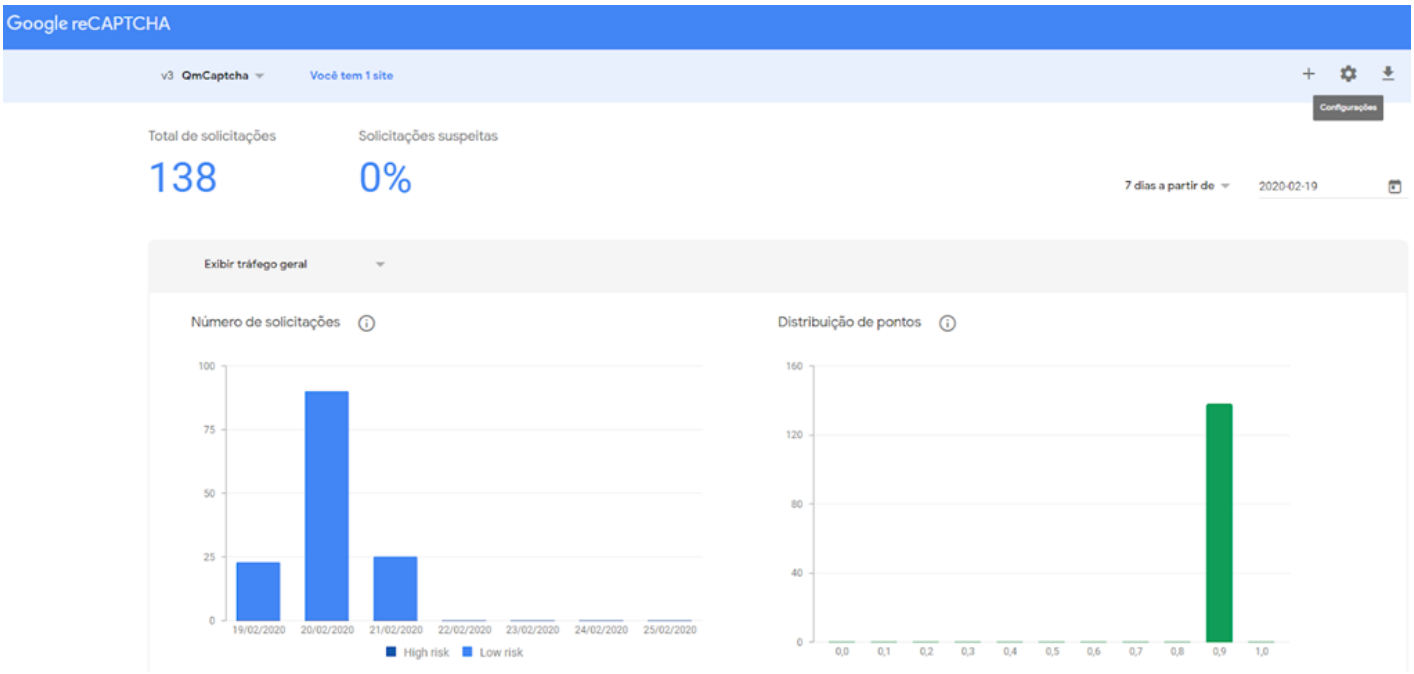


Clique em “criar”, e a página de criação do serviço será acionada.

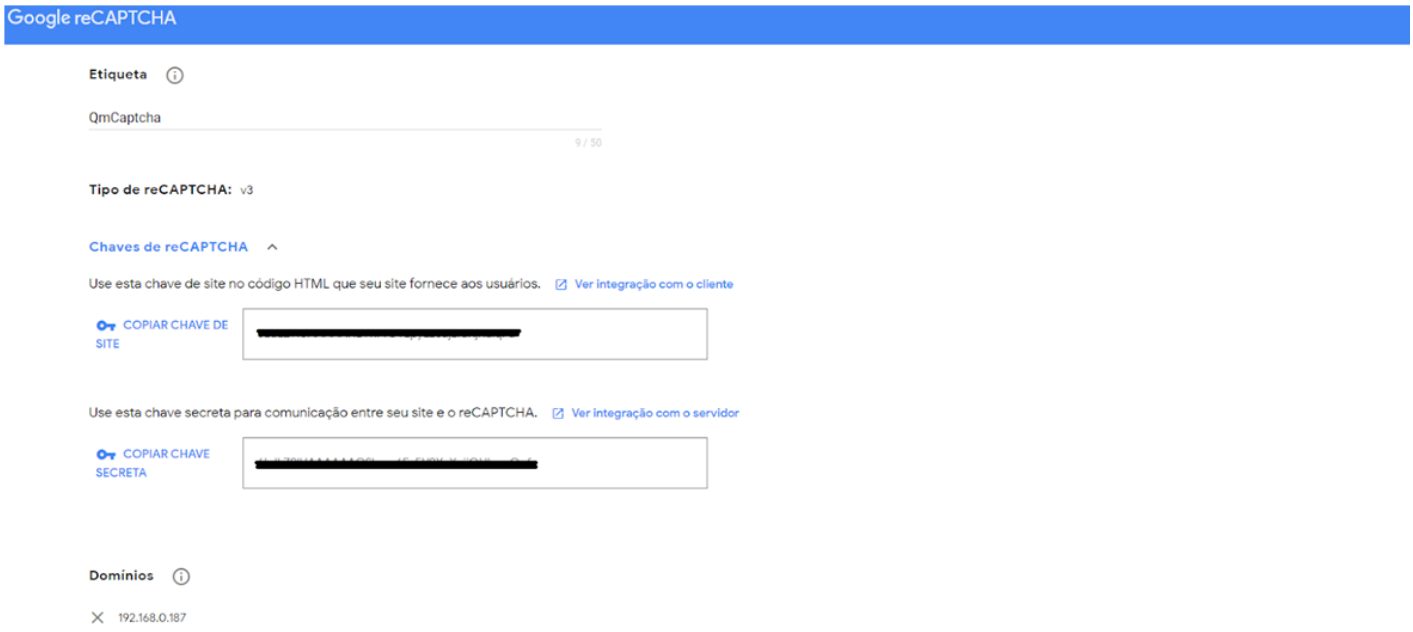


Na etiqueta informe um nome para o reCaptcha, por exemplo, “QMCaptcha”.
Selecione a versão 3 do serviço, e adicione o domínio onde o Quality Manager está sendo executado, ex: http://www.md2qualitymanager.com. Marque a opção “Aceitar os termos do serviço reCaptcha”.

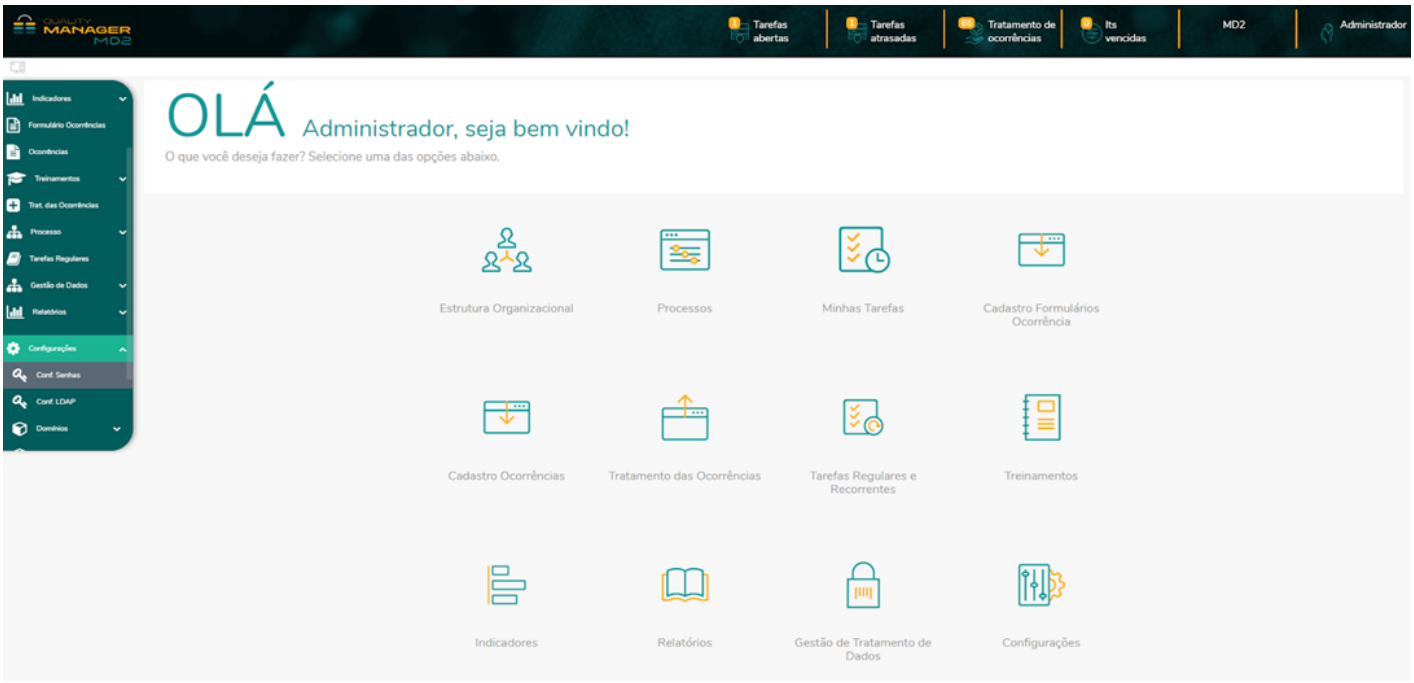
Pronto, o serviço reCaptcha foi criado no site do Google, sendo importante agora resgatarmos as chaves geradas. Para isso, clique na opção “Configurações”, e as chaves estarão disponíveis.



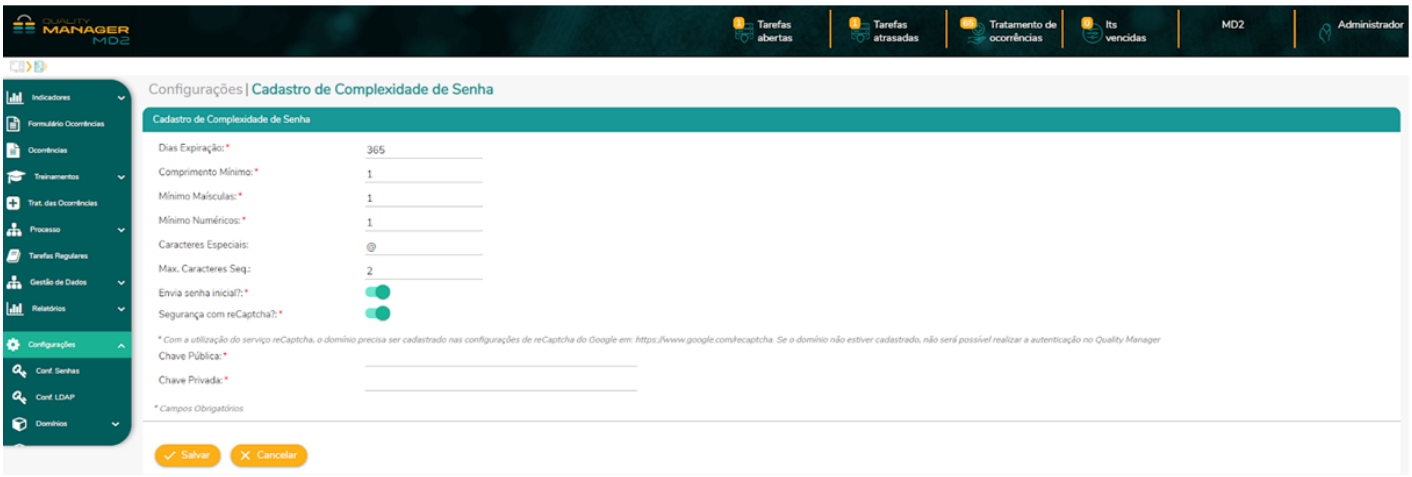
Copie as chaves para que posteriormente sejam informadas nas configurações do Quality Manager.



As chaves são informadas no menu de “Configuração de Senhas”, onde somente o administrador da ferramenta terá acesso.



Na interface de “Configuração de Senhas”, ative a opção “Segurança com reCaptcha”, e informa a chave pública e privada geradas no serviço reCaptcha. Os dois campos são obrigatórios caso a opção “Segurança com reCaptcha” esteja ativa, caso contrário, o Quality Manager realizará a autenticação sem a ** utilização do serviço.



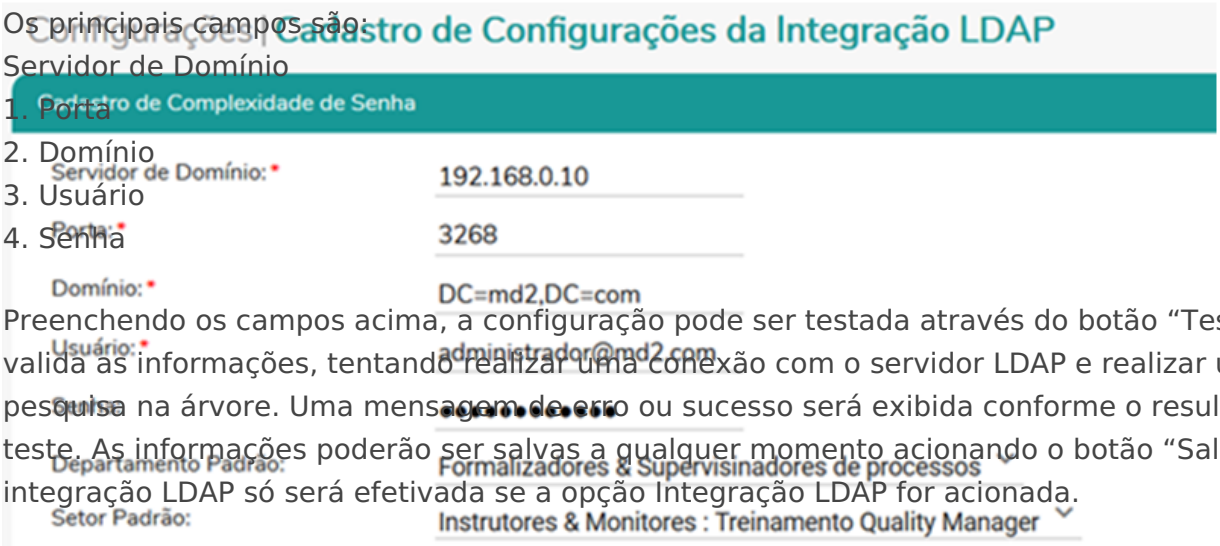
* É obrigatório ter ao menos uma conta do Google para utilização do serviço.
** Para realizar autenticação utilizando o serviço reCaptcha, é indispensável que o usuário tenha conexão com a internet.

Segurança – Integração LDAP

O Quality Manager possui duas formas de autenticar e autorizar um usuário do sistema.
São elas:
1. Autenticação standalone. Através dos usuários cadastrados na própria base de dados. Essa é a configuração padrão.
2. Autenticação LDAP.

O serviços de diretórios compatíveis são:
1. Active Directory (plataforma Windows)

Para habilitar a integração LDAP deve-se acessar ao sistema usando um usuário “Admin Full” ou um usuário com permissão de acesso e escrita na funcionalidade “Configuração LDAP” do menu “Configurações”. No formulário de Cadastro de Configurações da Integração LDAP deve ser informado as seguintes informações:
1. Servidor de Domínio: Endereço IP ou DNS do servidor LDAP
2. Porta: TPC de acesso ao serviço
3. Domínio: Nome do domínio LDAP no formato DC=[Nome Empresa],DC=[Sufixo, ex:com] ,DC=[Sufixo, ex:br].
4. Usuário: Usuário LDAP com permissão de pesquisa na árvore de domínio
5. Senha: Senha do usuário LDAP informado no campo anterior
6. Departamento Padrão: Departamento padrão do novo usuário a ser criado caso não exista na base local
7. Setor Padrão: Setor padrão do novo usuário a ser criado caso não exista na base local
8. Integração LDAP: Habilita ou não a integração LDAP com sistema Quality Manager



Os principais campos são:
Servidor de Domínio
1. Porta
2. Domínio
3. Usuário
4. Senha

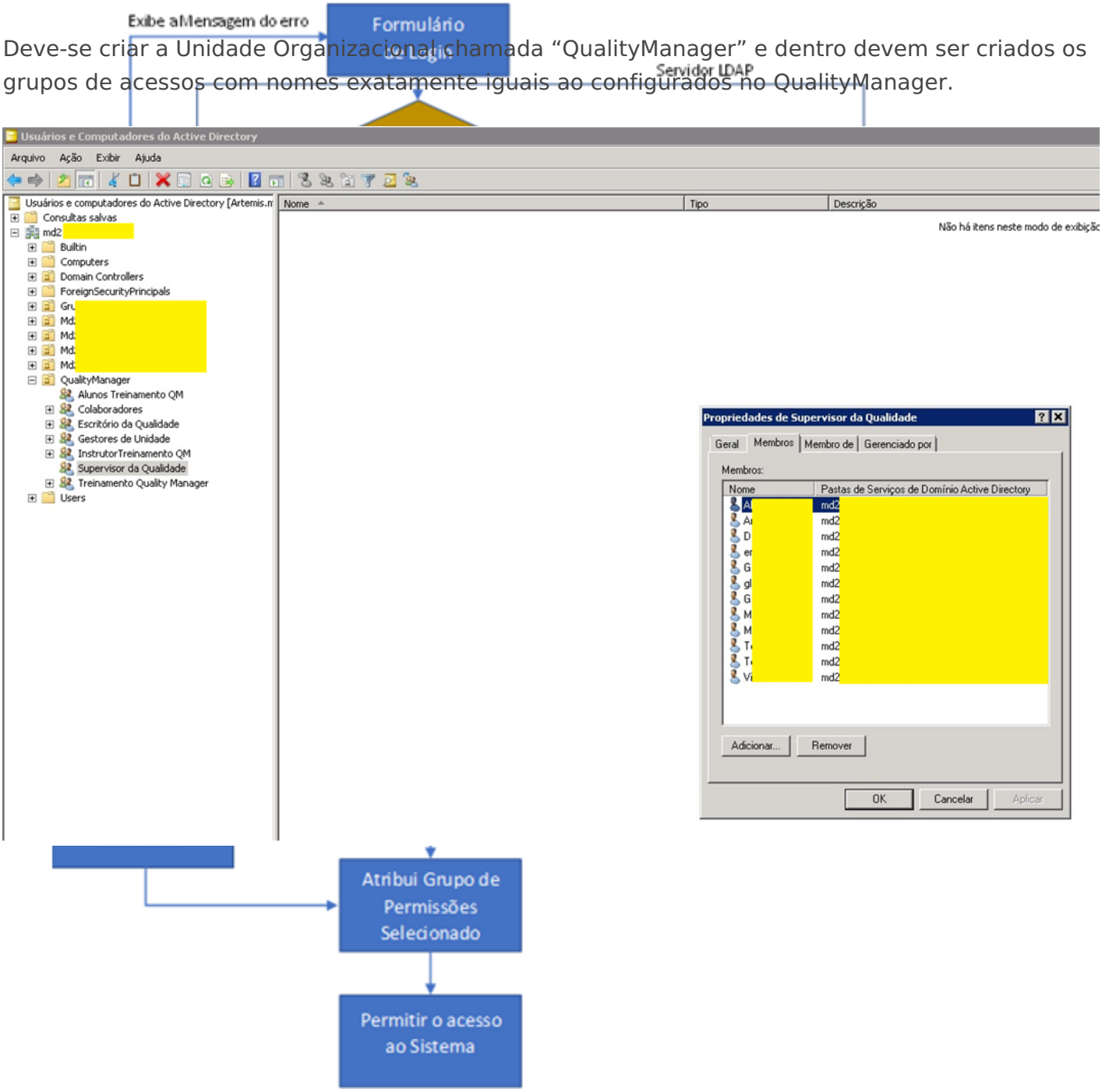
Preenchendo os campos acima, a configuração pode ser testada através do botão “Teste”. Este valida as informações, tentando realizar uma conexão com o servidor LDAP e realizar uma pesquisa na árvore. Uma mensagem de erro ou sucesso será exibida conforme o resultado do teste. As informações poderão ser salvas a qualquer momento acionando o botão “Salvar”, mas a integração LDAP só será efetivada se a opção Integração LDAP for acionada.

Uma vez habilitada a Integração LDAP, a tela de login envia os dados (usuário e senha) ao servidor LDAP configurado. Este servidor se encarregará de autenticar o usuário.

Caso o usuário for autenticado, será recuperado do mesmo servidor a hierarquia de grupos aos quais o usuário pertence para o Quality Manager realizar o processo de autorização seguindo o fluxo abaixo.

Exemplo de configuração no Active Directory:

Deve-se criar a Unidade Organizacional chamada “QualityManager” e dentro devem ser criados os grupos de acessos com nomes exatamente iguais ao configurados no QualityManager.

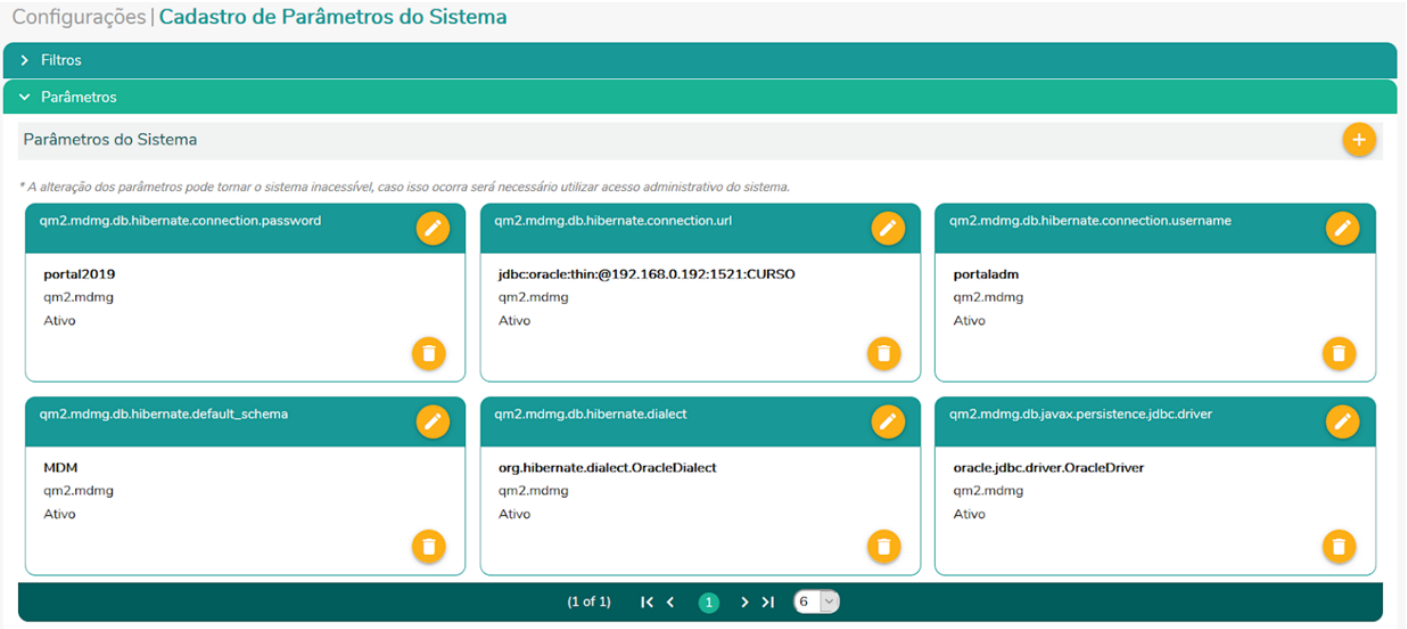


Dentro de cada grupo de acesso devem ser associados os respectivos usuários. Um cenário explicativo:
1. Foi instalado o QM, sem LDAP, e criado os usuários “usuario1” e “usuario2”.
2. O “usuario1” criou o “processo1” e o “usuario2” criou o “processo2”.
3. Foi habilitado o uso do LDAP no QM.
4. Somente o “usuario1” existe no LDAP. Vale ressaltar que o nome tem que ser exatamente igual “usuario1” no QM e “usuario1” no LDAP.
5. O “usuario2” não conseguirá logar no QM.
6. O “usuario1” conseguirá logar no QM, enxergará os seus processos normalmente. E se tiver permissão, conseguirá ver os processos do “usuario2” (sequer saberá que o “usuario2” não consegue se autenticar mais).
7. Cria-se o novo usuário “usuario3” no LDAP. Não é necessário cadastrar também no QM. No momento do primeiro login dele haverá a criação dele no QM, automaticamente.

OBS: Sobre a árvore do AD contendo os grupos é possível customizar dentro do Quality Manager na interface de parâmetros, um, em específico para setar o nome do diretório raiz dos grupos. Por exemplo, caso o cliente não desejar utilizar o esquema padrão com o diretório QualityManager setando esse o parâmetro **qm.app.ldap.rootLdapGroup** a aplicação passa a considerar o valor deste parâmetro no momento da autenticação.

Configurando Conexão de Banco de dados para módulos MDM Governance

O MD2 Quality Manager possui módulos de ferramentas MDM Governance que requerem a definição de acesso ao banco de dados. Estas definições são realizadas mediante a inclusão de parâmetros na interface de Cadastro de Parâmetros do Sistema para a unidade específica, conforme imagem.



Os valores seguem os padrões tecnológicos específicos e devem ser inseridos e alterados por profissionais especialistas. Exemplo para conectar a um banco de dados Oracle:

Descrição	Nome parâmetro	Valor (exemplo)	Grupo
Driver jdbc do banco de dados	qm2.mdmg.db.javax.persistence.jdbc.driver	oracle.jdbc.driver.OracleDriver	qm2.mdmg

URL jdbc de conexão ao banco Caso seja a conexão SID: "jdbc:oracle:thin:@<hostname ou IP>:<porta>**:**<sid>" Caso seja a conexão Service Name: "jdbc:oracle:thin:@<hostname ou IP>:<porta>**/**<service name>"	qm2.mdmg.db.hibernate.connection.url	jdbc:oracle:thin:@192.168.0.192:1521:CURSO	qm2.mdmg
Esquema/Owner das tabelas da solução MDM	qm2.mdmg.db.hibernate.default_schema	MDM	qm2.mdmg
Dialeto Hibernate	qm2.mdmg.db.hibernate.dialect	org.hibernate.dialect.OracleDialect	qm2.mdmg
Nome de usuário de banco de dados com privilégios de acesso, inclusão e alteração ao banco de dados da solução MDM	qm2.mdmg.db.hibernate.connection.username	usuario	qm2.mdmg
Senha do usuário de banco de dados	qm2.mdmg.db.hibernate.connection.password	senha	qm2.mdmg

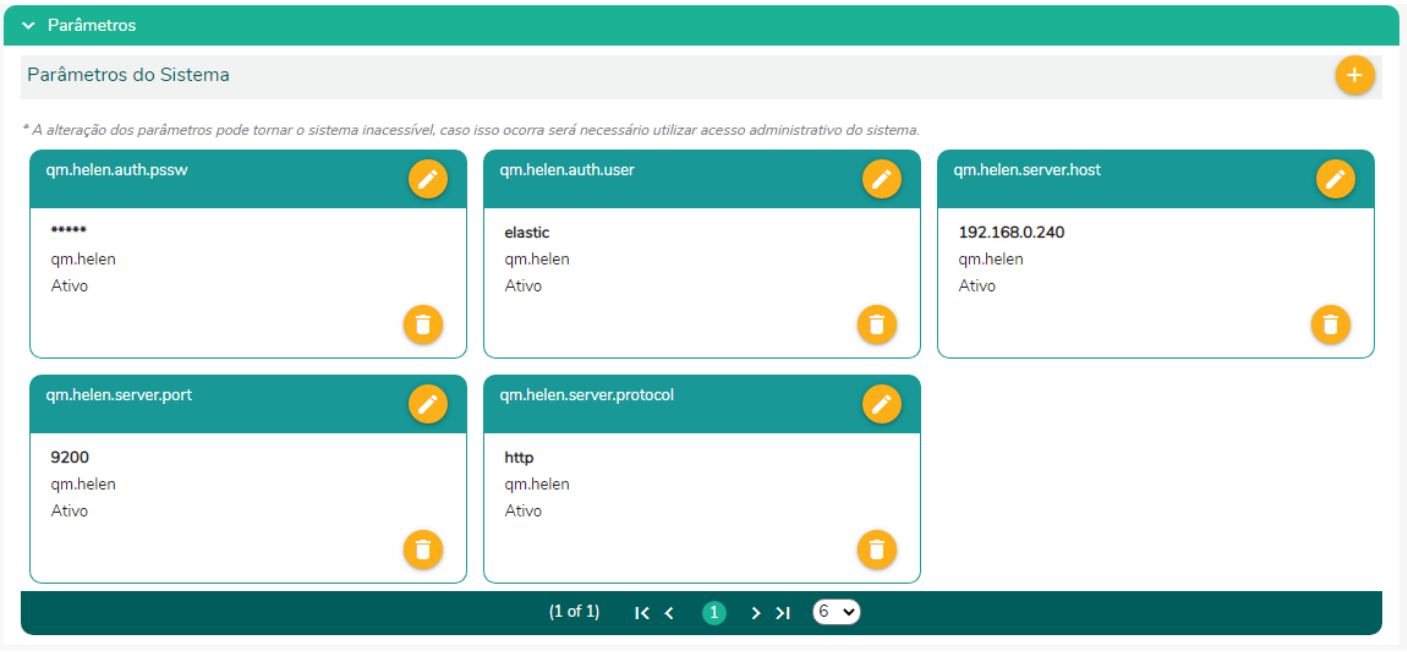
Exemplo para conectar a um banco de dados MS SQL Server:

Descrição	**Nome parâmetro**	**Valor (exemplo)**	**Grupo**
Driver jdbc do banco de dados	qm2.mdmg.db.javax.persistence.jdbc.driver	com.microsoft.sqlserver.jdbc.SQLServerDriver	qm2.mdmg
URL jdbc de conexão ao banco	qm2.mdmg.db.hibernate.connection.url	jdbc:sqlserver://192.168.0.192;DatabaseName=CURSO	qm2.mdmg
Esquema/Owner das tabelas da solução MDM	qm2.mdmg.db.hibernate.default_schema	MDM	qm2.mdmg
Dialeto Hibernate	qm2.mdmg.db.hibernate.dialect	org.hibernate.dialect.SQLServerDialect	qm2.mdmg
Nome de usuário de banco de dados com privilégios de acesso, inclusão e alteração ao banco de dados da solução MDM	qm2.mdmg.db.hibernate.connection.username	usuario	qm2.mdmg
Senha do usuário de banco de dados	qm2.mdmg.db.hibernate.connection.password	senha	qm2.mdmg

Uma vez cadastrados todos os parâmetros, o servidor Tomcat deverá ser reiniciado para poder efetivar a conexão ao banco de dados da solução MDM.

Configurando Conexão de Banco de dados para acesso ao ELK

O MD2 Quality Manager possui módulos que requerem a definição de acesso ao banco de dados do ELK. Estas definições são realizadas mediante a inclusão de parâmetros na interface de Cadastro de Parâmetros do Sistema para a unidade específica, conforme imagem.



Nome	Valor (Exemplo)	Grupo	Obs:
qm.helen.server.host	192.168.0.240	qm.helen	
qm.helen.server.protocol	http	qm.helen	
qm.helen.server.port	9200	qm.helen	
qm.helen.auth.user	usuário	qm.helen	
qm.helen.auth.pssw	senha	qm.helen	
qm.helen.timeout.con	10000	qm.helen	valores default
qm.helen.timeout.socket	60000	qm.helen	valores default

Back-up

Uma vez que a instalação e configuração é feita dentro da infraestrutura, recomenda-se fortemente que a máquina virtual (VM) da solução seja incluída na **política de backup** e no **sistema de monitoramento** para acompanhar a saúde e disponibilidade de aplicação.

Existem 2 cenários para execução de backups:
1. Através da própria ferramenta de virtualização. Fica a cargo do cliente estudar o melhor cenário, seja por exports full ou capturas de snapshots.
2. Através dos comandos de backups.

Em algumas situações, providenciar o backup da VM completa pode ser complexo, necessitando ser executado em janelas na produção. Nestes casos, recomendamos fazer o backup com maior frequência: 1. Do banco de dados através do comando:

```
$ pg_dump -h 127.0.0.1 -U md2net -d qualitymanager | gzip --best > ./bkp_DB_QM_$(date +\%Y\%m\%d_\%I_\%M_\%p).psql.gz
```

2. Do diretório de arquivos anexos:

```
/opt/qm_static/
```

3. Do diretório do WebServer Tomcat (contêm as configurações):

```
/var/lib/tomcat9/
```

4. Do diretório do database Postgres (contêm as configurações):

```
/etc/postgresql/9.5/main
```

Procedimento de restauração:

Caso o backup tenha sido feito através do “procedimento 2” (ver tópico “Backup”), a restauração deve ser substituindo os arquivos do diretórios que foram salvo e o banco de dados é restaurado através dos comandos:

```
$ sudo systemctl stop tomcat9.service
$ sudo systemctl restart postgresql.service
$ dropdb -h 127.0.0.1 -U md2net qualitymanager
$ createdb -h 127.0.1 -U md2net -E UTF8 -l pt_BR.utf8 -T template0 qualitymanager
```

```
$ gunzip -c bkp_DB_QM.psql.gz |  psql -h 127.0.0.1 -U md2net qualitymanager
$ sudo systemctl start tomcat9.service
```

Backup de Virtual Machine (VM) para Oracle VirtualBox

Soluções que estejam hospedadas através do servidor de virtualização, existe a alternativa de salvar a própria VM, seja ela inteira ou snapshot. Para os que são executam sob o Oracle VirtualBox, a gestão do snapshot pode ser feita através dos comandos abaixo:

```
VBoxManage snapshot <uuid|vmname> take <snapshot-name>
VBoxManage snapshot <uuid|vmname> restore <snapshot-name>
VBoxManage snapshot <uuid|vmname> delete <snapshot-name>
```

O primeiro comando é para geração do snapshot. O segundo para restauração de um snapshot previamente gerado e o terceiro para remoção de um snapshot, caso queira descartar.

Os arquivos snapshots ficam salvos no diretório "Snapshot", que fica armazenado o arquivo da máquina e disco (.vdi). Para cópias para locais externos, backups em nuvem ou disco de servidores de backup, deve-se fazer cópia dos arquivos ".vdi", ".vbox", ".vbox-prev" e ".log".

Dica: recomenda-se gerar o snapshot que remeta à data e hora que foi gerado.

```
c:\Program Files\Oracle\VirtualBox>VBoxManage.exe snapshot "vm-Template-teste1" take
"snapshot_%date:~-4,4%_%date:~-7,2%_%date:~-10,2%__%time:~0,2%_%time:~3,2%_%time:~6,2%"

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Snapshot taken. UUID: e92d6c82-80ba-43f2-b3eb-79ed0ae34c07
```

Gerenciar WebServer Tomcat

Iniciando o Tomcat:

```
$ sudo systemctl start tomcat9
```

Parando o Tomcat:

```
$ sudo systemctl stop tomcat9
```

Reiniciando o Tomcat:

```
$ sudo systemctl restart tomcat9
```

Gerenciar o banco de de dados Postgres

Iniciando o Postgresql:

```
$ sudo systemctl start postgresql
```

Parando o Postgresql:

```
$ sudo systemctl stop postgresql
```

Reiniciando o Postgresql:

```
$ sudo systemctl restart postgresql
```

Acompanhamento de logs via sistema

A aplicação gera logs em:

```
/var/log/tomcat9/*
```

O banco de dados em:

```
/var/log/postgresql/*
```

Estes podem ser monitorados para acompanhar a saúde da aplicação e também devem ser enviados para equipe de suporte, se necessário.

Acompanhamento de logs via aplicação

A solução contém um módulo para acompanhar alguns logs, úteis para auditoria. Apresenta informações de login, logon, erros de senha. Este componente também gerencia os acessos à aplicação. Um usuário só pode acessar apenas uma única sessão por vez.

Migração Legado GED

Ao realizar a atualização do QM para a versão 2.44.1 ou superior, vindo de uma versão anterior da 2.44.1 devemos rodar o jar "qm-tools.jar". Segue abaixo como utilizar.
java -jar <JAR GERADO*> mergeLegado --password=<senhaDB> --host=<ex:localhost> --port=<ex:5432> --database=<ex:qualitymanager> --username=<md2net>

java -jar qm-tools.jar mergeLegado --password=md2net2018 --host=192.168.0.183 --port=5432 --database=qualitymanager --username=md2net