

INSTITUTO INFNET
ESCOLA SUPERIOR DE TECNOLOGIA DA
INFORMAÇÃO
GRADUAÇÃO EM GESTÃO



PROJETO DE BLOCO
ARQUITETURA DE INFRAESTRUTURA DE
APLICAÇÕES

ALUNO: Alfredo Vieira Negraes Simões

E-MAIL: alfredo.simões@al.infnet.edu.br

TURMA: GGTI - NOITE



Instituto Infnet

CURSO DE GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

Alfredo Vieira Negraes Simões

Projeto de Bloco

ARQUITETURA DE INFRAESTRUTURA DE APLICAÇÕES

Rio de Janeiro

2018

Alfredo Vieira Negraes Simões

Projeto de Bloco

ARQUITETURA DE INFRAESTRUTURA DE APLICAÇÕES

Construção do projeto referente as disciplinas Virtualização de Desktop e Aplicações e Arquitetura de Infraestrutura de Aplicações da graduação em Gestão de Tecnologia da Informação apresentado ao Instituto INFNET como requisito parcial para a obtenção de grau na disciplina.

Projeto de Bloco

Arquitetura de Infraestrutura de Aplicações

Orientador: Carlos Nilton Araújo Correa

Rio de Janeiro

2018

SUMARIO

I. INTRODUÇÃO	6
II. OBJETIVO	6
III. JUSTIFICATIVA	6
CONTEXTO	7
1. ENTERPRISE RESOURCE PLANNING: ODOO.....	7
1.2. CARACTERÍSTICAS DE DESENVOLVIMENTO	8
ARQUITETURA RECOMENDADA.....	11
2. INFRAESTRUTURA DOCKER	11
2.1. REQUISITOS DE SISTEMA.....	12
2.2. CONFIGURAÇÃO.....	13
ARQUITETURA ALTERNATIVA.....	14
3. INFRAESTRUTURA VMWARE.....	14
3.1. REQUISITOS DE SISTEMA.....	15
3.2. CONFIGURAÇÃO.....	18
PLANEJAMENTO.....	19
4. CRONOGRAMA	19
4.1. GRAFICO DE GRANTT	19
DESENVOLVIMENTO & EXECUÇÃO	20
5. INSTALAÇÃO: ODOO	20
5.1. ETAPAS DO DESENVOLVIMENTO	20
5.2. EXECUÇÃO ANSIBLE.....	23
VERIFICAÇÃO	25
6. CONECTIVDADE	25
7. DOCKER.....	25
7.1 POSTGRESQL	26

7.2	ODOO	27
DOCUMENTAÇÃO.....		28
8.	GITHUB: REPOSITÓRIO DE CÓDIGO FONTE.....	28
8.2.	GITHUB: COMMIT & PUSH	30
8.3.	VERSIONAMENTO	31
8.4.	REPOSITÓRIO POPULADO	32
CONCLUSÃO.....		35
BIBLIOGRAFIA		36

I. INTRODUÇÃO

Construção de um Estudo de Caso sobre automatização e gestão através do uso da ferramenta ANSIBLE para instalação e configuração de um Sistema Distribuído.

II. OBJETIVO

Demonstrar conhecimento sobre:

- Propor soluções de arquitetura de infraestrutura para aplicações de acordo com o cenário.
- Utilizar ferramentas de automação de instalação de aplicações
- Levantar requisitos para um projeto de infraestrutura para aplicações

III. JUSTIFICATIVA

Permitir vivência simulada em uma situação problema e condução do processo de mudança para desenvolvimento de solução.

CONTEXTO

O presente projeto foi elaborado para o ambiente de uma pequena empresa que encontra dificuldades organizacionais e de gestão pela adoção de uma série de sistemas dispersos (com requisitos variados, pouca integração, etc) deseja mudar sua abordagem para centralizar e integrar sua gestão em sistema ERP coeso, porém on-premises, isto é, de servidor e recursos de TI hospedados e de gestão direta na própria empresa, devido a sua cultura e alguns requisitos legais, como são uma pequena empresa desejam fazê-lo pelo menor custo e/ou consumo de recursos possível à tal adoção.

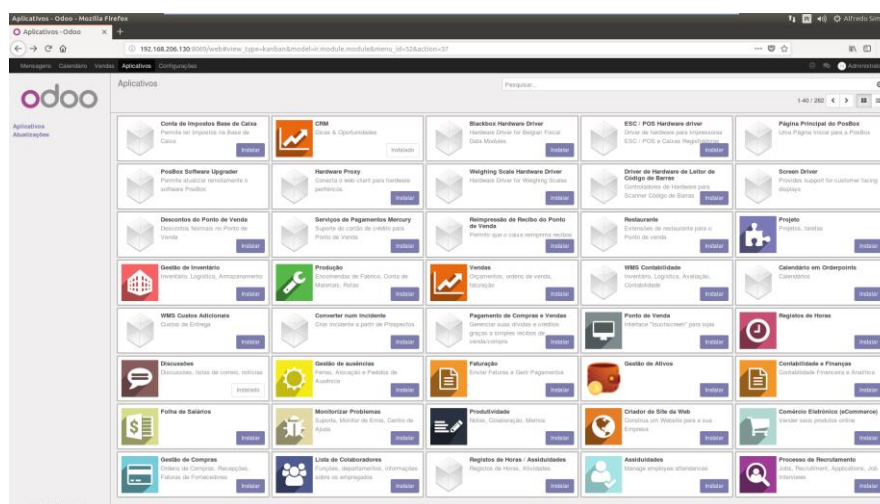
1. ENTERPRISE RESOURCE PLANNING: ODOO

O Odoo é uma ferramenta Open Source (disponível sob a GNU General Public License – Affero) de gestão empresarial ERP bem abrangente, dentre suas áreas funcionais podemos destacar contabilidade gerencial, contabilidade, gerenciamento de materiais, gestão de vendas e compras, tarefas automatizadas, gestão de recursos humanos, marketing, help desk, e ponto de venda, sua estrutura é modular, contando com 350 módulos disponíveis, totalmente baseado em arquitetura MVC (Model-View-Control) para a relação cliente-servidor.

1.2. Características de Desenvolvimento



O Odoo é altamente personalizável, sendo estas feitas a partir a seleção dos módulos que estão sendo carregados.



Em linhas gerais cada módulo Odoo pode conter elementos tais como:

- Objetos de negócios - Declarados como classes de Python, esses recursos são automaticamente persistidos por Odoo com base em sua configuração;
- Arquivos de dados - Arquivos XML ou CSV que declaram metadados (visualizações ou relatórios), dados de configuração (parametrização de módulos), dados de demonstração e etc;
- Controladores da Web - Manusear pedidos de navegadores da Web;
- Dados da web estática - Imagens, arquivos CSS ou javascript usados pela interface web ou site.

Existem variadas formas de se implementar o Odoo, podendo ser instalado ou não, variando conforme cada caso de uso pretendido, dentre as opções temos:

Online: consiste na maneira mais fácil de se utilizar, quer para teste ou execução real, totalmente gerenciado pela Odoo SA, trata-se da plataforma o SaaS da Odoo.

Pacotes de instalação: Odoo fornece instaladores para diferentes sistemas operacionais, tanto para as versões da Comunidade (open) quanto da Enterprise (comercial),

Instalação de fonte: permite maior flexibilidade em relação a instalações via pacotes, possibilitando substituir as configurações, usando parâmetros de linha de comando sem precisar editar um arquivo de configuração, bem como permitindo a execução de várias versões Odoo em um mesmo sistema, voltado especialmente para o desenvolvimento e personalização

Instalação via Docker: utilização do Odoo (e do PostgreSQL) dentro da estrutura de containers, utilizando isoladamente esses recursos com base em um kernel em comum (entre host e container), não sendo necessário a utilização de ambiente um de virtualização tradicional, com um S.O. no qual se instala a aplicação.

Todas elas segundo o seguinte padrão arquitetural:

Odoo Server – Camada de Aplicação

Pode-se dizer que esta camada é o núcleo do sistema, nela estão os elementos lógicos pertinentes ao negócio, suas funcionalidades existem para suportar e apreender as regras de negócio, nela ficam as instruções, funções, métodos, códigos e etc. para todo o sistema, seu foco não está na interface, ou nos dados, mas na lógica.

Odoo Database (PostgreSQL) – Camada de Banco de Dados

Esta camada recebe o papel computacional de repositório estruturado dos dados pertinentes ao sistema, implementação pelo PostgreSQL, um sistema gerenciador de banco de dados objeto (relacional), sendo este responsável por responder a demanda de instruções tais como:

- DML (Data Manipulation Language), isto é, utilizadas para o gerenciamento de dados dentro de objetos do banco, por exemplo, SELECT, INSERT, UPDATE, DELETE e etc;
- DDL (Data Definition Language) isto é, utilizadas para definir a estrutura de banco de dados ou esquema, por exemplo, CREATE, ALTER, TRUNCATE e etc.

- DCL (Data Control Language), isto é, utilizadas para definir privilégios de acesso declarações por exemplo, GRANT, REVOKE e etc.
- TCL (Transaction Control Language), isto é, utilizadas para gerenciar as mudanças (feitas por instruções DML), por exemplo, COMMIT, ROLLBACK e etc.

Desta forma alterações e manipulações dos dados são praticadas sem que seja necessário alterar todo o restante da arquitetura para tal.

Odoo Interface Web – Camada de Apresentação

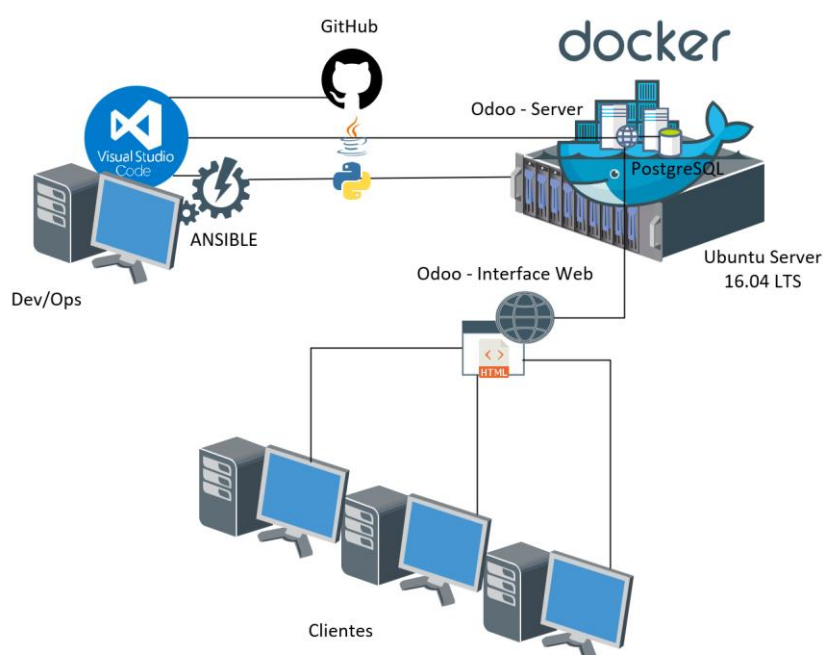
Esta camada é a interface direta entre o usuário e o sistema, por meio dela que ocorre a interação dos indivíduos com suas demandas e necessidades com as funcionalidades e soluções que o sistema oferece.

ARQUITETURA RECOMENDADA

Visto o modelo on-premises modelo tende a exigir investimento inicial mais alto na compra de hardware e software, bem como em alguns casos existe a necessidade de contratar profissionais capacitados para gerenciar o ambiente interno, optou-se por tecnologias em sua maioria Open Source, como: Sistema Operacional (Ubuntu Server), Implementação de Contêineres (Docker), ERP (Odoo) e Orquestração (Ansible), com um pequeno investimento no Gerenciamento de Código-fonte (GitHub), sendo em sua maior o investimento na aquisição de novo Hardware (servidor) em substituição ao legado, que já estava apresentando problemas para atender a demanda existente, bem como reduzir a carga sobre as estações de trabalho que também executavam muitas das aplicações; quanto a possibilidade de contratação profissionais se optará pelo ‘Outsourcing’ para as atividades pontuais.

2. INFRAESTRUTURA DOCKER

O diagrama a baixo apresenta visão simplificada e condensada das infraestruturas de desenvolvimento e operações com base na adoção de uma perspectiva DevOps da implantação no ambiente empresarial.



2.1. Requisitos de Sistema

2.1.1. Hardware

Considerando que o uso do sistema Odoo está associada ao uso de contêiner recomenda-se como referência o mínimo de Hardware similar ao servidor:

O PowerEdge R630 oferece densidade e produtividade, performance de 2 U em um chassi compacto de 1 U, parte da 13ª geração de servidores PowerEdge, o R630 é ideal para virtualização, gerando grandes bancos de dados transacionais ou executando aplicativos de negócios.



PowerEdge R630 - \$8,094.40	
CPU	2 x Intel® Xeon® E5-2640 v4 2.4GHz,25M Cache,8.0GT/s QPI,Turbo,HT,10C/20T (90W) Max Mem 2133MHz
RAM (Minimum)	2x 32GB RDIMM, 2400MT/s, Dual Rank, x4 Data Width
Add-in Network	1 x Intel Ethernet I350 QP 1Gb Network Card
Disk	4 x 800GB Solid State Drive SATA Read Intensive MLC 6Gbps 2.5in Hot-plug Drive, S3520

Os requisitos de hardware para o Odoo tendem a variar, visto que deve ser considerado o número de clientes que irão acessar o sistema e consumir seus recursos, havendo uma escalada de requisitos, segundo a seguinte logica:

Um mínimo de 2 CPUs 64 bits e 2 GB de RAM para atender até 5 usuários

Um mínimo de 4 CPUs 64 bits e 8 GB de RAM para atender até 20 usuários

Um mínimo de 8 CPUs 64 bits e 32 GB de RAM para atender mais de 100 usuários.

2.2. Configuração

2.2.1. Servidores

Ubuntu Server – Host Físico (PowerEdge R630 - \$8,094.40)

O Ubuntu Server 16.10 LTD associado ao Docker serão implementados em Host físico, visando melhor aproveitamento dos recursos computacionais e escalabilidade na demanda por serviços.

Odoo Server – Contêiner

O Odoo Server será implementando via contêiner (docker) disponibilizado pelo desenvolvedor da aplicação Odoo, visando simplificação das configurações e dependências, reduzindo cargas adicionais de trabalho de implantação e manutenção.

PostgreSQL SGBD – Contêiner

O PostgreSQL SGBD será implementando via contêiner (docker) disponibilizado pelo desenvolvedor da aplicação Odoo, visando simplificação de configurações e integração de componentes, também de modo a minimizar cargas adicionais de trabalho de implantação e manutenção.

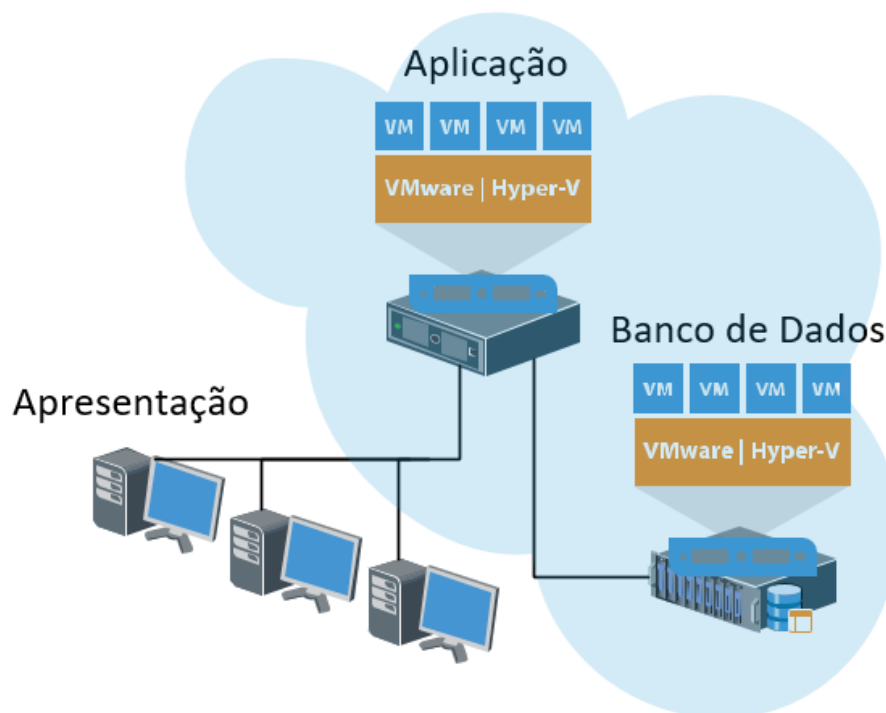
2.2.2. Cliente

Basicamente a única exigência será a existência de navegador web instalada nos dispositivos, visto que a Interface Web do Odoo é suportada por múltiplos navegadores para cada uma de suas versões. Nenhuma distinção é feita de acordo com a versão do navegador para estar atualizado, dentre as opções existentes fora elegida o navegador Mozilla Firefox como padrão, para manter uma experiência mais consistentes nos diferentes dispositivos que farão acesso ao Odoo.

ARQUITETURA ALTERNATIVA

Visto que o modelo on-premises modelo tende a exigir investimento inicial mais alto na compra de hardware e software, bem como em alguns casos existe a necessidade de contratar profissionais capacitados para gerenciar o ambiente interno, optou-se por tecnologias em sua maioria Open Source, como: Sistema Operacional (Ubuntu Server), ERP (Odoo) e Orquestração (Ansible), com um pequeno investimento no Gerenciamento de Código-fonte (GitHub), sendo em sua a maior o investimento na aquisição de novos Hardwares (servidores) em substituição ao legado, que já estava apresentando problemas para atender a demanda existente, bem como reduzir a carga sobre as estações de trabalho que também executavam muitas da aplicações, bem como aquisição das licenças para as soluções de Virtualização VMware; quanto a possibilidade de contratação profissionais se optará por consultores externos para as atividades pontuais.

3. INFRAESTRUTURA VMWARE



3.1. Requisitos de Sistema

3.1.1. Hardware

Considerando que o uso do sistema Odoo está associada ao uso de virtualização recomenda-se como referência 2 Servidores com o mínimo de Hardware similar aos servidores:

O PowerEdge R630 oferece densidade e produtividade, performance de 2 U em um chassi compacto de 1 U, parte da 13ª geração de servidores PowerEdge, o R630 é ideal para virtualização, gerando grandes bancos de dados transacionais ou executando aplicativos de negócios.



PowerEdge R630 - \$8,094.40	
CPU	2 x Intel® Xeon® E5-2640 v4 2.4GHz, 25M Cache, 8.0GT/s QPI, Turbo, HT, 10C/20T (90W) Max Mem 2133MHz
RAM (Minimum)	2x 32GB RDIMM, 2400MT/s, Dual Rank, x4 Data Width
Add-in Network	1 x Intel Ethernet I350 QP 1Gb Network Card
Disk	4 x 800GB Solid State Drive SATA Read Intensive MLC 6Gbps 2.5in Hot-plug Drive, S3520

O PowerEdge R730, um dos novos servidores PowerEdge de 13ª geração, oferece performance e utilização de armazenamento com uma configuração opcional de armazenamento híbrido no servidor que pode dar suporte ao armazenamento em camadas.



POWEREDGE R730XD - \$11,852.42	
CPU	2 x Intel® Xeon® E5-2650 v4 2.2GHz,30M Cache,9.60GT/s QPI,Turbo,HT,12C/24T (105W) Max Mem 2400MHz
RAM (Minimum)	4x 32GB RDIMM, 2400MT/s, Dual Rank, x4 Data Width
Add-in Network	2 x Intel Ethernet I350 QP 1Gb Network Card
Disk	2x 4TB 7.2K RPM SATA 6Gbps 3.5in Hot-plug Hard Drive 4x 800GB Solid State Drive SATA Mix Use 6Gbps 512n 2.5in Hot-plug Drive,3.5 HYB CARR, Hawk-M4E

Os requisitos de hardware para o Odoo tendem a variar, visto que deve ser considerado o número de clientes que irão acessar o sistema e consumir seus recursos, havendo uma escalada de requisitos, segundo a seguinte logica:

Um mínimo de 2 CPUs 64 bits e 2 GB de RAM para atender até 5 usuários

Um mínimo de 4 CPUs 64 bits e 8 GB de RAM para atender até 20 usuários

Um mínimo de 8 CPUs 64 bits e 32 GB de RAM para atender mais de 100 usuários.

Não podendo também ser desprezadas as necessidades de hardware referentes as plataformas de Virtualização da VMWare:

	Controlador de Serviços de Plataforma	Servidor do vCenter com um controlador de serviços de plataformas embutidos ou externos para um ambiente minúsculo (até 10 hosts, 100 máquinas virtuais)	vCenter Server com um Controlador de Serviços de Plataforma Integrada ou Externa para um Ambiente Pequeno (até 100 Hosts, 1000 Máquinas Virtuais)	Servidor do vCenter com um controlador de serviços de plataforma incorporada ou externa para um ambiente médio (até 400 hosts, 4.000 máquinas virtuais)	Servidor do vCenter com um controlador de serviços de plataformas embutidos ou externos para um grande ambiente (até 1.000 hosts, 10.000 máquinas virtuais)	vCenter Server com um Controlador de Serviços de Plataforma Integrada ou Externa para Ambiente X-Large (até 2.000 Hosts, 35.000 Máquinas Virtuais)
Número de CPUs	2	2	4	8	16	24
Memória	4 GB de RAM	10 GB de RAM	16 GB de RAM	24 GB de RAM	32 GB de RAM	48 GB de RAM

3.2. Configuração

3.2.1. Servidores

1. ESXi – Host Físico (PowerEdge R630 - \$8,094.40)

Odoo Server – Máquina Virtual

Host ESXi executando Máquina Virtual com o sistema operacional Ubuntu Server 16.10 LTD, utilizando o respectivo pacote de instalação e repositórios oficiais pacotes Community para instalação, visando configurações automaticamente de todas as dependências, instalação mais adequada para implantação gradual dos módulos, sendo usado a produção a longo prazo com trabalhos adicionais de implantação e manutenção.

2. ESXi – Host Físico (POWEREDGE R730XD - \$11,852.42)

PostgreSQL SGBD – Máquina Virtual

Host ESXi executando Máquina Virtual com o sistema operacional Ubuntu Server 16.10 LTD, utilizando os respectivos pacotes de instalação e repositórios oficiais pacotes para instalação.

3.2.2. Cliente

Basicamente a única exigência será a existência de navegador web instalada nos dispositivos, visto que a Interface Web do Odoo é suportada por múltiplos navegadores para cada uma de suas versões. Nenhuma distinção é feita de acordo com a versão do navegador para estar atualizado, dentre as opções existentes fora elegida os navegadores Mozilla Firefox como padrão, para manter uma experiência mais consistente nos diferentes dispositivos que farão acesso ao Odoo.

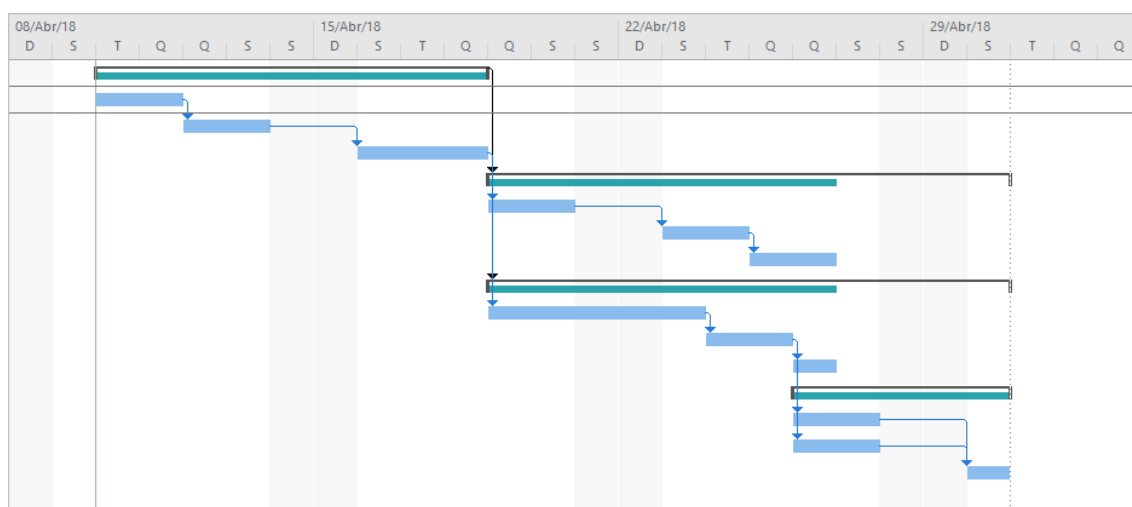
PLANEJAMENTO

4. CRONOGRAMA

Segue a baixo o cronograma simplificado para a realização das atividades de implantação do Sistema Odoo:

N	Nome da tarefa	dias	Início	Término	
1	Levantamento de Requisitos	7	Ter 10/04/18	Qua 18/04/18	
2	Entrevista	2	Ter 10/04/18	Qua 11/04/18	
3	Visita Técnica	2	Qui 12/04/18	Sex 13/04/18	2
4	Elaboração de Proposta	3	Seg 16/04/18	Qua 18/04/18	3
5	Planejamento	8	Qui 19/04/18	Seg 30/04/18	1
6	Elaboração da Linha de Base do Escopo	2	Qui 19/04/18	Sex 20/04/18	4
7	Elaboração da Linha de Base do Custo	2	Seg 23/04/18	Ter 24/04/18	6
8	Elaboração da Linha de Base do Cronograma	2	Qua 25/04/18	Qui 26/04/18	7
9	Desenvolvimento & Execução	8	Qui 19/04/18	Seg 30/04/18	1
10	Codificação	3	Qui 19/04/18	Seg 23/04/18	4
11	Debug	2	Ter 24/04/18	Qua 25/04/18	10
12	Implementação	1	Qui 26/04/18	Qui 26/04/18	11
13	Homologação	3	Qui 26/04/18	Seg 30/04/18	
14	Testes: DevOps	2	Qui 26/04/18	Sex 27/04/18	11
15	Testes: Usuários	2	Qui 26/04/18	Sex 27/04/18	11
16	Aceite final	1	Seg 30/04/18	Seg 30/04/18	14;15

4.1. Grafico de Grantt



DESENVOLVIMENTO & EXECUÇÃO

A opção feita foi por uma implementação mínima dos módulos ODOO e baseada na no uso de containers (**Instalação via Docker**) disponibilizados pela própria desenvolvedora, tanto para a aplicação quanto para o banco de dados (PostgreSQL) de modo a garantir uma implantação simples e de menor consumo de recursos computacionais.

5. INSTALAÇÃO: ODOO

A implantação e gerenciamento será realizado através da ferramenta de automatização de tarefas ANSIBLE (usando o ambiente ISE padrão o Visual Studio), sendo possível fazer o deploy de aplicações, provisionando de servidores, automatizar tarefas e outras funções, de modo a facilitar e agilizar o processo de implantação.

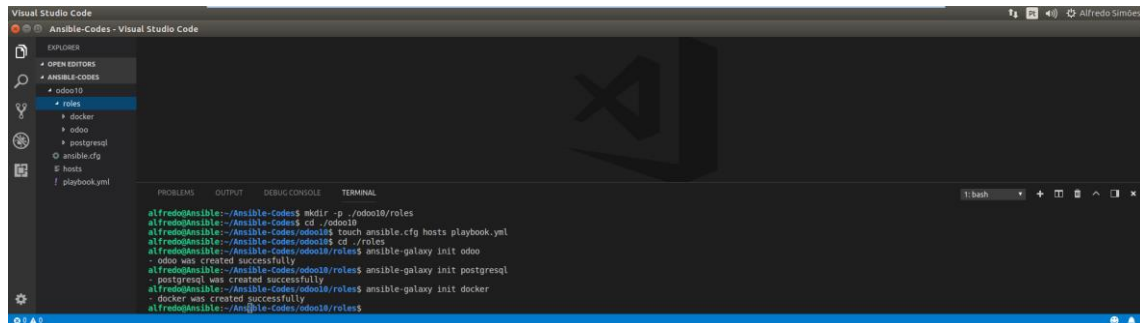
5.1. ETAPAS DO DESENVOLVIMENTO

O Ansible foi escolhido para o desenvolvimento por se tratar de uma ferramenta de automatização criada para gerenciar e executar configurações em múltiplas máquinas de uma vez, tendo entre seus principais atrativos em relação à automação:

- Linguagem simples, não necessitando de ter um notável conhecimento técnico para entender o que está sendo feito. Pelo fato da linguagem ser de fácil entendimento, é possível começar a criar serviços de automação de forma fácil e rápida.
- Conexão simples, utiliza apenas SSH para se conectar com os servidores e começar a orquestração.
- Execução simples, utiliza do princípio da idempotência, isto é, seus módulos só executarão uma ação caso mude um estado do sistema, caso contrário não tentara executar, passando de instrução a instrução até o fim sem geração de erros reexecução.

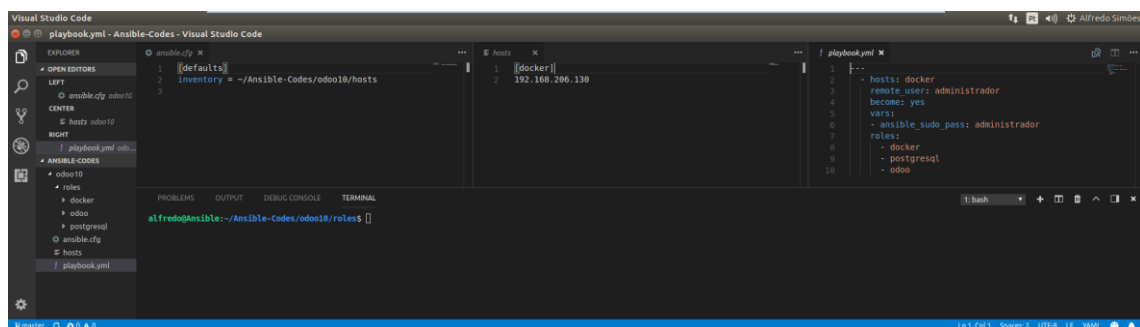
- Infraestrutura simples, utiliza uma arquitetura sem a presença de agentes (agentless), dessa forma não necessita de uma infraestrutura dedicada, fazendo com que o processo de automação seja mais eficiente e mais seguro.

5.1.1. Ansible: Criação da Estrutura de diretórios e Arquivos



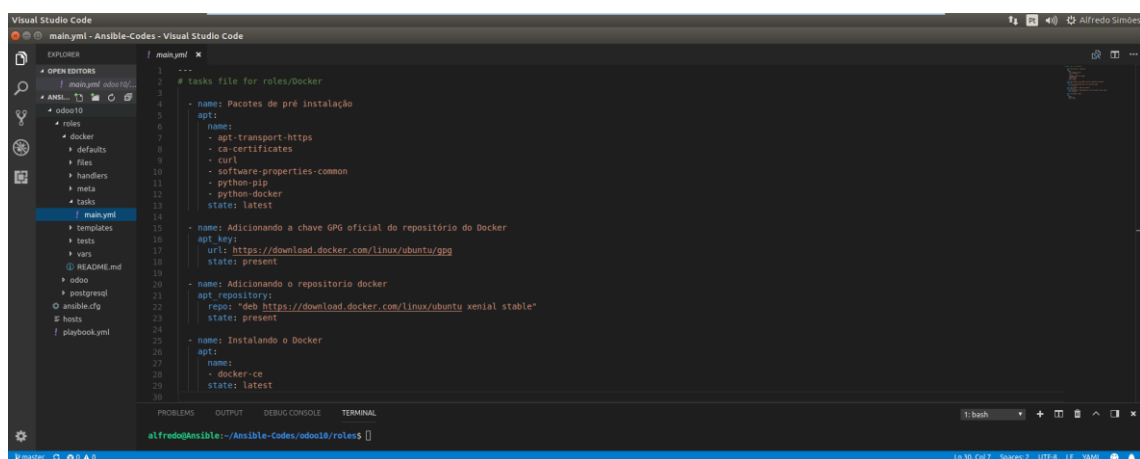
```
alfredo@Ansible:~/Ansible-Codes$ mkdir -p ./odoo18/roles
alfredo@Ansible:~/Ansible-Codes$ cd ./odoo18/
alfredo@Ansible:~/Ansible-Codes/odoo18$ touch ansible.cfg hosts playbook.yml
alfredo@Ansible:~/Ansible-Codes/odoo18$ cd ./roles/
alfredo@Ansible:~/Ansible-Codes/odoo18/roles$ ansible-galaxy init odoo
- odoo was created successfully
alfredo@Ansible:~/Ansible-Codes/odoo18/roles$ ansible-galaxy init postgresql
- postgresql was created successfully
alfredo@Ansible:~/Ansible-Codes/odoo18/roles$ ansible-galaxy init docker
- docker was created successfully
alfredo@Ansible:~/Ansible-Codes/odoo18/roles$
```

5.1.2. Ansible: Arquivos de Configuração Iniciais



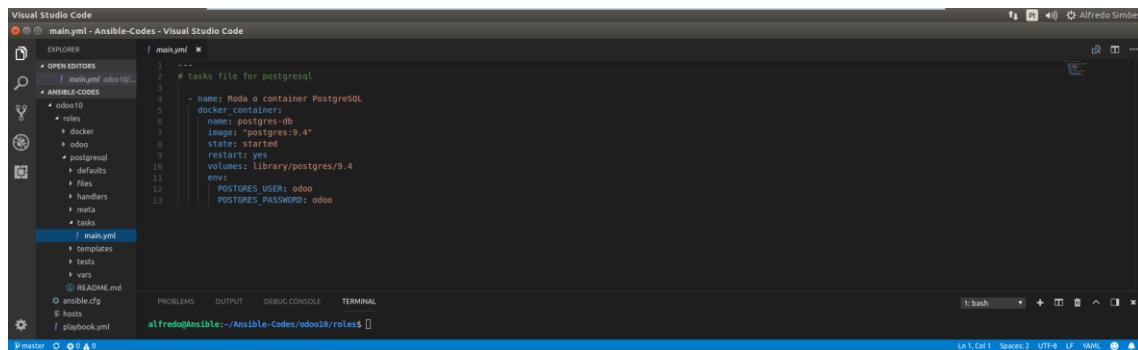
```
1 [[defaults]]
2 inventory = ~/Ansible-Codes/odoo18/hosts
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

5.1.3. Ansible: Arquivo de Instalação do Docker



```
1 ---
2 # tasks file for roles/Docker
3
4 - name: Pacotes de pré instalação
5   apt:
6     name:
7       - apt-transport-https
8       - ca-certificates
9       - curl
10      - software-properties-common
11      - python-pip
12      - python-docker
13      state: latest
14
15 - name: Adicionando a chave GPG oficial do repositório do Docker
16   apt_key:
17     url: https://download.docker.com/linux/ubuntu/gpg
18     state: present
19
20 - name: Adicionando o repositório docker
21   apt_repository:
22     repo: "deb https://download.docker.com/linux/ubuntu xenial stable"
23     state: present
24
25 - name: Instalando o Docker
26   apt:
27     name:
28       - docker-ce
29     state: latest
30
```

5.1.4. Ansible: Arquivo de Execução do Container PostgreSQL

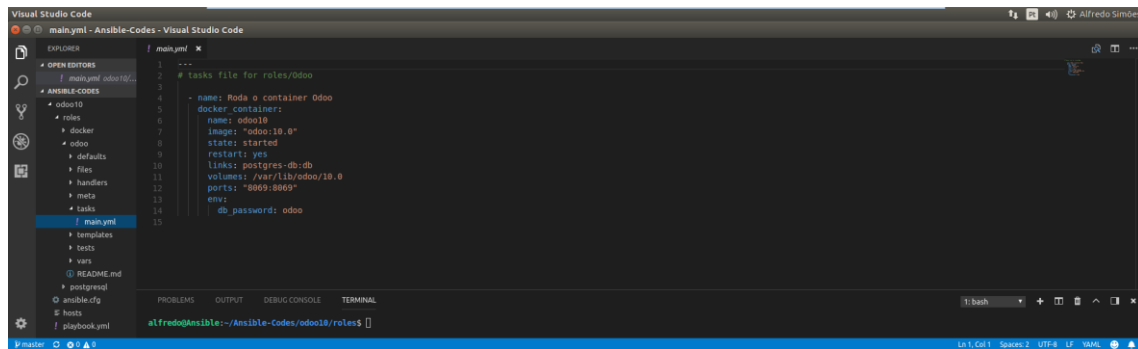


The screenshot shows the Visual Studio Code interface with an Ansible playbook named `main.yml` open. The left sidebar displays the Explorer view with a tree structure of files and folders. The main editor area shows the following Ansible code:

```
1 ---
2 # tasks file for postgresql
3
4 - name: Roda o container PostgreSQL
5   docker_container:
6     name: postgres-db
7     image: "postgres:9.4"
8     state: started
9     restart: yes
10    volumes: library/postgres/9.4
11    env:
12      POSTGRES_USER: odoo
13      POSTGRES_PASSWORD: odoo
```

The bottom status bar indicates the terminal is running `bash` in the `alfredo@Ansible:~/Ansible-Codes/odoo18/roles$` directory.

5.1.5. Ansible: Arquivo de Execução do Container Odoo



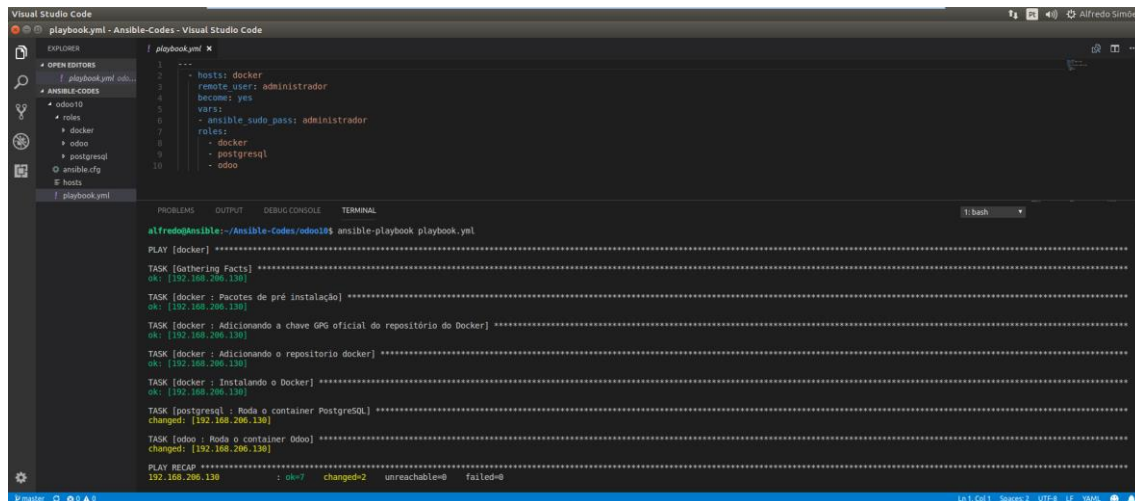
The screenshot shows the Visual Studio Code interface with an Ansible playbook named `main.yml` open. The left sidebar displays the Explorer view with a tree structure of files and folders. The main editor area shows the following Ansible code:

```
1 ---
2 # tasks file for roles/odoo
3
4 - name: Roda o container Odoo
5   docker_container:
6     name: odoo18
7     image: "odoo:18.0"
8     state: started
9     restart: yes
10    links: postgres-db:db
11    volumes: /var/lib/odoo/18.0
12    ports: "8069:8069"
13    env:
14      db_password: odoo
```

The bottom status bar indicates the terminal is running `bash` in the `alfredo@Ansible:~/Ansible-Codes/odoo18/roles$` directory.

5.2. EXECUÇÃO ANSIBLE

Execução do playbook desenvolvido para a implementação do Odoo:



```
1 ---
2   hosts: docker
3   remote_user: administrador
4   become: yes
5   vars:
6     - ansible_sudo_pass: administrador
7   roles:
8     - docker
9     - postgresql
10    - odoo
```

alfredo@alfredo:~/Ansible-Codes/odoo\$ ansible-playbook playbook.yml

PLAY [docker] *****

TASK [Gathering Facts] *****

ok: [192.168.206.130]

TASK [docker : Pacotes de pré instalação] *****

ok: [192.168.206.130]

TASK [docker : Adicionando a chave GPG oficial do repositório do Docker] *****

ok: [192.168.206.130]

TASK [docker : Adicionando o repositório docker] *****

ok: [192.168.206.130]

TASK [docker : Instalando o Docker] *****

ok: [192.168.206.130]

TASK [postgresql : Roda o container PostgreSQL] *****

changed: [192.168.206.130]

TASK [odoo : Roda o container Odoo] *****

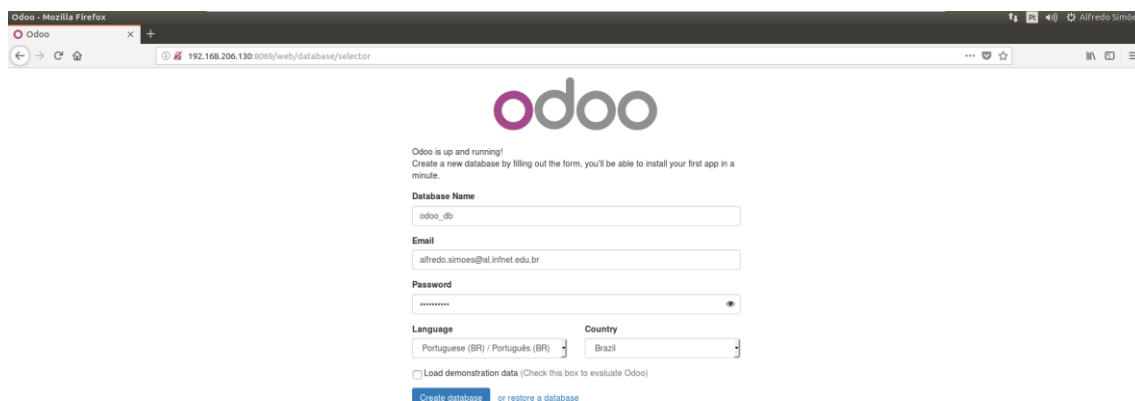
changed: [192.168.206.130]

PLAY RECAP *****

192.168.206.130 : ok=7 changed=2 unreachable=0 failed=0

Com o Odoo implementado é possível continuar o processo de configuração da ferramenta, acessível na porta 8069, para tal, é necessário acessar a aplicação utilizando o IP da host (192.168.206.130 – IP fixo) mais a porta, isto é, 192.168.206.130:8069 no endereço do navegador, como se segue a baixo:

5.2.1. Acesso Web Odoo: Criação do Banco de Dados da Aplicação



Odoo - Mozilla Firefox

Odoo

192.168.206.130:8069/web/database/selector

odoo

Odoo is up and running!
Create a new database by filling out the form, you'll be able to install your first app in a minute.

Database Name

odoo_db

Email

alfredo.simoos@al.infnet.edu.br

Password

Language

Portuguese (BR) / Portuguese (BR)

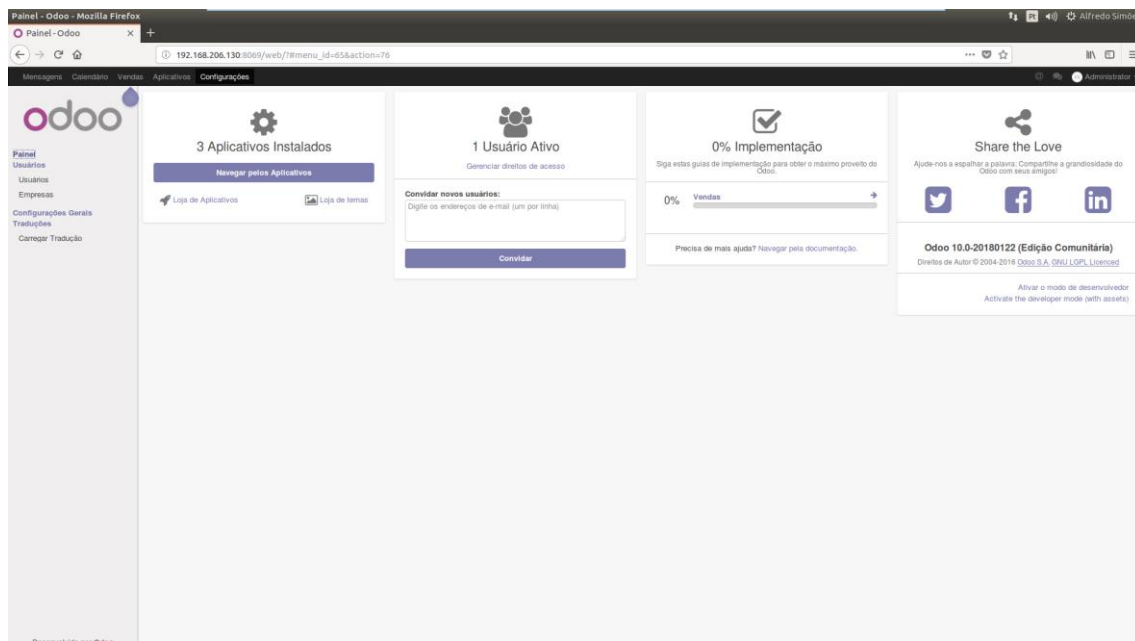
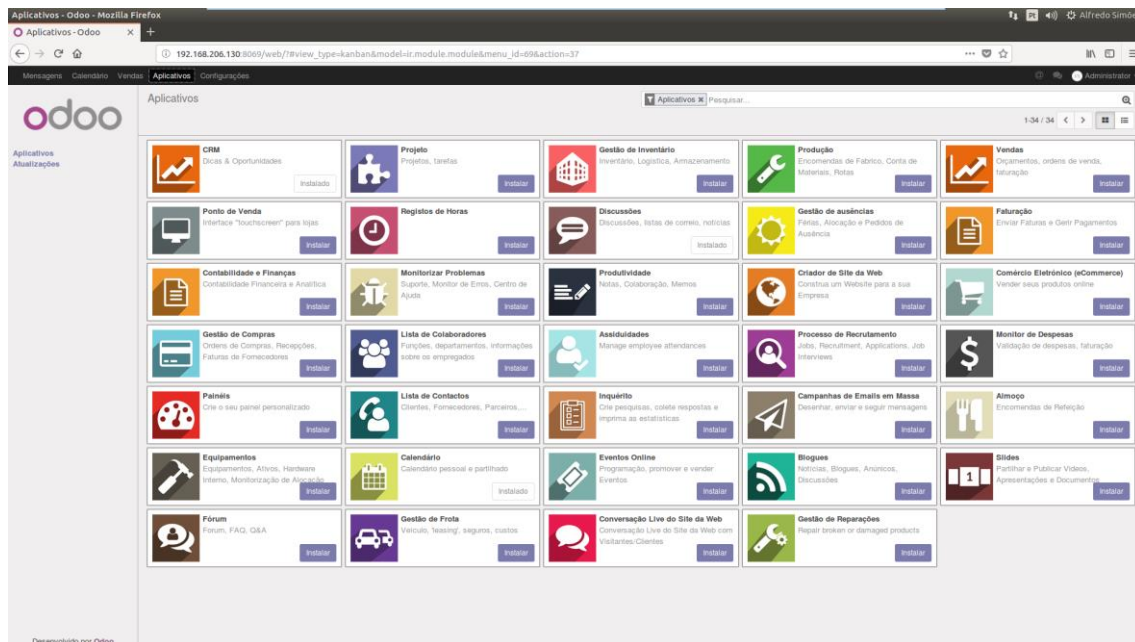
Country

Brazil

☐ Load demonstration data (Check this box to evaluate Odoo)

Create database or restore a database

5.2.2. Acesso Web Odoo: Acesso Administrativo a Aplicação



VERIFICAÇÃO

6. CONECTIVIDADE

Verificações da conectividade do servidor:

```
administrador@Docker:~$ ssh administrador@192.168.206.130
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 pacote pode ser atualizado.
0 atualização é uma atualização de segurança.

Last login: Tue Apr 10 07:46:04 2018 from 192.168.206.128
administrador@Docker:~$ ifconfig
docker0    Link encap:Ethernet  Endereço de HW 02:42:45:e0:db:fd
           inet end.: 172.17.0.1  Bcast:172.17.255.255  Masc:255.255.0.0
           UP BROADCAST MULTICAST  MTU:1500  Métrica:1
           pacotes RX:0 erros:0 descartados:0 excesso:0 quadro:0
           Pacotes TX:0 erros:0 descartados:0 excesso:0 portadora:0
           colisões:0 txqueuelen:0
           RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

```
administrador@Docker:~$ route
Tabela de Roteamento IP do Kernel

```

Destino	Roteador	MáscaraGen.	Opções	Métrica	Ref	Uso	Iface
default	PowerBox	0.0.0.0	UG	0	0	0	ens33
172.17.0.0	*	255.255.0.0	U	0	0	0	docker0
192.168.25.0	*	255.255.255.0	U	0	0	0	ens33
192.168.206.0	*	255.255.255.0	U	0	0	0	ens34

```
administrador@Docker:~$
```

7. DOCKER

Verificação do Serviço:

```
administrador@Docker:~$ sudo service docker status
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Seg 2018-04-09 20:02:09 -03; 51min ago
     Docs: https://docs.docker.com
    Main PID: 5738 (dockerd)
      Tasks: 48
     Memory: 1.1G
        CPU: 1min 13.618s
    CGroup: /system.slice/docker.service
            └─5738 /usr/bin/dockerd -H fd://
            └─5747 docker-containerd --config /var/run/docker/containerd/containerd.toml
            └─9142 docker-containerd-shim -namespace moby -workdir /var/lib/docker/containerd/daemon/io.containerd.runtime.v1.linux/moby/c32f40ad46f0f43f42003d61eff4f8af25a0b8dd51d16442db34c3
            └─9351 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8069 -container-ip 172.17.0.3 -container-port 8069
            └─9376 docker-containerd-shim -namespace moby -workdir /var/lib/docker/containerd/daemon/io.containerd.runtime.v1.linux/moby/5e0621c98ee566dbf837b9a2939b7d09b344ac7ed4b16be8733d3d34

Abr 09 20:07:30 Docker dockerd[5738]: time="2018-04-09T20:07:30-03:00" level=info msg="shim docker-containerd-shim started" address="/containerd-shim/moby/c32f40ad46f0f43f42003d61eff4f8af25a0b8dd51d16442db34c39880b7c381" module="containerd/tas
Abr 09 20:07:32 Docker dockerd[5738]: time="2018-04-09T20:07:32-03:00" level=info msg="shim reaped" id=5e0621c98ee566dbf837b9a2939b7d09b344ac7ed4b16be8733d3d34 module="containerd/tas
Abr 09 20:07:32 Docker dockerd[5738]: time="2018-04-09T20:07:32-03:00" level=info msg="Ignoring event" module=libcontainerd namespace=moby topic=/tasks/delete type="events.TaskDel
Abr 09 20:07:32 Docker dockerd[5738]: time="2018-04-09T20:07:32-03:00" level=info msg="shim docker-containerd-shim started" address="/containerd-shim/moby/5e0621c98ee566dbf837b9a2939b7d09b344ac7ed4b16be8733d3d34 module="containerd/tas
Abr 09 20:07:45 Docker dockerd[5738]: time="2018-04-09T20:07:45-03:00" level=info msg="shim reaped" id=c32f40ad46f0f43f42003d61eff4f8af25a0b8dd51d16442db34c39880b7c381 module="containerd/tas
Abr 09 20:07:45 Docker dockerd[5738]: time="2018-04-09T20:07:45-03:00" level=info msg="Ignoring event" module=libcontainerd namespace=moby topic=/tasks/delete type="events.TaskDel
Abr 09 20:07:45 Docker dockerd[5738]: time="2018-04-09T20:07:45-03:00" level=info msg="shim docker-containerd-shim started" address="/containerd-shim/moby/c32f40ad46f0f43f42003d61eff4f8af25a0b8dd51d16442db34c39880b7c381" module="containerd/tas
Abr 09 20:07:47 Docker dockerd[5738]: time="2018-04-09T20:07:47-03:00" level=info msg="shim reaped" id=5e0621c98ee566dbf837b9a2939b7d09b344ac7ed4b16be8733d3d34 module="containerd/tas
Abr 09 20:07:47 Docker dockerd[5738]: time="2018-04-09T20:07:47-03:00" level=info msg="Ignoring event" module=libcontainerd namespace=moby topic=/tasks/delete type="events.TaskDel
Abr 09 20:07:47 Docker dockerd[5738]: time="2018-04-09T20:07:47-03:00" level=info msg="shim docker-containerd-shim started" address="/containerd-shim/moby/5e0621c98ee566dbf837b9a2939b7d09b344ac7ed4b16be8733d3d34 module="containerd/tas
lines 1-25/25 (END)
```

```
administrador@docker:~$ sudo docker info
Containers: 2
  Running: 2
  Paused: 0
  Stopped: 0
Images: 2
Server Version: 18.03.0-ce
Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
  Volume: local
  Network: bridge host macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
  containerd version: cf9d4396dc68220d1cecb6686a6cc3aa5ce3667c
  runc version: 4f53a81fb7c994640722ac585fa9ca548971871
  init version: 949e6fa
Security Options:
  apparmor
  seccomp
  Profile: default
Kernel Version: 4.4.0-119-generic
Operating System: Ubuntu 16.04.4 LTS
OSType: linux
Architecture: x86_64
CPUs: 2
Total Memory: 3.842GiB
Name: docker
ID: GF30:XXBV:7NRZ:WLGW:OC44:YUN7:GW2X:CN6F:HGGV:COAL:NHSL:UAL6
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.0/8
Live Restore Enabled: false

WARNING: No swap limit support
administrador@docker:~$
```

Verificação das Imagens e Containers:

```
administrador@docker:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
postgres            9.4                ed5db6e669ff       3 weeks ago        263MB
odoo                 10.0               76cdf99f068c       3 weeks ago        860MB

administrador@docker:~$ sudo docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
de8621c98ee5       odoo:10.0          "/entrypoint.sh odoo" 34 minutes ago     Up 33 minutes      0.0.0.0:8069->8069/tcp, 8071/tcp    odoo10
c3740e46409        postgres:9.4        "docker-entrypoint.s" 34 minutes ago     Up 33 minutes      5432/tcp            postgres-db

administrador@docker:~$
```

7.1 Postgresql

Verificação dos logs para garantir o Banco de Dados está executando normalmente:

```
administrador@docker:~$
Success. You can now start the database server using:
    postgres -D /var/lib/postgresql/data
or
    pg_ctl -D /var/lib/postgresql/data -l logfile start

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.
waiting for server to start...LOG: could not bind IPv6 socket: Cannot assign requested address
HINT:  Is another postmaster already running on port 5432? If not, wait a few seconds and retry.
LOG: database system was shut down at 2018-04-09 23:06:59 UTC
LOG: MultiXact member wraparound protections are now enabled
LOG: database system is ready to accept connections
LOG: autovacuum launcher started
done
server started
CREATE DATABASE
CREATE ROLE

/usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*

LOG: received fast shutdown request
LOG: aborting any active transactions
LOG: autovacuum launcher shutting down
waiting for server to shut down...LOG: shutting down
LOG: database system is shut down
done
server stopped

PostgreSQL init process complete; ready for start up.

LOG: database system was shut down at 2018-04-09 23:07:02 UTC
LOG: MultiXact member wraparound protections are now enabled
LOG: database system is ready to accept connections
LOG: autovacuum launcher started
LOG: received smart shutdown request
LOG: autovacuum launcher shutting down
LOG: shutting down
LOG: database system is shut down
LOG: database system was shut down at 2018-04-09 23:07:30 UTC
LOG: MultiXact member wraparound protections are now enabled
LOG: database system is ready to accept connections
LOG: autovacuum launcher started
LOG: received smart shutdown request
LOG: autovacuum launcher shutting down
LOG: shutting down
LOG: database system is shut down
LOG: database system was shut down at 2018-04-09 23:07:45 UTC
LOG: MultiXact member wraparound protections are now enabled
LOG: database system is ready to accept connections
LOG: autovacuum launcher started
administrador@docker:~$
```

7.2 Odoo

Verificação dos logs para garantir a aplicação está executando normalmente:

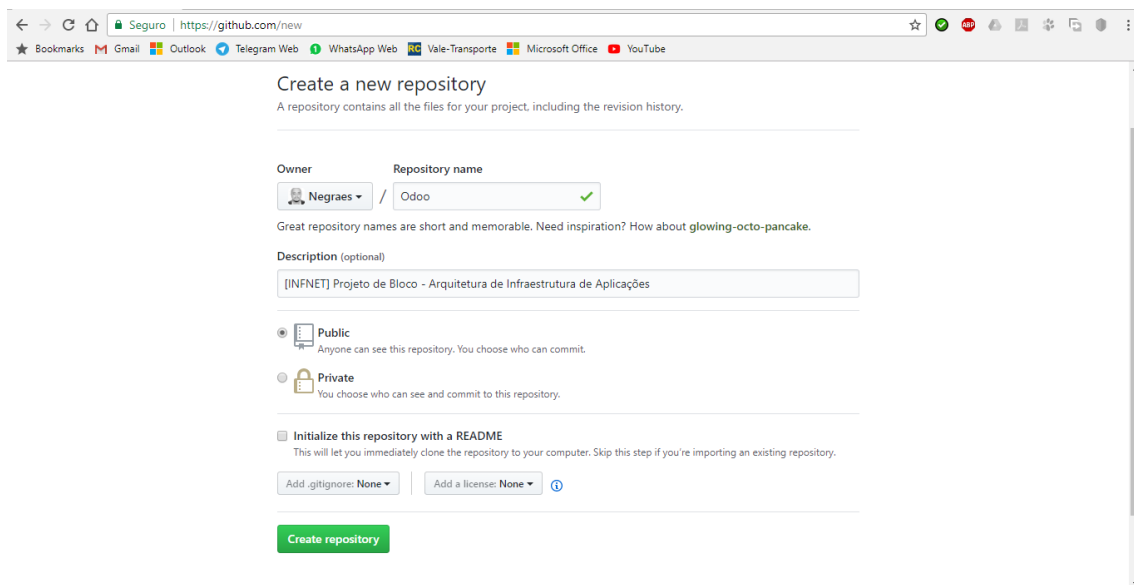
```
administrador@docker:~$ sudo docker logs odoo10
2018-04-09 23:06:59.897 1 INFO ? odoo: Odoo version 10.0-20180122
2018-04-09 23:06:59.897 1 INFO ? odoo: Using configuration file at /etc/odoo/odoo.conf
2018-04-09 23:06:59.897 1 INFO ? odoo: addons paths: ['/var/lib/odoo/addons/10.0', u'/mnt/extra-addons', u'/usr/lib/python2.7/dist-packages/odoo/addons']
2018-04-09 23:06:59.897 1 INFO ? odoo: database: odoo@172.17.0.2:5432
2018-04-09 23:06:59.916 1 INFO ? odoo.service.server: HTTP service (werkzeug) running on 0.0.0.0:8069
2018-04-09 23:07:31.733 1 INFO ? odoo.service.server: Initiating shutdown
2018-04-09 23:07:31.733 1 INFO ? odoo.service.server: Hit CTRL-C again or send a second signal to force the shutdown.
2018-04-09 23:07:33.372 1 INFO ? odoo: Odoo version 10.0-20180122
2018-04-09 23:07:33.372 1 INFO ? odoo: Using configuration file at /etc/odoo/odoo.conf
2018-04-09 23:07:33.372 1 INFO ? odoo: addons paths: ['/var/lib/odoo/addons/10.0', u'/mnt/extra-addons', u'/usr/lib/python2.7/dist-packages/odoo/addons']
2018-04-09 23:07:33.373 1 INFO ? odoo: database: odoo@172.17.0.2:5432
2018-04-09 23:07:33.393 1 INFO ? odoo.service.server: HTTP service (werkzeug) running on 0.0.0.0:8069
2018-04-09 23:07:46.702 1 INFO ? odoo.service.server: Initiating shutdown
2018-04-09 23:07:46.702 1 INFO ? odoo.service.server: Hit CTRL-C again or send a second signal to force the shutdown.
2018-04-09 23:07:48.143 1 INFO ? odoo: Odoo version 10.0-20180122
2018-04-09 23:07:48.143 1 INFO ? odoo: Using configuration file at /etc/odoo/odoo.conf
2018-04-09 23:07:48.143 1 INFO ? odoo: addons paths: ['/var/lib/odoo/addons/10.0', u'/mnt/extra-addons', u'/usr/lib/python2.7/dist-packages/odoo/addons']
2018-04-09 23:07:48.144 1 INFO ? odoo: database: odoo@172.17.0.2:5432
2018-04-09 23:07:48.161 1 INFO ? odoo.service.server: HTTP service (werkzeug) running on 0.0.0.0:8069
2018-04-09 23:11:32.074 1 INFO ? odoo.addons.report.models.report: Will use the wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2018-04-09 23:11:32.555 1 INFO ? odoo.http: HTTP Configuring static files
2018-04-09 23:11:32.565 1 INFO ? odoo.http: Generating nondb routing
2018-04-09 23:11:32.579 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET / HTTP/1.1" 303 -
2018-04-09 23:11:32.631 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web HTTP/1.1" 303 -
2018-04-09 23:11:32.716 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web/database/selector HTTP/1.1" 200 -
2018-04-09 23:11:32.854 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web/static/lib/fontawesome/css/font-awesome.css HTTP/1.1" 200 -
2018-04-09 23:11:32.872 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web/static/lib/jquery/jquery.js HTTP/1.1" 200 -
2018-04-09 23:11:32.882 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web/static/lib/bootstrap/css/bootstrap.css HTTP/1.1" 200 -
2018-04-09 23:11:32.901 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web/static/lib/bootstrap/js/model.js HTTP/1.1" 200 -
2018-04-09 23:11:32.917 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web/static/lib/bootstrap/js/tooltip.js HTTP/1.1" 200 -
2018-04-09 23:11:32.931 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web/static/lib/bootstrap/js/dropdown.js HTTP/1.1" 200 -
2018-04-09 23:11:32.943 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:32] "GET /web/static/src/img/logo2.png HTTP/1.1" 200 -
2018-04-09 23:11:32.945 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:33] "GET /web/static/src/img/fontawesome-webfont.woff?v=4.5.0 HTTP/1.1" 200 -
2018-04-09 23:11:32.979 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:11:33] "GET /web/static/src/img/favicon.ico HTTP/1.1" 200 -
2018-04-09 23:14:47.927 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:47] "GET / HTTP/1.1" 303 -
2018-04-09 23:14:47.945 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:47] "GET /web HTTP/1.1" 303 -
2018-04-09 23:14:47.985 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:47] "GET /web/database/selector HTTP/1.1" 200 -
2018-04-09 23:14:48.000 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:48] "GET /web/static/lib/fontawesome/css/font-awesome.css HTTP/1.1" 200 -
2018-04-09 23:14:48.004 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:48] "GET /web/static/lib/bootstrap/css/bootstrap.css HTTP/1.1" 200 -
2018-04-09 23:14:48.006 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:48] "GET /web/static/lib/jquery/jquery.js HTTP/1.1" 200 -
2018-04-09 23:14:48.018 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:48] "GET /web/static/lib/bootstrap/js/model.js HTTP/1.1" 200 -
2018-04-09 23:14:48.023 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:48] "GET /web/static/lib/bootstrap/js/tooltip.js HTTP/1.1" 200 -
2018-04-09 23:14:48.026 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:48] "GET /web/static/lib/bootstrap/js/dropdown.js HTTP/1.1" 200 -
2018-04-09 23:14:48.032 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:48] "GET /web/static/src/img/logo2.png HTTP/1.1" 200 -
2018-04-09 23:14:48.132 1 INFO ? werkzeug: 192.168.206.128 - - [09/Apr/2018 23:14:48] "GET /web/static/lib/fontawesome/fonts/fontawesome-webfont.woff?v=4.5.0 HTTP/1.1" 200 -
2018-04-09 23:14:50.884 1 INFO ? None odoo.service.db: Create database 'odoo_db'.
2018-04-09 23:14:52.256 1 INFO ? odoo db odoo.modules.loading: Loading 1 modules...
2018-04-09 23:14:52.379 1 INFO ? odoo db odoo.modules.registry: module base: creating or updating database tables
2018-04-09 23:14:53.065 1 INFO ? odoo db odoo.models: Computing parent left and right for table ir.ui.menu...
2018-04-09 23:14:53.790 1 INFO ? odoo db odoo.models: Computing parent left and right for table res.partner.category...
2018-04-09 23:14:57.539 1 INFO ? odoo db odoo.models: Storing computed values of ir.ui.view.fields.model_data_id
2018-04-09 23:14:57.541 1 INFO ? odoo db odoo.models: Storing computed values of ir.attachment.fields.res_name
2018-04-09 23:14:57.543 1 INFO ? odoo db odoo.models: Storing computed values of res.partner.bank.fields.sanitized_acc_number
2018-04-09 23:14:57.543 1 INFO ? odoo db odoo.models: Storing computed values of res.users.fields.share
```

DOCUMENTAÇÃO

O GitHub é uma plataforma para hospedagem de código-fonte com controle de versão usando o Git, permitindo a desenvolvedores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source a partir de qualquer lugar do mundo.

8. GitHub: Repositório de Código Fonte

8.1.1. Criação



The screenshot shows the GitHub 'Create a new repository' page in a web browser. The browser's address bar shows 'https://github.com/new'. The page title is 'Create a new repository' with a subtitle 'A repository contains all the files for your project, including the revision history.' The form includes fields for 'Owner' (set to 'Negraes') and 'Repository name' (set to 'Odoo' with a green checkmark). Below these is a 'Description (optional)' field containing '[INFNET] Projeto de Bloco - Arquitetura de Infraestrutura de Aplicações'. There are two radio buttons for visibility: 'Public' (selected) and 'Private'. At the bottom, there is a checkbox for 'Initialize this repository with a README' and two dropdown menus for 'Add .gitignore: None' and 'Add a license: None'. A green 'Create repository' button is at the bottom.

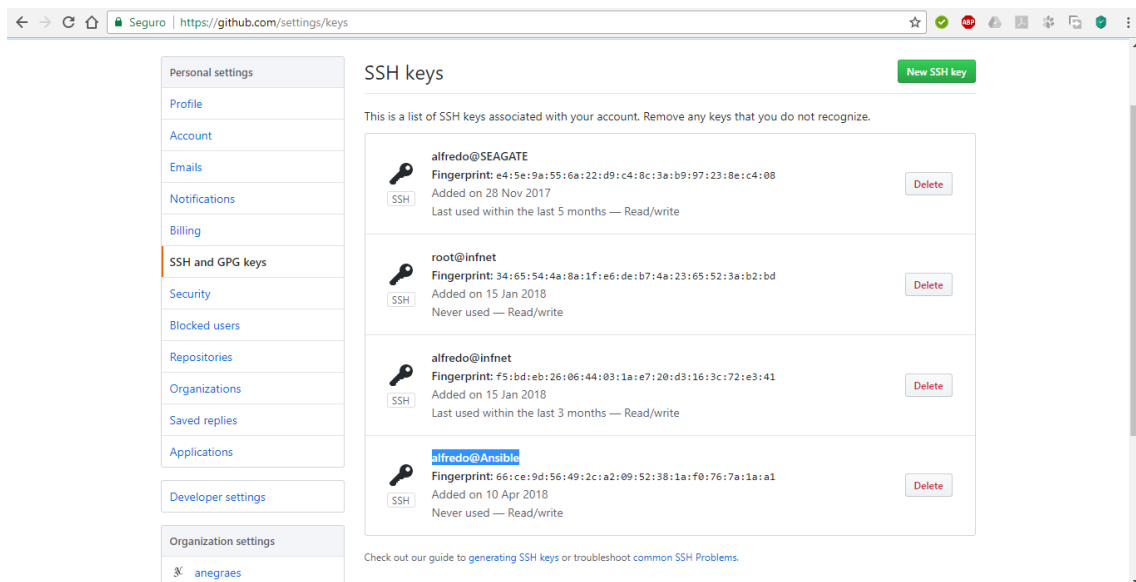
8.1.2. Permissões

Configurações SSH (chaves) para acesso e transferências de códigos fonte:

Host

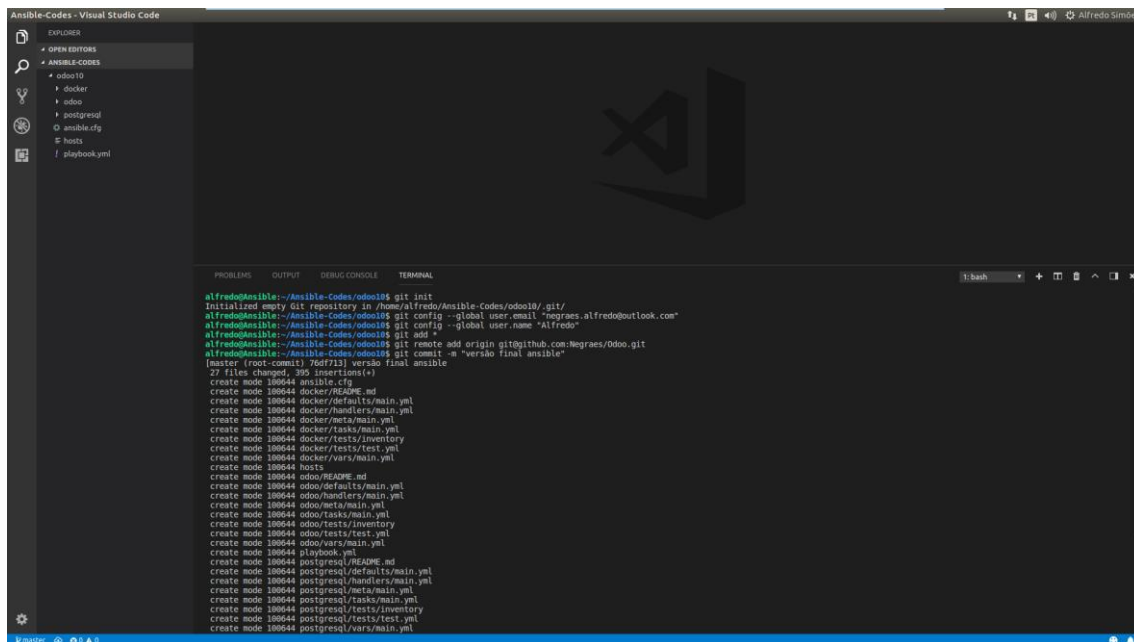
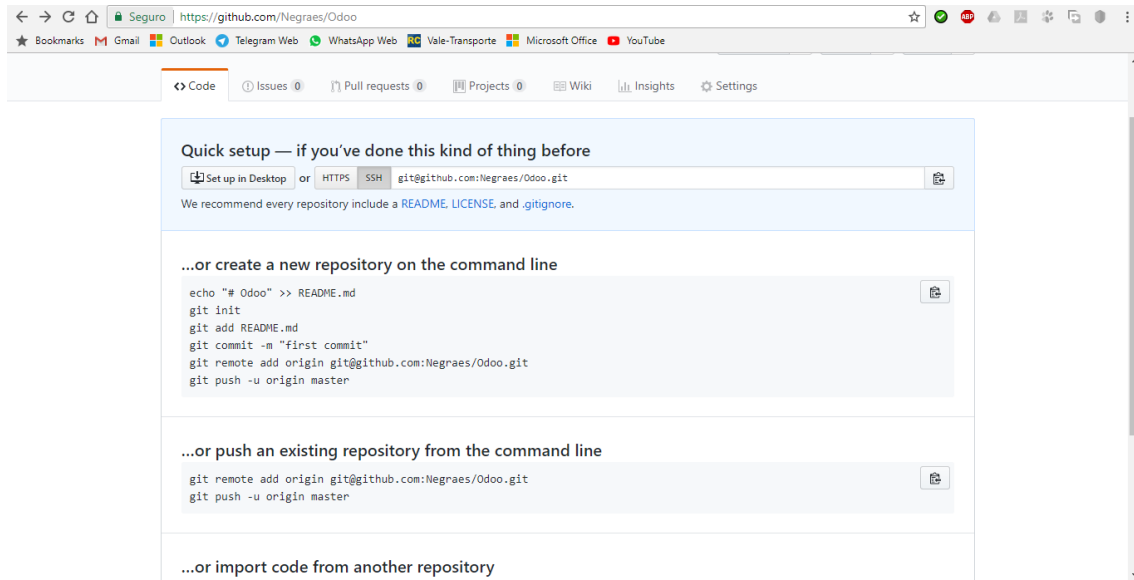
```
alfredo@Ansible: ~/ssh
alfredo@Ansible:~/ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAU1Xxb0L8ki3YBeKlwEmW2Sk3PaLOUT8xbn9uaT2gwZuL/B1DN7CDog13pfManT2T0//f3TU3x64wZ/L8czoUVMX1e-U04/hohGLJY6D3VMsc7kCV0VS8YBY7G5JHf252Vv718Vfj7oe1Ryq8
1UIt6IB/1ZefH9WIMMIbemaQ0wgrYLUeWcQ03VABp3r9XZ7x1C6aCAdkm95owLuJ5UWbXmT03VbH71Lc5vVvmmq3qT1ga/knCzXDE2uY3y90397yhgYYEtu6DzAwR/Jq5wLWjz1cVGxWpx1CoLmLm6F/8ne95FZ5iWgNTDonayzqIQ1b2+8KF2F
edmiNistrador@docker
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQC91zUVOjZUMNLLk5TD3HnCERCm9r1pUW92j1j7KqmMuqKst/puVenlaqWfFEmBCBgUj/16kxET3ft1AAS18g40Z09IppreyFZcu8I3Ck140TcowiFwZ7VqzYB+4/qvK1Jpp545x1RcMleU6AeDy
53+2qkX//xMJLk6hntkw6mf3hCTQxF5Cd19snudKZsL8R1ZeuF2e0BhG1Bm7IoySUI51BmcdB12f8dxxkUZaKT8FYrY0T1ZUF7tPwupGXHQKRP0sC+jvyUarwv0178V+L54BCZP0RyeRwW23a5Eya3GjAEKjU6WJmGN5J0qoFM9MndAaQVvYLhwh/
root@docker
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQ00AnSoGJ0yU8aDcC8CNI1Ddeas1B4C0UkR30U7N141AxtkN3/faw58c99P7/Bap7Cf1wbz9wJvUWbpl1-dm:HyqWZK0Dy38VH-7w@rg80EdmZKavn01CPL/F1G/6qv1bxZvTZzNgwymbusAz
03BF1om17/8B2Ayo9mLL1RRWfA1101F2g8xkDmR35vWbP2+joe1Nq3ohnd1BaAInSE1yr4TzRias16AllUzytaVMT+SoeR0Cg3hup6XURTfinkf33aEGcxtR51fCHIM4L34t4GtTYG1ZrnuFLTA9XE+vdKCb+rZK3uPNa/sssE0ND5mWCIYyqr
alfredo@Ansible
alfredo@Ansible:~/ssh$
```

GitHub

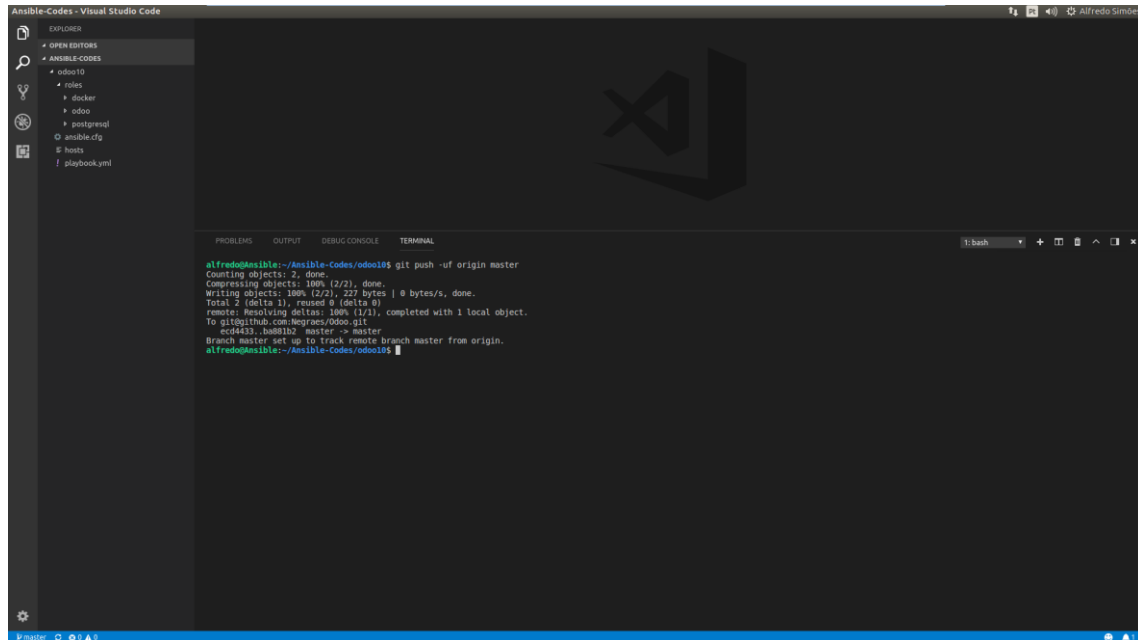


8.2. Github: Commit & Push

8.2.1. Configurações de Sincronização



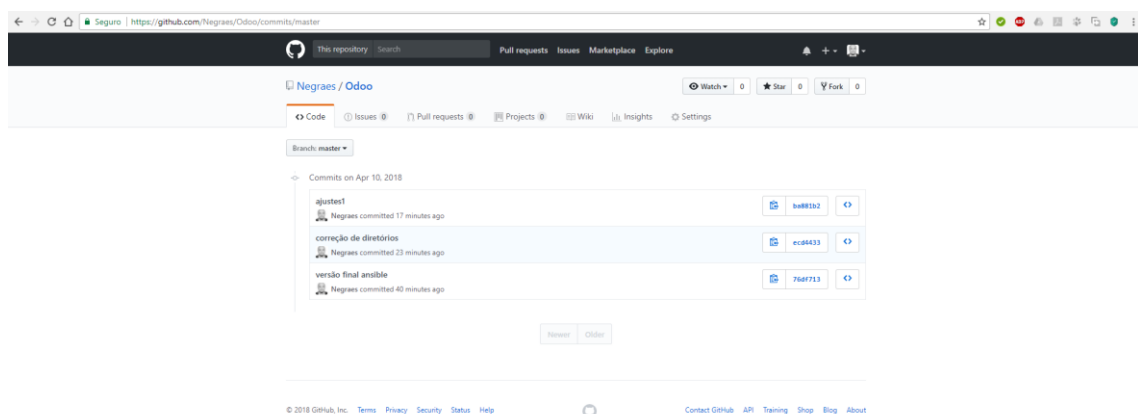
8.2.2. Sincronização com o Repositório:



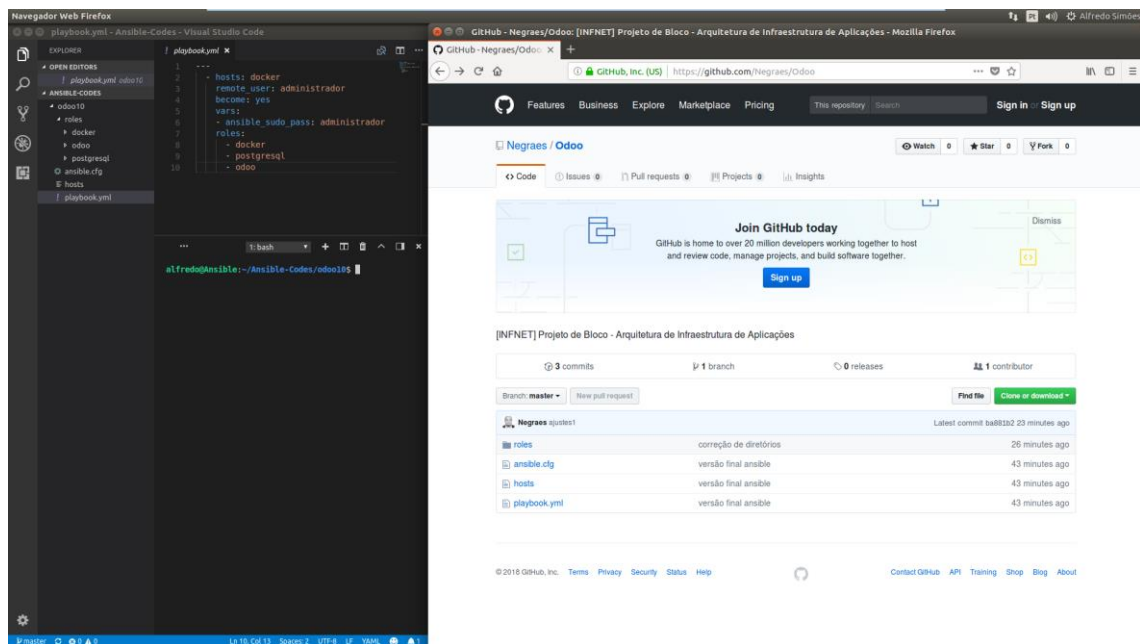
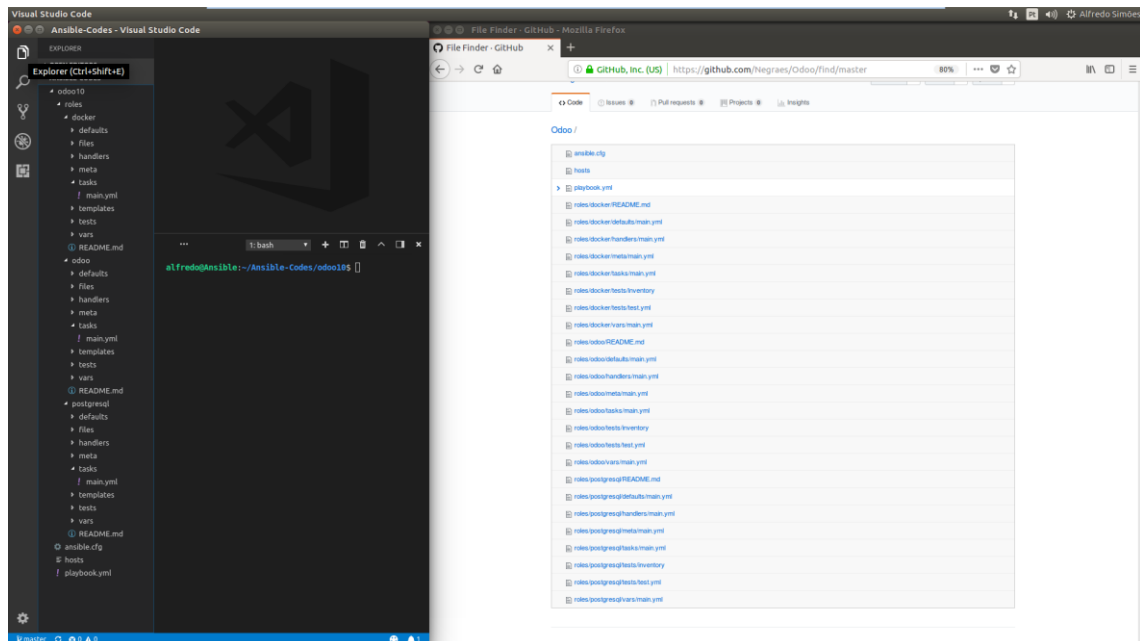
The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the output of a `git push -uf origin master` command. The output indicates that the push was successful, with objects being counted, compressed, and written to the remote repository. The terminal text is as follows:

```
alfredo@Ansible:~/Ansible-Codes/odoo10$ git push -uf origin master
Counting objects: 2, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 222 bytes | 0 bytes/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Negraes/Odoo.git
   ec4433..ba881b2 master -> master
Branch master set up to track remote branch master from origin.
alfredo@Ansible:~/Ansible-Codes/odoo10$
```

8.3. Versionamento

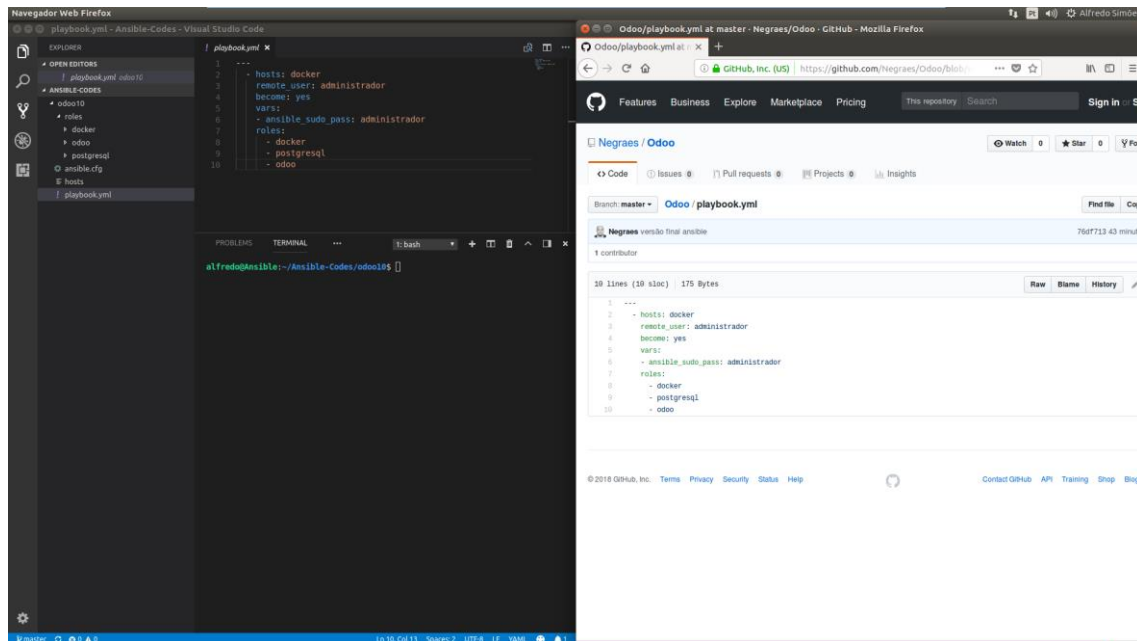


8.4. Repositório Populado

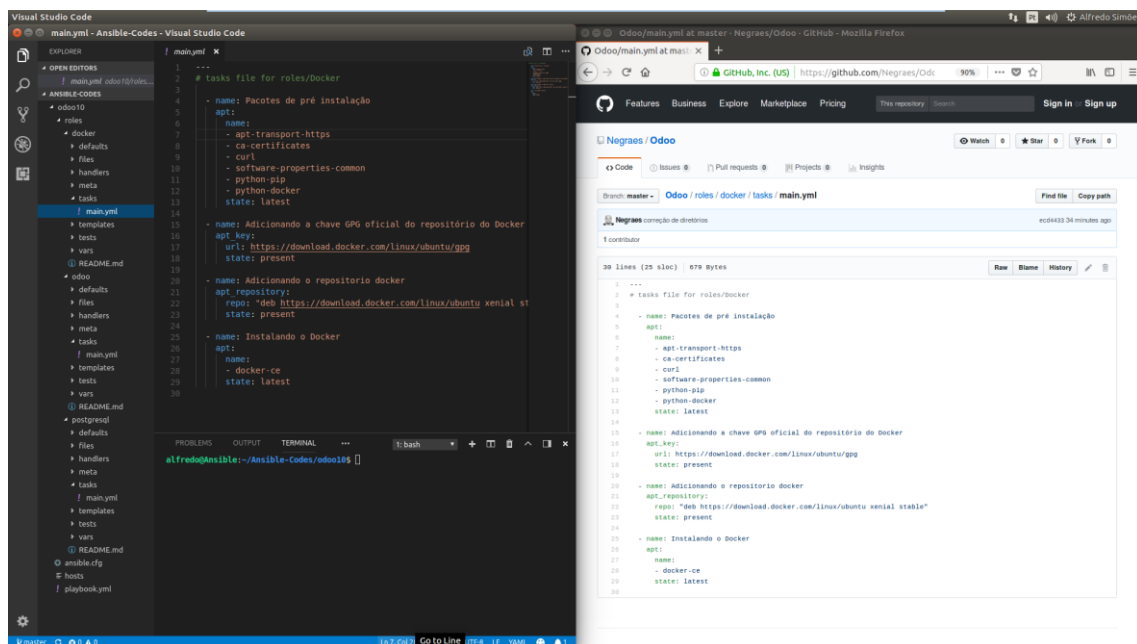


<https://github.com/Negraes/Odoo>

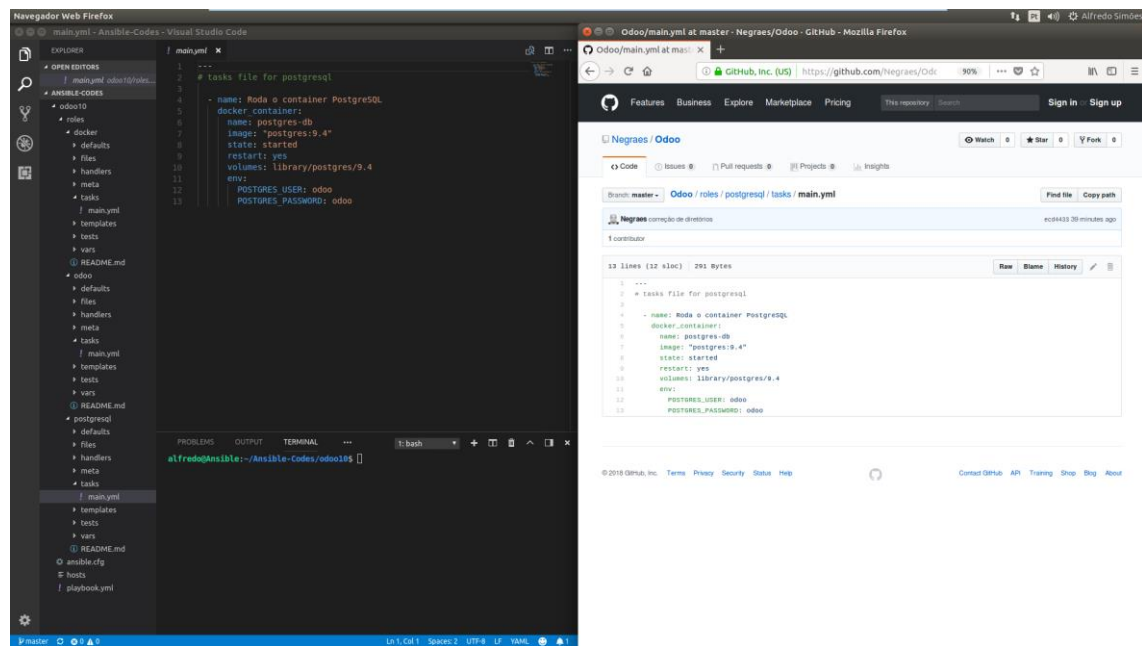
8.4.1. Playbook



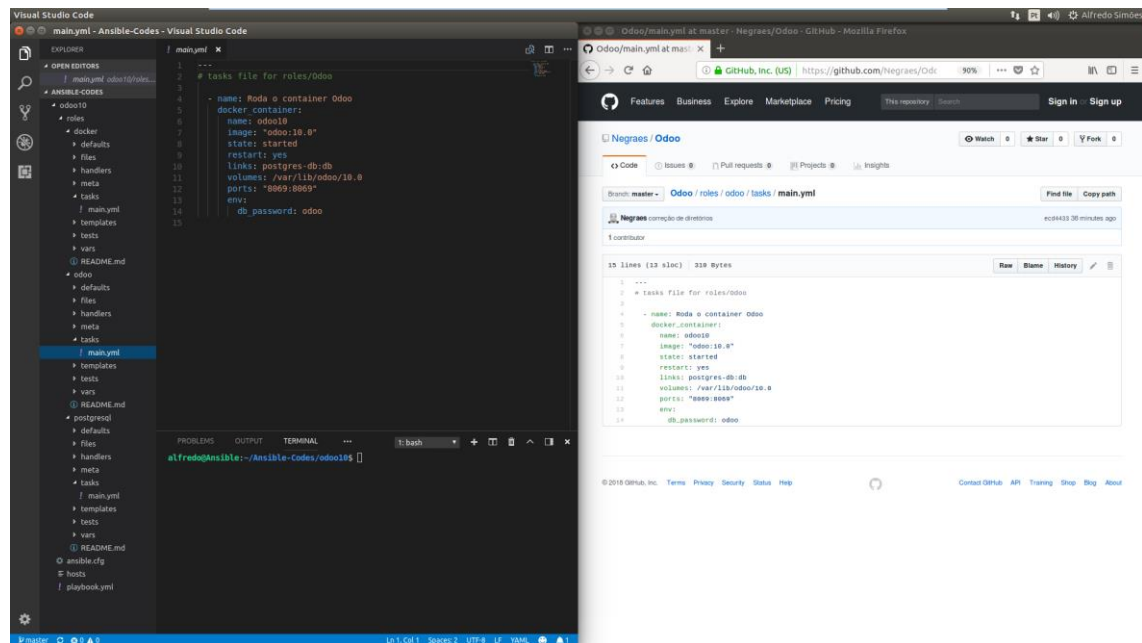
8.4.2. Docker



8.4.3. PostgreSQL



8.4.4. Odoo



CONCLUSÃO

O provisionamento de um servidor de aplicação tende a ser um processo longo e complexo, porém com a associação das tecnologias certas, como uma ferramenta de orquestração como o Ansible e uma de implementação de containers como o Docker pode ser um diferencial na facilitação desse processo inicial e a redução sensível da complexidade e demora para implantação uma vez que se pode reaproveitar o código original que iniciou o processo de base no qual foi modelado, podendo assim ser replicado inúmeras vezes sem erros e com pouca ou nenhuma modificação de ambiente a ambiente.

O custo de recursos que sejam eles computacionais, financeiros ou humanos também é sensivelmente impactado por implantações orientadas em uma perspectiva DevOps associada ao ecossistema Open Source, dispondo ampla gama soluções amplamente testadas, documentadas e muito das vezes otimizadas.

Graças a essas características principais, a execução do presente trabalho transcorreu sem grandes dificuldades e contanto com vastas condições para pesquisa e testes, permitindo, mesmo com imprevistos no cronograma original, a execução conforme e a tempo sem ultrapassar o tempo ou recursos disponíveis.

BIBLIOGRAFIA

- **WALKER**, Matt. *CEH Certified ethical hacker*. New York: McGraw-Hill, 2014.
- **TIPTON**, Harold F., **HERNANDEZ**, Steven. *Official (ISC)² guide to the CISSP CBK*. Boca Raton: CRC Press, 2013.
- **PELTIER**, Thomas R. *Information security policies, procedures, and standards: guideline for effective information security management*. Florida: Auerbach Publications, 2002
- **ISACA**. *CERTIFIED INFORMATION SECURITY MANAGER: CISM review manual*. ILLINOIS, EUA: ISACA, 2012
- **MUNIZ**, Joseph; **LAKHANI**, Aamir. *Web Penetration Testing with Kali Linux*. Reino Unido: Packt Publishing, 2013.
- **STUTTARD**, Dafydd; **PINTO**, Marcus. *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. USA: Wiley, 2011.
- **SIQUEIRA**, Luciano Antonio. *Certificação LPI-1 (101 - 102)*. Brasil: Alta Books, 2014.
- **SIQUEIRA**, Luciano Antonio. *Certificação LPI-2 (201 - 202)*. Brasil: Alta Books, 2014.