

Instalando Produto - Odhen Retail

O repositório utilizado é o GitLab

<http://gitlab.teknisa.com/>

Os dados de login são os mesmos da VPN e é necessário estar conectado nela para acessar (ele nem entra kk), é necessário solicitar acesso ao grupo para ver os repositórios (procure seu gestor).

O produto engloba os módulos - <http://gitlab.teknisa.com/odhen/management>.

E um dos módulos é o Retail -

<http://gitlab.teknisa.com/odhen/man>

<http://gitlab.teknisa.com/odhen/manfrontend>

<http://gitlab.teknisa.com/odhen/manbackend>

Mas não é o único, existe o CADEST, ACL e afins.



GitLab

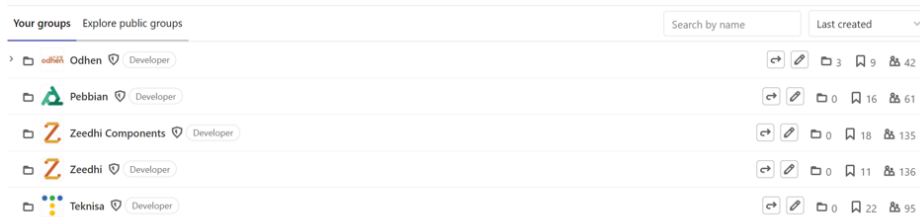
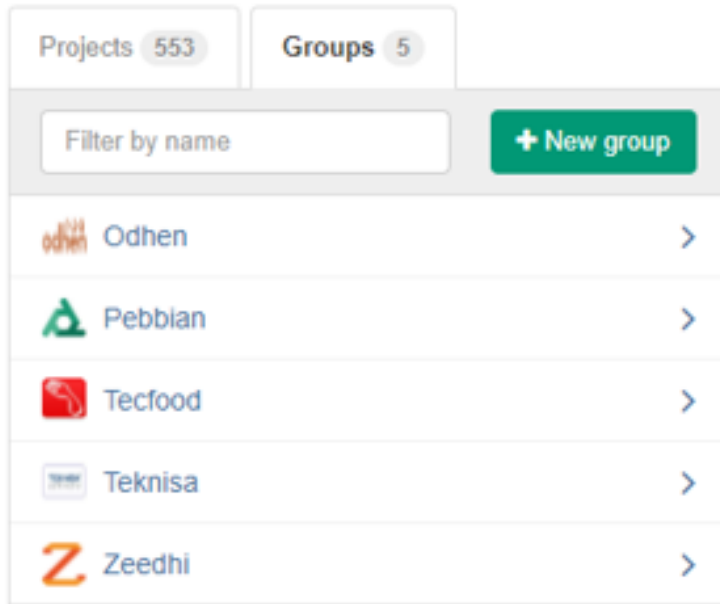
A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

LDAP	Standard
LDAP Username	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	
<input type="button" value="Sign in"/>	

1. Grupos Relacionados nos Repositórios (Code e GitLab)



2. Gerar chaves de acesso (SSH)

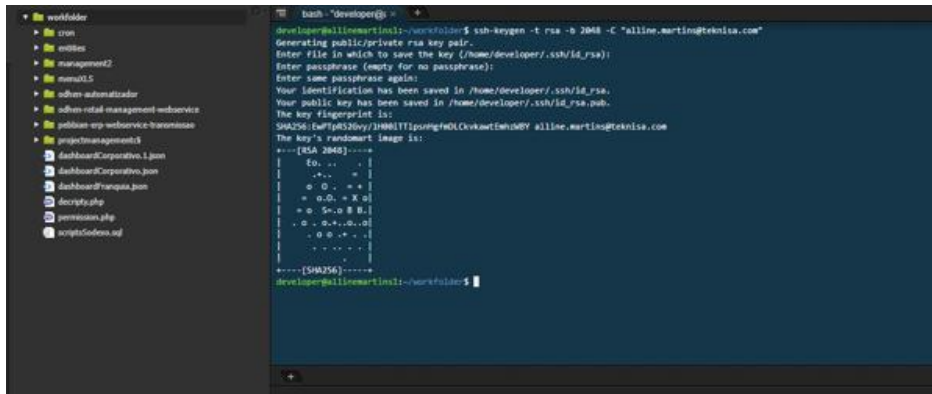
Abrir o cloud9 <http://nomesobrenome2.zeedhi.com/> ou VSCode no <https://nomesobrenome84.zeedhi.com/> (sem o 84 caso seja para o php7.4) no diretório ~/workfolder rodar os comandos abaixo:

Comandos para config do git, não ficar pedindo dados de conexão quando tentar conectar o tempo todo

```
git config --global user.name "Fulano de Tal"
git config --global user.email "nome.sobrenome@teknisa.com"
git config -l
```

```
ssh-keygen -t rsa -b 2048 -C "nome.sobrenome@teknisa.com"
```

Obs.: O cloud e a chave são pessoais, então é importante trocar os nomes acima com o pessoal de vcs

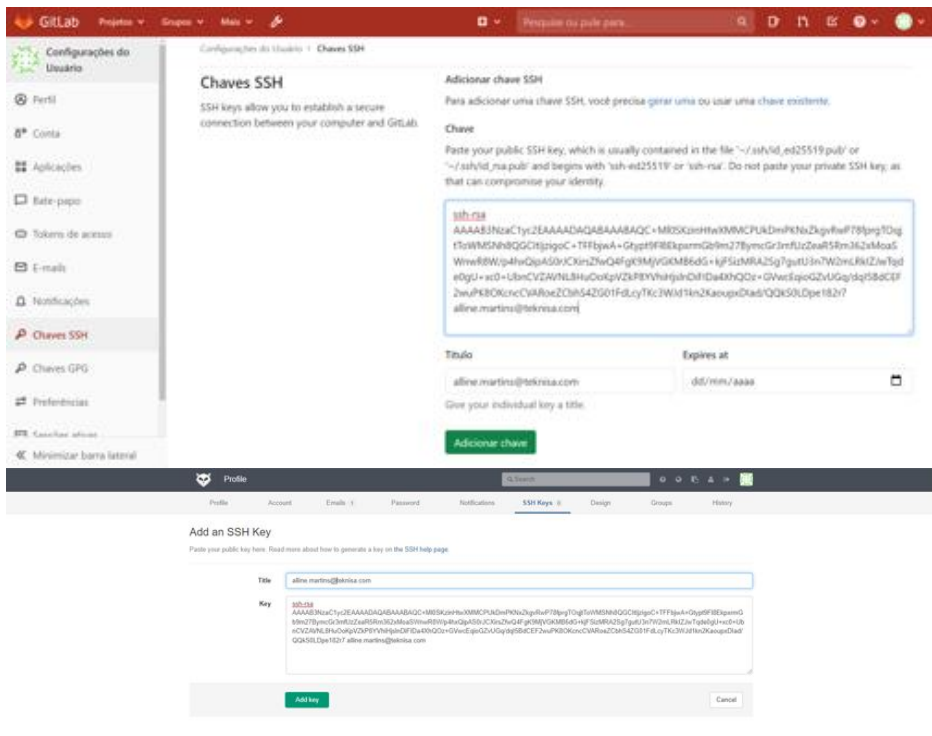


Após rodar irá pedir 3x para que seja dado o ENTER, após isso rodar o seguinte comando para pegar a chave pública e ser possível colocar nos Code e GitLab

cat ~/.ssh/id_rsa.pub



Copiar a chave e colocar tanto no GitLab, quanto no Code



Para confirmar se está conectando com os repositórios, rodar os comandos abaixo:

```

ssh -T git@code.zeedhi.com
ssh -T git@gitlab.teknisa.com
  
```

```

developer@allinemartins1:~/workfolder$ ssh -T git@code.zeedhi.com
The authenticity of host 'code.zeedhi.com (192.168.122.56)' can't be established.
ECDSA key fingerprint is SHA256:77tEQK8/RTD4YgBnOt5RrYmMJ0qvVu/JweqJPdud44.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'code.zeedhi.com,192.168.122.56' (ECDSA) to the list of known hosts.
Welcome to GitLab, Alline Cristiane Dias Martins!
developer@allinemartins1:~/workfolder$ ssh -T git@gitlab.teknisa.com
Welcome to GitLab, @allinemartins!
developer@allinemartins1:~/workfolder$ █

```

3. Instalando o Produto - Odhen Retail

<http://gitlab.teknisa.com/odhen/management>

Todo o processo será realizado através da ide (Cloud9 ou VSCode) e vamos clonar através do SSH

Para clonar através do SSH, deve-se rodar o seguinte código:

```
git clone git@gitlab.teknisa.com:odhen/management.git
```

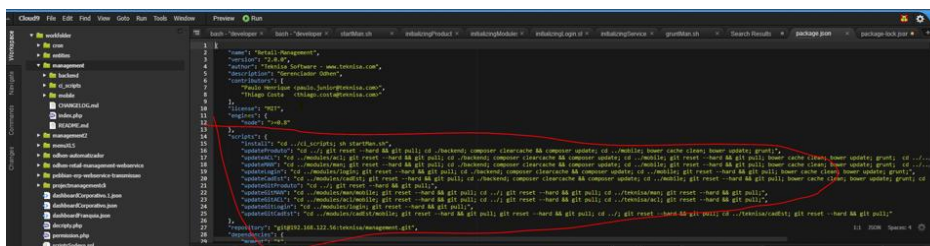
```

developer@allinemartins1:~/workfolder$ git clone git@gitlab.teknisa.com:odhen/management.git
Cloning into 'management'...
remote: Enumerating objects: 29626, done.
remote: Total 29626 (delta 0), reused 0 (delta 0), pack-reused 29626
Receiving objects: 100% (29626/29626), 32.66 MiB | 1.37 MiB/s, done.
Resolving deltas: 100% (17952/17952), done.
developer@allinemartins1:~/workfolder$ █

```

Dentro do MOBILE há um arquivo onde há algumas configurações referente ao projeto, nele é possível verificar alguns scripts pré-definidos para que sejam instalado os Micro Serviços, os principais que precisamos são o LOGIN, ACL (Controle de Acesso) e o Modulo de Vendas (MAN) =>

/management/mobile/package.json



Para pegar tudo de forma manual é importante deixar esse arquivo aqui /management/ci_scripts/initializingModules.sh todo comentado (**ctrl + ;**), pois nele baixa algumas dependências ao rodar o comando abaixo dentro da pasta MOBILE /management/mobile

```
cd ~/workfolder/management/mobile
npm run install
```

Aqui vai rodar o **composer**, o **bower** e o **grunt** somente do Produto, agora se faz necessário clonar o Login e ACL

4. Instalando o Login

<http://gitlab.teknisa.com/teknisa/loginsetup>

Por se tratar de um micro serviço é necessário **criar a pasta *MODULES*** dentro do Projeto principal, e assim dentro dessa pastar clonar o projeto de LOGIN

```
mkdir ~/workfolder/management/modules
cd ~/workfolder/management/modules
git clone git@gitlab.teknisa.com:teknisa/loginsetup.git
mv loginsetup login
```

```
cd ~/workfolder/management/modules/login/backend
composer install
```

```
cd ~/workfolder/management/modules/login/mobile
npm install
bower install
grunt
```

Após isso já é possível ver a primeira imagem do Produto

```
/*PARA QUEM PRECISAR*/
npm install grunt
npm install moment
/*****/
```

<http://allinemartins1.zeedhi.com/workfolder/management/>



5. Instalando ACL (Controle de Acesso)

1. Clonar o módulo ACL
2. `cd ~/workfolder/management/modules`
3. `git clone git@gitlab.teknisa.com:teknisa/acl.git`
4. `cd ~/workfolder/management/modules/acl/backend`
`composer install`
5. Clonar o módulo ACL Frontend dentro da pasta ACL.
6. `cd ~/workfolder/management/modules/acl`
7. `git clone git@gitlab.teknisa.com:teknisa/acl-frontend.git`
8. `mv acl-frontend mobile`
9. `cd ~/workfolder/management/modules/acl/mobile`
10. `npm install`
11. `bower install`
`grunt`
12. Criar uma pasta teknisa dentro de modules, clonar o repositório do Backend do ACL, após clonar, renomear a pasta ACLFRONTEND apenas para MOBILE.
13. `mkdir ~/workfolder/management/modules/teknisa`
14. `cd ~/workfolder/management/modules/teknisa`
15. `git clone git@gitlab.teknisa.com:teknisa/acl-backend.git`
`mv acl-backend acl`

/*PARA QUEM PRECISAR*/

`npm install grunt`

`npm install moment`

/*******/

Após esses comandos é possível já acessar o sistema, agora é a vez do MAN (Vendas)

6. Instalando MAN (Vendas)

<http://gitlab.teknisa.com/odhen/man>

<http://gitlab.teknisa.com/odhen/manfrontend>

<http://gitlab.teknisa.com/odhen/manbackend>

1. Clonar o backend (configs)
2. `cd ~/workfolder/management/modules`
3. `git clone git@gitlab.teknisa.com:odhen/man.git`
4. `cd ~/workfolder/management/modules/man/backend`
`composer install`

5. Clonar o Frontend do Man
6. `cd ~/workfolder/management/modules/man`
7. `git clone git@gitlab.teknisa.com:odhen/manfrontend.git`
8. `mv manfrontend mobile`
9. `cd ~/workfolder/management/modules/man/mobile`
10. `npm install`
11. `bower install`
`grunt`

12. Clonar o Backend do MAN (assets)
13. `cd ~/workfolder/management/modules/teknisa`
14. `git clone git@gitlab.teknisa.com:odhen/manbackend.git`
15. `mv manbackend man`
`cd ~/workfolder/management/modules/teknisa/man`

/*PARA QUEM PRECISAR*/

`npm install grunt`

`npm install moment`

/*******/

Notas de rodapé -

Para instalar o CADEST - [Instalando o cadEst](#)

Para instalar o CADFIN - [Instalando o cadFin](#)

Entenda a atualização do módulo - [Atualização do Projeto Odhen Retail](#)

É possível “resetar“ se vc fizer alguma coisa errada, apagando a vendor nos backend e rodando um `composer install`, ou a pasta `bower_components` e rodando um `bower install` depois.

Esse script abaixo atualiza os módulos MAN, CNC, CADEST e ACL, é recomendado rodar

antes de entrar em algum repositório de tarefa (dando play no pm por ex) e depois de pausar/finalizar a tarefa.

```
cd ~/workfolder/management/backend
git checkout master
git reset --hard
git pull
git pull origin master
composer update
```

```
cd ~/workfolder/management/ci_scripts
git checkout master
git reset --hard
git pull
git pull origin master
```

```
cd ~/workfolder/management/mobile
git checkout master
git reset --hard
git pull
git pull origin master
bower update
grunt
```

```
cd ~/workfolder/management/modules/acl/backend
git checkout PROD_2272_2.24.3
cd ~/workfolder/management/modules/acl/mobile
git checkout PROD_2272_2.24.3
grunt
```

```
cd ~/workfolder/management/modules/cadEst/backend
git checkout master
git reset --hard
git pull
git pull origin master
composer update
```

```
cd ~/workfolder/management/modules/cadEst/mobile
git checkout master
git reset --hard
git pull
git pull origin master
bower update
grunt buildAclStruct
grunt build
grunt buildEnvironment
grunt zhWords
grunt zhIdGenerator
grunt
```

```
cd ~/workfolder/management/modules/cnc/backend
git checkout master
git reset --hard
git pull
git pull origin master
composer update
```

```
cd ~/workfolder/management/modules/cnc/mobile
git checkout master
git reset --hard
git pull
git pull origin master
bower update
grunt
```

```
cd ~/workfolder/management/modules/teknisa/cnc
git checkout master
git reset --hard
git pull
git pull origin master
cd ~/workfolder/management/modules/man/backend
git checkout master
git reset --hard
git pull
git pull origin master
composer update
```

```
cd ~/workfolder/management/modules/man/mobile
git checkout master
git reset --hard
git pull
git pull origin master
bower update
grunt
```

```
cd ~/workfolder/management/modules/teknisa/man
git checkout master
git reset --hard
git pull
git pull origin master
```
