

INSTITUTO INFNET
ESCOLA SUPERIOR DE TECNOLOGIA DA INFORMAÇÃO
GRADUAÇÃO EM REDES DE COMPUTADORES



**Fundamentos da Arquitetura de Infraestrutura de
Aplicações**

ALUNO: FERNANDO LUCIANO MOURA

E-MAIL: fernando.moura@al.infnet.edu.br

Matrícula: 03341021701

TURMA: GRC-PR MANHÃ

**Projeto Final - Arquitetura e Infraestrutura de
Aplicações**

Sumário

Objetivo	4
Introdução	4
Justificativa	4
1. - Informações sobre como o projeto da aplicação aceita requisições de desenvolvimento e correção de bugs por parte dos usuários e stakeholders.....	5
2. Detalhes técnicos sobre como o código-fonte da aplicação é gerenciado.....	7
3. Informações sobre como compilar/construir os componentes da aplicação e transformá-la em uma unidade instalável	9
3.1. Instalação de Servidor OpenERP.....	9
3.2. Dependências Odoo Web.....	10
3.3. Instalação do Componente Gdata.....	10
4. Uma estimativa da infraestrutura e ferramentas de gerenciamento de configuração que podem ser usadas para gerenciar a instalação da aplicação.	11
5. Um esboço de um plano de implantação da aplicação em sua estrutura corporativa.	12
6. Escrevendo um playbook Ansible que instale a aplicação escolhida por você para realização do TP1.....	13
7. Documentando em um pequeno relatório a criação de um repositório público para armazenar o playbook desenvolvido no TP da etapa 2.	18
8. Documentando a execução de uma aplicação distribuída, composta por dois ou mais containers Docker.	21
8.1. Execução da Aplicação e de Seus Respectivos Containers.....	22
9. Apresentação.....	23
10. Apresentação da Empresa.	23
11. Premissas do Nosso Projeto.....	24
12. Requisitos do Projeto.....	24
13. Por que VMware.....	25
14. Comparações entre a forma como você pretende implantar seu ambiente e outras abordagens, usando outras ferramentas.	28
15. Uma proposta inicial de como organizar a infraestrutura de sua aplicação, de acordo com o sistema de virtualização em que você pretende implementá-la.	31

16.	Cronograma de Estimativa de Implantação da Aplicação.....	31
17.	Implementação da Nossa Aplicação na Infraestrutura da Empresa genstar.....	32
18.	Uma pequena descrição textual teórica sobre a arquitetura da solução de virtualização que você pretende utilizar.	33
19.	Comparações entre a forma como você pretende implantar seu ambiente e outras abordagens, usando outras ferramentas.....	34
20.	Um planejamento passo a passo (com descrições de cada etapa) de como será feita a implantação da aplicação distribuída virtualizada.	35
21.	Iniciando o cliente Web.....	38
22.	Referências dos downloads e/ou versões de todos os componentes da solução implementada.....	42
22.1.	Instalação das dependências do Python para Odoo.....	43
22.2.	Criamos usuário e grupo Odoo e efetuamos a instalação do Gdata.....	45
22.3.	Instalação do WKHTMLTOPDF para o Odoo (Dependência necessária)	46
23.	Funcionamento da Aplicação.....	50
24.	Se o prazo estabelecido no início do projeto foi adequado para a execução (não há problema se levou mais tempo que o planejado, mas se levou, explique as razões).....	51
25.	Se os recursos planejados (quantidade de memória, disco, etc.) foram suficientes para colocar a solução em funcionamento.	52
26.	Se as funcionalidades previstas na solução original funcionaram a contento. Se não foram, explique as razões.....	52
27.	Quais seriam as melhorias futuras que poderiam ser feitas no projeto executado, adotando soluções adicionais, aperfeiçoando instalações ou tornando o cenário mais complexo.	52
28.	Conclusão.....	53
29.	Referências Bibliográficas.....	53
30.	Link do Projeto.....	53

Objetivo

Apresentar o OpenERP, sistema de gestão, que é rotineiramente utilizado não só por grandes empresas, mas também por empresas muito pequenas e empresas independentes. Esta diversidade é uma ilustração de flexibilidade do software: a coordenação bastante elegante entre as expectativas funcionais das pessoas sobre o software e uma grande facilidade de uso. E essa diversidade também é encontrada em vários setores e profissões que usam o software, incluindo os produtos agrícolas, têxteis, leilões públicos, TI e associações comerciais.

Justificativa

Para fazer com que um sistema ERP totalmente esteja disponível para pequenas e médias empresas, a redução de custos é a primeira prioridade. Open source software torna possível reduzir consideravelmente os custos de desenvolvimento por meio da reutilização agressiva de bibliotecas de software de código aberto, para eliminar intermediários (distribuidores), com todas as suas despesas de vendas caras, para cortar custos de venda por publicação gratuita do software, e reduzir consideravelmente a sobrecarga de marketing.

Introdução

Riscos e os custos de integração são importantes barreiras a todas as vantagens que você ganha a partir de tais sistemas. É por isso que, hoje, algumas pequenas e médias empresas utilizam ERP. Além disso, os maiores fornecedores de ERP, como SAP, Microsoft e Oracle não ter sido capaz de conciliar o poder e a cobertura abrangente de um sistema de ERP com a simplicidade e flexibilidade desejadas pelos usuários. Mas é exatamente isso que pequenas e médias empresas estão procurando. Os processos de software open source, e os novos modelos de negócios adotados por seus desenvolvedores, fornecer uma nova maneira de resolver tais questões de custo e de qualidade para este tipo de software corporativo.

1. - Informações sobre como o projeto da aplicação aceita requisições de desenvolvimento e correção de bugs por parte dos usuários e stakeholders.

O OpenERP que agora passou a ser chamado de Odoo, tem uma boa quantidade de pessoas que formam equipes e comunidades que fornecem apoio para a sua manutenção. Esta comunidade formada por colaboradores que prestam ajuda no que diz respeito as requisições que são feitas e possíveis incidentes que venham a acontecer no desenvolvimento do sistema.

Afinal, quem são os diferentes integrantes da comunidade do OpenERP?

Relatórios de erros e processamento de bugs Bug Tracker

Conforme descrito na seção Launchpad, Community Platform, o OpenERP usa o Launchpad como plataforma de rastreamento de bugs.

Qualquer pessoa é livre para relatar novos bugs ou dar feedback sobre os já existentes no OpenPlayer do OpenPlayer. O único requisito é inscrever-se no Launchpad e juntar-se à equipe OpenERP Community , que requer apenas alguns cliques.

☒ Colaboradores

São as pessoas que querem ajudar a melhorar o projeto, adicionar funcionalidade e melhorar a estabilidade. Qualquer pessoa pode contribuir com o projeto, reportar bugs, propondo algumas melhorias e postar patches.

A equipe da comunidade está disponível no Launchpad: <https://launchpad.net/~OpenERP-comunidade>

☒ Equipes de Colaboradores da Comunidade

Committers oficiais são pessoas autorizadas a comitar nos repositórios da comunidade compartilhados. Essas pessoas são aprovadas como committers pela comunidade, uma vez que tenham desde manchas de boa qualidade e / ou as contribuições do projeto o suficiente.

ELES São Autorizados a:

- ✓ Propor e postar seus próprios patches contra relatórios de bugs.
- ✓ Reveja os patches dos contribuintes, comentários e / ou melhorá-los.

- ✓ Comprometer remendo de boa qualidade ou contribuição no repositório da comunidade.
- ✓ Escrever uma notícia sobre o Planeta OpenERP RSS
- ✓ A lista da equipe contribuintes comunidade incluem as seguintes equipas principais (** obter a lista completa aqui **: <https://launchpad.net/~OpenERP-community-revisor> / + participação), mais específicas para uma determinada categoria de recursos OpenERP:

☒ **OpenERP Core Team**

A equipe principal OpenERP os funcionários oriundos do OpenERP SA responsáveis pela qualidade do repositório oficial. Eles garantem a estabilidade ea coerência da versão oficial, revendo todos os patches e compromete.

O Core Team OpenERP está disponível no Launchpad: <https://launchpad.net/~OpenERP>

☒ **Equipes de especialistas**

Para ajudar os desenvolvedores e colaboradores tomar as decisões corretas quando melhorar OpenERP, montamos equipes de especialistas em diferentes domínios de gestão. Somente as pessoas que têm uma forte experiência em OpenERP eo domínio relacionado pode ser aplicado como um especialista. Temos equipes de contadores, especialistas em produção, técnicos, especialistas em gestão de serviços,...

Os desenvolvedores podem contactar os nossos especialistas lista quando eles precisam de feedback sobre as características específicas a serem desenvolvidas. Entre em contato com nossos especialistas apenas para novas questões relacionadas com o desenvolvimento. Eles não fornecem ajuda sobre os recursos atuais do OpenERP. A maioria dos nossos especialistas têm posições muito superiores na empresa em que trabalham, de modo que eles não têm tempo para gastar fornecer ajuda ou apoio.

☒ **Equipe de tradutores**

As equipes de tradução estão no comando da tradução do OpenERP em um idioma específico.

Quando criamos uma especificação

<<https://blueprints.launchpad.net/openobject/+addspec>>para um novo recurso no Launchpad (chamado de projeto), podemos atribuir uma equipe especializada como um projetista da especificação. Então, podemos clicar no feedback pedido em nosso projeto e atribuir isto a uma equipa especializada. Eles vão receber um email de notificação e vai discutir o recurso solicitado. A equipe vai melhorar suas especificações diretamente em seu projeto.

2. Detalhes técnicos sobre como o código-fonte da aplicação é gerenciado.

Os códigos do OpenERP, por serem parte de um Sistema Opensource, sofrem constantes alterações e diante disso, precisam ser organizados e gerenciados da melhor forma possível de modo que sejam disponibilizados para que possam ser consultados, alterados e aperfeiçoados.

☒ **Repositório**

O repositório é a pasta do projeto. Todo repositório tem uma pasta oculta.git. Isso é o que mostra para o git e para você que existe um repositório naquela pasta.

☒ **Commit**

Um commit é um grupo de alterações no código. Toda vez que você quiser "salvar" as alterações feitas por você no repositório, você commita essas mudanças. Um commit contém as alterações que foram feitas nele e uma mensagem descritiva, além de informações meta (data, autor, etc).

Faça commit de alterações já completas ou que possam ser completadas por alguém. Nunca separe alterações em pequenos commits de poucas mudanças.

☒ **Branches**

Branches são separações de código. O branch padrão do projeto no Odoo é a versão atual sempre. Branches normalmente são utilizados para separar alterações grandes ou novas funcionalidades do projeto, por exemplo: Existe um projeto de blog, os desenvolvedores já fizeram quase toda a parte do blog, mas existem alterações para fazer no sistema de usuários do blog e algumas a fazer no sistema de posts do blog. Para isso, cria-se uma branch "usuarios" e uma "posts" (ou algo do tipo) e fazem-se as alterações nessas branches, um time trabalha em cada uma dessas branches, enquanto isso, outro time continua trabalhando em pequenas mudanças ou bugfixes na branch master.

☒ **Merge**

Um merge é a união de duas branches, normalmente, merges são feitos na branch master. No exemplo do blog, quando a alteração do blog for terminada, alguém vai unir essas alterações na branch master para que elas possam finalmente fazer parte do projeto de fato. Os merges costumam dar bastante problema, pois os códigos podem (e provavelmente vão entrar em conflito). Se houverem alterações no mesmo arquivo ou o git não conseguir definir se alguma linha deve ou não entrar no projeto por motivo de conflito, essas alterações deverão ser corrigidas manualmente.

☒ **Clone**

Um clone de um repositório funciona como uma branch de um repositório online em um repositório local. Ou seja, quando se deseja trabalhar em um repositório hospedado no github, clona-se esse repositório para o seu computador, para poder trabalhar nele.

☒ **Pull**

É uma atualização do repositório local. Se estiver trabalhando com vários desenvolvedores chega um momento que você necessita buscar as modificações que eles fizeram, um pull baixa as últimas alterações remotas para o seu computador.

☒ **Push**

Envia (ou tenta enviar) o código para o repositório online.

☒ Fork

O fork é como um clone, porém dentro do github. Isso quer dizer que o repositório não vai ser baixado para seu computador, vai ser feita uma cópia para a sua conta, no fork é possível fazer qualquer modificação.

☒ Pull Request

Um pull request é um pedido que se faz ao dono do repositório para que esse atualize o código dele com o seu código. Ou seja, você pede para que o dono do projeto ao qual você quer contribuir adicione suas modificações ao projeto oficial.

3. Informações sobre como compilar/construir os componentes da aplicação e transformá-la em uma unidade instalável.

Instalação e configuração do servidor PostgreSQL no Ubuntu

Usamos o seguinte comando no prompt de comando do seu sistema para instalar o postgresql

```
fernanddo@openerp-fernanddo:/$ sudo apt-get install postgresql
```

Para uma interface gráfica de usuário do PostgreSQL

```
fernanddo@openerp-fernanddo:/$ sudo apt-get install pgadmin3
```

3.1. Instalação de Servidor OpenERP

Instalando os pacotes necessários

Python 2.6 ou posterior é necessário para OpenERP. Ele está embutido no Ubuntu versão 10.04 ou superior. Algumas bibliotecas Python também são necessárias, conforme listado abaixo.

Em uma distribuição Linux baseada no Debian podemos instalar todas as dependências necessárias com este único comando

```
fernanddo@openerp-fernanddo:/$ sudo apt-get install python-dateutil  
python-docutils python-feedparser python-jinja2 python-ldap python-
```

```
libxslt1 python-lxml python-mako python-mock python-openid python-
psycpg2 python-psutil python-pybabel python-pychart python-pydot python-
pyparsing python-reportlab python-simplejson python-tz python-unittest2
python-vatnumber python-vobject python-webdav python-werkzeug python-xlwt
python-yaml python-zsi poppler-utils python-pip python-pypdf python-
passlib python-decorator gcc python-dev mc bzip python-setuptools python-
markupsafe python-reportlab-accel python-zsi python-yaml python-argparse
python-openssl python-egenix-mxdatetime python-usb python-serial lptools
make python-pydot python-psutil python-paramiko poppler-utils python-
pdftools antiword python-requests python-xlsxwriter python-suds python-
psycogreen python-ofxparse python-gevent
```

3.2. Dependências Odoo Web

```
sudo apt-get install -y npm
sudo ln -s /usr/bin/nodejs /usr/bin/node
sudo npm install -g less less-plugin-clean-css
```

Efetamos o download do Repositório

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |
sudo apt-key add -
```

3.3. Instalação do Componente Gdata

```
Sudo wget https://pypi.python.org/packages/a8/70/
bd554151443fe9e89d9a934a7891aaffc63b9cb5c7d608972919a002c03c/gdata-
2.0.18.tar.gz
sudo tar zxvf gdata-2.0.18.tar.gz
sudo chown -R odoo: gdata-2.0.18
sudo -s
cd gdata-2.0.18/
python setup.py install
exit
```

Obtemos o último Odoo 10 do depósito github, seguindo os seguintes procedimentos:

```
cd /opt/odoo
sudo wget https://github.com/odoo/odoo/archive/10.0.zip
sudo unzip 10.0.zip
```

Ou fazemos um Git Clone Odoo

```
git clone --depth=1 --branch=10.0 https://github.com/odoo/odoo.git /opt/odoo/odoo
```

Passamos para a instalação do WKHTMLTOPDF

<https://www.getopenerp.com/wkhtmltopdf-for-odoo8/>

A partir daí já podemos iniciar o servidor Odoo

```
cd /opt/odoo/odoo-10.0  
./odoo-bin
```

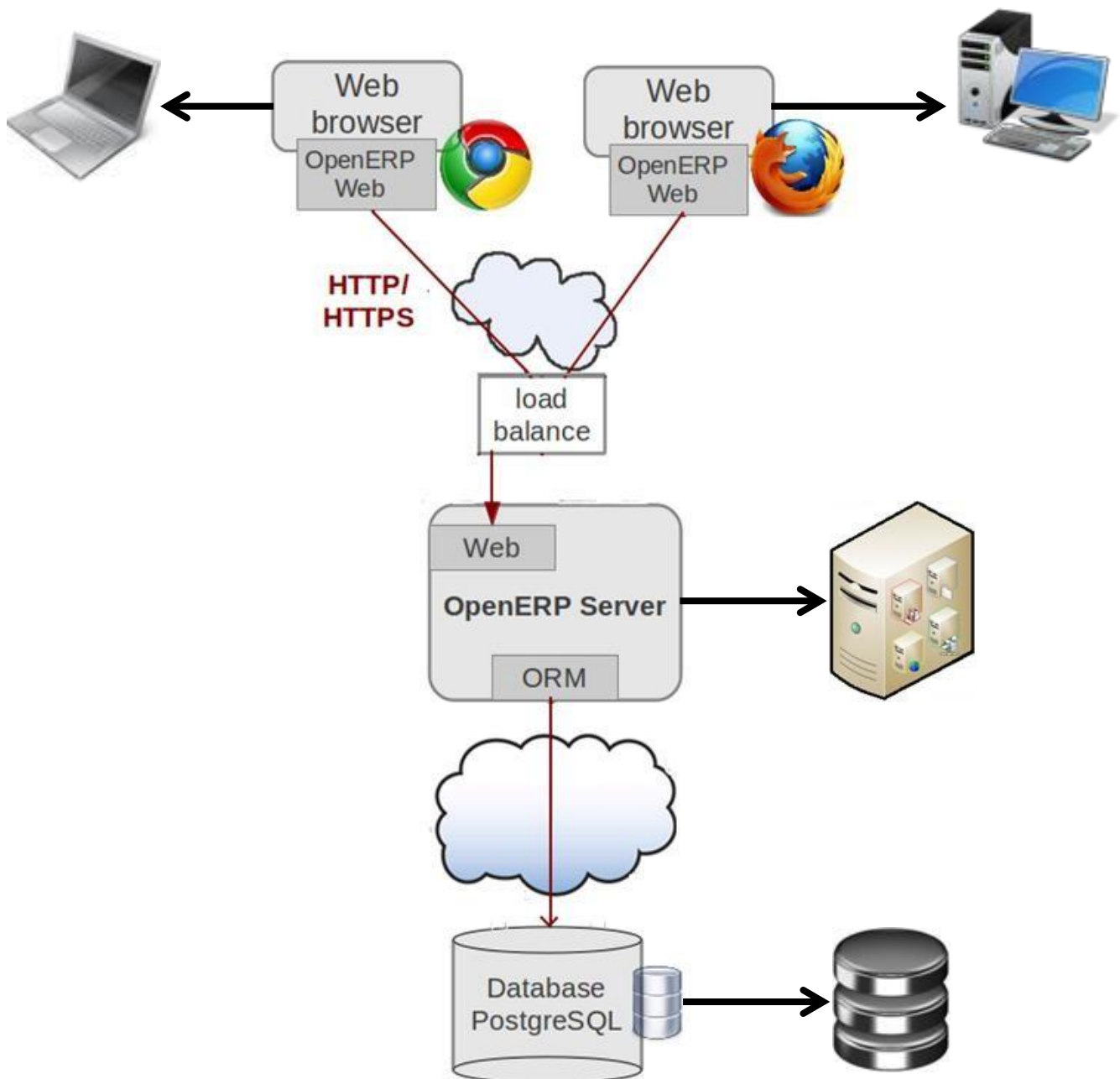
Evidente que essas são as etapas principais de instalação e compilação dos componentes do OpenERP (Odoo), existem outros procedimentos necessários para seu perfeito funcionamento como cópias de arquivos entre diretórios, permissões em arquivos e pastas, edição de arquivos .conf, os quais não citamos pois o importante era mencionar apenas os elementos principais da aplicação.

4. Uma estimativa da infraestrutura e ferramentas de gerenciamento de configuração que podem ser usadas para gerenciar a instalação da aplicação.

A arquitetura completa do OpenERP inclui os seguintes elementos:

- ✓ Um servidor de banco de dados. Postgresql 9.4 ou superior
- ✓ Um Servidor de Aplicação OpenERP. Linux Ubuntu 16.04
- ✓ Um Servidor de Cliente web OpenERP. Linux Ubuntu 16.04
- ✓ VMWare Workstation Pro para Instalação e Gerenciamento dos Servidores

5. Um esboço de um plano de implantação da aplicação em sua estrutura corporativa.



6. Escreva um playbook Ansible que instale a aplicação escolhida por você para realização do TP1. Seu playbook deve automatizar o máximo possível de passos de instalação e configuração, e deve incluir comentários (e diretivas "name") que permitam compreender facilmente o que está sendo realizado.

[group vars/odoo](#)

```
+pacotes_odoo: [
+    "adduser",
+    "ghostscript",
+    "postgresql-client",
+    "python",
+    "python-pip",
+    "python-imaging",
+    "python-pychart",
+    "python-libxslt1",
+    "xfonts-base",
+    "xfonts-75dpi",
+    "libxrender1",
+    "libxext6",
+    "fontconfig",
+    "python-zsi",
+    "python-lasso",
+    "libzmq3-dev",
+    "libpq-dev",
+    "python-dev",
+    "libffi-dev",
+    "libxml2-dev",
+    "libxslt1-dev",
+    "libldap2-dev",
+    "libsasl2-dev",
+    "libssl-dev",
+    "libjpeg8-dev",
+]
```

Hosts

```
[odoo]
192.168.2.140
```

Ansible.cfg

```
[defaults]
+[defaults]
+inventory = inventory/
+library = library/
+roles_path = roles:/etc/ansible/roles
+remote_user = ubuntu
+log_path = ~/ansible.log
+vault_password_file = ~/.ssh/id_rsa
+ansible_managed = fernanddo.it managed
+retry_files_enabled = False
+host_key_checking = False
+private_key_file = ~/.ssh/id_rsa
+
+[privilege_escalation]
+become = True
+become_method = sudo
+become_user = root
+
+[persistent_connection]
+connect_timeout = 30
+connect_retries = 30
+connect_interval = 1
```

playbook.yml

```
---
- hosts: odoo
  gather_facts: no
  pre_tasks:
    - name: instalar python 2
      raw: sudo bash -c "test -e /usr/bin/python || (apt -y update && apt
-y install python-minimal aptitude)"
    - name: gather facts
      setup:
    - name: faz apt update e upgrade dos pacotes
      apt:
        upgrade: yes
        update_cache: yes
        cache_valid_time: 3600
  tasks:
```

```

- name: instala as dependencias do odoo
  apt:
    name: "{{ item }}"
    state: present
  with_items: "{{ pacotes_odoo }}"
- name: cria o link nodejs
  file:
    src: /usr/bin/nodejs
    dest: /usr/bin/node
    state: link
- name: instala os pacotes less via npm
  npm:
    name: "{{ item }}"
    global: yes
  with_items:
    - less
    - less-plugin-clean-css
- name: adiciona o repositório postgres
  apt_repository:
    repo: deb http://apt.postgresql.org/pub/repos/apt/ xenial-pgdg
main
    state: present
- name: adiciona a chave do repositório postgres
  apt_key:
    url: "https://www.postgresql.org/media/keys/ACCC4CF8.asc"
    state: present
- name: instala o pacote postgres
  apt:
    name: postgresql-9.6
    update_cache: yes

```

Como sabemos, o ansible faz utilização do serviço ssh para que sejam aplicadas as devidas configurações em todos os servidores que por ele forem gerenciados. Por isso primeiramente iremos instalar o serviço ssh.

```
fehr@ubuntu: ~  
fehr@ubuntu:~$ sudo apt-get install openssh-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Suggested packages:  
  ssh-askpass rssh molly-guard monkeysphere  
The following NEW packages will be installed:  
  openssh-server  
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.  
Need to get 0 B/338 kB of archives.  
After this operation, 912 kB of additional disk space will be used.  
Preconfiguring packages ...  
Selecting previously unselected package openssh-server.  
(Reading database ... 176119 files and directories currently installed.)  
Preparing to unpack .../openssh-server_1:7.2p2-4ubuntu2.2_amd64.deb ...  
Unpacking openssh-server (1:7.2p2-4ubuntu2.2) ...  
Processing triggers for ufw (0.35-0ubuntu2) ...  
Processing triggers for systemd (229-4ubuntu17) ...  
Processing triggers for ureadahead (0.100.0-19) ...  
Processing triggers for man-db (2.7.5-1) ...  
Setting up openssh-server (1:7.2p2-4ubuntu2.2) ...  
Creating SSH2 RSA key; this may take some time ...  
2048 SHA256:rvLnzpd2uClGAbOqG5FhdhPPfIZq4/j+xmKVL9vx28Y root@ubuntu (RSA)  
Creating SSH2 DSA key; this may take some time ...  
1024 SHA256:snwyGI3PEoauwF2LoXSUYjMQ5NpcJXOLm9sNF0XPY5E root@ubuntu (DSA)  
Creating SSH2 ECDSA key; this may take some time ...  
256 SHA256:YkjYQvNUy+2f8akKIdotda5mv420U/g3gS/3VPYjLBo root@ubuntu (ECDSA)  
Creating SSH2 ED25519 key; this may take some time ...  
256 SHA256:U4NpvuEb5lQCNCCbZQwuNyMrNseDKV+11lXq9wisLo0 root@ubuntu (ED25519)  
Processing triggers for systemd (229-4ubuntu17) ...  
Processing triggers for ureadahead (0.100.0-19) ...  
Processing triggers for ufw (0.35-0ubuntu2) ...
```

Logo após essa instalação, optamos por uma forma de efetuarmos o login, sem a necessidade de senha. Além do mais, para eliminar o login com senha usamos chaves SSH, que oferecem um sistema de autenticação muito mais seguro

```
fehr@ubuntu: ~  
fehr@ubuntu:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/fehr/.ssh/id_rsa):  
Created directory '/home/fehr/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/fehr/.ssh/id_rsa.  
Your public key has been saved in /home/fehr/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:7OM2Y4NzJbrBLpzuXtCofYjPLBRagDvBNy8Jd5x+Xp4 fehr@ubuntu  
The key's randomart image is:  
+---[RSA 2048]---+  
.. ..  
oo.+ +  
o+.*  
o oo+ o .  
 . oo.+ S .  
 . B +.E .  
 oo.Bo+o  
  ===+B.  
  ++o**+.  
+---[SHA256]-----+  
fehr@ubuntu:~$  
ubuntu
```


Agora executamos um comando para nosso servidor SSH entender que a chave que acabamos de gerar é autorizada para logins remotos. Assim, armazenamos uma cópia da chave em um caminho padrão para o serviço SSH:

```
fehr@ubuntu:~$ cp -p ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

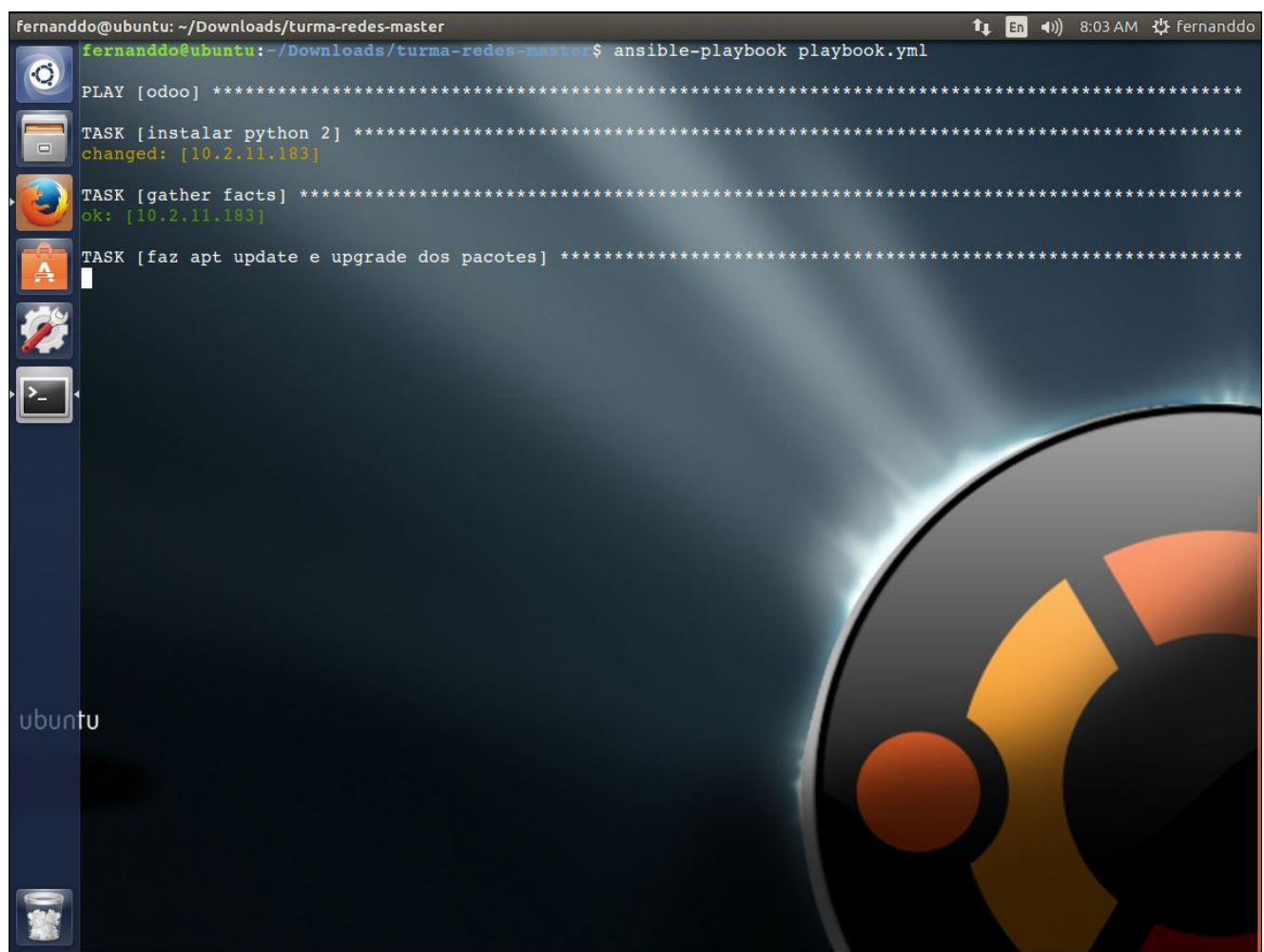
Após isso efetuamos um teste de acesso para verificação

```
fehr@ubuntu$ ssh localhost
```

```
Last login: wed Jun 21 17:22:33 2017 from localhost
```

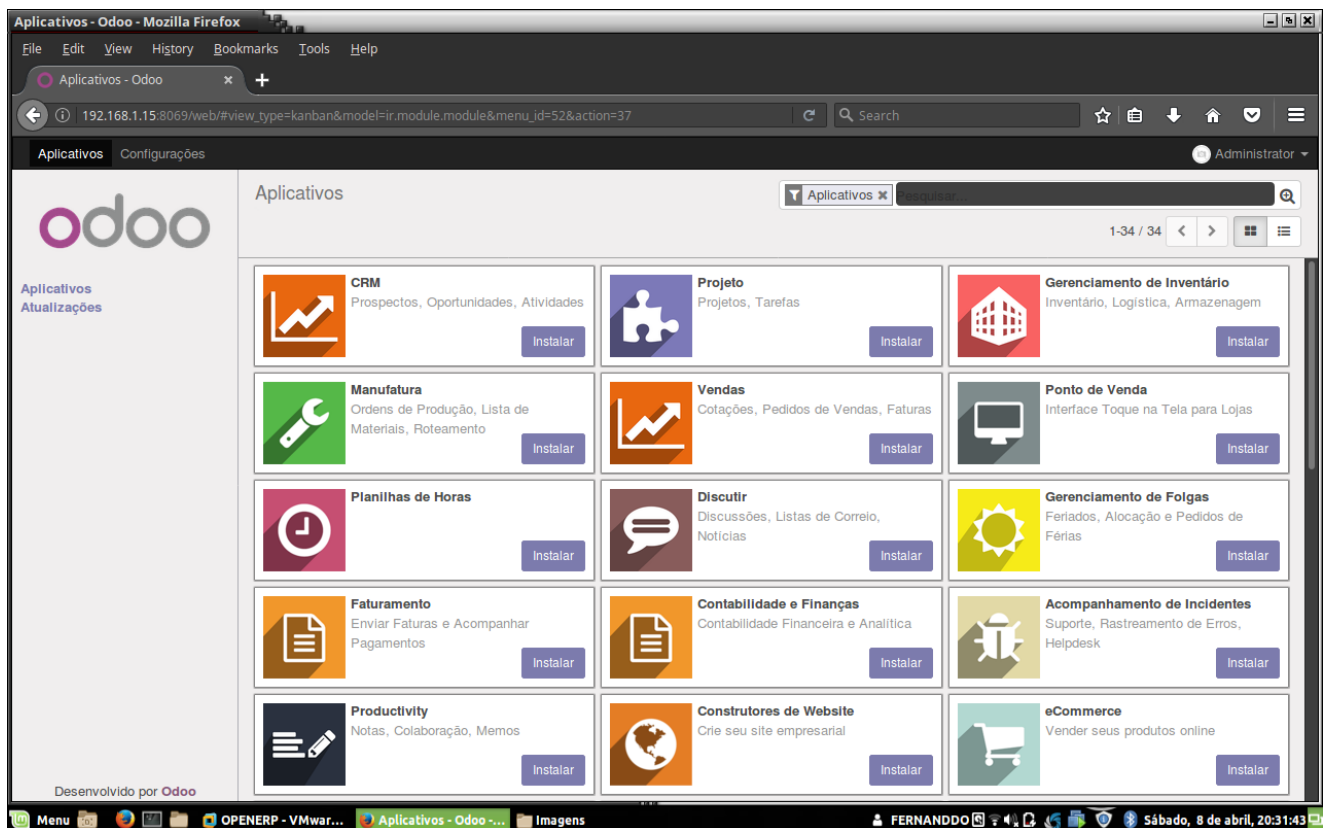
```
fehr@ubuntu$ [Nota: Você está dentro da sessão SSH]
```

Assim, já podemos dar início ao nosso Playbook para instalação de nossa aplicação



Optamos por abreviar as informações por entendermos que já possuímos uma quantidade suficiente de informações para a instalação de nossa aplicação.

Assim, nossa aplicação já se encontra funcionando e acessível



Certamente que tivemos que abreviar um pouco do processo para que não ficasse tantas informações desnecessárias, certamente com o Playbook, nosso ambiente se torna mais dinâmico e automatizado, tenho em mãos configurações que podem nos auxiliar na automatização de qualquer servidor que quisermos, sempre com eficiência e produtividade.

7. Documente em um pequeno relatório a criação de um repositório público para armazenar o playbook desenvolvido no TP da etapa 2. Você também precisa capturar as telas do processo de inicialização, do primeiro commit e do primeiro push do seu código. Faça também uma alteração simples no seu código (pode ser até mesmo a modificação de um README.md) e documente o processo de commit e push. Por fim, inclua no seu relatório o link para seu repositório, que será consultado pelo professor e precisa refletir as operações documentadas em seu texto.

Para esta etapa, efetuamos os comandos necessários para que nosso objetivo fosse alcançado e assim nosso projeto estivesse disponibilizado de forma pública, onde pode ser consultado. Os comandos necessários estão descritos assim:

```
echo "# ProjetodeBloco" >> README.md
```

```
git init
git add README.md
git commit -m "first commit"
git remote add origin
https://github.com/FernanddoMoura/ProjetodeBloco.git
git push -u origin master
```

...or push an existing repository from the command line

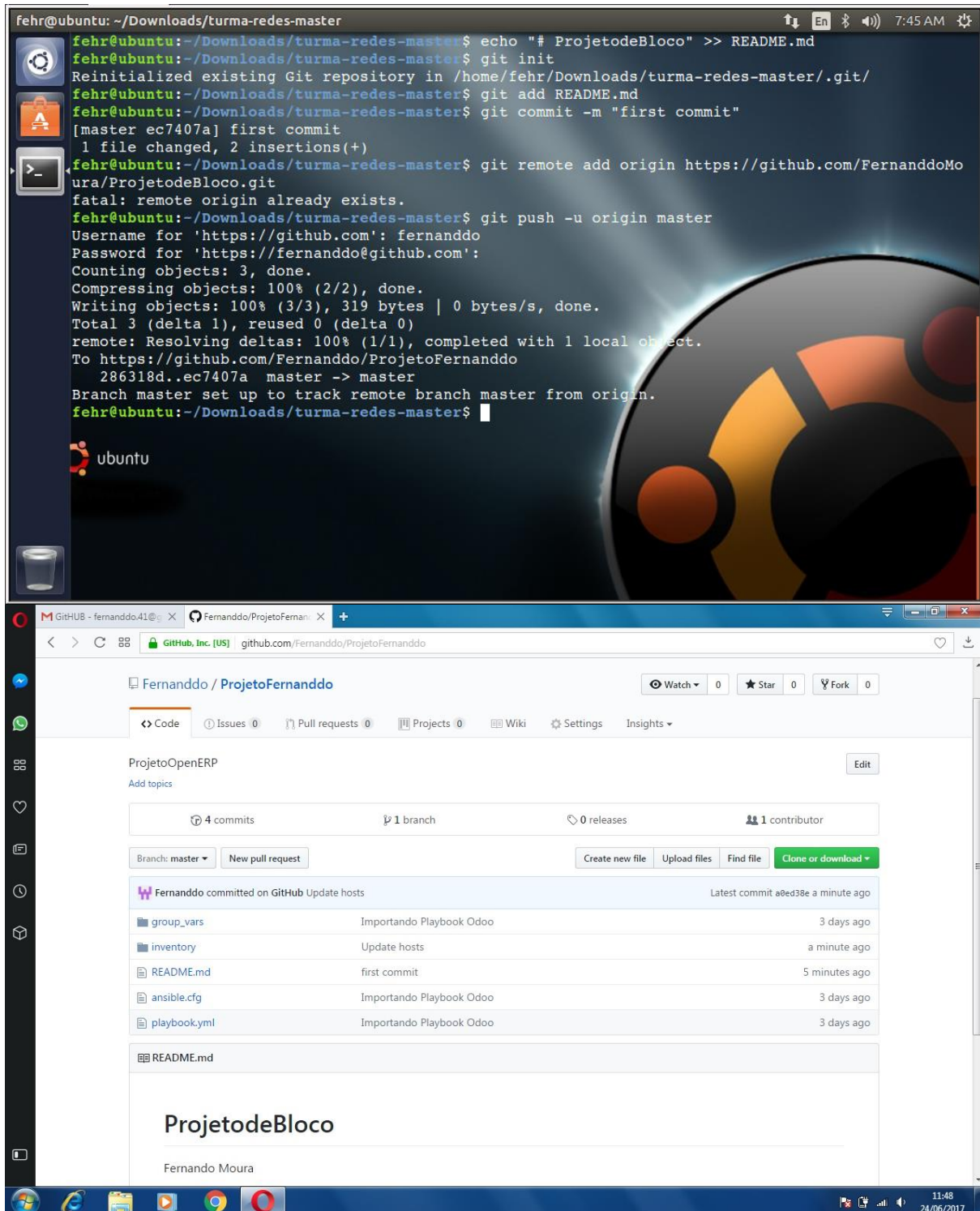
```
git remote add origin
https://github.com/FernanddoMoura/ProjetodeBloco.git
git push -u origin master
```

Dessa forma nosso pimeiro commit fica assim:

A terminal window on an Ubuntu system showing the execution of git commands. The window title is 'fehr@ubuntu: ~/Downloads/turma-redes-master'. The terminal output shows the creation of a README.md file, initialization of a git repository, adding the README.md file, committing it with the message 'first commit', adding a remote origin, and pushing the master branch to the remote repository. The background of the terminal window features a large, stylized graphic of a globe with orange and yellow segments.

```
fehr@ubuntu: ~/Downloads/turma-redes-master
fehr@ubuntu:~/Downloads/turma-redes-master$ echo "# ProjetodeBloco" >> README.md
fehr@ubuntu:~/Downloads/turma-redes-master$ git init
Reinitialized existing Git repository in /home/fehr/Downloads/turma-redes-master/.git/
fehr@ubuntu:~/Downloads/turma-redes-master$ git add README.md
fehr@ubuntu:~/Downloads/turma-redes-master$ git commit -m "first commit"
[master 286318d] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
fehr@ubuntu:~/Downloads/turma-redes-master$ git remote add origin https://github.com/FernanddoMoura/ProjetodeBloco.git
fatal: remote origin already exists.
fehr@ubuntu:~/Downloads/turma-redes-master$ git push -u origin master
Username for 'https://github.com': Fernanddo
Password for 'https://Fernanddo@github.com':
Counting objects: 11, done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (11/11), 1.62 KiB | 0 bytes/s, done.
Total 11 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Fernanddo/ProjetoFernanddo
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
fehr@ubuntu:~/Downloads/turma-redes-master$
```

Após termos nossa sequência de comandos necessária acontecendo de forma bem sucedida, nosso projeto já recebe nossas primeiras alterações conforme figura acima. Assim após as alterações que vão acontecendo, já efetuamos nosso segundo commit, e push, conforme a figura a seguir:



The image shows a terminal window and a web browser. The terminal window displays the following commands and output:

```
fehr@ubuntu: ~/Downloads/turma-redes-master
fehr@ubuntu:~/Downloads/turma-redes-master$ echo "# ProjetoBloco" >> README.md
fehr@ubuntu:~/Downloads/turma-redes-master$ git init
Reinitialized existing Git repository in /home/fehr/Downloads/turma-redes-master/.git/
fehr@ubuntu:~/Downloads/turma-redes-master$ git add README.md
fehr@ubuntu:~/Downloads/turma-redes-master$ git commit -m "first commit"
[master ec7407a] first commit
1 file changed, 2 insertions(+)
fehr@ubuntu:~/Downloads/turma-redes-master$ git remote add origin https://github.com/FernandoMoura/ProjetoBloco.git
fatal: remote origin already exists.
fehr@ubuntu:~/Downloads/turma-redes-master$ git push -u origin master
Username for 'https://github.com': fernando
Password for 'https://fernando@github.com':
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 319 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Fernando/ProjetoFernando
286318d..ec7407a master -> master
Branch master set up to track remote branch master from origin.
fehr@ubuntu:~/Downloads/turma-redes-master$
```

The web browser shows the GitHub repository page for 'Fernando / ProjetoFernando'. The page displays the repository name, a table of commits, and the README content.

File	Commit Message	Time Ago
group_vars	Importando Playbook Odoo	3 days ago
inventory	Update hosts	a minute ago
README.md	first commit	5 minutes ago
ansible.cfg	Importando Playbook Odoo	3 days ago
playbook.yml	Importando Playbook Odoo	3 days ago

The README content is as follows:

```
ProjetoBloco

Fernando Moura
```

Assim nosso projeto fica disponibilizado no seguinte caminho:

<https://github.com/Fernando/ProjetoFernando>

8. Documente a execução de uma aplicação distribuída, composta por dois ou mais containers Docker. Seu trabalho deve incluir um texto de pelo menos uma página explicando que aplicação vai ser executada e quais containers serão usados para isto. O restante da entrega deve incluir uma documentação completa da execução dos containers, e uma captura de tela mostrando o resultado alcançado.

Para atender os requisitos desta tarefa, seguimos com a orientação de escolha e execução do Odoo (OpenERP), que segue em nosso projeto desde o início para que possamos ter não somente organização no mesmo como o desejo de obtermos sucesso no funcionamento de nossa aplicação. Para isso iremos verificar sua execução e seus devidos containers que serão descritos de uma forma detalhada para que tudo ocorra de modo que a mesma possa rodar perfeitamente.

A Seguir veremos a aplicação do Odoo – OpenERP sendo instalada, assim como os containers necessários ao seu funcionamento:

Container Python 3.5:

```
$ sudo apt-get install python3.5
```

Container do gerenciador de pacotes pip:

```
$ sudo apt-get install python-pip
```

Container Docker

```
$ sudo apt-get install docker-compose
```

Container do Postgres

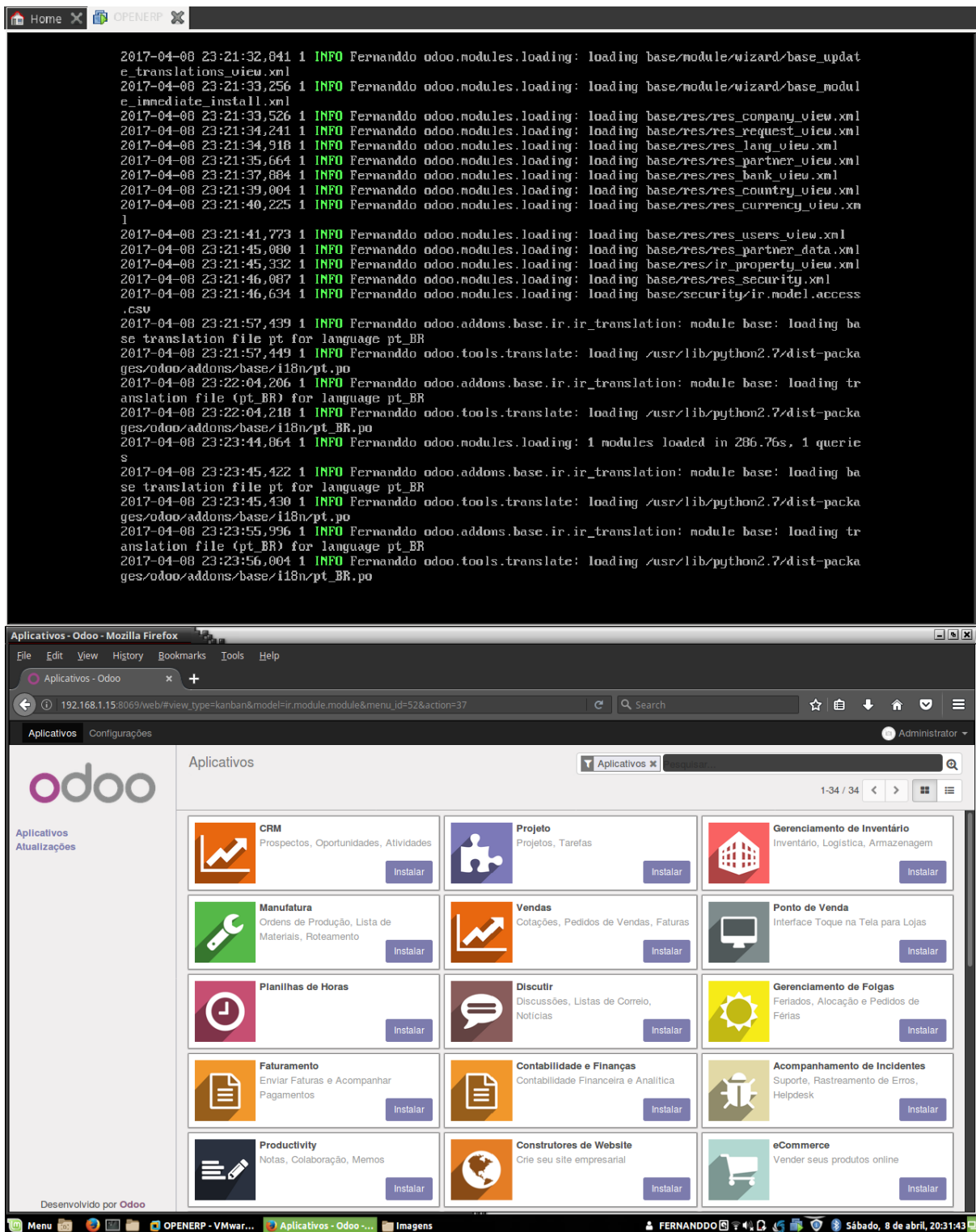
```
$ docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo --  
name db postgres:9.4
```

Container Odoo

```
$ docker run -p 8069:8069 --name odoo --link db:db -t odoo
```


8.1. Execução da Aplicação e de Seus Respectivos Containers

Podemos verificar a seguir a execução dos containers da aplicação.



The image displays two windows from a desktop environment. The top window is a terminal titled 'OPENERP' showing a series of log messages from the Odoo application. The logs indicate the loading of various modules, including base modules like 'base/module/wizard/base_update_translations_view.xml' and 'base/module/wizard/base_module_install.xml', and resource modules like 'base/res/res_company_view.xml', 'base/res/res_request_view.xml', 'base/res/res_lang_view.xml', 'base/res/res_partner_view.xml', 'base/res/res_bank_view.xml', 'base/res/res_country_view.xml', 'base/res/res_currency_view.xml', 'base/res/res_users_view.xml', 'base/res/res_partner_data.xml', 'base/res/ir_property_view.xml', and 'base/res/res_security.xml'. It also shows the loading of addons, including the base.ir.ir_translation module and the base.ir.ir_translation module for the Portuguese language (pt_BR). The logs are timestamped and include the user 'Fernando'.

The bottom window is a web browser titled 'Aplicativos - Odoo - Mozilla Firefox' showing the Odoo application interface. The browser address bar shows the URL '192.168.1.15:8069/web/#view_type=kanban&model=ir_module.module&menu_id=52&action=37'. The interface displays a grid of application modules, each with an icon, a title, a brief description, and an 'Instalar' (Install) button. The modules include:

- CRM**: Prospectos, Oportunidades, Atividades
- Projeto**: Projetos, Tarefas
- Gerenciamento de Inventário**: Inventário, Logística, Armazenagem
- Manufatura**: Ordens de Produção, Lista de Materiais, Roteamento
- Vendas**: Cotações, Pedidos de Vendas, Faturas
- Ponto de Venda**: Interface Toque na Tela para Lojas
- Planilhas de Horas**
- Discutlr**: Discussões, Listas de Correio, Notícias
- Gerenciamento de Folgas**: Feriados, Alocação e Pedidos de Férias
- Faturamento**: Enviar Faturas e Acompanhar Pagamentos
- Contabilidade e Finanças**: Contabilidade Financeira e Analítica
- Acompanhamento de Incidentes**: Suporte, Rastreamento de Erros, Helpdesk
- Productivity**: Notas, Colaboração, Memos
- Construtores de Website**: Crie seu site empresarial
- eCommerce**: Vender seus produtos online

The bottom of the browser window shows the system tray with the date 'Sábado, 8 de abril, 20:31:43' and the user 'FERNANDO'.

9. Apresentação

Para que nossa carreira em tecnologia da informação continue em crescimento é preciso apresentarmos novos desafios diariamente, avaliando como podemos contribuir para o crescimento da empresa que estamos e criando projetos de TI que vão aperfeiçoar os processos e as relações de trabalho. Para buscar destaque de verdade, precisamos aprender a dominarmos a arte de defender suas ideias e convencer as pessoas responsáveis a implementar as soluções nas quais acreditamos. Encontrar prioridades estratégicas para a empresa e explorar as ideias que contribuam de forma efetiva para os objetivos do negócio. Mostrar sintonia com a visão de seus gestores e o alinhamento estratégico do negócio serão essenciais para conseguir aprovação de um projeto de TI.

10. Apresentação da Empresa

A Genstar é uma empresa com foco em serviços gerenciados de tecnologia; Movemo-nos para o aperfeiçoamento contínuo de soluções em tecnologia aplicada à negócios, é assim que buscamos nosso crescimento e de quem valoriza nossa empresa. Foi fundada em 2008, visando atender em parceria, empresas de todos os tamanhos. A Genstar se especializou em maximizar retorno sobre capital investido em tecnologia, otimizando processos, agregando eficiência, qualidade e redução de custos operacionais. Objetivando resultados positivos, a Genstar pôde construir parcerias duradouras com seus clientes. Nossas soluções incluem Mobilidade, Cloud Services, Consultoria no desenvolvimento de soluções móveis e baseadas em nuvem, Service Desk, Field Service, Monitoria e controle de infraestrutura e Segurança da Informação.

As Práticas que nos ajudarão a alcançar uma visão bem sucedida são:

- 1) Adquirir tecnologia que é adaptável interoperáveis e de preferência baseadas em tecnologia open-source;
- 2) Promover a colaboração de todas as entidades na organização Companhia do Café que estão focados em alinhando soluções de TI com os objetivos de negócios;
- 3) Selecionar soluções de arquitetura aberta, que possam acomodar o compartilhamento de dados e a interligação de tecnologias;

- 4) Tornar a informação facilmente acessível através de redes compartilhadas;
- 5) Reduzir o tempo de transação para os usuários de serviços de TI;
- 6) Agressivamente procurar oportunidades para alavancar os investimentos em tecnologia que beneficiem múltiplos departamentos;
- 7) Padronizar o uso da tecnologia no âmbito de entidades organizacionais;
- 8) Conceber e adquirir sistemas que maximizam o uso da informação e tecnologia atual;
- 9) Fazer investimentos adequados em treinamento para a equipe de suporte de TI e usuários; e
- 10) Implementar estratégias para recrutar agressivamente e reter profissionais de TI.

11. Premissas do Nosso Projeto

Sendo uma empresa com foco em serviços gerenciados de tecnologia; Movemo-nos para o aperfeiçoamento contínuo de soluções em tecnologia aplicada a negócios. É assim que buscamos nosso crescimento e de quem valoriza nossa empresa. Buscamos em um momento, acompanhar a tendência da atualidade que nos leva a buscar dois importantes alvos que são:

- ✓ Implantação, desenvolvimento e treinamento da crescente tecnologia da virtualização, onde utilizaremos a Aplicação VMWare Workstation Pro.
- ✓ Implementação e desenvolvimento de uma aplicação, acordo com as necessidades do mercado administrativo, sendo essa aplicação o Odoo (Antes chamado de OpenERP).

Já seguindo o caminho da virtualização, implementaremos o nosso Sistema ERP, que poderá ser disponibilizado tanto no ambiente local, como em nuvem, se assim for apresentada a necessidade e com uma vantagem considerável por ser uma aplicação que roda tanto em Windows como em Linux

12. Requisitos do Projeto

A virtualização de servidor permite que vários sistemas operacionais sejam executados em um único servidor físico como máquinas virtuais, cada uma com acesso aos recursos computacionais do servidor subjacente.

A virtualização de rede para reprodução completa de uma rede física no software. Os aplicativos são executados na rede virtual exatamente da mesma maneira como se

estivessem em uma rede física.

Implantar desktops como serviços gerenciados oferece a oportunidade de responder com mais rapidez a mudanças de oportunidades e necessidades.

Reestruturação da Infraestrutura de Rede para atender a demanda do Planejamento de todo o ambiente.

Investimento considerado em melhorias do Serviço de Internet, com um plano de segurança bem elaborado para que os acessos externos sejam protegidos.

Implementação do Odoo (OpenERP), sendo este desenvolvido de acordo com as necessidades que surgirem, sendo um Sistema ERP que irá atender completamente às necessidades administrativas da Genstar e assim poderá ser disponibilizado para outras empresas que demonstrarem interesse.

Contexto

Uma análise no mercado de tecnologia, nos fez identificar a necessidade de investimento em uma aplicação ERP, que engloba muitas funções que são imprescindíveis à administração de uma empresa, muitos deles possui um alto custo, seja de aquisição, manutenção e até mesmo treinamento. Nossa proposta envolve a migração das aplicações para virtualização e Implantação de um Sistema ERP Opensource.

Proposta

Apresentação do Odoo (OpenERP) que trará um novo conceito de aplicação ERP para empresas, que possui um código aberto e totalmente opensource, ou seja, não agregará nenhum custo para a empresa.

13. Por que VMware

Fornecedora confiável da plataforma escolhida por mais de 500 mil clientes globalmente, a VMware é pioneira em virtualização e inovadora em nuvem e mobilidade corporativa. Com liderança comprovada permite que possamos executar, gerenciar, conectar e proteger aplicativos entre nuvens e dispositivos em um ambiente operacional comum, a fim de obter liberdade e controle.

➤ O que a VMWare oferece ?

vSphere - O vSphere nos possibilita obtermos o que há de melhor em desempenho, disponibilidade e eficiência da infraestrutura e dos aplicativos.

O VMware Integrated OpenStack - fornece controle dos recursos de nuvem por meio de APIs do OpenStack. Incentiva o desenvolvedor a inovar ao implantar e gerenciar com rapidez e facilidade um OpenStack de nível de produção na sua infraestrutura VMware

O VMware vCenter Server - oferece uma plataforma centralizada para gerenciar seus ambientes VMware vSphere para que você possa automatizar e oferecer uma infraestrutura virtual com confiança.

O VMware NSX - Plataforma de virtualização de rede do data center definido por software. Permite a criação de redes inteiras em software e as incorpora à camada do hipervisor, abstraídas do hardware físico subjacente. Todos os componentes de rede podem ser provisionados em minutos, sem a necessidade de modificar o aplicativo.

➤ **Qual será o Custo?**

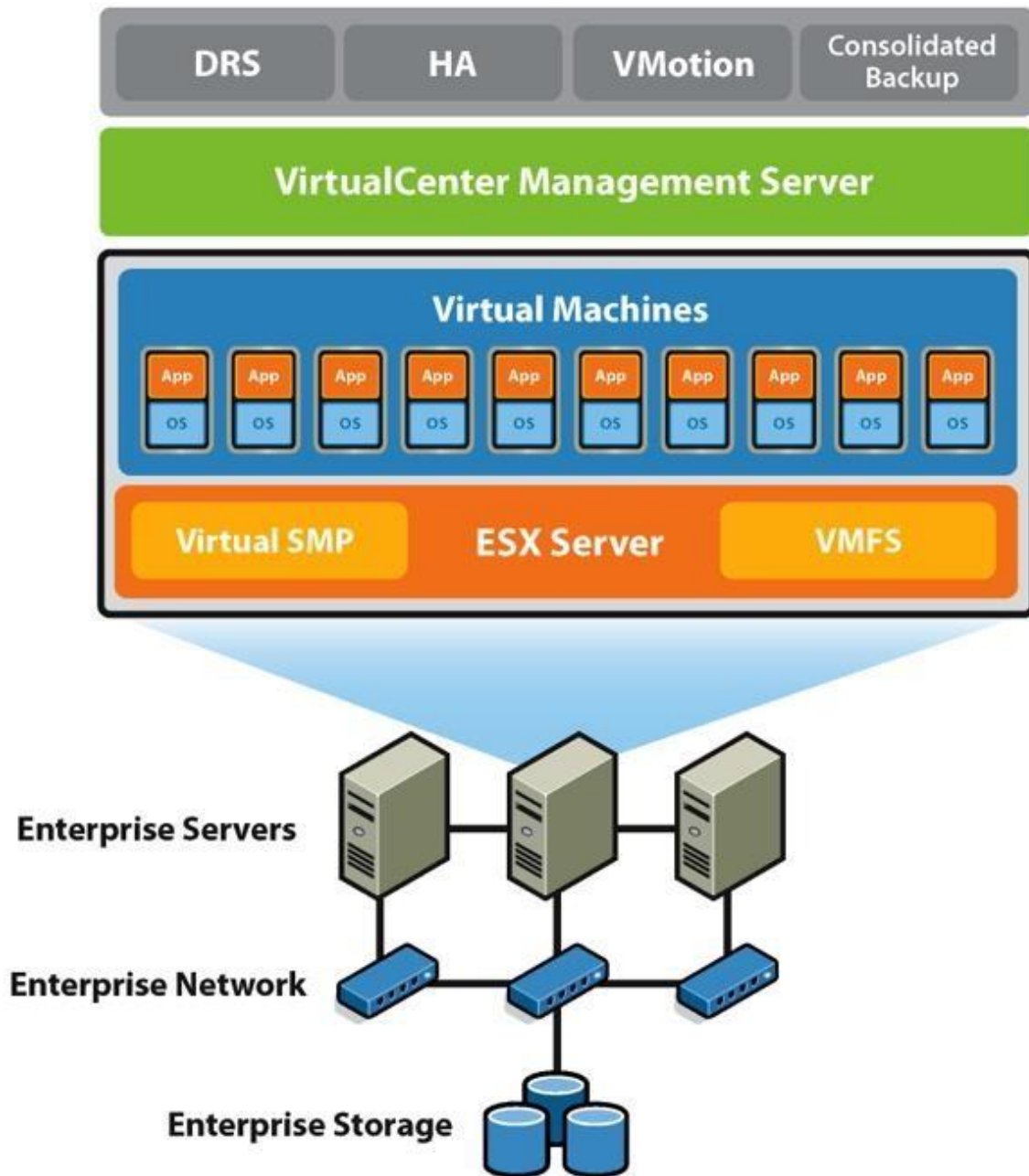
Para essa reestruturação o investimento ficará em torno dos \$ 15.000,00 sendo que este valor é uma previsão devido ao fato dos preços dos produtos VMware estarem sob eventuais reajustes e também pelo fato da variação cambial do dólar estar sofrendo constantes mudanças.

➤ **Qual o Cenário Necessário para Essa Implantação?**

O cenário por estar em fase de mudanças e significativas alterações, a Equipe de TI ainda avalia quais os equipamentos necessários para essa implantação, pois é muito importante do ponto de vista dos negócios evitar possíveis erros que podem fazer com que o projeto não tenha êxito ou traga custos maiores do que o planejado.

- ☒ Falta de redundâncias
- ☒ Super Dimensionamento das máquinas virtuais
- ☒ Super Dimensionamento dos servidores hospedeiros
- ☒ Subdimensionamento do ambiente, principalmente de disco
- ☒ Licenciamento de Aplicações
- ☒ Configuração de rede sem redundâncias
- ☒ Não monitorar a infraestrutura adequadamente

VMware Infrastructure

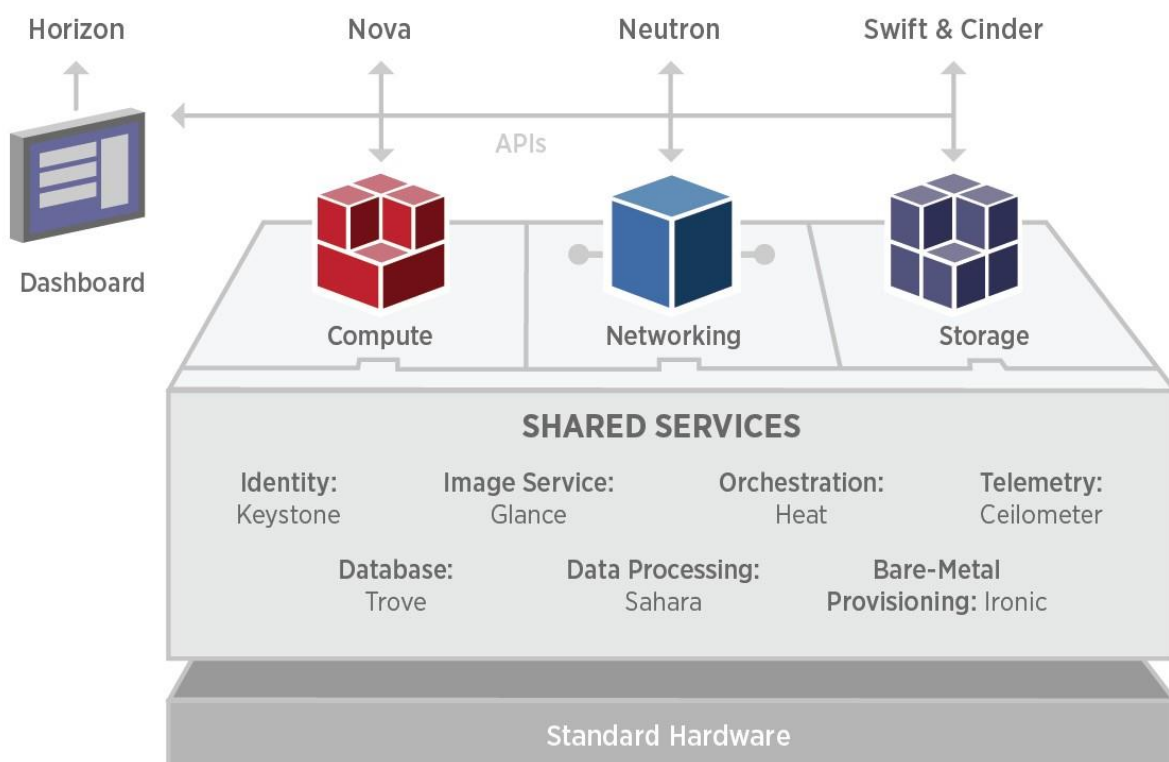


Esta ilustração nos demonstra como fica o cenário da Infraestrutura da Empresa Genstar.com, onde percebemos como ficarão disponíveis os servidores, de Banco de Dados, Web e Aplicação em nosso ambiente, além dos Storages, bem como a rede e as Máquinas Virtuais e o vCenter. Assim, migraremos nossa aplicação para a nuvem, tornando o seu funcionamento e sua disponibilidade mais eficientes e gerenciáveis, proporcionando maior segurança e confiabilidade para a empresa.

14. Comparações entre a forma como você pretende implantar seu ambiente e outras abordagens, usando outras ferramentas.

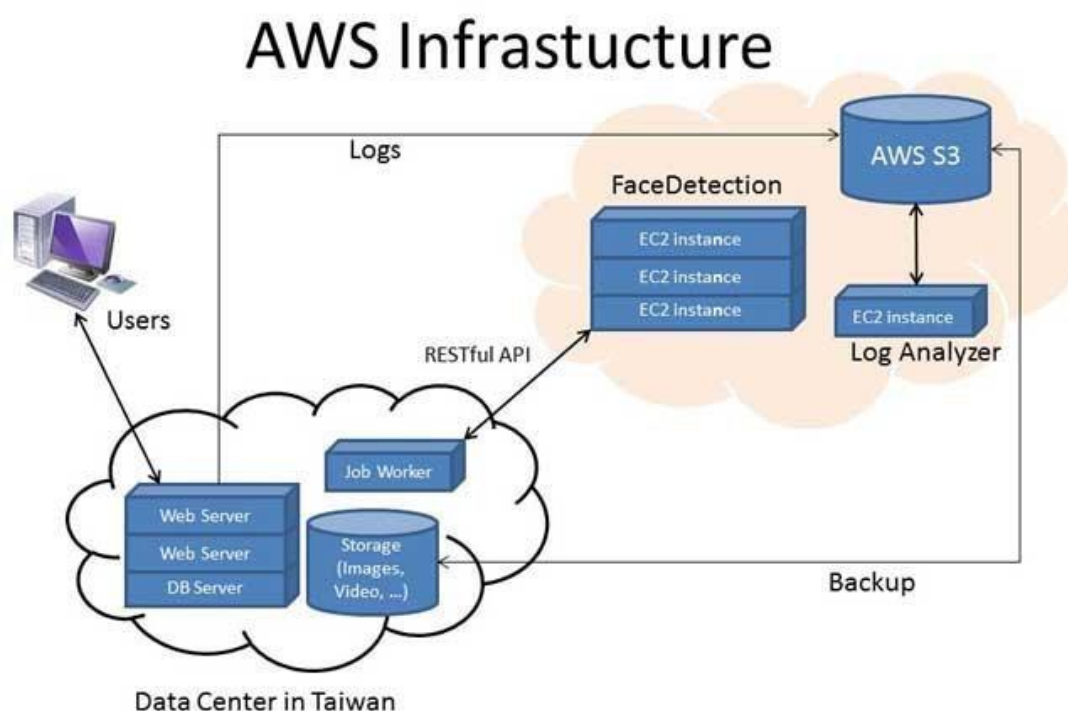
Além da Virtualização de nosso ambiente através da implantação de uma arquitetura VMWare local, diretamente em nossos servidores, poderemos optar por outras duas formas de efetuarmos essa implantação, além de estabelecermos um breve comparativo entre as possibilidades para termos uma noção do que temos em vista para planejamento.

Openstack - O OpenStack é um sistema operacional em nuvem que controla grandes conjuntos de recursos de computação, armazenamento e rede em todo um datacenter, todos gerenciados através de um painel que dá aos administradores o controle ao capacitar seus usuários a fornecer recursos através de uma interface web. O OpenStack controla grandes pools de recursos de computação, armazenamento e rede em um datacenter, gerenciados através de um painel ou através da API OpenStack . OpenStack trabalha com a empresa popular e tecnologias de código aberto tornando-o ideal para infra-estrutura heterogênea.



Através da arquitetura acima podemos entender um pouco sobre como podemos disponibilizar nossa aplicação através da nuvem Openstack, que nos oferece uma forma amplamente segura e gerenciável que nos oferecerá excelente funcionalidade e disponibilidade.

AWS Amazon - A Amazon Web Services oferece um amplo conjunto de produtos de nuvem globais, como computação, armazenamento, bancos de dados, análise, redes, dispositivos móveis, ferramentas do desenvolvedor, ferramentas de gerenciamento, IoT, segurança e aplicações empresariais. Esses serviços ajudam as empresas a se mover mais rapidamente, baixar os custos de TI e escalar.



Essa possibilidade de rodarmos nossa aplicação na Amazon AWS nos oferece uma ampla eficiência e acessibilidade com segurança e confiabilidade oferecidos pela Amazon. Essa é sem dúvida uma possibilidade muito interessante, caso pensemos em um planejamento de migração total de nossa aplicação e infraestrutura.

Apesar de a solução VMware vCloud possuir um custo de licenciamento considerável, por outro lado, temos a facilidade de uso, o suporte do fabricante e a maior disponibilidade de recursos humanos capacitados para manter a solução. O VMware vCloud por ser uma solução “all-in-one” (mesmo sendo uma suíte composta por vários produtos), uma solução de nuvem completa, é menos flexível, as API’s são proprietárias e a interação com as mesmas geralmente é realizada através de Perl, PowerCli e Powershell. Por isso inicialmente temos por positiva essa implantação em nosso ambiente para darmos

estrutura a nossa aplicação distribuída.

O Gerenciamento do código fonte está disponibilizado no GitHub, por se tratar de uma aplicação de código aberto, podemos encontrar uma variedade de códigos desenvolvidos e parâmetros de adequação a cada módulo de instalação do OpenERP(Odoo), bem como os códigos necessários para instalação de outras instâncias como NFe, Localização Brasileira dentre outros. Assim também se aplica ao seu processo de desenvolvimento, pois, sempre iremos encontrar códigos disponíveis que nos ajudarão a fazer com que essa aplicação seja desenvolvida e conseqüentemente sofra um considerável processo de evolução. Não existe um ambiente de desenvolvimento integrado. O código Python deve ser editado em um editor externo. A lógica da aplicação (ou seja, os fluxos de trabalho e estrutura de dados) pode ser alterada através da interface do cliente.

Como pré-requisitos, para a exigência da aplicação, infelizmente não há realmente uma documentação oficial que lhe dê números exatos. É um pouco de um palpite calculado.

1. Se dispomos inicialmente de 8 GB de RAM com 4 núcleos, ele deve funcionar muito bem. Com garantia de termos espaço suficiente disponível também! Por exemplo 500GB em disco.
2. O PostgreSQL continua funcionando bastante agradável e a quantidade de registros não tem muito efeito. Tenha a impressão de que o PostgreSQL precisa de memória RAM suficiente para poder processar todos os dados. Muitas coisas podem ser sintonizadas no banco de dados para melhorar o desempenho. Dê uma olhada nesta documentação:
3. A maioria das pessoas usa o **Ubuntu 14.04** e funciona excelente, ainda não houve problemas com o **Ubuntu 14.04**, mas o aconselhável seria o **Ubuntu 16.04** que é uma versão mais atualizada e com boa estabilidade também. Podemos usar também o **CentOS**, mas há menos documentação disponível para isso. Quanto ao Windows, não seria aconselhável, quando estamos rolando para um cliente ou tantos usuários, os resultados não são tão bons no Windows.
4. Tanto o Google Chrome quanto o Firefox fazem um ótimo trabalho. Em teoria, todos os navegadores são suportados por Odoo, mas não aconselhamos o Internet Explorer ou Edge, estes têm mais problemas do que o Google Chrome e o Firefox.

15. Uma proposta inicial de como organizar a infraestrutura de sua aplicação, de acordo com o sistema de virtualização em que você pretende implementá-la.

Motivados pela possibilidade de redução de custos com hardware, componentes de rede e até mesmo de energia elétrica, encontramos na virtualização a solução de muitos problemas.

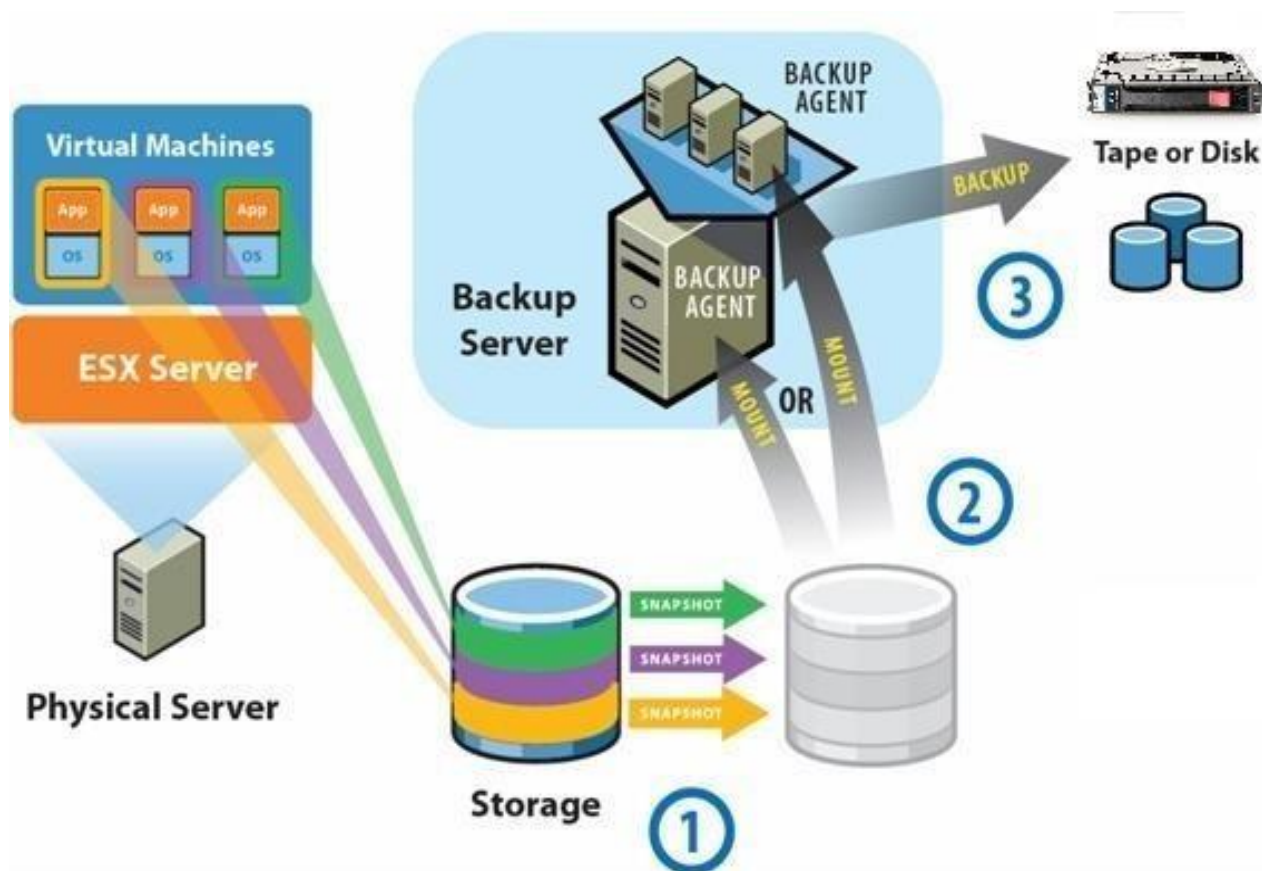
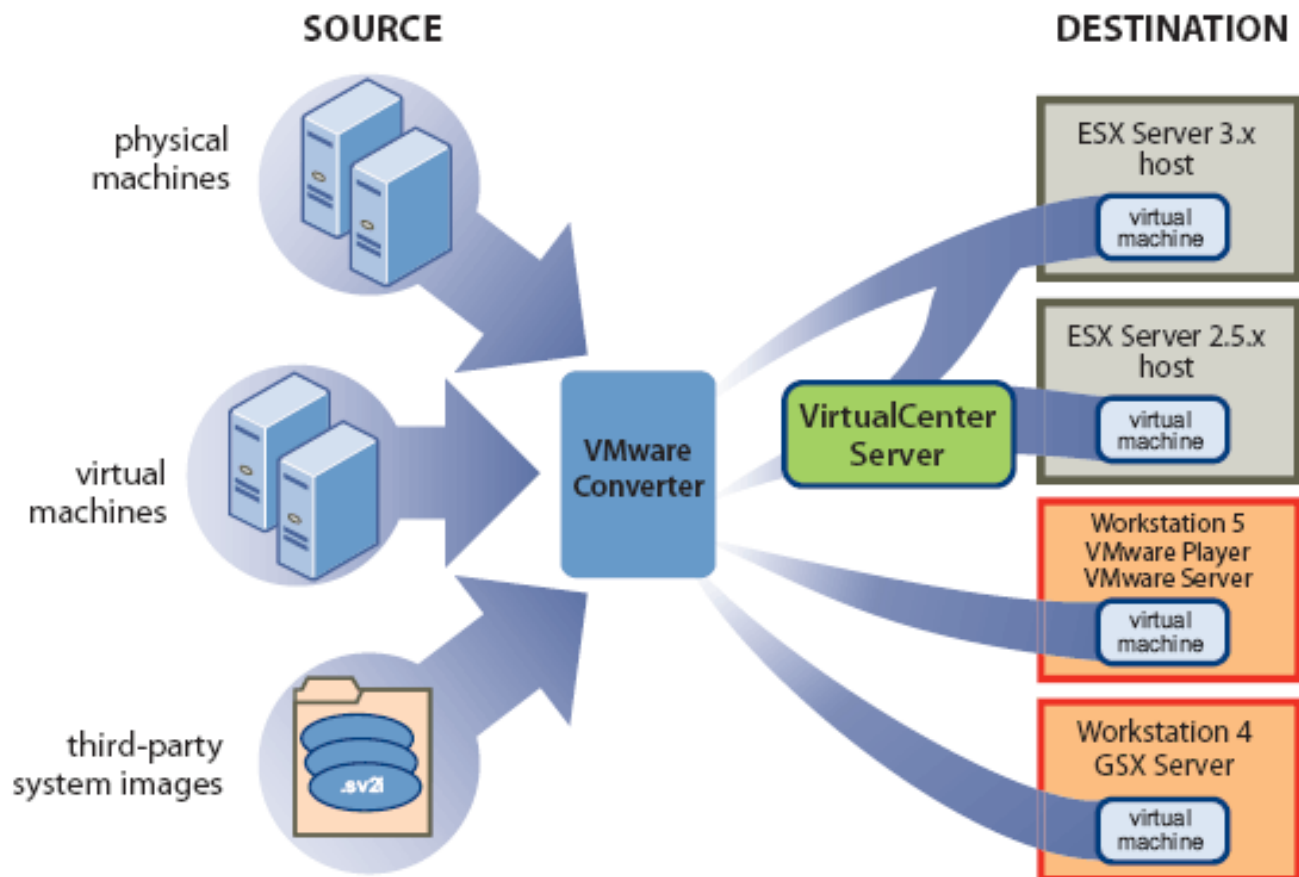
A virtualização também nos proporciona a vantagem de executarmos nossas tarefas de forma mais rápida e segura. A virtualização de servidores pode ser explicada conceitualmente como a reorganização de ambientes operacionais de servidores físicos em servidores virtuais. Essa técnica tem inúmeras vantagens e gera vários resultados positivos para as empresas e os departamentos de TI. Além disso, essa tecnologia leva a otimização de processos, alta confiabilidade, maior desempenho, produtividade e escalabilidade. E, nos dias de hoje, com a concorrência acirrada, aumentar a produtividade, pode fazer toda a diferença nos números da empresa. Dessa forma temos um plano ilustrativo de como será a nossa infraestrutura com essa migração.

Diante de tantos benefícios, é difícil imaginarmos que essa realidade mudará tão cedo.

16. Cronograma de Estimativa de Implantação da Aplicação

FASE DE DESENVOLVIMENTO	INÍCIO EM	EVOLUÇÃO	TÉRMINO EM	SITUAÇÃO
Mapeamento do Processo	03/04/2017	100%	04/04/2017	●
Desenho do Processo	05/04/2017	100%	29/04/2017	●
Aderência do Processo	30/04/2017	100%	04/05/2017	●
Desenvolvimnto dos Requisitos	05/05/2017	100%	14/05/2017	●
Desenvolvimento das Migrações	15/05/2017	100%	24/05/2017	●
Treinamento dos Usuários Chave	25/05/2017	40%	04/06/2017	●
Treinamento dos Usuários Finais	05/06/2017	30%	14/06/2017	●
Homologação da Solução	15/06/2017	50%	19/06/2017	●
Preparação para a Operação	20/06/2017	40%	29/06/2017	●
Operação e Acompanhamento	30/06/2017	0%	03/07/2017	●

17. Implementação da Nossa Aplicação na Infraestrutura da Empresa genstar



18. Uma pequena descrição textual teórica sobre a arquitetura da solução de virtualização que você pretende utilizar.

Para nossa aplicação, iremos dispor da arquitetura de virtualização tipo 2, pois é talvez o conceito mais fácil de entender e que nos atenderá em nosso projeto. Neste cenário, teremos o computador anfitrião físico executando um sistema operacional padrão sem modificações, que será o Ubuntu Linux e a camada de virtualização corre em cima desse SO, sendo na verdade uma aplicação hóspede, no caso o Odoo (OpenERP). Nesta arquitetura, o VMM fornece a cada máquina virtual todos os serviços do sistema físico, incluindo uma BIOS virtual, dispositivos virtuais e memória virtual. Isso tem o efeito de levar o sistema hóspede a acreditar que está a ser executado diretamente sobre o hardware do sistema, ao invés de numa máquina virtual dentro de uma aplicação.

O próprio SO fornecerá a camada de abstração (conhecida como Hipervisor de Tipo 2), de tal forma que permite que outros sistemas residam no seu interior, criando máquinas virtuais. Esta arquitetura pode então ser chamada como Virtualização Hospedada (Hosted Virtualization) já que um OS a “hospeda”, como se pode ver na figura.

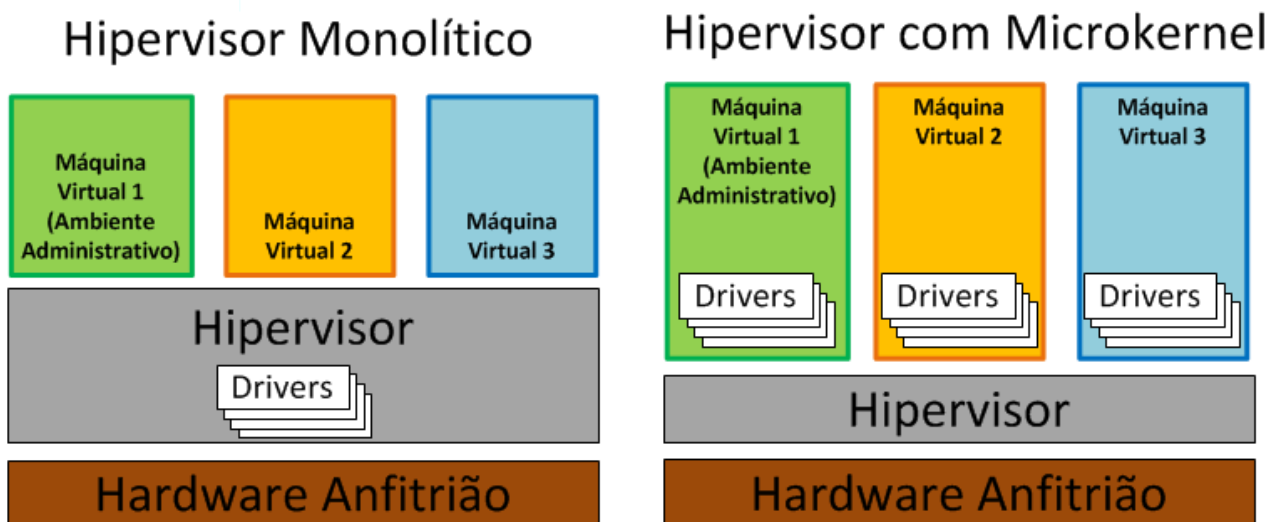


19. Comparações entre a forma como você pretende implantar seu ambiente e outras abordagens, usando outras ferramentas.

Na Arquitetura de Nível 2, as múltiplas camadas de abstração entre os SO hóspedes e o hardware anfitrião subjacente não são propícias a altos níveis de desempenho das máquinas virtuais. No entanto, esta técnica tem a vantagem de não ser necessária nenhuma alteração no SO anfitrião nem em nenhum dos hóspedes e não requer nenhum suporte especial de virtualização da CPU.

Na Arquitetura de Nível 1, Uma vez que tem acesso de baixo nível directo ao hardware, um hipervisor deste tipo é mais eficiente do que uma aplicação hospedada e oferece maior desempenho, pois utiliza menos recursos (sem ciclos de CPU ou espaço de memória separado, como no caso de um verdadeiro SO anfitrião).

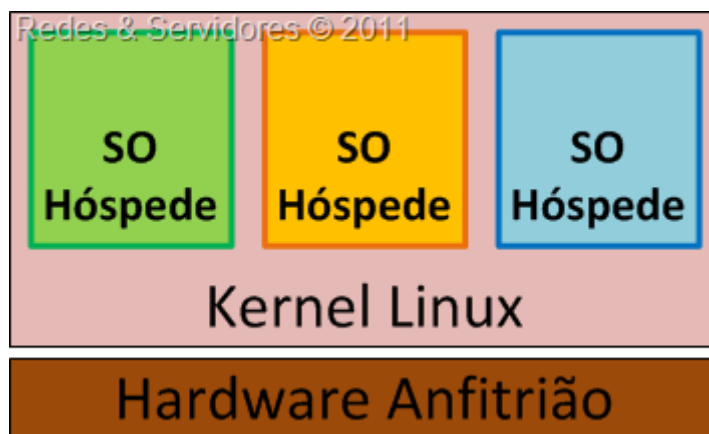
A desvantagem deste modelo é que há dependência do hipervisor para os drivers (pelo menos no caso do hipervisor monolítico). Além disso, a maioria das implementações desta abordagem requerem suporte específica de virtualização a nível do hardware.



Hipervisor Incorporado

Na Virtualização ao Nível do Núcleo, o sistema operativo anfitrião é executado num núcleo (kernel) especialmente modificado que contém extensões projectadas para gerir e controlar várias máquinas virtuais, cada uma contendo um sistema operativo hóspede.

A camada de virtualização é incorporada num núcleo de sistema operativo e cada hóspede executa o seu próprio núcleo, apesar de se aplicarem restrições na medida em que os sistemas operativos hóspedes devem ter sido compilados para o mesmo hardware em que o kernel está em execução.



O benefício real desta abordagem é que o código do hipervisor é dramaticamente mais leve que o do Tipo 1 ou Tipo 2. Com o hipervisor embutido no kernel Linux, os sistemas operativos hóspedes beneficiam de excelente desempenho de disco e E/S de rede.

20. Um planejamento passo a passo (com descrições de cada etapa) de como será feita a implantação da aplicação distribuída virtualizada.

Iremos mostrar um pequeno descritivo, com pelo menos um diagrama mostrando os componentes da aplicação, mais uma página completa de texto descrevendo a arquitetura e suas funcionalidades.

Para acessar OpenERP podemos:

Usar apenas um navegador web apontado para o servidor do cliente-web OpenERP. O melhor é usar o navegador da Web se o servidor OpenERP está a uma certa distância (como em outra localidade), porque é mais tolerante com atrasos. O cliente web também é mais fácil de manter, porque ele geralmente é instalado nos computadores dos usuários.

Um sistema OpenERP é formado a partir de três componentes principais:

O servidor de banco de dados PostgreSQL, que contém todas as bases de dados, cada um dos quais contém todos os dados e a maioria dos elementos da configuração do sistema, OpenERP

O OpenERP usa o Postgres como seu banco de dados, é o único suportado. Quando as instalações do software necessário são feitas, você deve criar um usuário do PostgreSQL. Este usuário deve ser o mesmo que o usuário do sistema. O OpenERP usará esse usuário para se conectar ao PostgreSQL.



☒ **Base de dados**

Sem criar e configurar um usuário do PostgreSQL para OpenERP conforme descrito abaixo, não poderemos criar um banco de dados usando o OpenERP Client. O superusuário padrão para PostgreSQL é chamado postgres . Talvez seja necessário fazer login como esse usuário primeiro. Tornar este novo usuário um superusuário. Só então você poderá criar um banco de dados usando o OpenERP Client. Em resumo, openerp é o novo usuário criado no PostgreSQL

☒ **O servidor de aplicação OpenERP**

Para acessarmos o OpenERP podemos:

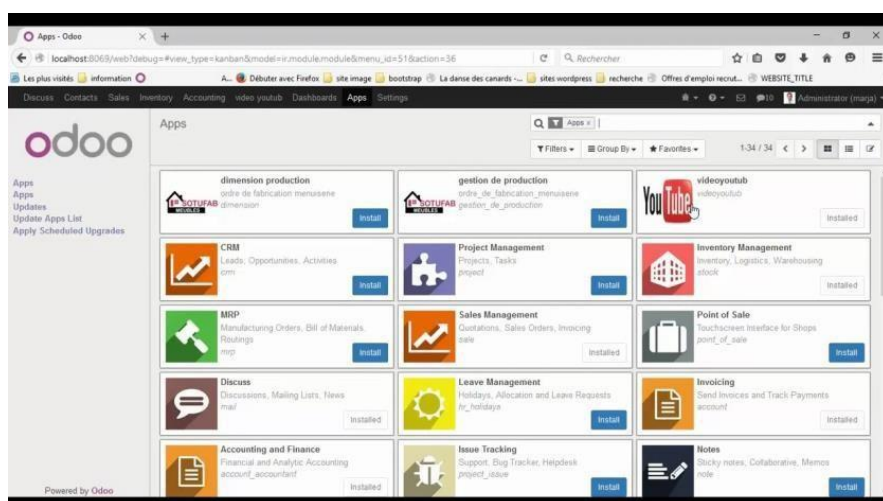
Usarmos um web browser padrão para acesso ao OpenERP client-web server, ou usar um aplicativo cliente (o cliente GTK) instalado em cada computador.

De um lado seria melhor com o aplicativo cliente (chamado de cliente GTK por causa da tecnologia) se você estiver usando um servidor local (como no mesmo prédio). Neste caso, o cliente GTK será mais sensível, de modo mais satisfatório para utilizar. Há pouca diferença funcional entre os dois clientes OpenERP - o cliente web eo cliente GTK no presente. O cliente web oferece mais funcionalidade, por exemplo, o recurso de Inteligência Corporativa, e a vista Gantt. Os dois métodos

de acesso dão instalações muito semelhantes, podendo ser usados ambos no mesmo servidor ao mesmo tempo. É melhor usar o navegador web se o servidor OpenERP estiver a alguma distância (como em outra localidade) porque é mais tolerante aos atrasos de tempo entre os dois do que o cliente GTK. O cliente da web também é mais fácil de manter, porque geralmente já está instalado nos computadores dos usuários.



Tela de Login para Acesso ao OpenERP através do Client GTK



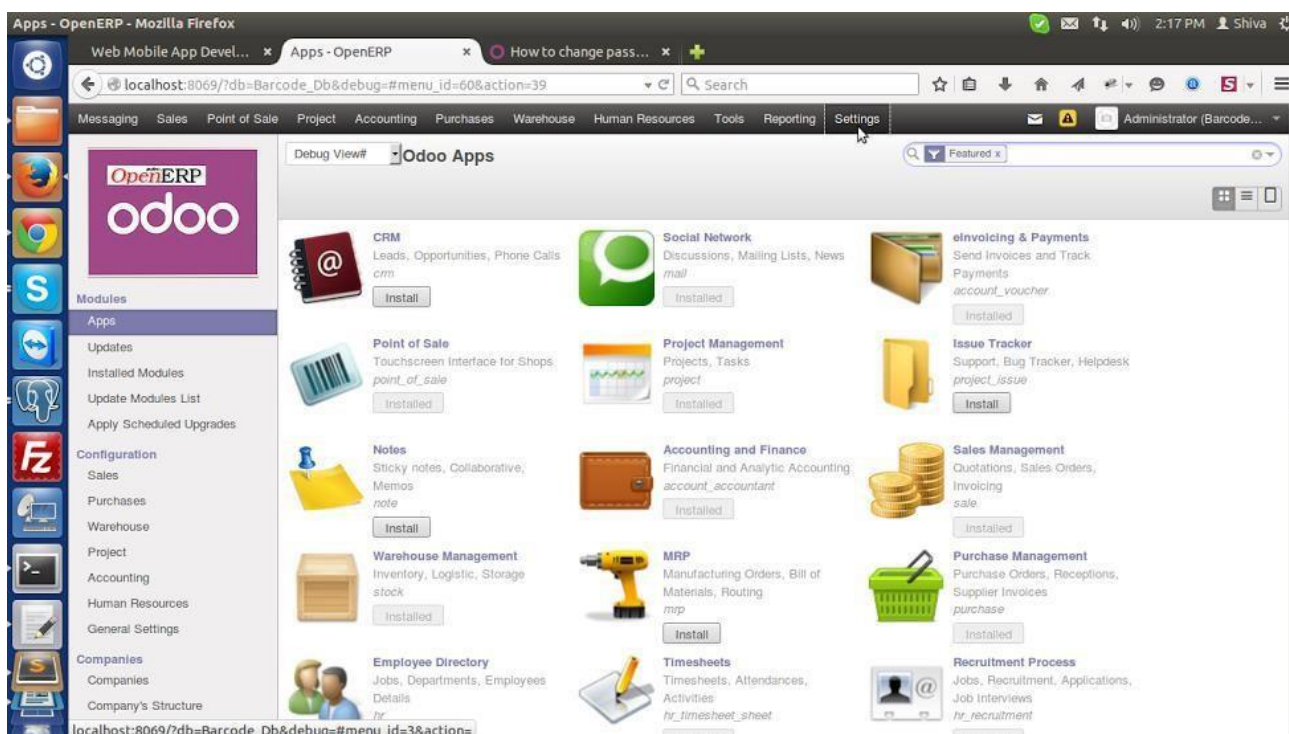
Tela de Login para Acesso ao OpenERP através do Client WEB (Obs. O OpenErp passou a ser chamado Odoo, como vemos na imagem ao lado.)

❌ **O Servidor WEB OpenERP**

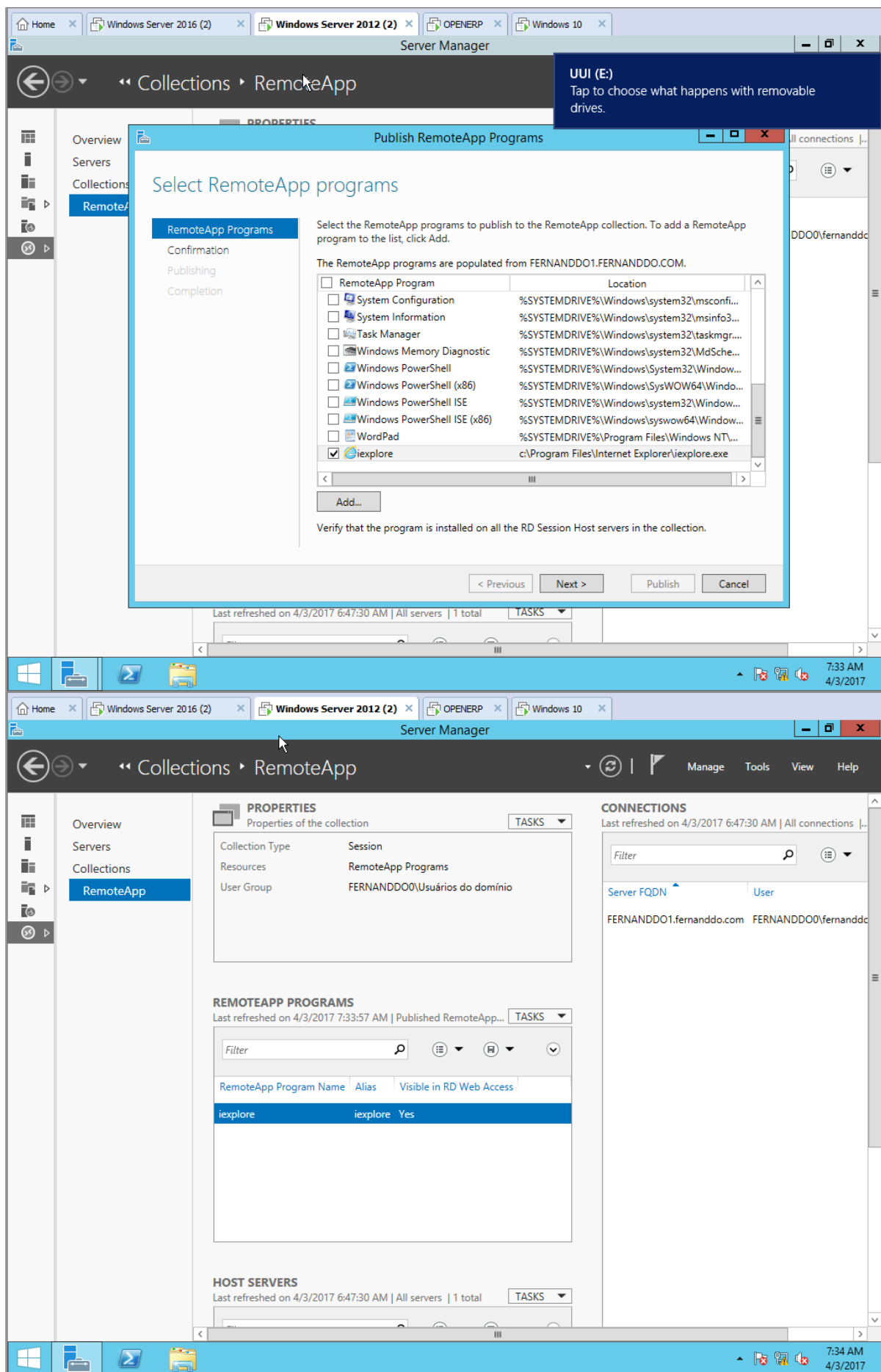
Devemos instalar configurar e executar o OpenERP Server antes de usar o OpenERP Web. O cliente precisa do servidor web para rodar. Devemos instalar o aplicativo do servidor no seu computador, ou em um servidor independente, acessível através da rede. O servidor web, um aplicativo separado chamado o objeto aberto no cliente web, que permite a conexão ao OpenERP em navegadores web padrão. O cliente web agora está embutido no principal OpenERP Server, e não necessita de implantação separada.

21. Iniciando o cliente Web

O servidor web que está sendo inicializado e as configurações salvas, você pode iniciar o Cliente Web OpenERP. Podemos usar um navegador de livre escolha para conectar-se a OpenERP Web. Se o cliente web está instalado no mesmo computador que o servidor, podemos navegar para <http://localhost:8069> para se conectar à versão web OpenERP. Se o servidor estiver instalado em um computador separado, basta sabermos o nome ou endereço IP do servidor na rede e navegar para [http:// <server_address>: 8069](http://<server_address>:8069) para conexão ao OpenERP. Lembrando que 8069 é a porta padrão do OpenERP.

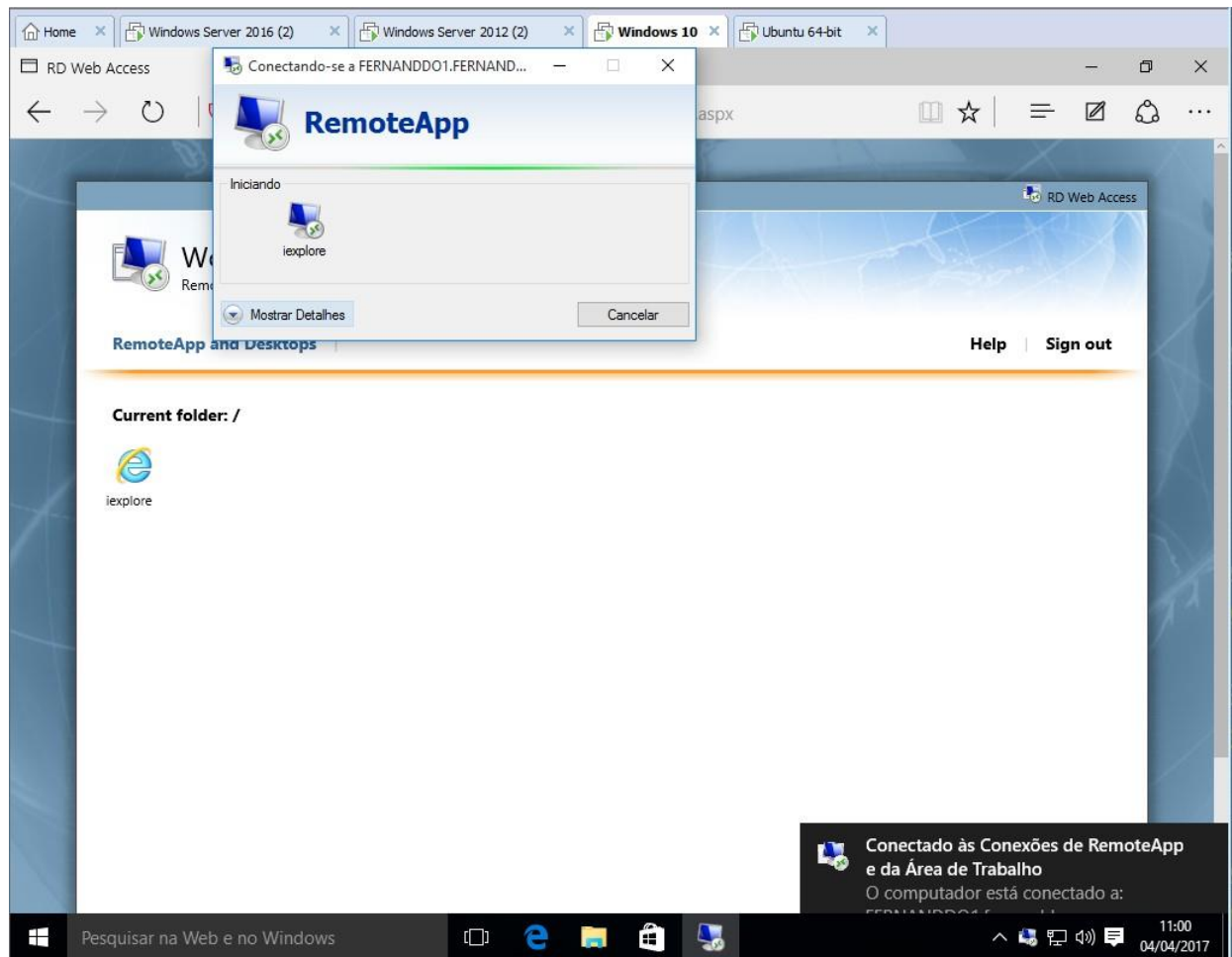


Uma das formas em que nossa aplicação estará disponibilizada em nossa infraestrutura, através do Serviço Remoteapp, aplicação esta citada como sendo o OpenErp, que passou a ser chamada de Odoo. Já foi visto que o Odoo tem seu acesso via Cliente GTK ou Interface Web, mas em nosso caso específico será disponibilizado em nosso domínio fernando.com pelo acesso web e com isso publicaremos o Internet Explorer no Servidor RDS e em seguida será feita a sua configuração para que o mesmo ao ser acessado pelos usuários já seja direcionado para a aplicação.

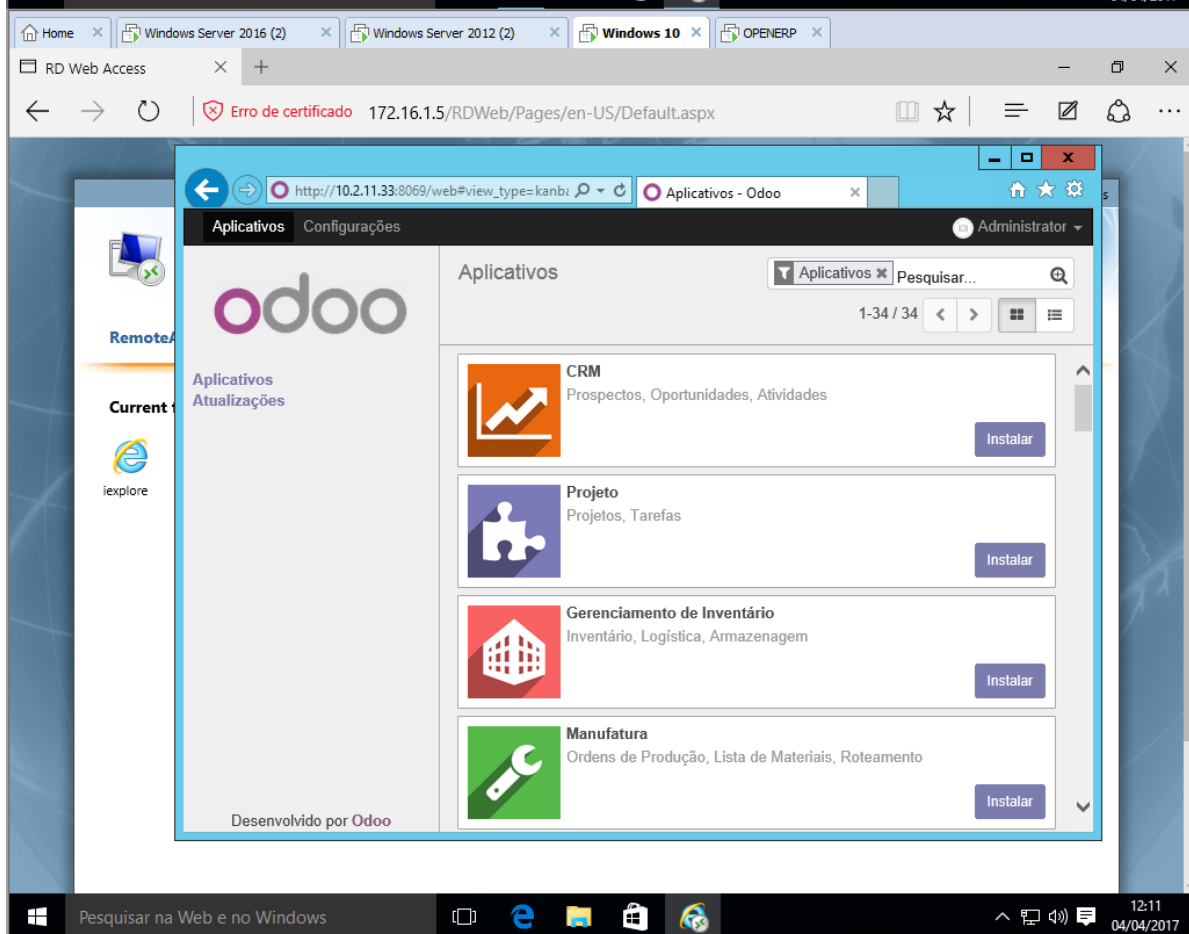
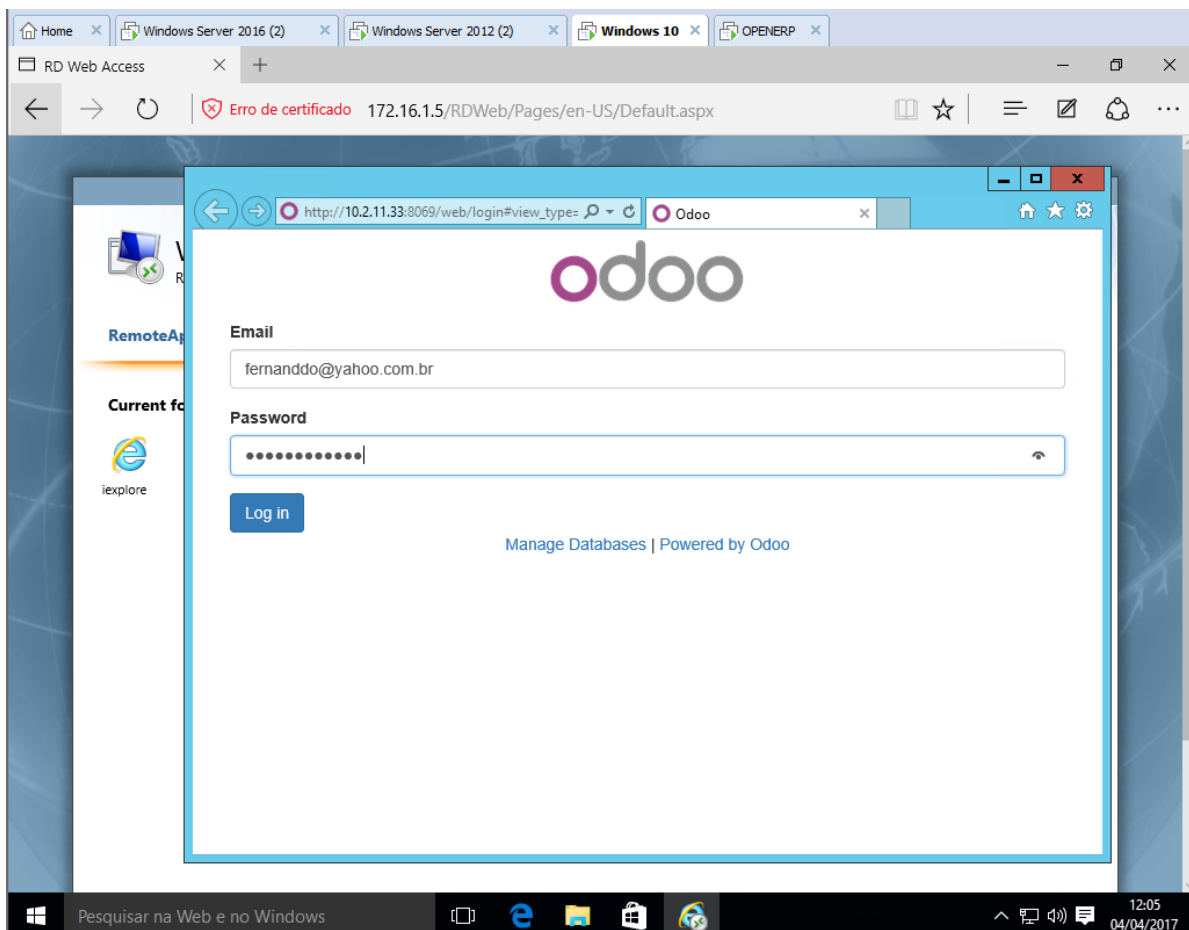


Nas figuras acima, podemos verificar o Internet Explorer sendo publicado no servidor RDS e já sendo disponibilizado em nosso domínio. Depois de publicado, efetuamos a

configuração dos parâmetros necessários para que quando acessado pelo usuário do domínio, este já busque a aplicação e apresente a tela de login para que esta seja devidamente acessada como podemos ver a seguir:



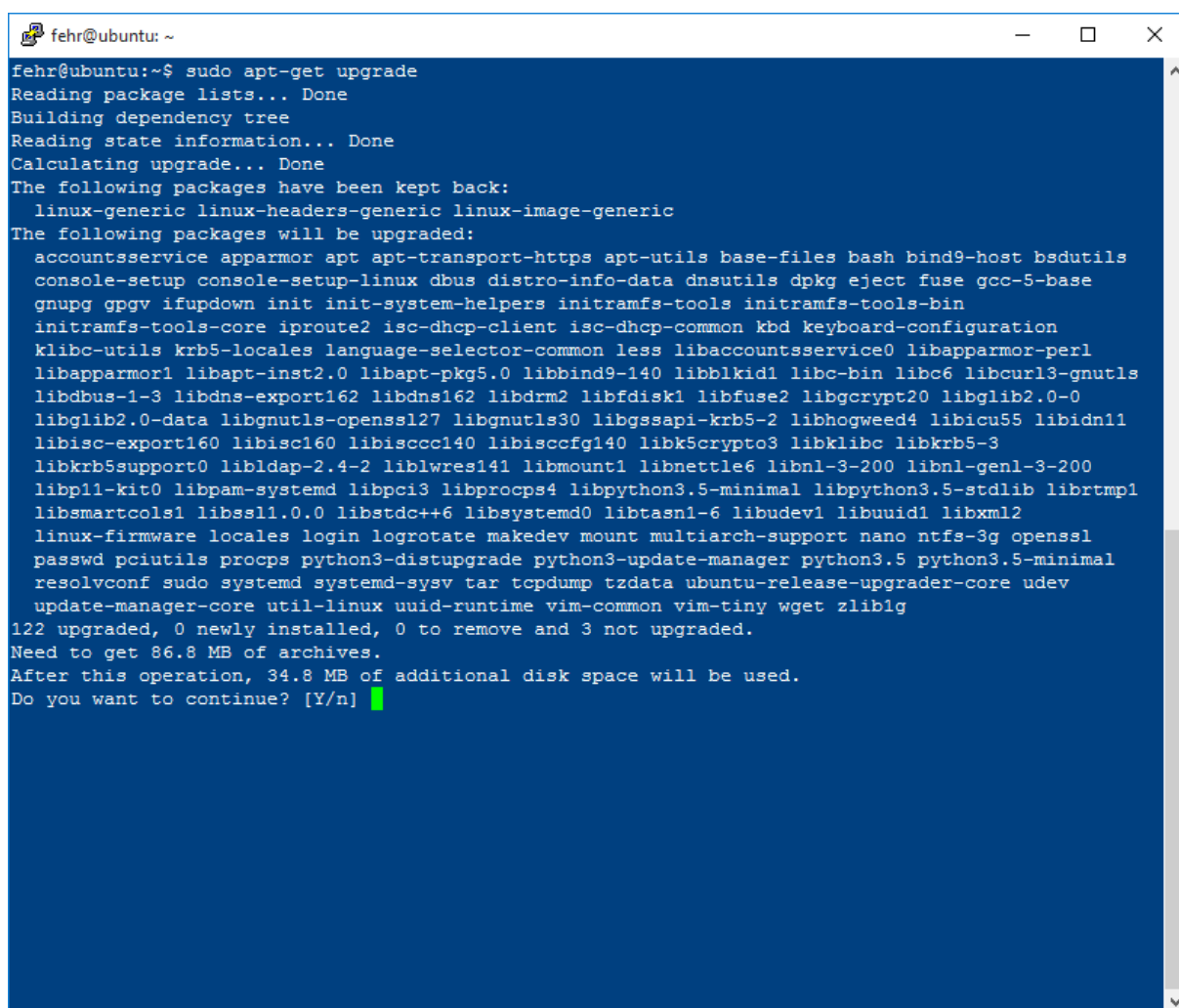
Quando o usuário do domínio faz o acesso ao servidor RDS (172.16.1.5/rdweb), que está em nosso domínio, podemos ver o início ao acesso a nossa aplicação web (Odoo), que logo estará disponibilizada como veremos a seguir:



22. Referências dos downloads e/ou versões de todos os componentes da solução implementada.

Como já apresentado em nosso projeto, estamos em processo de desenvolvimento do OpenERP (Odoo), o qual já vimos as etapas anteriores e agora nesta etapa vamos acompanhar mais um dos seus processos de instalação conforme a sequência nos mostra a seguir:

Como nossa instalação será em um servidor Ubuntu 16.04 vamos sequenciar essa instalação da seguinte forma:

A terminal window titled 'fehr@ubuntu: ~' with standard window controls. It displays the output of the command 'sudo apt-get upgrade'. The output shows that several packages are kept back (linux-generic, linux-headers-generic, linux-image-generic) and a large number of other packages are to be upgraded. It lists 122 packages to be upgraded, 0 newly installed, and 0 to be removed. It also indicates that 86.8 MB of archives are needed and that 34.8 MB of additional disk space will be used. The prompt 'Do you want to continue? [Y/n]' is shown with a green cursor.

```
fehr@ubuntu:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-generic linux-headers-generic linux-image-generic
The following packages will be upgraded:
  accountsservice apparmor apt apt-transport-https apt-utils base-files bash bind9-host bsdtls
  console-setup console-setup-linux dbus distro-info-data dnsutils dpkg eject fuse gcc-5-base
  gnupg gpgv ifupdown init init-system-helpers initramfs-tools initramfs-tools-bin
  initramfs-tools-core iproute2 isc-dhcp-client isc-dhcp-common kbd keyboard-configuration
  klibc-utils krb5-locales language-selector-common less libaccountsservice0 libapparmor-perl
  libapparmor1 libapt-inst2.0 libapt-pkg5.0 libbind9-140 libblkid1 libbc-bin libbc6 libcurl3-gnutls
  libdbus-1-3 libdns-export162 libdns162 libdrm2 libfdisk1 libfuse2 libgcrypt20 libglib2.0-0
  libglib2.0-data libgnutls-openssl27 libgnutls30 libgssapi-krb5-2 libhogweed4 libicu55 libidn11
  libisc-export160 libisc160 libisccc140 libisccfg140 libk5crypto3 libklibc libkrb5-3
  libkrb5support0 libldap-2.4-2 liblwres141 libmount1 libnettle6 libnl-3-200 libnl-genl-3-200
  libp11-kit0 libpam-systemd libpci3 libprocps4 libpython3.5-minimal libpython3.5-stdlib librtmp1
  libsmartcols1 libssl1.0.0 libstdc++6 libsystemd0 libtasn1-6 libudev1 libuuid1 libxml2
  linux-firmware locales login logrotate makedev mount multiarch-support nano ntfs-3g openssl
  passwd pciutils procps python3-distupgrade python3-update-manager python3.5 python3.5-minimal
  resolvconf sudo systemd systemd-sysv tar tcpdump tzdata ubuntu-release-upgrader-core udev
  update-manager-core util-linux uuid-runtime vim-common vim-tiny wget zlib1g
122 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 86.8 MB of archives.
After this operation, 34.8 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

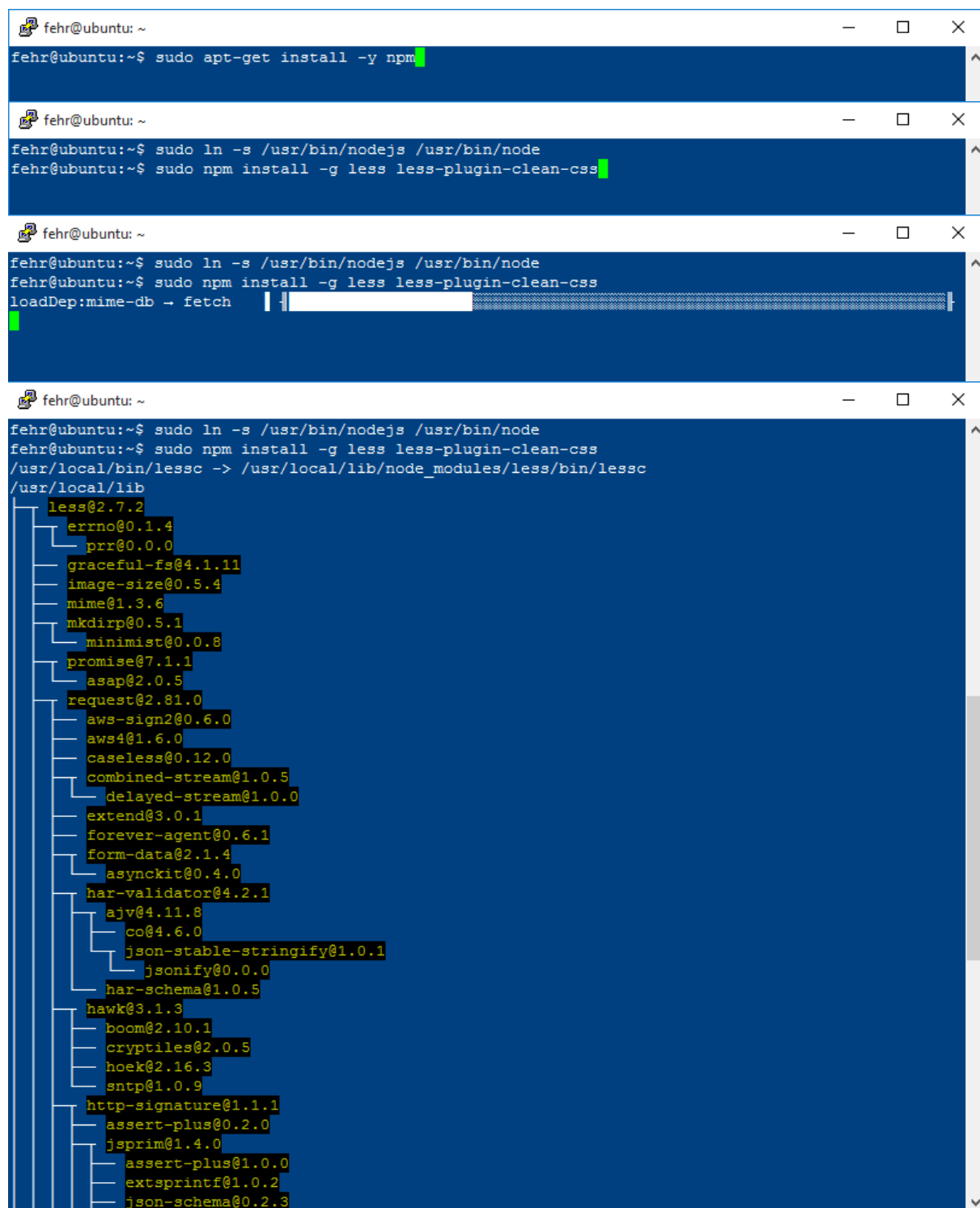
Após efetuarmos o comando `sudo apt-get update`, efetuamos as atualizações necessárias para que todas as fontes do apt estejam ok prosseguimos conforme a figura acima do comando para a atualização do sistema para que todos os repositórios estejam ok. Assim damos prosseguimento as etapas de instalação de nossa aplicação.

22.1. Instalação das dependências do Python para Odoo

```
fehr@ubuntu: ~  
fehr@ubuntu:~$ sudo apt-get install python-dateutil python-docutils python-feedparser python-jinja2  
python-ldap python-libxslt1 python-lxml python-mako python-mock python-openid python-psycopg2 python-  
psutil python-pybabel python-pychart python-pydot python-pyparsing python-reportlab python-simple  
json python-tz python-unittest2 python-vatnumber python-vobject python-webdav python-werkzeug python-  
xlwt python-yaml python-zsi poppler-utils python-pip python-pypdf python-passlib python-decorato  
r gcc python-dev mc bzr python-setuptools python-markupsafe python-reportlab-accel python-zsi pytho  
n-yaml python-argparse python-openssl python-egenix-mxdatetime python-usb python-serial lptools mak  
e python-pydot python-psutil python-paramiko poppler-utils python-pdftools antiword python-requests  
python-xlswriter python-suds python-psycogreen python-ofxparse python-gevent  
  
libdpkg-perl libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libexpat1-dev  
libfakeroot libfile-fcntllock-perl libgcc-5-dev libgd3 libglib2.0-dev libglib2.0-glib libglib2.0-  
libgstreamer1.0-0 libgvc6 libgvcpr2 libijs-0.35 libindicate-gtk3 libindicate5 libisl15 libitml  
libjbig2dec0 libjs-jquery libksba8 liblcms2-2 libllvm3.8 liblsan0 libltdl7 libmng2 libmpc3  
libmpfr4 libmpx0 libmysqlclient20 libnptl0 libogg0 libopus0 liborc-0.4-0 libpaper-utils  
libpaper1 libpathplan4 libpciaccess0 libperl5.22 libpoppler58 libpq5 libpython-all-dev  
libpython-dev libpython-stdlib libpython2.7 libpython2.7-dev libpython2.7-minimal  
libpython2.7-stdlib libqt4-dbus libqt4-declarative libqt4-designer libqt4-help libqt4-network  
libqt4-opengl libqt4-script libqt4-scripttools libqt4-sql libqt4-sql-mysql libqt4-svg  
libqt4-test libqt4-xml libqt4-xmlpatterns libqtassistantclient4 libqtcore4 libqtdbus4 libqtgui4  
libqtwebkit4 libquadmath0 libssh2-1 libstdc++-5-dev libtheora0 libtidy-0.99-0 libtsan0  
libtxc-dxtn-s2tc0 libubsan0 libvisual-0.4-0 libvorbis0a libvorbisenc2 libvpx3 libwebp5  
libwebpmux1 libx11-xcb1 libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0  
libxcb-sync1 libxmu6 libxpm4 libxshmfence1 libxslt1.1 libxt6 libxxf86vm1 libyaml-0-2  
linux-libc-dev lptools make manpages-dev mc mc-data mysql-common patch perl perl-modules-5.22  
pinentry-curses poppler-data poppler-utils python python-all python-all-dev python-antlr  
python-babel python-babel-localedata python-bs4 python-bzrlib python-cairo python-ffi-backend  
python-chameleon python-chardet python-configobj python-crypto python-cryptography  
python-dateutil python-dbus python-decorator python-dev python-docutils python-ecdsa  
python-egenix-mxdatetime python-egenix-mxtools python-enum34 python-feedparser python-funcsigs  
python-gevent python-gi python-gobject python-gobject-2 python-gpgme python-greenlet  
python-gtk2 python-httplib2 python-httpplib2 python-idna python-indicate python-ipaddress  
python-jinja2 python-keyring python-launchpadlib python-lazr.restfulclient python-lazr.uri  
python-ldap python-libxml2 python-libxslt1 python-linecache2 python-lxml python-mako  
python-markupsafe python-minimal python-mock python-ndg-httpsclient python-oauth  
python-ofxparse python-openid python-openssl python-paramiko python-passlib python-pbr  
python-pdftools python-pil python-pip python-pip-whl python-pkg-resources python-psutil  
python-psycogreen python-psycopg2 python-pyasn1 python-pybabel python-pychart python-pydot  
python-pygments python-pyinotify python-pyparsing python-pypdf python-qt4 python-renderpm  
python-reportlab python-reportlab-accel python-requests python-roman python-secretstorage  
python-serial python-setuptools python-simplejson python-sip python-six python-stdnum  
python-suds python-traceback2 python-tz python-unittest2 python-urllib3 python-usb  
python-utidylib python-vatnumber python-vobject python-wadllib python-webdav python-werkzeug  
python-wheel python-xdg python-xlswriter python-xlwt python-yaml python-zope.interface  
python-zsi python2.7 python2.7-dev python2.7-minimal qdbus qt-at-spi qtchooser qtcore4-l10n  
rename unzip xz-utils  
0 upgraded, 275 newly installed, 0 to remove and 3 not upgraded.  
Need to get 158 MB of archives.  
After this operation, 675 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

O odoo funciona basicamente em arquitetura python e na figura acima vemos como acontece essa instalação das dependências e bibliotecas python para que não corramos o risco de falta de componentes e assim o seu funcionamento ficar comprometido e consequentemente causando impacto na produção e nos negócios.

A Seguir efetuaremos as instalações do NPM, que é o gerenciador de pacotes que vem pré-instalado com a plataforma de servidor Node.js.



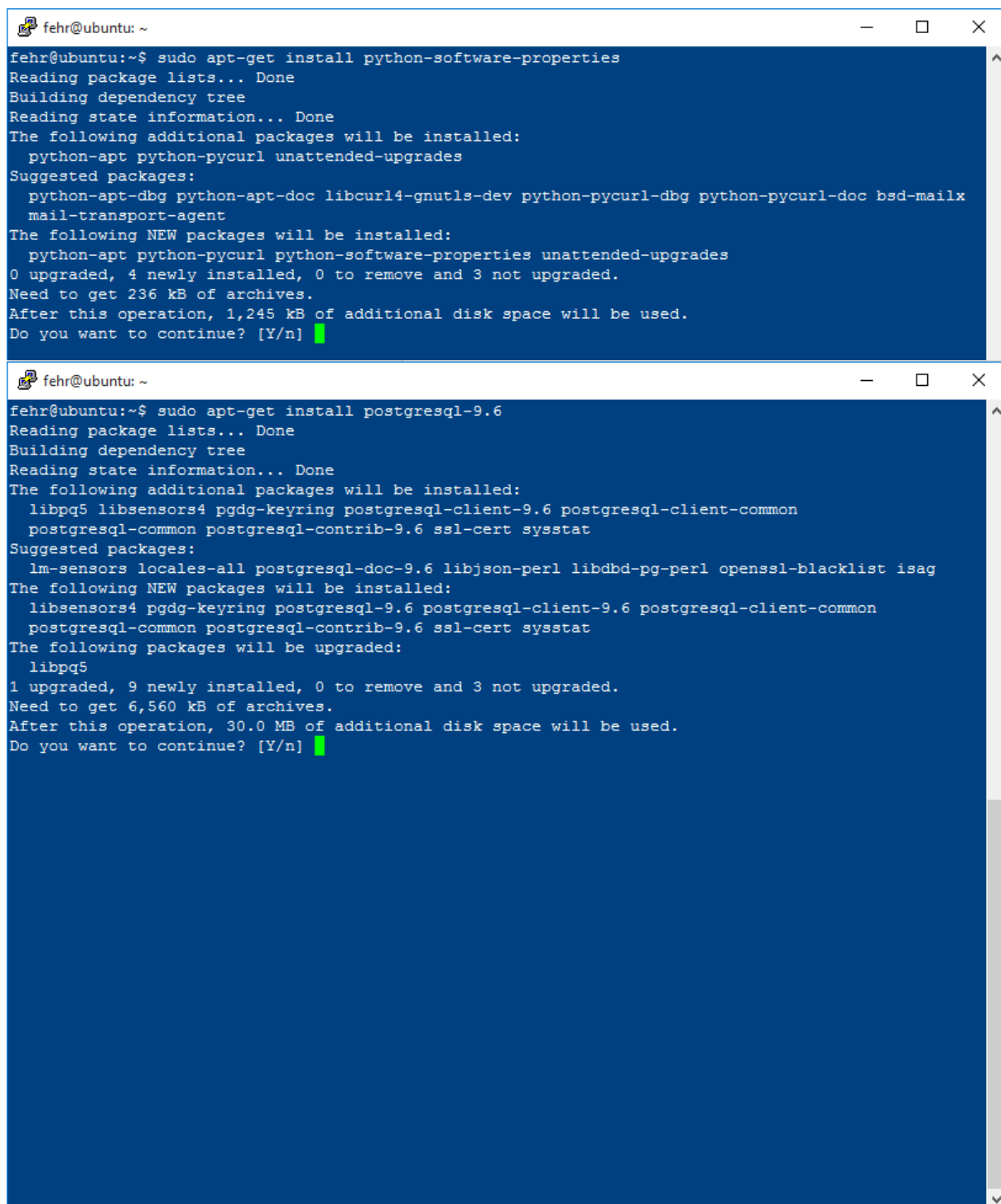
```
fehr@ubuntu: ~  
fehr@ubuntu:~$ sudo apt-get install -y npm  
  
fehr@ubuntu: ~  
fehr@ubuntu:~$ sudo ln -s /usr/bin/nodejs /usr/bin/node  
fehr@ubuntu:~$ sudo npm install -g less less-plugin-clean-css  
  
fehr@ubuntu: ~  
fehr@ubuntu:~$ sudo ln -s /usr/bin/nodejs /usr/bin/node  
fehr@ubuntu:~$ sudo npm install -g less less-plugin-clean-css  
loadDep:mime-db → fetch  
  
fehr@ubuntu: ~  
fehr@ubuntu:~$ sudo ln -s /usr/bin/nodejs /usr/bin/node  
fehr@ubuntu:~$ sudo npm install -g less less-plugin-clean-css  
/usr/local/bin/lessc -> /usr/local/lib/node_modules/less/bin/lessc  
/usr/local/lib  
less@2.7.2  
├── errno@0.1.4  
├── prr@0.0.0  
├── graceful-fs@4.1.11  
├── image-size@0.5.4  
├── mime@1.3.6  
├── mkdirp@0.5.1  
├── minimist@0.0.8  
├── promise@7.1.1  
├── asap@2.0.5  
├── request@2.81.0  
├── aws-sign2@0.6.0  
├── aws4@1.6.0  
├── caseless@0.12.0  
├── combined-stream@1.0.5  
├── delayed-stream@1.0.0  
├── extend@3.0.1  
├── forever-agent@0.6.1  
├── form-data@2.1.4  
├── async@0.4.0  
├── har-validator@4.2.1  
├── ajv@4.11.8  
├── co@4.6.0  
├── json-stable-stringify@1.0.1  
├── jsonify@0.0.0  
├── har-schema@1.0.5  
├── hawk@3.1.3  
├── boom@2.10.1  
├── cryptiles@2.0.5  
├── hoek@2.16.3  
├── sntp@1.0.9  
├── http-signature@1.1.1  
├── assert-plus@0.2.0  
├── jsprim@1.4.0  
├── assert-plus@1.0.0  
├── extsprintf@1.0.2  
└── json-schema@0.2.3
```

O NPM é uma ferramenta que auxilia um desenvolvedor a instalar e gerenciar programas de terceiros. Não confundamos o npm com o commonJS require (). O NPM não é utilizado para carregar o código javascript, ele é usado para instalar o

código e gerenciar as dependências a partir da linha de comando. Os pacotes encontrados no npm podem variar de simples bibliotecas auxiliares como o underscore.js, ao automatizador de tarefas grunt.js.

22.2. Instalação do Postgresql

Como é evidente, o OpenERP necessita de um banco de dados para rodar e o que este utiliza é o Postgresql cuja instalação verificamos da seguinte forma:



```
fehr@ubuntu: ~  
fehr@ubuntu:~$ sudo apt-get install python-software-properties  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  python-apt python-pycurl unattended-upgrades  
Suggested packages:  
  python-apt-dbg python-apt-doc libcurl4-gnutls-dev python-pycurl-dbg python-pycurl-doc bsd-mailx  
  mail-transport-agent  
The following NEW packages will be installed:  
  python-apt python-pycurl python-software-properties unattended-upgrades  
0 upgraded, 4 newly installed, 0 to remove and 3 not upgraded.  
Need to get 236 kB of archives.  
After this operation, 1,245 kB of additional disk space will be used.  
Do you want to continue? [Y/n]   
  
fehr@ubuntu: ~  
fehr@ubuntu:~$ sudo apt-get install postgresql-9.6  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libpq5 libsensors4 pgdg-keyring postgresql-client-9.6 postgresql-client-common  
  postgresql-common postgresql-contrib-9.6 ssl-cert sysstat  
Suggested packages:  
  lm-sensors locales-all postgresql-doc-9.6 libjson-perl libdbd-pg-perl openssl-blacklist isag  
The following NEW packages will be installed:  
  libsensors4 pgdg-keyring postgresql-9.6 postgresql-client-9.6 postgresql-client-common  
  postgresql-common postgresql-contrib-9.6 ssl-cert sysstat  
The following packages will be upgraded:  
  libpq5  
1 upgraded, 9 newly installed, 0 to remove and 3 not upgraded.  
Need to get 6,560 kB of archives.  
After this operation, 30.0 MB of additional disk space will be used.  
Do you want to continue? [Y/n]   

```

Criamos os usuários de banco de dados para OpenERP (Odoo)

```
sudo su postgres
```

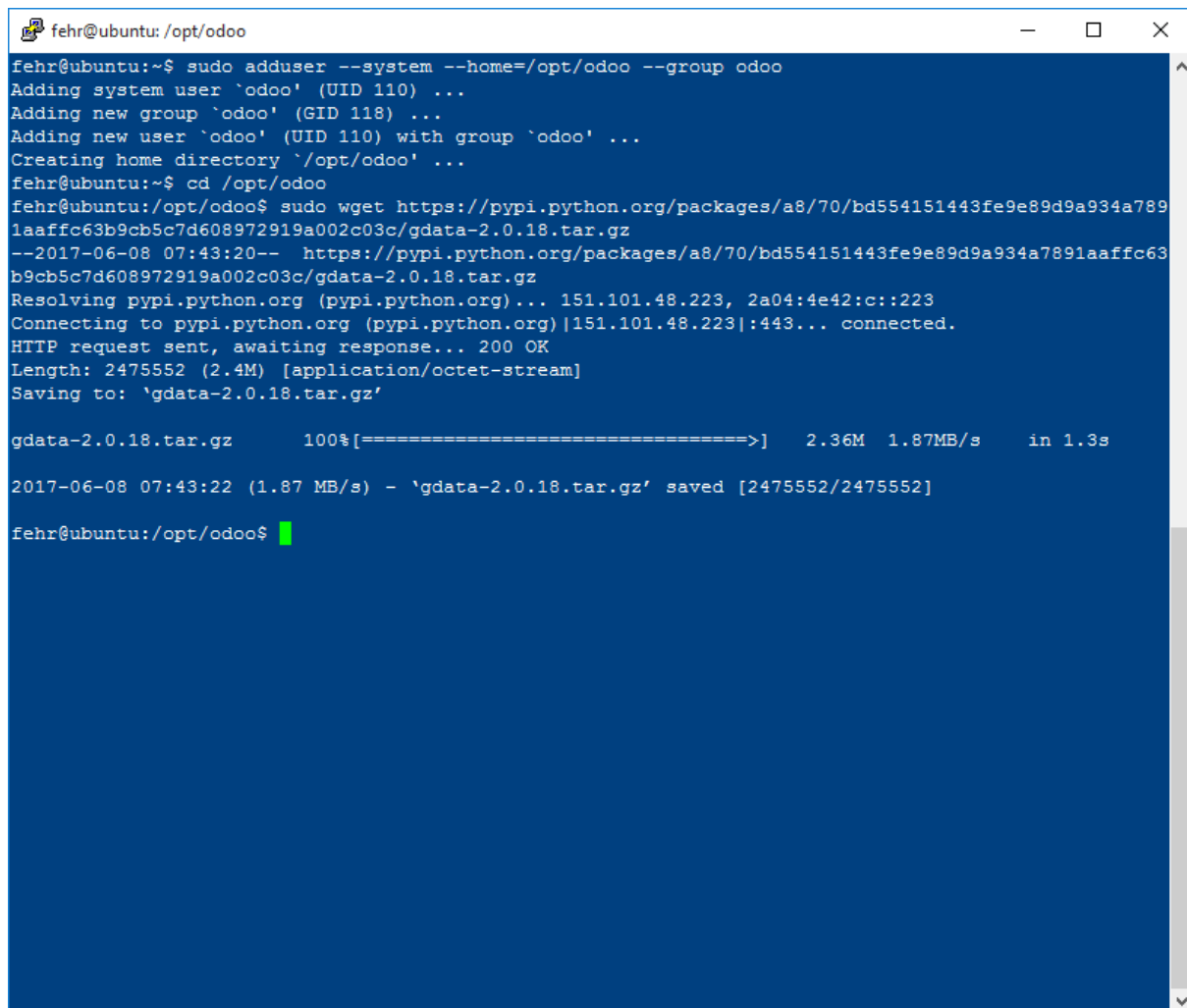
```
cd
```

```
createuser -s odoo
```

```
createuser -s fernando
```

```
exit
```

22.3. Criamos usuário e grupo Odoo e efetuamos a instalação do Gdata



```
fehr@ubuntu: /opt/odoo
fehr@ubuntu:~$ sudo adduser --system --home=/opt/odoo --group odoo
Adding system user `odoo' (UID 110) ...
Adding new group `odoo' (GID 118) ...
Adding new user `odoo' (UID 110) with group `odoo' ...
Creating home directory `/opt/odoo' ...
fehr@ubuntu:~$ cd /opt/odoo
fehr@ubuntu:/opt/odoo$ sudo wget https://pypi.python.org/packages/a8/70/bd554151443fe9e89d9a934a7891aaffc63b9cb5c7d608972919a002c03c/gdata-2.0.18.tar.gz
--2017-06-08 07:43:20-- https://pypi.python.org/packages/a8/70/bd554151443fe9e89d9a934a7891aaffc63b9cb5c7d608972919a002c03c/gdata-2.0.18.tar.gz
Resolving pypi.python.org (pypi.python.org)... 151.101.48.223, 2a04:4e42:c::223
Connecting to pypi.python.org (pypi.python.org)|151.101.48.223|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2475552 (2.4M) [application/octet-stream]
Saving to: `gdata-2.0.18.tar.gz'

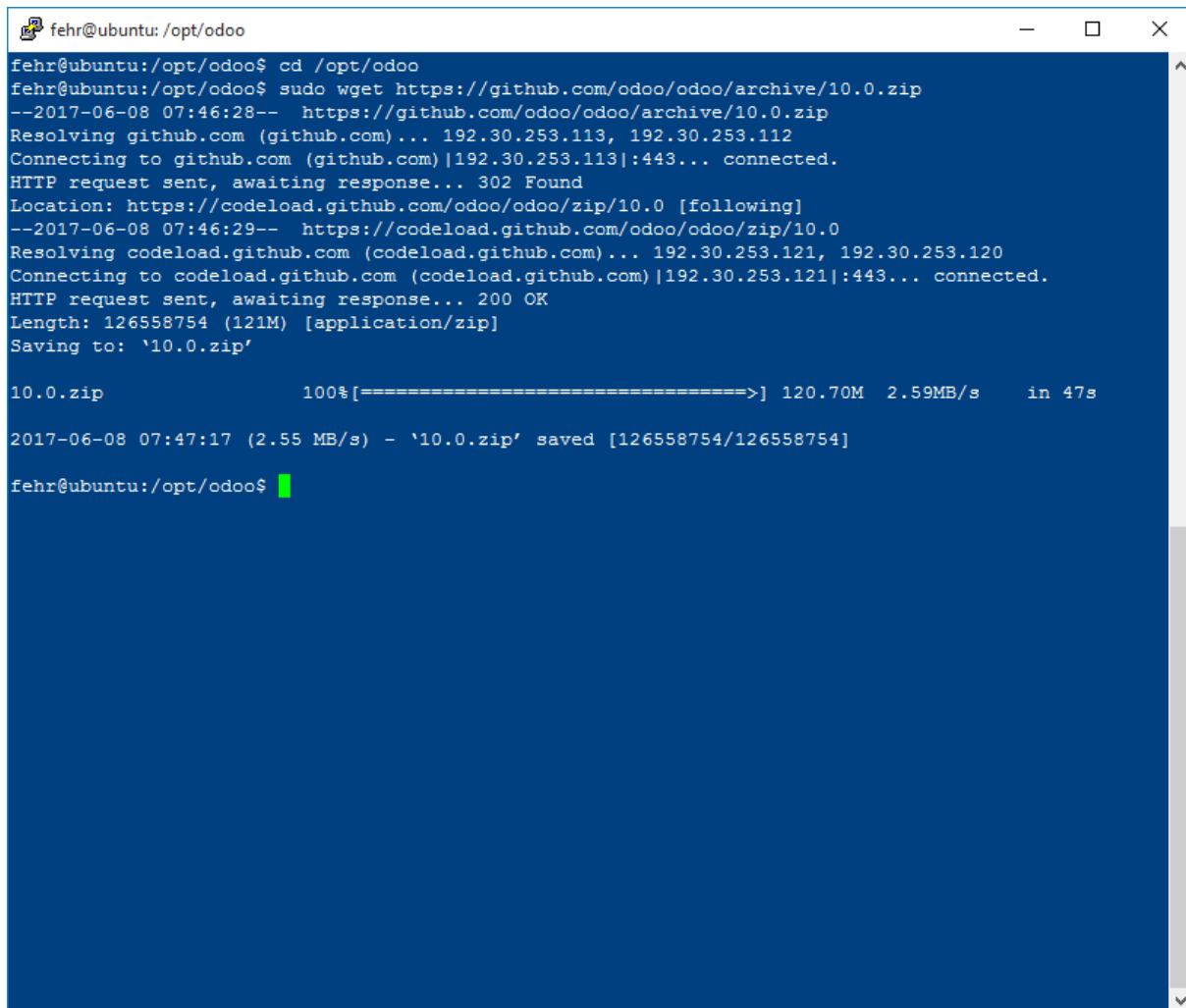
gdata-2.0.18.tar.gz      100%[=====>] 2.36M  1.87MB/s  in 1.3s

2017-06-08 07:43:22 (1.87 MB/s) - `gdata-2.0.18.tar.gz' saved [2475552/2475552]

fehr@ubuntu:/opt/odoo$
```

Após criarmos o usuário odoo e o grupo odoo, efetuamos a instalação do Gdata e após isso, vamos para o Download da última versão do OpenERP.

Efetuamos a busca da última versão do Odoo 10 do depósito github. Baixamos o arquivo Zip do URL: "https://github.com/odoo/odoo/tree/10.0". Transferimos o mesmo arquivo para o diretório / opt / odoo no servidor através do ftp. Caso contrário, procedemos da seguinte forma:

A terminal window titled 'fehr@ubuntu: /opt/odoo' with standard window controls. The terminal output shows the user navigating to /opt/odoo and using 'sudo wget' to download 'https://github.com/odoo/odoo/archive/10.0.zip'. It shows the connection to github.com, the 302 Found response, and the redirection to 'https://codeload.github.com/odoo/odoo/zip/10.0'. The download then proceeds from codeload.github.com with a 200 OK response. A progress bar indicates the file is 120.70M and downloading at 2.59MB/s. The final status shows the file '10.0.zip' is saved with a size of 126558754 bytes.

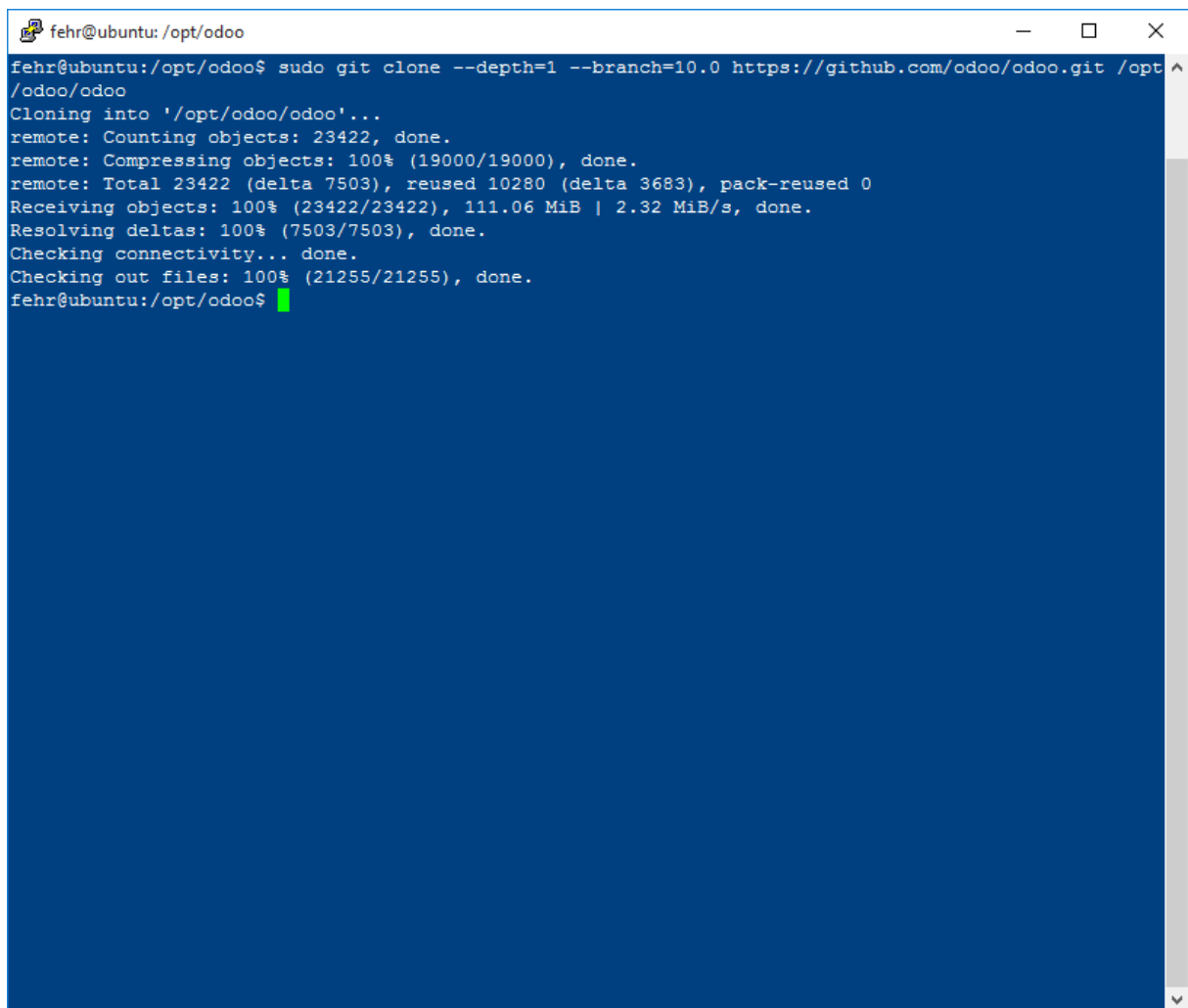
```
fehr@ubuntu: /opt/odoo$ cd /opt/odoo
fehr@ubuntu: /opt/odoo$ sudo wget https://github.com/odoo/odoo/archive/10.0.zip
--2017-06-08 07:46:28-- https://github.com/odoo/odoo/archive/10.0.zip
Resolving github.com (github.com)... 192.30.253.113, 192.30.253.112
Connecting to github.com (github.com)|192.30.253.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/odoo/odoo/zip/10.0 [following]
--2017-06-08 07:46:29-- https://codeload.github.com/odoo/odoo/zip/10.0
Resolving codeload.github.com (codeload.github.com)... 192.30.253.121, 192.30.253.120
Connecting to codeload.github.com (codeload.github.com)|192.30.253.121|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 126558754 (121M) [application/zip]
Saving to: '10.0.zip'

10.0.zip                100%[=====>] 120.70M  2.59MB/s   in 47s

2017-06-08 07:47:17 (2.55 MB/s) - '10.0.zip' saved [126558754/126558754]

fehr@ubuntu: /opt/odoo$
```

Após concluirmos o procedimento, prosseguimos com nossa instalação, para que consigamos chegar a conclusão da mesma.

A terminal window with a dark blue background and white text. The window title bar shows 'fehr@ubuntu: /opt/odoo' and standard window controls. The terminal output shows the command 'sudo git clone --depth=1 --branch=10.0 https://github.com/odoo/odoo.git /opt/odoo' being executed. The output includes progress bars for counting, compressing, and receiving objects, as well as resolving deltas and checking out files. The prompt 'fehr@ubuntu:/opt/odoo\$' is visible at the bottom.

```
fehr@ubuntu:/opt/odoo$ sudo git clone --depth=1 --branch=10.0 https://github.com/odoo/odoo.git /opt/odoo
Cloning into '/opt/odoo/odoo'...
remote: Counting objects: 23422, done.
remote: Compressing objects: 100% (19000/19000), done.
remote: Total 23422 (delta 7503), reused 10280 (delta 3683), pack-reused 0
Receiving objects: 100% (23422/23422), 111.06 MiB | 2.32 MiB/s, done.
Resolving deltas: 100% (7503/7503), done.
Checking connectivity... done.
Checking out files: 100% (21255/21255), done.
fehr@ubuntu:/opt/odoo$
```

Como vemos na figura acima, fazemos a clonagem do repositório do GitHub. Direcionando para o nosso repositório local criado para receber a aplicação e assim já entrando na fase final de instalação do OpenERP (Odoo).

Isso é importante pois nos possibilita efetuar o download da aplicação já com as suas atualizações e correções para que não tenhamos problemas futuros com relação à instalação da mesma em nosso ambiente.

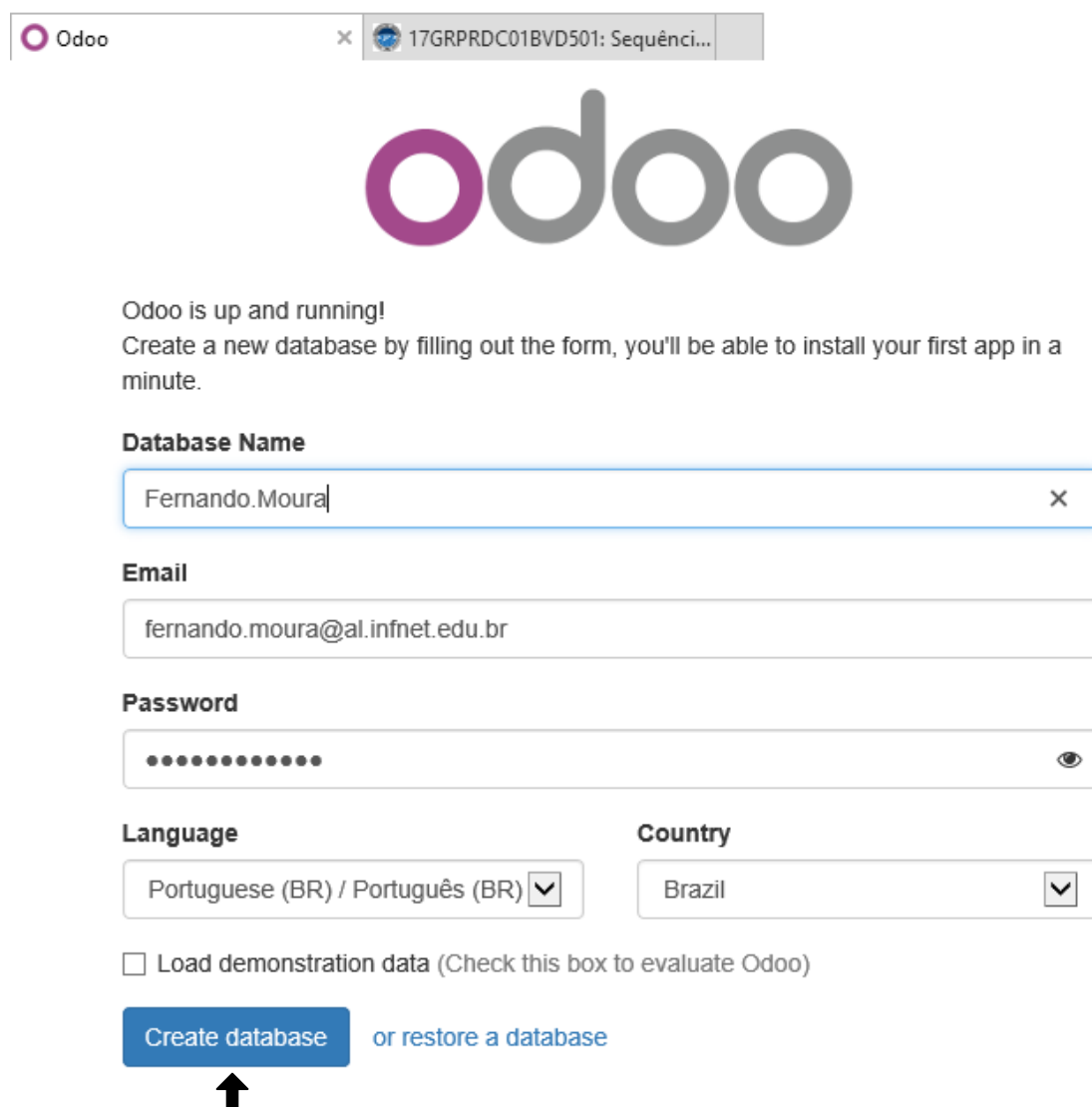
22.4. Instalação do WKHTMLTOPDF para o Odoo (Dependência necessária)

```
fehr@ubuntu: ~  
fehr@ubuntu:/opt/odoo$ cd  
fehr@ubuntu:~$ wget https://downloads.wkhtmltopdf.org/0.12/0.12.4/wkhtmltox-0.12.4_linux-generic-amd64.tar.xz  
--2017-06-08 07:59:55-- https://downloads.wkhtmltopdf.org/0.12/0.12.4/wkhtmltox-0.12.4_linux-generic-amd64.tar.xz  
Resolving downloads.wkhtmltopdf.org (downloads.wkhtmltopdf.org)... 137.74.141.30  
Connecting to downloads.wkhtmltopdf.org (downloads.wkhtmltopdf.org)|137.74.141.30|:443... connected  
HTTP request sent, awaiting response... 200 OK  
Length: 14541904 (14M) [application/octet-stream]  
Saving to: 'wkhtmltox-0.12.4_linux-generic-amd64.tar.xz'  
  
wkhtmltox-0.12.4_linux-g 100%[=====] 13.87M 1.13MB/s in 16s  
  
2017-06-08 08:00:12 (891 KB/s) - 'wkhtmltox-0.12.4_linux-generic-amd64.tar.xz' saved [14541904/14541904]  
  
fehr@ubuntu:~$
```

```
fehr@ubuntu: ~/wkhtmltox  
fehr@ubuntu:~$ cd wkhtmltox/  
fehr@ubuntu:~/wkhtmltox$ sudo cp -Rav * /usr/local/  
'bin/wkhtmltoimage' -> '/usr/local/bin/wkhtmltoimage'  
'bin/wkhtmltopdf' -> '/usr/local/bin/wkhtmltopdf'  
'include/wkhtmltox' -> '/usr/local/include/wkhtmltox'  
'include/wkhtmltox/image.h' -> '/usr/local/include/wkhtmltox/image.h'  
'include/wkhtmltox/dllbegin.inc' -> '/usr/local/include/wkhtmltox/dllbegin.inc'  
'include/wkhtmltox/pdf.h' -> '/usr/local/include/wkhtmltox/pdf.h'  
'include/wkhtmltox/dllend.inc' -> '/usr/local/include/wkhtmltox/dllend.inc'  
'lib/libwkhtmltox.so.0.12.4' -> '/usr/local/lib/libwkhtmltox.so.0.12.4'  
'lib/libwkhtmltox.so.0' -> '/usr/local/lib/libwkhtmltox.so.0'  
'lib/libwkhtmltox.so.0.12' -> '/usr/local/lib/libwkhtmltox.so.0.12'  
'lib/libwkhtmltox.so' -> '/usr/local/lib/libwkhtmltox.so'  
'share/man/man1' -> '/usr/local/share/man/man1'  
'share/man/man1/wkhtmltopdf.1.gz' -> '/usr/local/share/man/man1/wkhtmltopdf.1.gz'  
'share/man/man1/wkhtmltoimage.1.gz' -> '/usr/local/share/man/man1/wkhtmltoimage.1.gz'  
fehr@ubuntu:~/wkhtmltox$
```

23. Funcionamento da Aplicação

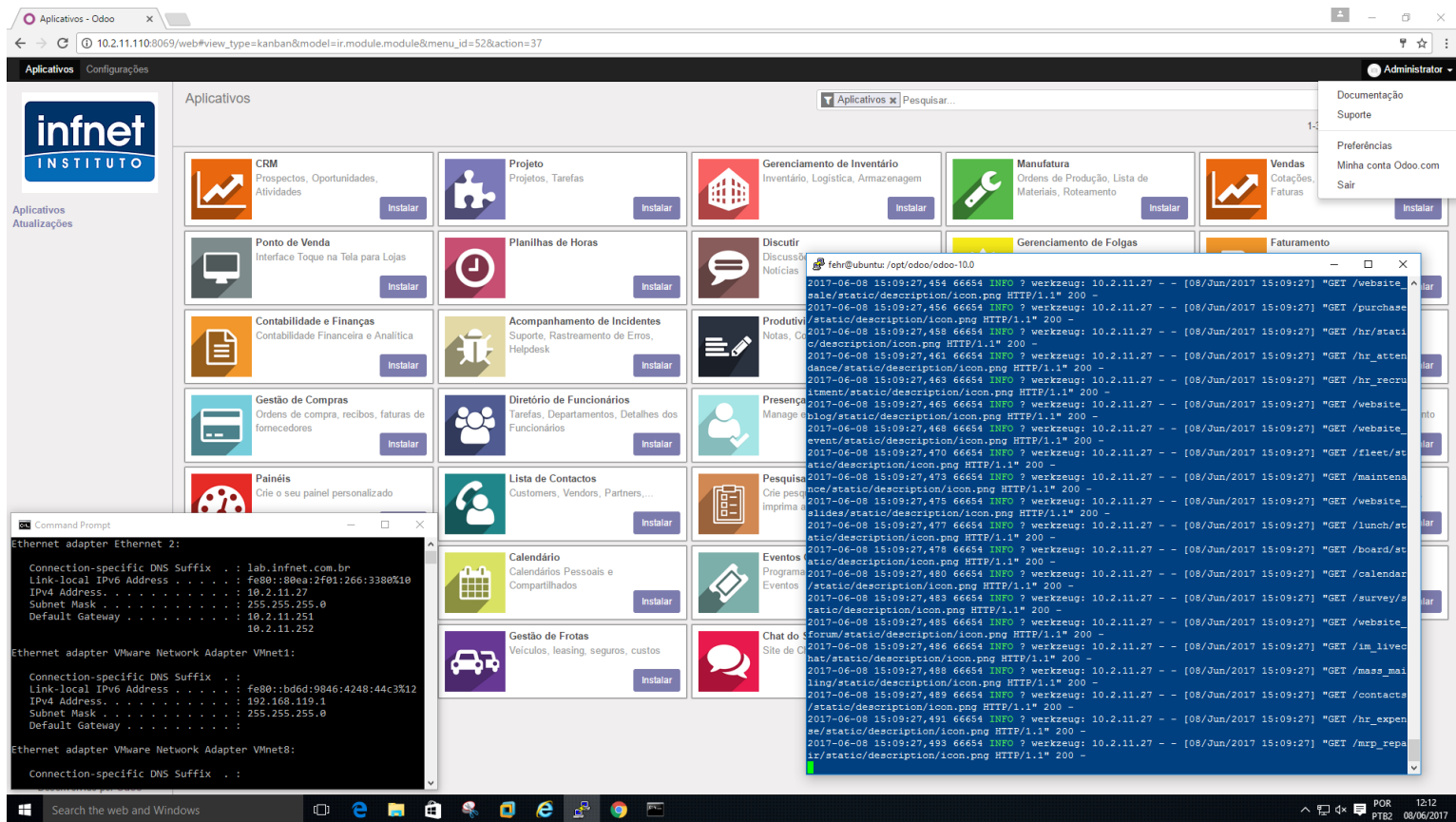
Após seguirmos passo a passo os procedimentos da instalação de nossa aplicação, chegamos a etapa tão esperada. Podemos verificar na figura abaixo a sua tela inicial, onde podemos já observar os campos de criação dos usuários e seus respectivos logins e perfis.



The screenshot shows the Odoo installation interface. At the top, there's a browser window title bar with 'Odoo' and a tab titled '17GRPRDC01BVD501: Sequênci...'. Below the title bar is the large 'odoo' logo. The text 'Odoo is up and running!' is followed by instructions: 'Create a new database by filling out the form, you'll be able to install your first app in a minute.' The form contains several fields: 'Database Name' with the value 'Fernando.Moura', 'Email' with 'fernando.moura@al.infnet.edu.br', 'Password' (masked with dots), 'Language' set to 'Portuguese (BR) / Português (BR)', and 'Country' set to 'Brazil'. There is a checkbox for 'Load demonstration data (Check this box to evaluate Odoo)'. At the bottom, there is a blue button labeled 'Create database' and a link 'or restore a database'. A black arrow points upwards to the 'Create database' button.

Após preenchermos a tela com os dados referentes aos usuários que farão uso de nossa aplicação distribuída, veremos a interface de acesso do OpenERP, onde para cada setor ou departamento, se farão as instalações de cada módulo que seja necessário para o ambiente de produção de cada um.

Estando pronta a criação do usuário, podemos ver a interface da nossa aplicação já apresentada após o primeiro acesso.



24. Se o prazo estabelecido no início do projeto foi adequado para a execução (não há problema se levou mais tempo que o planejado, mas se levou, explique as razões).

Certamente, o nosso prazo foi cumprido dentro das previsões, tudo correu dentro do previsto, o nosso projeto OpenERP apesar de ter sofrido um pequeno impacto por conta de alguns imprevistos, não nos trouxe problema em relação ao prazo estabelecido, tendo seus requisitos atendidos e já se encontrando plenamente funcional em nosso ambiente.

25. Se os recursos planejados (quantidade de memória, disco, etc.) foram suficientes para colocar a solução em funcionamento. Se não foram, explique as razões e sugira uma nova configuração.

Outro fator que merece um destaque foi que todos os recursos inicialmente planejados, atenderam plenamente as necessidades funcionais de nossa aplicação. Claro evidente que de acordo com nosso planejamento futuro, havendo uma necessidade de aumento de demanda, já temos um plano de upgrade de nossos recursos, pois por se tratar de uma aplicação que atende plenamente vários setores da empresa, podemos afirmar que isso se trata apenas de uma questão de tempo.

26. Se as funcionalidades previstas na solução original funcionaram a contento. Se não foram, explique as razões.

Como vimos em figuras pontuais anteriormente, nossa aplicação não somente teve seu funcionamento perfeitamente pleno, como teve todos os seus módulos apresentando perfeito funcionamento em todas elas, conseguindo atender a todas as áreas da empresa para as quais foi desenvolvida, com pleno atendimento das expectativas e atingindo os objetivos de cumprir todos os requisitos exigidos pelas mesmas com máximo nível de satisfação e eficiência.

27. Quais seriam as melhorias futuras que poderiam ser feitas no projeto executado, adotando soluções adicionais, aperfeiçoando instalações ou tornando o cenário mais complexo.

Certamente, por estar ainda em fase inicial e de implantação, nosso projeto certamente precisará sofrer alterações e aperfeiçoamento, pois contando que a demanda crescerá, novas solicitações ocorrerão, já temos em planejamento a implantação de um Load Balance, como também uma forma de eliminar o possível failover que pode ocorrer, através de uma infraestrutura de redundância, como também o VPN que será necessário para os funcionários que estiverem trabalhando na forma homeoffice ou para algum fornecedor que precisar dispor de utilização de nosso ambiente, como também a implantação de um servidor de backup para o banco de dados

28. Conclusão

A virtualização seja ela de serviços, aplicativos ou de servidores hoje já deixou de ser uma tendência para ser uma realidade em muitos setores e áreas, seja dentro ou fora da área da tecnologia. Podemos definir virtualização como o uma forma de se executar vários serviços, programas, ou até mesmo sistemas operacionais em um único equipamento físico. A virtualização possibilita inclusive simular hardwares diferentes em um único equipamento, como roteadores, switches, servidores, celulares, etc. Hoje praticamente é impossível fazer qualquer serviço na internet sem que se tenha passado ou utilizado mesmo sem o seu conhecimento por um ou mais servidores ou serviços virtualizados. Um grande exemplo são os serviços de cloud computing. Onde servidores em sua maioria virtualizados formam uma grande rede fisicamente distribuídas, onde temos servidores dedicados para cada serviço específico que se precisa realizar.

29. Referências Bibliográficas

<https://aws.amazon.com/pt/>

<https://www.vmware.com>

<https://www.openstack.org>

<https://github.com>

<https://www.odoo.com>

<https://www.ansible.com>

<https://www.python.org>

<https://www.postgresql.org>

30. Link do Projeto

<https://github.com/Fernanddo/ProjetoFernanddo>