INSTALAÇÃO DO G-HOSP Rails 6

Passo a passo para instalar o **G-HOSP** atualizado para Rails 6 em clientes que já **estão usando o sistema.**

Como saber se o servidor é apto para a instalação do RAILS 6

Requisitos mínimos para instalação:

- Banco de dados: Postgres 9.6 no mínimo
- Sistema operacional: Debian 8 ou superior, Ubuntu 16 ou superior
- Node v10 ou superior
- Yarn v1 ou superior

Como avaliar estes requisitos

- **Postgres 9.6**: Acesse o banco de dados e execute o seguinte comando:

```
SELECT version();
```

Se o postgres não for 9.6 ou superior, não pode ser realizada a migração para rails 6 ainda.
Primeiro, deve ser feito o processo de atualização da versão do banco de dados.

- **Sistema operacional:** No servidor, execute o seguinte comando:

```
cat /etc/os-release
```

ou

```
lsb_release -a
```

Se o sistema operacional for abaixo do debian 8 ou abaixo do Ubuntu 16, existem grandes possibilidades de problemas com os demais requisitos, node, postgres ou yarn. Por isso, caso seja necessário, primeiro solicitar ao cliente a formatação e atualização do servidor para Debian 8 (preferencialmente).

- **Node:** No servidor, execute o seguinte comando:

```
node -v
```

Se o node não estiver instalado, você pode adiantar a instalação do mesmo (está presente neste manual, mais para baixo). Se ocorrem problemas na instalação

do node, estes primeiros devem ser corrigidos e somente após isto, migrar para rails 6.

- **Yarn:** No servidor, execute o seguinte comando:

```
yarn -v
```

Mesma coisa que o node, realizar a instalação conforme o manual abaixo ensina.

Sumário:

Instale a versão nova do Ruby - 2.7.1

```
inovadora@localhost:~$ rvm install 2.7.1
```

Gere a chave de acesso ao repositório do rails 6

Primeiramente, confira se a chave já não está criada:

```
inovadora@localhost:~$ cd ~/.ssh
inovadora@localhost:~/.ssh$ ls
```

Se já existir algum arquivo com nome relacionado a rails 6, pode pular este passo.
Caso não, execute:

```
inovadora@localhost:~/.ssh$ ssh-keygen
```

Esse comando irá solicitar o nome da nova chave, informe: *rails6.*
Segue exemplo:

```
inovadora@localhost:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/inovadora/.ssh/id_rsa): r
```

Nas demais perguntas é só teclar “Enter”.

Em seguida execute:

```
inovadora@localhost:~/.ssh$ cat rails6.pub
```

Você verá uma saída no terminal com conteúdo semelhante a este aqui:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCyAHjyS8PhRHVI1Mb+RnW8rxf5097Vaiw0mfuK
KzKdEFhPprOVtT0R3A5HFaZlzLBreKOLiYzSEAW8mfA8HgijUNTQwRdJ93cdNLFxy8KL
7ncaPdPdMkgtcljWXzx3B/Cp9SzZNTxBDK5qnBSQfcQfSPfHlkijhTlcL4UORcg3G95/
fcLta9B/zNH6G0ta1OZ0Boi8Qf8tPslm2Su1 ...
```

Essa “chave” deve ser adicionada ao bitbucket, na conta do **G-MUS**, na Sessão de Chaves.
Confirme que você está logado na conta do G-MUS. Seguem os acessos:

Novamente no terminal, realize a configuração para uso de múltiplas chaves através do arquivo config. Segue exemplo de como fazer:
Execute:

```
inovadora@localhost:~$ cd ~/.ssh
inovadora@localhost:~/.ssh$ nano config
```

Cole dentro deste arquivo a seguinte configuração:

```
Host bitbucket.org
  Hostname bitbucket.org
  IdentityFile ~/.ssh/id_rsa
Host bitbucket-rails6
  Hostname bitbucket.org
  IdentityFile ~/.ssh/rails6
```

Clone o repositório do rails 6

Execute:

```
inovadora@localhost:~$ git clone
git@bitbucket-rails6:inovadora-ti/g-hosp-rails6.git ~/g-hosp_rails6
```

Configurações do g-hosp rails 6

Entre na pasta do g-hosp rails 6 e inicie setando a versão 20.09 e clonando o rgloader:

```
inovadora@localhost:~$ cd ~/g-hosp_rails6
inovadora@localhost:~/g-hosp_rails6$ git checkout master
inovadora@localhost:~/g-hosp_rails6$ git clone
git@bitbucket.org:inovadora/rgloader.git
```

Configuração de acesso ao banco de dados

Em seguida, realize a configuração de acesso ao banco de dados. Para isso, copie o .ghosp.yml da pasta do g-hosp normal para o rails 6.
Um exemplo seria fazer desta forma:

```
inovadora@localhost:~/g-hosp_rails6$ cp ~/g-hosp/.ghosp.yml
~/g-hosp_rails6/.ghosp.yml
```

No arquivo .ghosp.yml do rails 6, você deve adicionar mais um trecho de parametrização. Basta copiar um bloco que esteja pronto, colar, e renomear o ambiente para "*test*" ***(DEIXAR ALINHADO COM AS LINHAS)***.
Segue exemplo abaixo de como fazer:

```
inovadora@localhost:~/g-hosp_rails6$ nano .ghosp.yml

(copie o production todo)
production:
   adapter:  postgresql
   encoding: utf8
   database: nome_banco_dados_exemplo
   username: inovadora
   password: senha_banco_dados_exemplo
   host: localhost
   porta_mordomo: 7001
   schema_search_path: gestho, public, mordomo, prescricao, ct
gemus
   port: 5432
   ip_mordomo: localhost
   database_mordomo: nome_banco_dados_exemplo
```

```
(cole mais a baixo, e renomeie o ambiente para "test")
test:
   adapter:  postgresql
   encoding: utf8
   database: nome_banco_dados_exemplo
   username: inovadora
   password: senha_banco_dados_exemplo
   host: localhost
```

```
   porta_mordomo: 7001
   schema_search_path: gestho, public, mordomo, prescricao, ct
gemus
   port: 5432
   ip_mordomo: localhost
   database_mordomo: nome_banco_dados_exemplo
```

Configuração do puma.rb de cada módulo

Agora você deve configurar os módulos do rails 6 individualmente. No presente momento, os módulos são Recepção, Prescrição e Cadastros Gerais.

Primeiro, gere o arquivo puma.rb para cada um dos módulos. Segue:

```
inovadora@localhost:~/g-hosp_rails6$ . scripts/gera_puma_rb.sh
inovadora@localhost:~/g-hosp_rails6$ . scripts/gera_puma_rb.sh
inovadora@localhost:~/g-hosp_rails6$ . scripts/gera_puma_rb.sh
```

Em seguida, edite os arquivos gerados e renomeie o socket para sigla do módulo + rails6. Abaixo o exemplo:

```
inovadora@localhost:~/g-hosp_rails6$ nano cg/config/puma.rb
rc/config/puma.rb pr/config/puma.rb
```

```
(Exemplo de como é o arquivo)
# config/puma.rb
threads 4,5
# workers 2
environment 'production'
bind 'unix:///tmp/cg.socket'
preload_app!
# daemonize
pidfile 'tmp/pids/puma.pid'
state_path 'tmp/pids/puma.state'
activate_control_app 'unix:///tmp/cgctl.sock'
# prune_bundler
stdout_redirect 'log/puma.log', 'log/puma_error.log', true
worker_timeout 60
```

```
(Exemplo de como ele tem que ficar)

# config/puma.rb
threads 4,5
# workers 2
environment 'production'
bind 'unix:///tmp/cg_rails6.socket'
preload_app!
# daemonize
pidfile 'tmp/pids/puma.pid'
state_path 'tmp/pids/puma.state'
activate_control_app 'unix:///tmp/cgrails6ctl.sock'
# prune_bundler
stdout_redirect 'log/puma.log', 'log/puma_error.log', true
worker_timeout 60
```

Faça esse mesmo procedimento para todos os demais módulos.

Criação de pastas tmp e pids

Dando sequência, em cada um dos módulos, crie a pasta tmp e dentro dela, a pasta pids. Confira um exemplo de como fazer isso:

```
inovadora@localhost:~/g-hosp_rails6$ mkdir cg/tmp rc/tmp pr/tm
cg/tmp/pids rc/tmp/pids pr/tmp/pids
```

Copiar Relatórios

O próximo passo é copiar os relatórios personalizados para os módulos. Observe como isso pode ser feito:

```
inovadora@localhost:~/g-hosp_rails6$ cp -r  ~/g-hosp/cg/rels/*
~/g-hosp_rails6/cg/rels/
inovadora@localhost:~/g-hosp_rails6$ cp -r  ~/g-hosp/rc/rels/*
~/g-hosp_rails6/rc/rels/
inovadora@localhost:~/g-hosp_rails6$ cp -r  ~/g-hosp/pr/rels/*
~/g-hosp_rails6/pr/rels/
```

ou

```
inovadora@localhost:~/g-hosp_rails6$ cp -r ../g-hosp/cg/rels/*
```

```
./cg/rels/
inovadora@localhost:~/g-hosp_rails6$ cp -r ../g-hosp/rc/rels/*
./rc/rels/
inovadora@localhost:~/g-hosp_rails6$ cp -r ../g-hosp/pr/rels/*
./pr/rels/
```

O passo a seguir é opcional. Vamos copiar os anexos do PR. Só execute-o se na pasta “pr/public”, houver uma pasta com nome “anexos”.

```
inovadora@localhost:~/g-hosp_rails6$ cd pr/public
inovadora@localhost:~/g-hosp_rails6$ rsync -rlptog
../../../g-hosp/pr/public/anexos .
```

Node e Yarn

Vida que segue, agora tu deves instalar o Node e o Yarn caso já não ainda não estejam instalados.
Para verificar se o Node já está instalado, execute:

```
inovadora@localhost:~/ node -v
```

Se o retorno for algo como: “Comando não encontrado”, realize o procedimento abaixo:

```
inovadora@localhost:~/ sudo su
root@localhost:~/ curl -sL https://deb.nodesource.com/setup_14
sudo bash -
root@localhost:~/ apt install nodejs
root@localhost:~/ exit
```

Para verificar se o Yarn já está instalado, execute:

```
inovadora@localhost:~/ yarn -v
```

Se o retorno for algo como: “Comando não encontrado”, realize o procedimento abaixo:

```
inovadora@localhost:~/ sudo su
root@localhost:~/ curl -sS
https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
root@localhost:~/ echo "deb https://dl.yarnpkg.com/debian/ sta
main" | sudo tee /etc/apt/sources.list.d/yarn.list
root@localhost:~/ apt update && sudo apt install yarn
root@localhost:~/ exit
```

Bundle nos módulos novos

Agora, em cada um dos módulos da pasta rails 6, execute a seguinte sequência de comandos:

```
inovadora@localhost:~/g-hosp_rails6$ cd cg
inovadora@localhost:~/g-hosp_rails6/cg$ bundle install
inovadora@localhost:~/g-hosp_rails6/cg$ EDITOR=nano rails
credentials:edit
```

Se houverem problemas com o comando bundle, verifique qual a versão que o rvm está usando do ruby e se necessário, ajuste para a 2.7.1.

```
inovadora@localhost:~/$ rvm list
inovadora@localhost:~/$ rvm use <versao>
```

Outra alternativa é recarregar as funções do rvm. Use o comando:

```
inovadora@localhost:~/$ source /home/$USER/.rvm/scripts/rvm
```

O último comando do bloco, vai abrir um arquivo para edição, é só salvar e fechar. Continue com os seguintes comandos:

```
inovadora@localhost:~/g-hosp_rails6/cg$ RAILS_ENV=production r
assets:clean
inovadora@localhost:~/g-hosp_rails6/cg$ RAILS_ENV=production r
assets:precompile
```

Execute o mesmo procedimento para todos os módulos do Rails 6, atualmente é no CG, RC e PR.

Config dos sockets
urar NGINX, Ajustar nome

Esse passo aqui pode variar um pouco de cliente para cliente, mas abaixo segue um exemplo de como ele deve ser feito. Primeiro, edite o arquivo g-hosp dentro da pasta /etc/nginx/sites-enabled.

```
inovadora@localhost:~/$ sudo nano /etc/nginx/sites-enabled/g-h
```

Nele, vai estar configurado todos os virtual host’s, um para cada módulo usado pelo cliente em questão. Vamos mexer somente nos que forem migrados para rails 6. Abaixo segue um exemplo do que deve ser alterado:

```
(modelo de como vai estar ai no seu arquivo)

upstream cg_upstream
   {
     server unix:///tmp/cg.socket;
   }

server
   {
     listen 3000;
     server_name localhost;
     root /home/inovadora/g-hosp/cg/public;
     location /
     {
          proxy_pass http://cg_upstream;
          proxy_set_header Host $host:3000;
          proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
          client_max_body_size 45M;
          proxy_read_timeout 3000;
     }
     location ~* ^/assets/
     {
          expires 1y;
          add_header Cache-Control public;
          add_header Last-Modified "";
          add_header ETag "";
          break;
     }
   }
```

Mude para ficar da seguinte forma (partes em cinza):

```
upstream cg_upstream
   {
     server unix:///tmp/cg_rails6.socket;
   }
```

```
server
   {
     listen 3000;
     server_name localhost;
     root /home/inovadora/g-hosp_rails6/cg/public;
     location /
     {
          proxy_pass http://cg_upstream;
          proxy_set_header Host $host:3000;
          proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
          client_max_body_size 45M;
          proxy_read_timeout 3000;
     }
     location ~* ^/assets/
     {
          expires 1y;
          add_header Cache-Control public;
          add_header Last-Modified "";
          add_header ETag "";
          break;
     }
   }
```

Ajustar o arquivo módulos de cada um dos G-hosp’s

Dentro da pasta g-hosp, vai existir um arquivo com nome: módulos.
Copie esse arquivo para dentro da pasta rails 6.

```
inovadora@localhost:~/g-hosp_rails6$ cp ../g-hosp/modulos .
```

Edite este arquivo do rails 6.

```
inovadora@localhost:~/g-hosp_rails6$ nano modulos
```

Deixe marcado somente os módulos ativos em rails 6.

```
cg|*
cp|
ct|
fn|
ft|
gd|
hh|
mn|
mp|
nt|
pc|
ph|
pm|
pn|
pr|*
rc|*
rd|
sc|
st|
ts|
```

Agora acesse a pasta do g-hosp e desmarque os módulos marcados no rails6

```
inovadora@localhost:~/$ cd g-hosp
inovadora@localhost:~/g-hosp$ nano modulos
```

Atualização do sistema

Realize o processo de atualização do g-hosp_rails6 via scripts/web_atualiza_tudo. Esse procedimento não será detalhado aqui, mas deve ser realizado da 20.09 para a 20.11, e depois da 20.11 para a última versão disponível. Dependendo da situação, você também precisará atualizar o g-hosp normal. Se o cliente já estiver acima da versão 20.11, atualize primeiro o g-hosp_rails6 e depois o g-hosp normal. Caso esteja abaixo, primeiro atualize o g-hosp normal até a última versão, e depois o rails 6.

Lembrando que só há migrações para serem rodados no G-HOSP normal até a 20.11, depois todas elas passam a ser concentradas no G-HOSP Rails 6.

Excluir Rules do banco de dados

Rode o script **excluir_rules_views_rails_6** no banco de dados. Segue exemplo de como isso deve ser feito.

```
inovadora@localhost:~/g-hosp_rails6$ psql nome_do_banco_exempl
cg/db/comandos_locais_migracao/excluir_rules_views_rails_6.sql
```

MUITO CUIDADO.

Se o cliente for integrado, G-MUS com G-HOSP, você deverá refazer a integração de banco de dados. Os comandos SQL que devem ser rodados estão dentro da pasta do G-HOSP Rails 6:

g-hosp_rails6/cg/db/migracao_manual/020_views_gemus.sql

Finalizando e reiniciando os serviços:

Após todo estes procedimentos, reinicie o nginx:

```
inovadora@localhost:~$ sudo /etc/init.d/nginx restart
```

e também reinicie os módulos, com o script_web_stop tudo e web_start_tudo