



# João Pedro Jesus dos Santos

**Projeto de Bloco - Arquitetura de Infraestrutura de  
Aplicações**

# Sumário

- Capítulo 1 - História e Aplicação - 3
  - História - 4
  - Aplicação timesheet - 5/11
- Capítulo 2 - Tecnologias - 12
  - Projeto virtualizado (LAMP) - 13
  - Pré-requisito - 14
  - Diagrama físico - 15/16
  - Diagrama virtualizado - 17/18
  - Gerenciamento código fonte - 19
  - Arquitetura da solução - 20
  - Comparação da solução de virtualização - 21
- Capítulo 3 - Implementação - 22
  - Passo a passo - 23/24
  - Cronograma - 25
  - Processo de implementação - 26 /87
- Capítulo 4 - Resultados - 88
  - Considerações finais - 89

# Capítulo 1 - História e aplicação

# Expert informática

A Expert Informática atua no mercado de TI há 10 anos na cidade do Rio de Janeiro, focado no sistemas para empresas que atuam no ramo de previdência. Atualmente a empresa possui 50 funcionários, devido uma grande demanda de requisições que estavam sendo solicitadas no último mês foram contratados 15 novos funcionários.

Sendo assim uma parte da alta gerência da empresa estava em dúvida se seria necessário essa quantidade de contratação e que os antigos funcionários já conseguiram realizar as novas requisições, portanto foi se decidido criar um timesheet para avaliar o nível, agilidade e quantas solicitações cada funcionário está realizando mensalmente.

# Aplicação Timesheet

## JavaScript/HTML5 Timesheet (PHP, MySQL)

[DayPilot for JavaScript](#) - HTML5 Calendar/Scheduling Components for JavaScript/Angular/React

Employee:

DEMO		June 2020																							
Day	Total	12 AM	1 AM	2 AM	3 AM	4 AM	5 AM	6 AM	7 AM	8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	7 PM	8 PM	9 PM	10 PM	11 PM
6/1/2020	0:00 hours																								
6/2/2020	0:00 hours																								
6/3/2020	0:00 hours																								
6/4/2020	0:00 hours																								
6/5/2020	0:00 hours																								
6/6/2020	0:00 hours																								
6/7/2020	0:00 hours																								
6/8/2020	0:00 hours																								
6/9/2020	0:00 hours																								
6/10/2020	0:00 hours																								
6/11/2020	0:00 hours																								

Generated using [DayPilot UI Builder](#).


# Aplicação Timesheet

## Acessando o Timesheet

[illegible]

# Aplicação Timesheet

## Selecione o funcionário

Employee: Pedro 

**DEMO**

Day

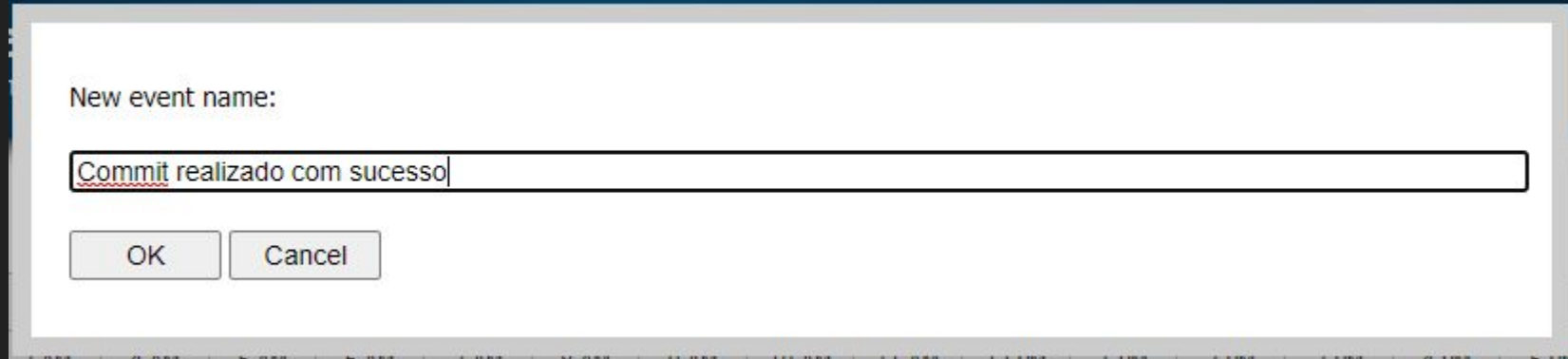
6/1/2020 1:00 hour

Clique em algum quadrado para cadastrar atividade

[illegible]

# Aplicação Timesheet

Adicionando evento



A screenshot of a software dialog box titled "New event name:". The dialog box has a white background and a thin blue border. Inside, there is a text input field containing the text "Commit realizado com sucesso". Below the input field, there are two buttons: "OK" and "Cancel". The "OK" button is on the left and the "Cancel" button is on the right. Both buttons have a light gray background and a thin black border.

New event name:

Commit realizado com sucesso

OK Cancel



# Aplicação Timesheet

## Visualizando o total de horas

[illegible]

**Obs: Para deletar a atividade somente no banco**

[illegible]

# Timesheet

O timesheet é uma forma de monitorar e avaliar as atividades já realizadas pelos funcionários.

Na imagem acima temos os seguintes campos a serem preenchido pelos programadores:

Funcionários: Nomes do funcionários para escolher

Data: O dia em que foi feito a atividade

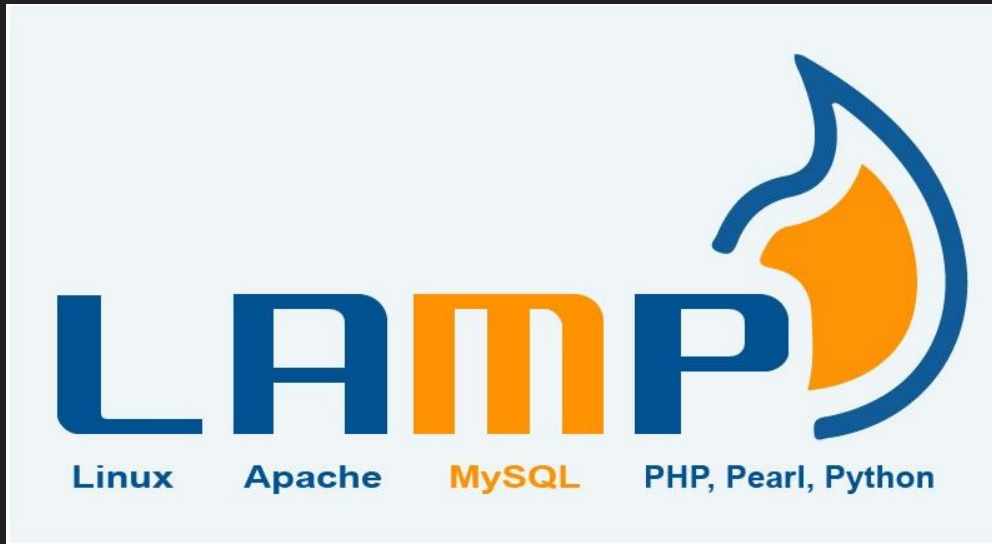
Horas: A hora em que foi feito a atividade

Descrição: A descrição da atividade

## Capítulo 2 - Tecnologías

# Projeto virtualizado (LAMP)

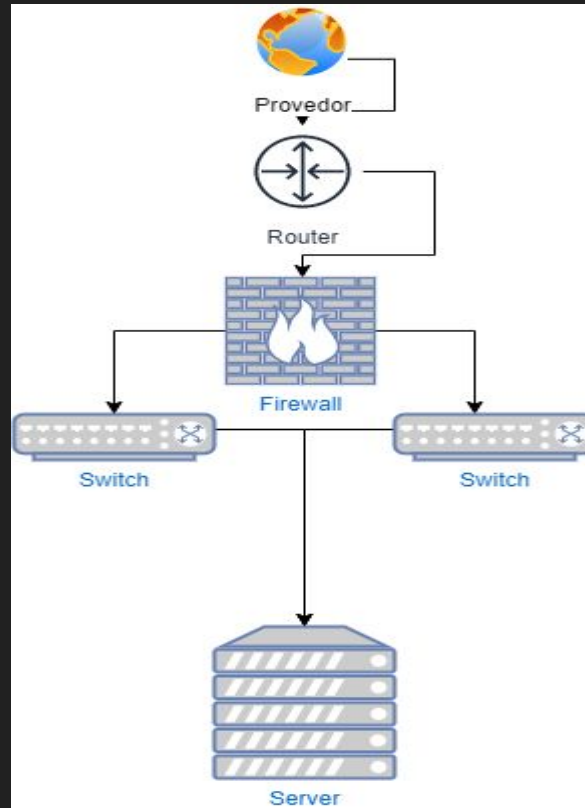
LAMP é uma combinação de softwares livres e de código aberto. O acrônimo LAMP refere-se às primeiras letras de: Linux, Apache, MariaDB ou MySQL e PHP ou Python, componentes principais para viabilizar o desenvolvimento de aplicações web de propósito geral, de alta disponibilidade e de alto desempenho.



# Pré-requisito

- 1 Servidor PowerEdge T140
  - Processador: Intel® Celeron G4930 3.2GHz, 2M cache, 2C/2T, no turbo
  - Memória: 8GB UDIMM DDR4 de 2666 MT/s (2x)
  - Armazenamento: 1TB SATA cabeado, 6 Gbps, 7200 RPM e 3,5"
- Internet
- 2 Switch (8 portas)
- Licença ESXI

# Diagrama físico

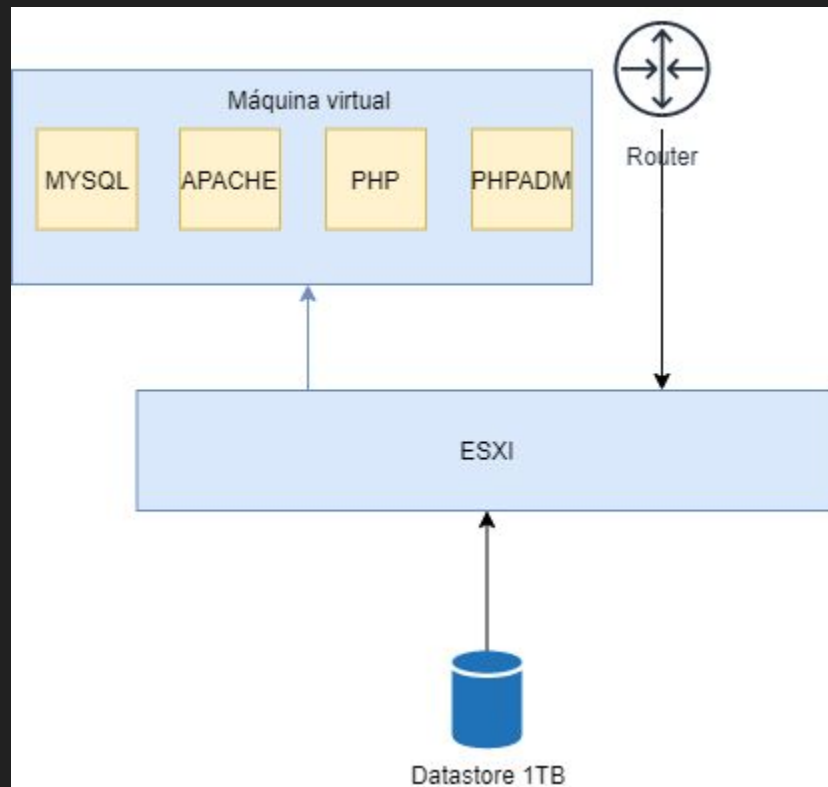


# Diagrama físico

- O recurso de firewall do roteador para criar mais uma barreira para proteger a conexão
- 2 Switchs para caso um falhe o outro não deixe o serviço cair



# Diagrama virtualizado



# Diagrama virtualizado

- O servidor já possui um HD de 1 Tera, onde irão ficar armazenado tudo (SO, isos, BD e etc).
- Hypervisor: ESXI
- No ubuntu 18.04 terá:
  - Banco de dados - MySql
  - Servidor Web - Apache Server
  - Phpmyadmin
  - Php
  - Ansible - automatização

# Gerenciamento do código-fonte

O gerenciamento do código-fonte será feita no github, onde os desenvolvedores irão atualizar código. A cada nova atualização de código, o desenvolvedor deverá preencher um documento sinalizando o que foi alterado (O código só poderá ser jogado para o servidor depois de uma revisão e autorização de um supervisor).

Tendo o código atualizado no repositório central, basta o próximo desenvolvedor baixar para a sua máquina e realizar as suas atualizações.

Git Hub: <https://github.com/J-Pedro/assessment-virtualizacao>

# Arquitetura da solução de virtualização

## SDDC (Software-Defined Data Center) - Local

Nesse modelo de Data Center, toda a infraestrutura é virtualizada e entregue como serviço. Ou seja, por meio de um software específico é possível separar a infraestrutura de hardwares em blocos para uso próprio ou fornecimento dos recursos e capacidades para diversos usuários simultaneamente.

### Por que data center físico?

Foi decidido por optar por um data center físico, pois a empresa já possui servidores virtualizados com aplicações rodando. E essa aplicação não irá consumir muito recurso, portanto não seria necessário criar essa aplicação na nuvem.

# Comparação da solução de virtualização

Nuvem pública (AWS) - As nuvens públicas são a maneira mais comum de implantar a computação em nuvem. Os recursos de nuvem (como servidores e armazenamento) pertencem a um provedor de serviço de nuvem terceirizado, são operados por ele e entregues pela Internet. O AWS é um exemplo de nuvem pública. Ou seja, caso fosse decidido optar pela AWS teríamos utilizado as soluções:

EC2 - Criação de instância

RDS - Banco de dados nativo da amazon (opcional)

VPC - Uma nuvem virtual privada para gerenciarmos nossa rede

# Capítulo 3 - Implementação

# Passo a passo

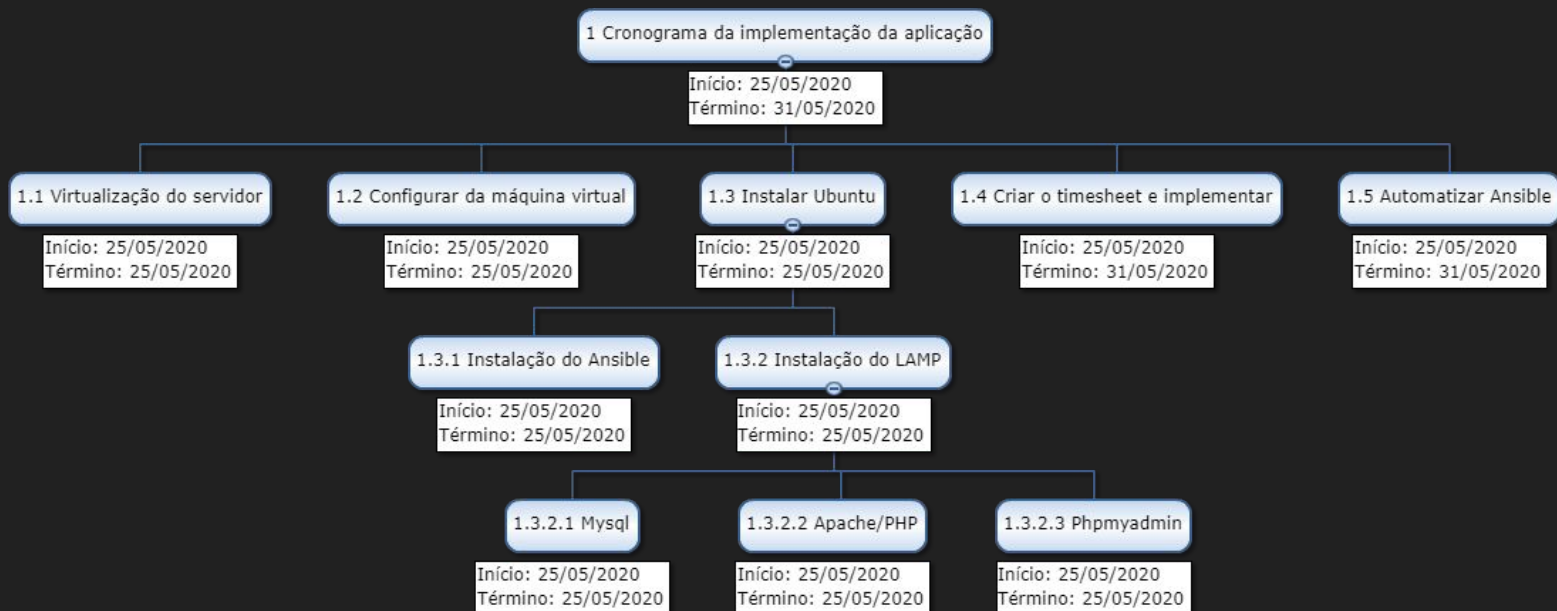
- Virtualização do servidor (VMware workstation - Virtualização é uma técnica que permite que uma aplicação de um sistema operacional (ou mesmo um sistema operacional inteiro) rode dentro de outro sistema. Nesse caso vamos instalar o ubuntu dentro do Esxi
- Configurar da máquina virtual - Antes de realizar a instalação do Ubuntu vamos informar as configurações da máquina virtual (Vcpu, Vram, quantidade de disco).
- Instalar Ubuntu - Nessa iremos realizar a instalação do SO Ubuntu
- Instalação do Lamp ( Linux + Apache + mysql + php ) - A definição de LAMP já foi definida antes, mas lembrando LAMP é uma combinação de softwares livres e de código aberto. O acrônimo LAMP refere-se as primeiras letras de: Linux, Apache, MariaDB ou MySQL e PHP ou Python.

# Passo a passo

- Instalação do Ansible - O Ansible é uma ferramenta que realiza a automação de várias tarefas como: Instalação e atualização de pacotes, poder encapsular a aplicação e poder rodar em outros servidores e etc.
- Criar o timesheet e implementar - E por fim teremos o timesheet que é a nossa aplicação e rodará no sistema.



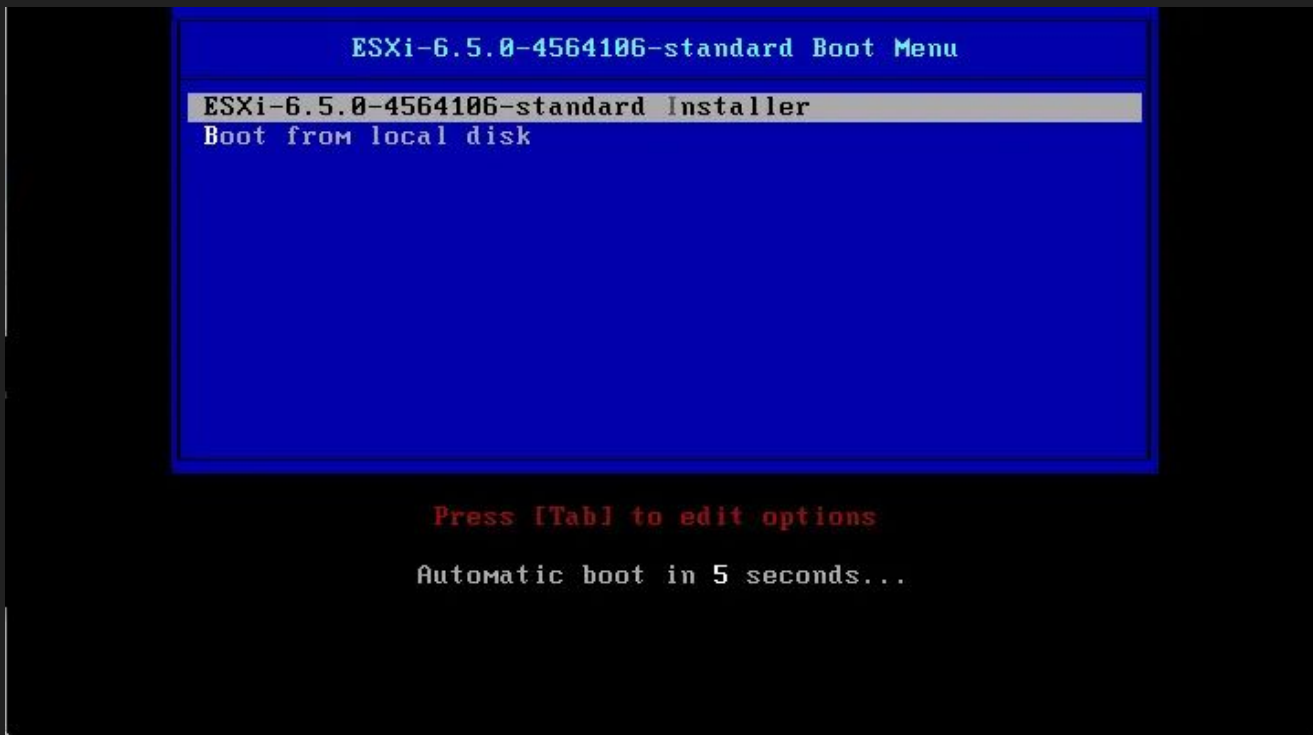
# Cronograma



# Processo de implementação

## Instalação ESXi

Inicie o servidor ou VM pelo CD e escolha a opção de boot pela ISO do ESXi.



# Processo de implementação

## Instalação ESXi

Espere carregar todos os módulos.



# Processo de implementação

## Instalação ESXI

Na tela de boas vindas tecle ENTER para continuar.



# Processo de implementação

## Instalação ESXI

Aceite a licença de uso teclando F11.



# Processo de implementação

## Instalação ESXI

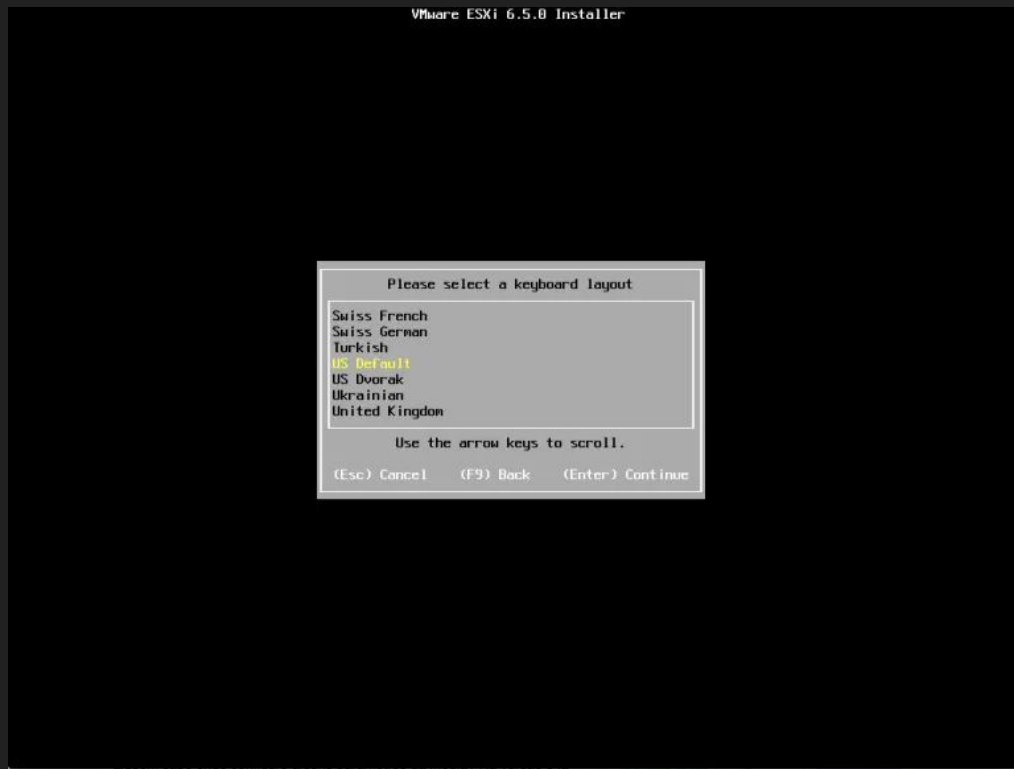
Selecione o disco onde vai ser instalado o sistema e tecele ENTER.



# Processo de implementação

## Instalação ESXI

Selecione o layout do teclado que deseja usar e tecle ENTER.



# Processo de implementação

## Instalação ESXI

Insira a senha que deseja usar e tecle ENTER



A screenshot of the ESXi root password setup screen. The window has a title bar that says "Enter a root password". Inside the window, there are two lines of text: "Root password: \*\*\*\*\*" and "Confirm password: \*\*\*\*\*". Below these, it says "Passwords match.". At the bottom of the window, there are three options: "(Esc) Cancel", "(F9) Back", and "(Enter) Continue".



# Processo de implementação

## Instalação ESXi

Tecle F11 para começar a instalar o ESXi



# Processo de implementação

## Instalação ESXi

O ESXi começa a ser instalado



# Processo de implementação

## Instalação ESXi

Após concluir a instalação tecle ENTER para reiniciar.



# Processo de implementação

## Instalação ESXi

Após o sistema reiniciar, ele irá pegar um endereço IP dinamicamente caso a sua rede possua um servidor DHCP. Caso não tenha um servidor DHCP nós podemos configurar o endereço IP manualmente

```
VMware ESXi 6.5.0 (VMKernel Release Build 4564106)  
  
VMware, Inc. VMware Virtual Platform  
  
2 x Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz  
8 GiB Memory
```

```
Download tools to manage this host from:  
http://192.168.1.220/ (DHCP)  
http://\[fd79:e45b:30e3::9091\] (DHCP)  
http://\[fe80::20c:29ff:fe16:67781\] (STATIC)
```

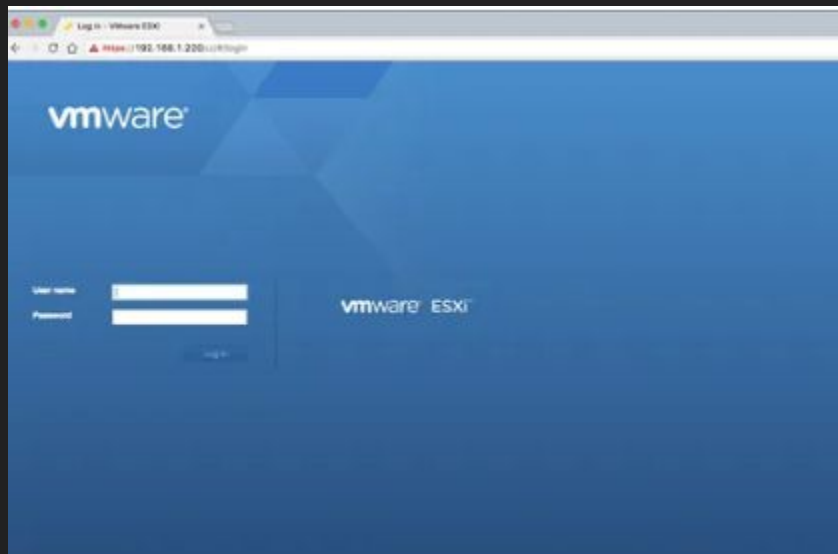
# Processo de implementação

## Instalação ESXi

Depois que tiver com o endereço IP, podemos acessar o nosso ESXi direto pelo navegador.

Para acessar o servidor via navegador, insira o seguinte endereço, troque pelo seu IP.

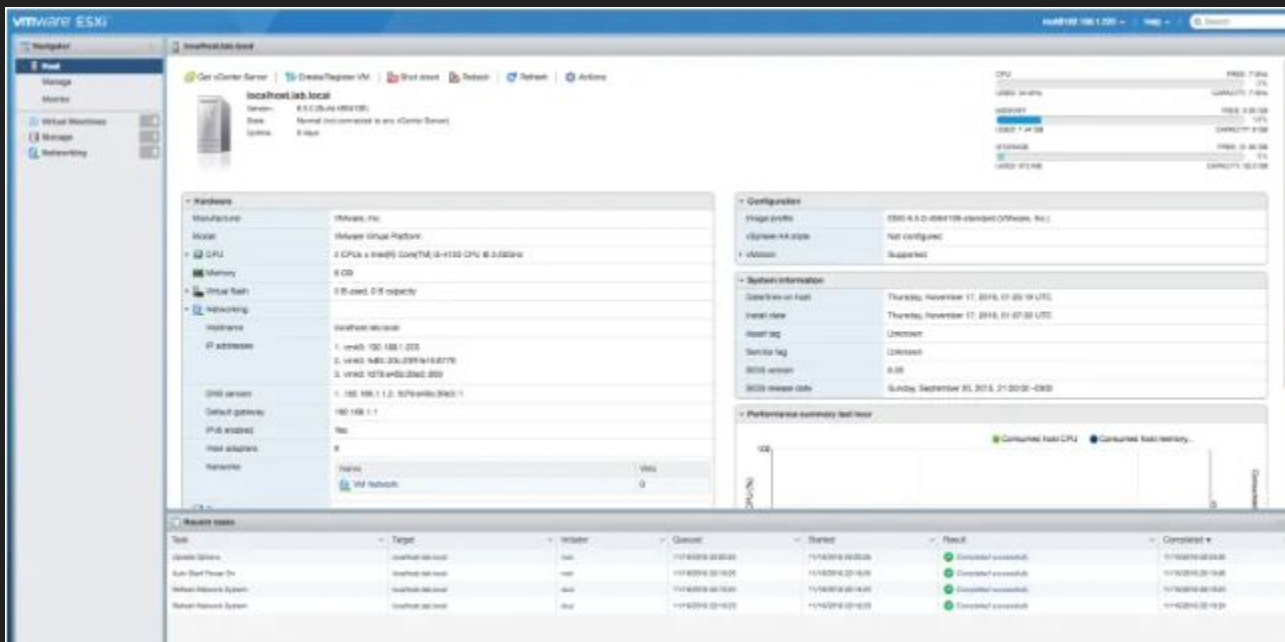
`http://IP_DO_SERVIDOR/ui`



# Processo de implementação

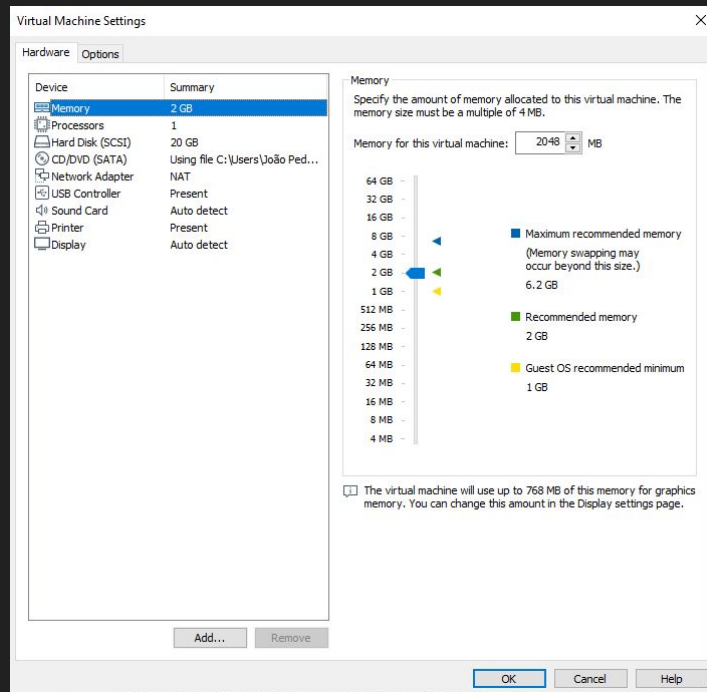
## Instalação ESXi

Pronto, servidor pronto para usar



# Processo de implementação

Como não tenho acesso ao ESXI, segue abaixo as configurações da VMware Workstation e faça no ESXI



# Processo de implementação

Sudo apt-get update para realizar atualização dos pacotes

```
analista@ubuntu: ~  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Thu Jun  4 21:42:45 UTC 2020  
  
System load:  0.0          Processes:            151  
Usage of /:   20.4% of 19.56GB  Users logged in:     1  
Memory usage: 12%          IP address for ens33: 192.168.119.138  
Swap usage:   0%  
  
* MicroK8s gets a native Windows installer and command-line integration.  
  
    https://ubuntu.com/blog/microk8s-installers-windows-and-macos  
  
44 packages can be updated.  
0 updates are security updates.  
  
Last login: Thu Jun  4 21:30:56 2020  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
analista@ubuntu:~$ sudo apt-get update
```



# Processo de implementação

Ssh key gen para criar um chave ssh (O ssh foi instalado na instalação do próprio ubuntu 18.04)

```
analista@ubuntu: ~  
You can enable repos with yum-config-manager --enable <repo>  
analista@ubuntu:~$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/analista/.ssh/id_rsa):  
Created directory '/home/analista/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/analista/.ssh/id_rsa.  
Your public key has been saved in /home/analista/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:LZjvpAxlJCNY3q09TbXb+WBXlW6y9SM4Cu/TsjnYmDg analista@ubuntu  
The key's randomart image is:  
+---[RSA 2048]-----+  
| . . . o |  
| + . . . . o |  
| . o + o . . . |  
| . O + o . = |  
| = S o o == o |  
| . + o o o . = |  
| . . X o . . o |  
| E B B . . |  
| + o + = |  
+---[SHA256]-----+  
analista@ubuntu:~$
```

# Processo de implementação

Precisamos sinalizar para nosso servidor SSH que a chave que acabamos de gerar é autorizada para logins remotos.

```
analista@ubuntu: ~  
|   E B B.. |  
|   + o+=   |  
+-----[SHA256]-----+  
analista@ubuntu:~$ ls -la  
total 36  
drwxr-xr-x 6 analista analista 4096 Jun  4 21:54 .  
drwxr-xr-x 3 root      root    4096 Jun  2 02:44 ..  
-rw-r--r-- 1 analista analista 220 Apr  4 2018 .bash_logout  
-rw-r--r-- 1 analista analista 3771 Apr  4 2018 .bashrc  
drwx----- 2 analista analista 4096 Jun  4 21:30 .cache  
drwx----- 3 analista analista 4096 Jun  4 21:30 .gnupg  
-rw-r--r-- 1 analista analista 807 Apr  4 2018 .profile  
drwxr-xr-x 2 analista analista 4096 Jun  4 21:50 .rpmdb  
drwx----- 2 analista analista 4096 Jun  4 21:54 .ssh  
-rw-r--r-- 1 analista analista  0 Jun  4 21:48 .sudo_as_admin_successful  
analista@ubuntu:~$ cd .ssh  
analista@ubuntu:~/.ssh$ ls -la  
total 16  
drwx----- 2 analista analista 4096 Jun  4 21:54 .  
drwxr-xr-x 6 analista analista 4096 Jun  4 21:54 ..  
-rw----- 1 analista analista 1679 Jun  4 21:54 id_rsa  
-rw-r--r-- 1 analista analista 397 Jun  4 21:54 id_rsa.pub  
analista@ubuntu:~/.ssh$ cd ..  
analista@ubuntu:~$ touch ~/.ssh/authorized_keys
```

# Processo de implementação

Agora colocamos o id\_rsa.pub dentro do arquivo authorized\_keys

```
analista@ubuntu: ~  
+----[SHA256]-----+  
analista@ubuntu:~$ ls -la  
total 36  
drwxr-xr-x 6 analista analista 4096 Jun  4 21:54 .  
drwxr-xr-x 3 root      root      4096 Jun  2 02:44 ..  
-rw-r--r-- 1 analista analista  220 Apr  4 2018 .bash_logout  
-rw-r--r-- 1 analista analista 3771 Apr  4 2018 .bashrc  
drwx----- 2 analista analista 4096 Jun  4 21:30 .cache  
drwx----- 3 analista analista 4096 Jun  4 21:30 .gnupg  
-rw-r--r-- 1 analista analista  807 Apr  4 2018 .profile  
drwxr-xr-x 2 analista analista 4096 Jun  4 21:50 .rpmdb  
drwx----- 2 analista analista 4096 Jun  4 21:54 .ssh  
-rw-r--r-- 1 analista analista    0 Jun  4 21:48 .sudo_as_admin_successful  
analista@ubuntu:~$ cd .ssh  
analista@ubuntu:~/.ssh$ ls -la  
total 16  
drwx----- 2 analista analista 4096 Jun  4 21:54 .  
drwxr-xr-x 6 analista analista 4096 Jun  4 21:54 ..  
-rw----- 1 analista analista 1679 Jun  4 21:54 id_rsa  
-rw-r--r-- 1 analista analista  397 Jun  4 21:54 id_rsa.pub  
analista@ubuntu:~/.ssh$ cd ..  
analista@ubuntu:~$ touch ~/.ssh/authorized_keys  
analista@ubuntu:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
analista@ubuntu:~$
```

# Processo de implementação

Agora vamos instalar pacotes necessários para a instalação do ansible

```
analista@ubuntu: ~  
  
https://ubuntu.com/blog/microk8s-installers-windows-and-macos  
  
44 packages can be updated.  
0 updates are security updates.  
  
Last login: Thu Jun  4 21:42:46 2020 from 192.168.119.1  
analista@ubuntu:~$ exit  
logout  
Connection to localhost closed.  
analista@ubuntu:~$ sudo apt install software-properties-common  
[sudo] password for analista:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  python3-software-properties  
The following packages will be upgraded:  
  python3-software-properties software-properties-common  
2 upgraded, 0 newly installed, 0 to remove and 43 not upgraded.  
Need to get 33.8 kB of archives.  
After this operation, 0 B of additional disk space will be used.  
Do you want to continue? [Y/n]
```

# Processo de implementação

## Adicionando repositório ansible

```
analista@ubuntu:~$ sudo apt-add-repository --yes --update ppa:ansible/ansible
Hit:1 http://br.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://br.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://br.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://br.archive.ubuntu.com/ubuntu bionic-security InRelease
Get:5 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease [15.9 kB]
Get:6 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic/main amd64 Packages
[692 B]
Get:7 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic/main Translation-en
[472 B]
Fetched 17.0 kB in 2s (10.4 kB/s)
```

# Processo de implementação

## Instalando ansible

```
analista@ubuntu: ~  
Fetches 17.0 kB in 2s (10.4 kB/s)  
Reading package lists... Done  
analista@ubuntu:~$ sudo apt install ansible  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  python-asn1crypto python-cffi-backend python-crypto python-cryptography  
  python-enum34 python-httplib2 python-idna python-ipaddress python-jinja2  
  python-markupsafe python-paramiko python-pkg-resources python-pyasn1  
  python-setuptools python-six python-yaml sshpass  
Suggested packages:  
  python-crypto-doc python-cryptography-doc python-cryptography-vectors  
  python-enum34-doc python-jinja2-doc python-gssapi python-setuptools-doc  
The following NEW packages will be installed:  
  ansible python-asn1crypto python-cffi-backend python-crypto  
  python-cryptography python-enum34 python-httplib2 python-idna  
  python-ipaddress python-jinja2 python-markupsafe python-paramiko  
  python-pkg-resources python-pyasn1 python-setuptools python-six python-yaml  
  sshpass  
0 upgraded, 18 newly installed, 0 to remove and 43 not upgraded.  
Need to get 7,370 kB of archives.  
After this operation, 66.5 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

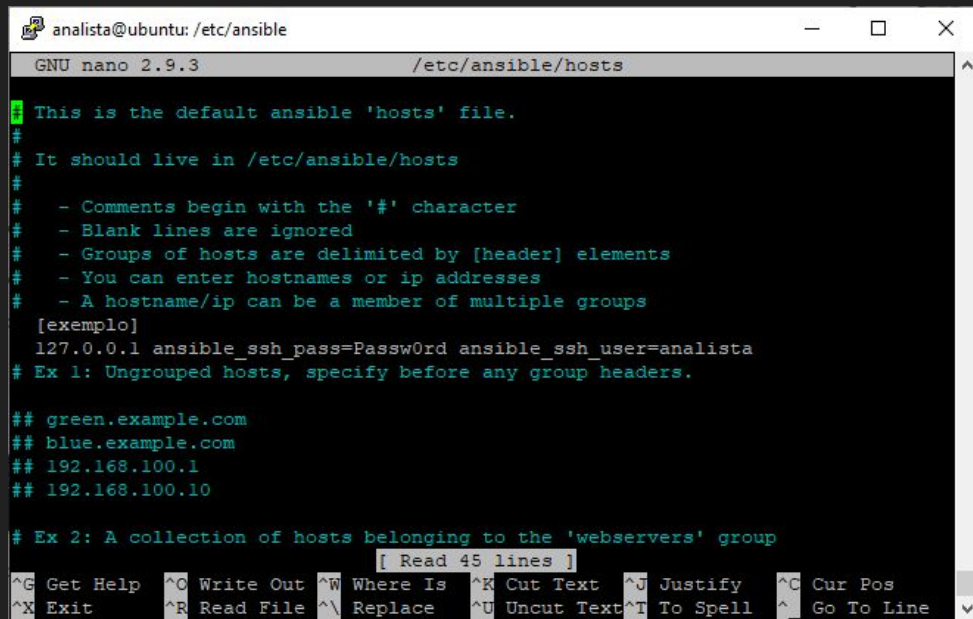
# Processo de implementação

## Configurando hosts do ansible

```
analista@ubuntu: /etc/ansible
drwxr-xr-x 3 root root      4096 Feb  3 18:24 update-manager
drwxr-xr-x 2 root root      4096 Feb  3 18:24 update-motd.d
drwxr-xr-x 2 root root      4096 May 24 2019 update-notifier
drwxr-xr-x 2 root root      4096 Jun  2 02:27 vim
drwxr-xr-x 4 root root      4096 Feb  3 18:24 vmware-tools
lrwxrwxrwx 1 root root          23 Feb  3 18:22 vtrgb -> /etc/alternatives/vtrgb
-rw-r--r-- 1 root root      4942 Apr  8 2019 wgetrc
drwxr-xr-x 4 root root      4096 Feb  3 18:23 X11
drwxr-xr-x 4 root root      4096 Feb  3 18:22 xdg
drwxr-xr-x 5 root root      4096 Jun  4 21:49 yum
-rw-r--r-- 1 root root          477 Mar 16 2018 zsh_command_not_found
analista@ubuntu:/etc$ cd /ansible
-bash: cd: /ansible: No such file or directory
analista@ubuntu:/etc$ cd /ansible/hosts
-bash: cd: /ansible/hosts: No such file or directory
analista@ubuntu:/etc$ cd /etc/ansible
analista@ubuntu:/etc/ansible$ ls -la
total 36
drwxr-xr-x  3 root root  4096 Jun  4 22:26 .
drwxr-xr-x 95 root root  4096 Jun  4 22:26 ..
-rw-r--r--  1 root root 19985 May 12 20:30 ansible.cfg
-rw-r--r--  1 root root  1016 May 12 20:30 hosts
drwxr-xr-x  2 root root  4096 May 12 20:31 roles
analista@ubuntu:/etc/ansible$ sudo nano /etc/ansible/hosts
```

# Processo de implementação

Agora vamos informar o nosso servidor local



The screenshot shows a terminal window titled 'analista@ubuntu: /etc/ansible'. Inside, the GNU nano 2.9.3 editor is open to the file /etc/ansible/hosts. The file contains default configuration and example entries for hosts. The cursor is positioned at the end of the line '# Ex 2: A collection of hosts belonging to the 'webservers' group'. The bottom of the window displays a status bar with various keyboard shortcuts.

```
analista@ubuntu: /etc/ansible
GNU nano 2.9.3 /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# [exemplo]
# 127.0.0.1 ansible_ssh_pass=Passw0rd ansible_ssh_user=analista
# Ex 1: Ungrouped hosts, specify before any group headers.
#
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
#
# Ex 2: A collection of hosts belonging to the 'webservers' group
#
[ Read 45 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```



# Processo de implementação

Agora vamos testar se o inventário está funcionando

```
analista@ubuntu:/etc/ansible$ sudo ansible exemplo -m ping -u analista
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 127.0.0.1 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A future Ansible release will default to using the
discovered platform python for this host. See https://docs.ansible.com/ansible/
2.9/reference_appendices/interpreter_discovery.html for more information. This
feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
127.0.0.1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
```

# Processo de implementação

Agora vamos instalar o apache

```
analista@ubuntu:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
0 upgraded, 9 newly installed, 0 to remove and 43 not upgraded.
Need to get 1,621 kB of archives.
After this operation, 6,562 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

# Processo de implementação

Agora vamos instalar o Mysql

```
analista@ubuntu: ~  
analista@ubuntu:~$ sudo apt install -y mysql-server mysql-client php-mysql  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libaiol libcgi-fast-perl libcgi-pm-perl libencode-locale-perl  
  libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl  
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl  
  liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7  
  mysql-client-core-5.7 mysql-common mysql-server-5.7 mysql-server-core-5.7  
  php-common php7.2-common php7.2-mysql  
Suggested packages:  
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinycal  
The following NEW packages will be installed:  
  libaiol libcgi-fast-perl libcgi-pm-perl libencode-locale-perl  
  libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl  
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl  
  liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client  
  mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server  
  mysql-server-5.7 mysql-server-core-5.7 php-common php-mysql php7.2-common  
  php7.2-mysql  
0 upgraded, 26 newly installed, 0 to remove and 43 not upgraded.  
Need to get 20.7 MB of archives.  
After this operation, 164 MB of additional disk space will be used.
```

# Processo de implementação

Agora vamos instalar o PHP

```
analista@ubuntu:~$ sudo apt-get install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
php-mysql is already the newest version (1:7.2+60ubuntu1).
The following additional packages will be installed:
  libapache2-mod-php7.2 libsodium23 php7.2 php7.2-cli php7.2-json
  php7.2-opcache php7.2-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.2 libsodium23 php php7.2 php7.2-cli
  php7.2-json php7.2-opcache php7.2-readline
0 upgraded, 9 newly installed, 0 to remove and 43 not upgraded.
Need to get 3,113 kB of archives.
After this operation, 10.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

# Processo de implementação

Reiniciando o Apache

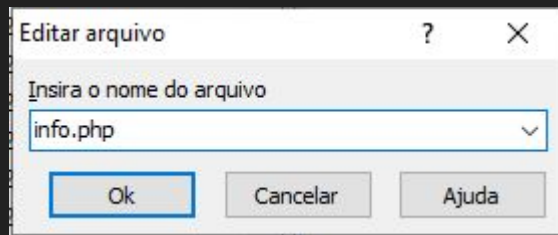
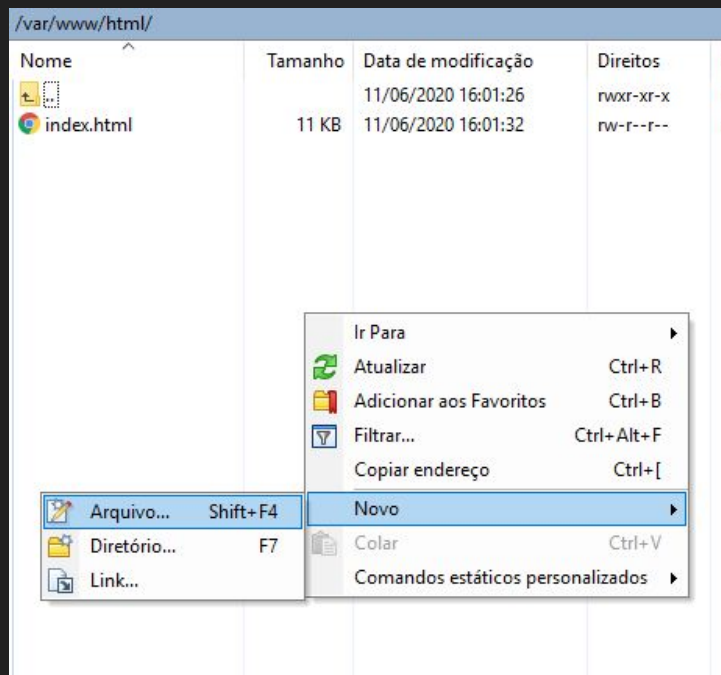
```
analista@ubuntu:~$ sudo service apache2 restart  
analista@ubuntu:~$
```

Dando permissão para adicionarmos arquivo na pasta do site

```
analista@ubuntu:~$ sudo chown -R $USER:root /var/www
```

# Processo de implementação

Criar um novo arquivo via WinScp



Insira essas linhas no arquivo



# Processo de implementação

No navegador acesse pelo “seuip/info.php” para ter informações sobre o php instalado

192.168.119.138/info.php

PHP Version 7.2.24-0ubuntu0.18.04.6



System	Linux ubuntu 4.15.0-101-generic #102-Ubuntu SMP Mon May 11 10:07:26 UTC 2020 x86_64
Build Date	May 26 2020 13:09:11
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqld.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini,

# Processo de implementação

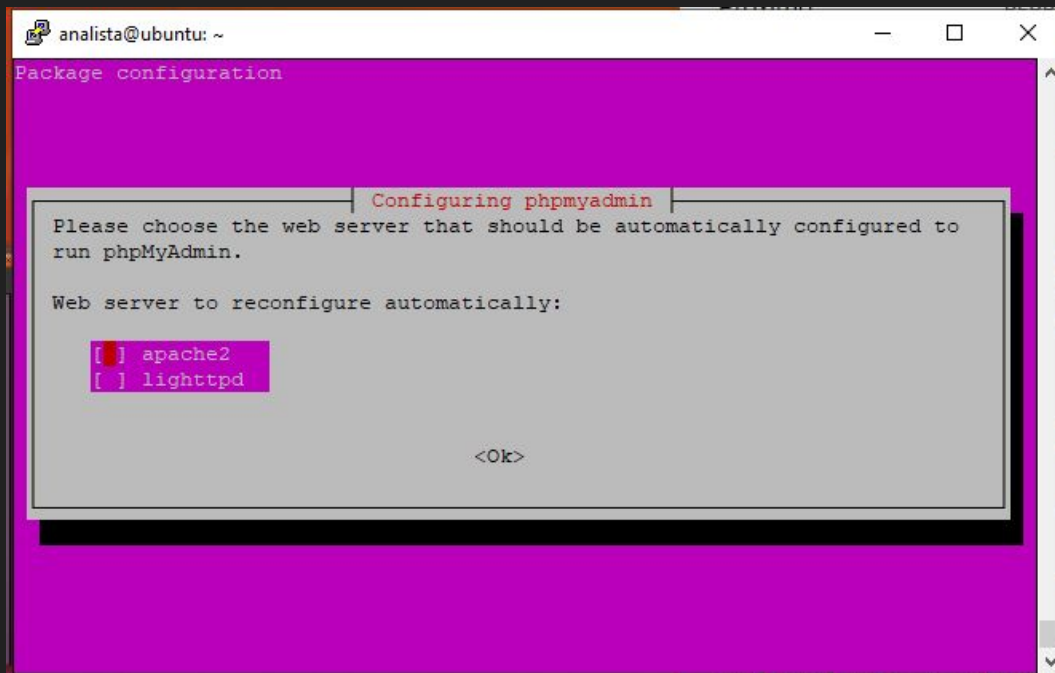
Agora vamos instalar o phpmyadmin

```
analista@ubuntu:~$ sudo apt-get install phpmyadmin
```



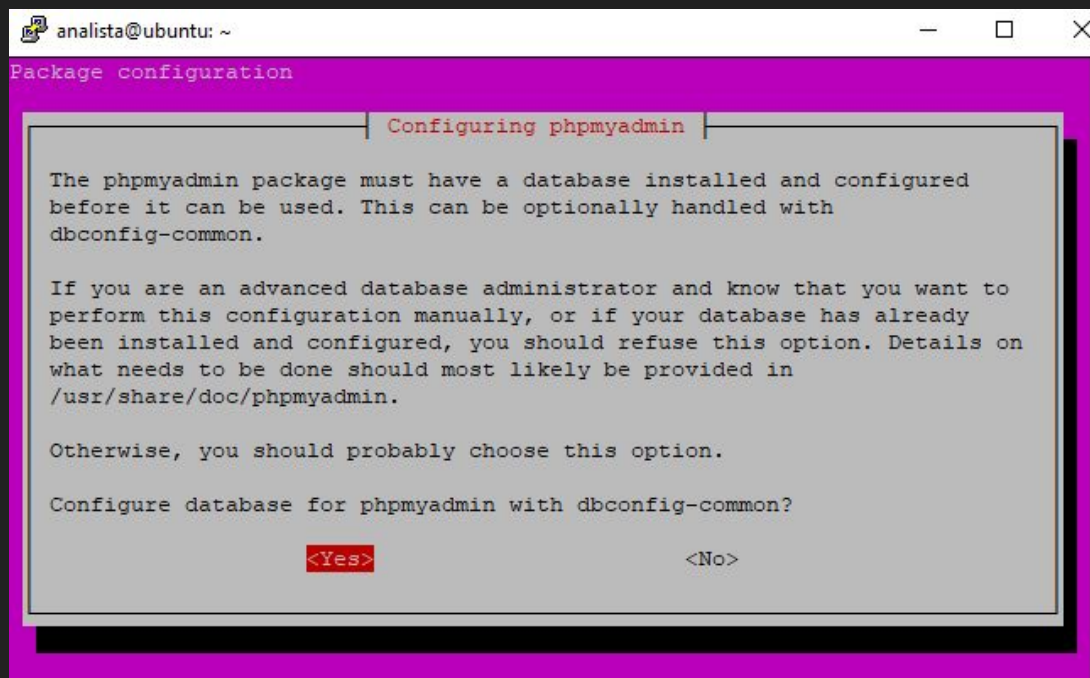
# Processo de implementação

Na instalação do phpmyadmin irá abrir essa tela de configuração, selecionamos o apache2



# Processo de implementação

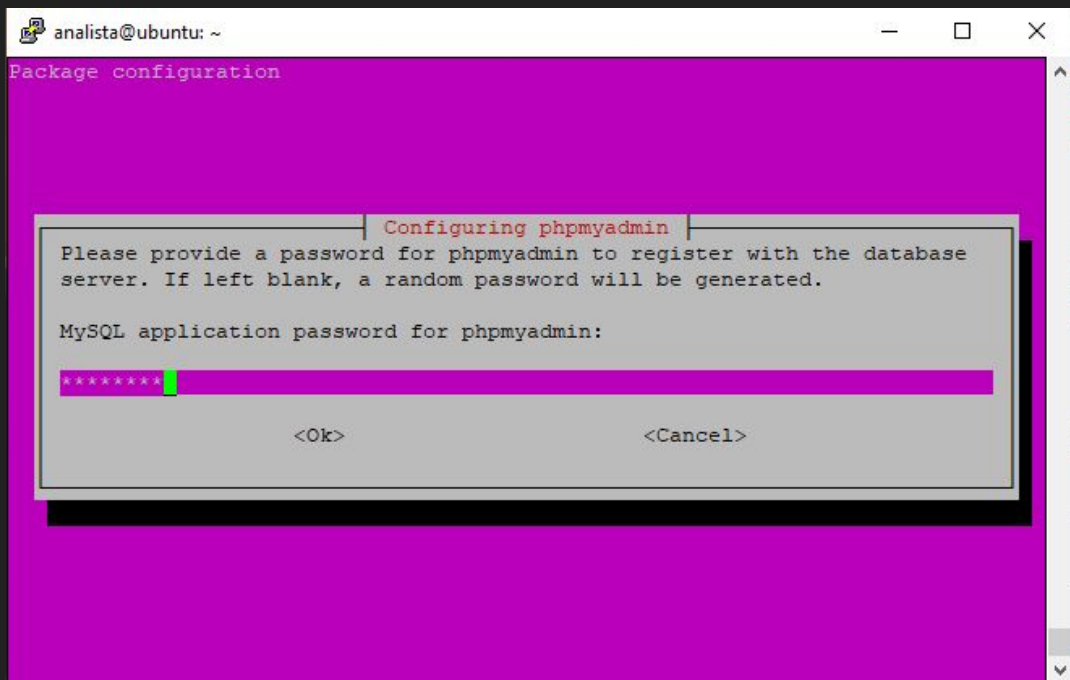
Configurar sobre o nosso mysql já criado, clicamos em “yes”

A terminal window titled 'analista@ubuntu: ~' with standard window controls. The prompt is 'Package configuration'. A box titled 'Configuring phpmyadmin' contains the following text: 'The phpmyadmin package must have a database installed and configured before it can be used. This can be optionally handled with dbconfig-common. If you are an advanced database administrator and know that you want to perform this configuration manually, or if your database has already been installed and configured, you should refuse this option. Details on what needs to be done should most likely be provided in /usr/share/doc/phpmyadmin. Otherwise, you should probably choose this option. Configure database for phpmyadmin with dbconfig-common?'. At the bottom of the box are two options: '<Yes>' (highlighted in red) and '<No>'.

```
analista@ubuntu: ~  
Package configuration  
Configuring phpmyadmin  
  
The phpmyadmin package must have a database installed and configured  
before it can be used. This can be optionally handled with  
dbconfig-common.  
  
If you are an advanced database administrator and know that you want to  
perform this configuration manually, or if your database has already  
been installed and configured, you should refuse this option. Details on  
what needs to be done should most likely be provided in  
/usr/share/doc/phpmyadmin.  
  
Otherwise, you should probably choose this option.  
  
Configure database for phpmyadmin with dbconfig-common?  
  
<Yes> <No>
```

# Processo de implementação

Senha para acessar como phpmyadmin



# Processo de implementação

Para acessarmos ao nosso phpmyadmin devemos dar permissão no apache

```
analista@ubuntu:~$ sudo nano /etc/apache2/apache2.conf
```

Inserimos esses comandos

```
#include phpmyadmin file  
Include /etc/phpmyadmin/apache.conf
```


Reiniciamos o serviço do apache

```
analista@ubuntu:~$ sudo service apache2 restart
```

# Processo de implementação

Agora temos acesso ao phpmyadmin

192.168.119.138/phpmyadmin/index.php



Bemvindo ao phpMyAdmin

Língua - *Language*

Português - Portuguese ▼

Entrada ⓘ

Utilizador :

Palavra-passe:

Executar

# Processo de implementação

Agora vamos definir uma senha ao usuário root no mysql

```
analista@ubuntu:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 32
Server version: 5.7.30-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> update user set authentication_string=PASSWORD("senha123") where user = '
root';
```

# Processo de implementação

Para conseguirmos acessar via phpmyadmin com o usuario root, devemos mudar uma configuração no banco de dados

```
mysql> select user, plugin from user;
```

user	plugin
root	auth_socket
mysql.session	mysql_native_password
mysql.sys	mysql_native_password
debian-sys-maint	mysql_native_password
phpmyadmin	mysql_native_password

```
5 rows in set (0.00 sec)
```

```
mysql> update user set plugin="mysql_native_password" where user = 'root';
```

```
Query OK, 1 row affected (0.00 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql>
```

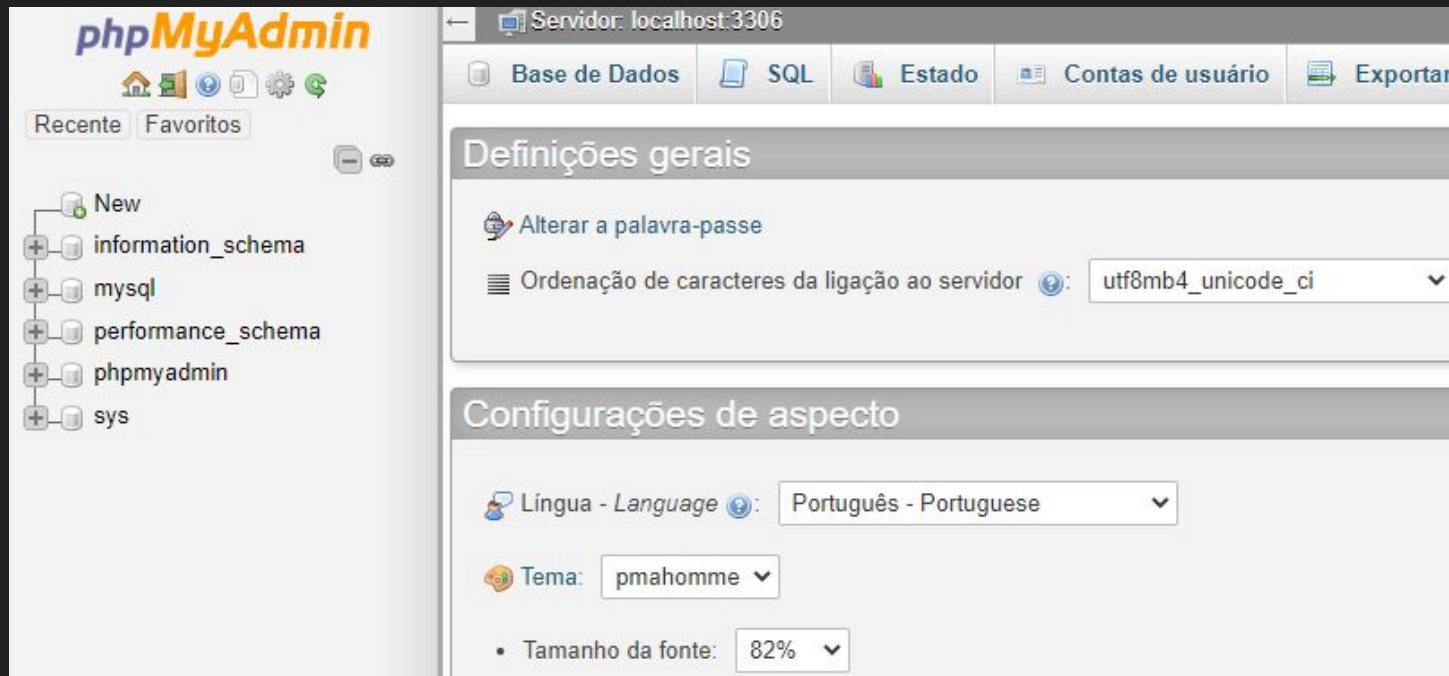
```
mysql> flush privileges;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql>
```

# Processo de implementação















Agora temos acesso ao phpmyadmin com o usuário root





# Processo de implementação

Agora na pasta do site mova os arquivos da aplicação

/var/www/html/				
Nome	Tamanho	Data de modificação	Direitos	Proprie...
		11/06/2020 16:01:26	rw-r--r--	analista
 js		11/06/2020 16:40:11	rw-rw-r--	analista
 license		11/06/2020 16:40:11	rw-rw-r--	analista
 readme		11/06/2020 16:40:11	rw-rw-r--	analista
 _db.php	1 KB	09/09/2019 13:26:54	rw-rw-r--	analista
 _db_mysql.php	2 KB	09/09/2019 13:26:54	rw-rw-r--	analista
 _db_sqlite.php	2 KB	09/09/2019 13:14:38	rw-rw-r--	analista
 backend_create.php	1 KB	19/06/2014 09:41:42	rw-rw-r--	analista
 backend_events.php	1 KB	09/09/2019 13:26:54	rw-rw-r--	analista
 backend_move.php	1 KB	19/06/2014 13:09:32	rw-rw-r--	analista
 backend_resize.php	1 KB	19/06/2014 09:41:42	rw-rw-r--	analista
 backend_resources.php	1 KB	09/09/2019 13:26:54	rw-rw-r--	analista
 backend_update.php	1 KB	19/06/2014 09:41:42	rw-rw-r--	analista
 index.html	6 KB	09/09/2019 14:15:06	rw-rw-r--	analista

# Processo de implementação

Configure o arquivo de conexão

```
<?php
$host = "127.0.0.1";
$port = 3306;
$username = "root";
$password = "senha123";
$database = "timesheet";
```

# Processo de implementação

Crie o banco de dados e tabelas

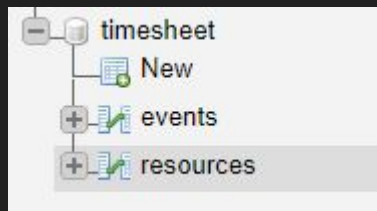




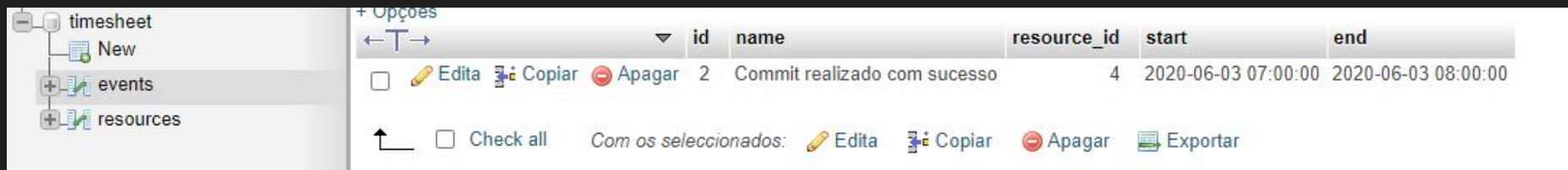


Tabela de funcionários

+ Opções					
				id	name
<input type="checkbox"/>		Edita	 Copiar	 Apagar	1 Pedro
<input type="checkbox"/>		Edita	 Copiar	 Apagar	2 Juan
<input type="checkbox"/>		Edita	 Copiar	 Apagar	3 Thelma
<input type="checkbox"/>		Edita	 Copiar	 Apagar	4 Bruno

# Processo de implementação

Gravado no banco com sucesso



+ Opções		id	name	resource_id	start	end
<input type="checkbox"/>	Editar  Copiar  Apagar	2	Commit realizado com sucesso	4	2020-06-03 07:00:00	2020-06-03 08:00:00

☐ Check all Com os seleccionados: Editar Copiar Apagar Exportar

# Processo de implementação

Agora vamos criar o ansible para automatizar, segue os primeiros comandos

```
analista@ubuntu:~$ mkdir timesheet-ansible  
analista@ubuntu:~$ cd timesheet-ansible  
analista@ubuntu:~/timesheet-ansible$ touch playbook.yml  
analista@ubuntu:~/timesheet-ansible$ touch hosts  
analista@ubuntu:~/timesheet-ansible$
```

# Processo de implementação

Configurando playbook.yml

```
--  
- hosts: timesheet  
  remote_user: analista  
  become: yes  
  roles:  
    - server  
    - php  
    - mysql
```

# Processo de implementação

## Configurando hosts

```
analista@ubuntu: ~/timesheet-ansible
GNU nano 2.9.3 hosts

[timesheet]
127.0.0.1 ansible_ssh_pass=Passw0rd ansible_ssh_user=analista
```

## Testando

```
analista@ubuntu:~/timesheet-ansible$ sudo ansible-playbook -i hosts playbook.yml -k -K
[sudo] password for analista:
SSH password:
BECOME password[defaults to SSH password]:

PLAY [timesheet] *****

TASK [Gathering Facts] *****
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 127.0.0.1 should use /usr/bin/python3,
but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future
Ansible release will default to using the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more
information. This feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
ok: [127.0.0.1]

PLAY RECAP *****
127.0.0.1 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescue
d=0 ignored=0

analista@ubuntu:~/timesheet-ansible$
```

# Processo de implementação

Criando roles que serão executadas no nosso ansible

```
analista@ubuntu:~/timesheet-ansible$ mkdir roles
analista@ubuntu:~/timesheet-ansible$ cd roles
analista@ubuntu:~/timesheet-ansible/roles$ ansible-galaxy init server
- Role server was created successfully
analista@ubuntu:~/timesheet-ansible/roles$ ansible-galaxy init php
- Role php was created successfully
analista@ubuntu:~/timesheet-ansible/roles$ ansible-galaxy init mysql
- Role mysql was created successfully
analista@ubuntu:~/timesheet-ansible/roles$
```



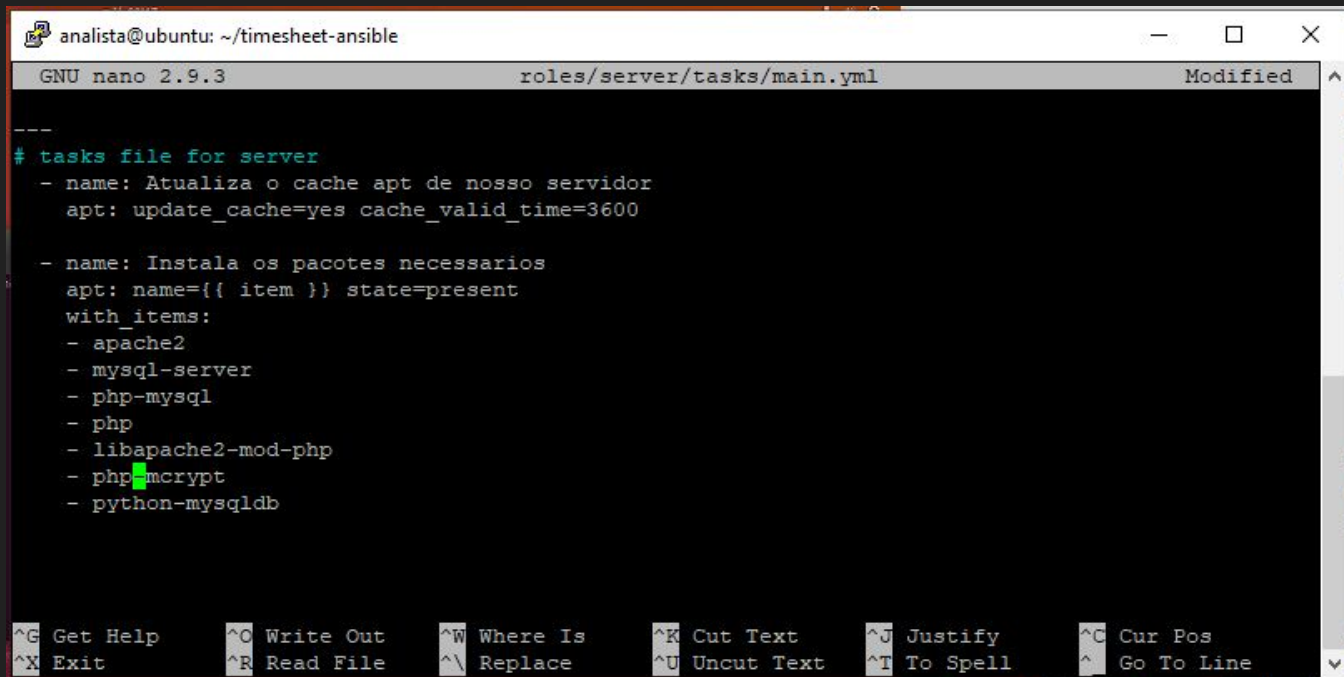
# Processo de implementação

Configurando role do server

```
analista@ubuntu:~/timesheet-ansible/roles$ nano /server/tasks/main.yml
```

# Processo de implementação

## Configurando task do server



The screenshot shows a terminal window titled 'analista@ubuntu: ~/timesheet-ansible'. The window contains the GNU nano 2.9.3 editor editing the file 'roles/server/tasks/main.yml'. The file content is as follows:

```
---
# tasks file for server
- name: Atualiza o cache apt de nosso servidor
  apt: update_cache=yes cache_valid_time=3600

- name: Instala os pacotes necessarios
  apt: name={{ item }} state=present
  with_items:
    - apache2
    - mysql-server
    - php-mysql
    - php
    - libapache2-mod-php
    - php-mcrypt
    - python-mysqldb
```

The bottom of the window displays a status bar with various keyboard shortcuts for nano editor operations:

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell	^_ Go To Line

# Processo de implementação

## Configurando task do php

```
analista@ubuntu: ~/timesheet-ansible
GNU nano 2.9.3 roles/p
--
# tasks file for php

- name: Instala as extensoes PHP necessarias
  apt: name={{ item }} state=present
  with_items:
    - php7.2-gd
```

# Processo de implementação

Configurando defaults do mysql para passar algumas variáveis

```
analista@ubuntu: ~/timesheet-ansible
GNU nano 2.9.3 roles/mysql/defaults/main.yml

---
# defaults file for mysql

# nome do banco de dados da aplicacao
tm_mysql_db: timesheet
# login a ser criado para a aplicacao
tm_mysql_user: root
# senha a ser atribuida para o login da aplicacao
tm_mysql_password: senha123
```

Configurando o tasks do mysql

```
analista@ubuntu: ~/timesheet-ansible
GNU nano 2.9.3 roles/mysql/tasks/main.yml

---
# tasks file for mysql

- name: Cria um usuario de banco para o WordPress
  mysql_user:
    name={{ tm_mysql_user }}
    password={{ tm_mysql_password }}
    priv='*.*:ALL'
```

# Processo de implementação

## Rodando playbook.yml

```
analista@ubuntu: ~/timesheet-ansible
analista@ubuntu:~/timesheet-ansible$ sudo ansible-playbook -i hosts playbook.yml -k -K
[sudo] password for analista:
SSH password:
BECOME password[defaults to SSH password]:

PLAY [timesheet] *****

TASK [Gathering Facts] *****
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 127.0.0.1 should use /usr/bin/python3, but is using
/usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will
default to using the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
ok: [127.0.0.1]

TASK [server : Atualiza o cache apt de nosso servidor] *****
ok: [127.0.0.1]

TASK [server : Instala os pacotes necessarios] *****
[DEPRECATION WARNING]: Invoking "apt" only once while using a loop via squash actions is deprecated. Instead
of using a loop to supply multiple items and specifying 'name: "{{ item }}"', please use 'name: ['apache2',
'mysql-server', 'php-mysql', 'php', 'libapache2-mod-php', 'python-mysqldb']' and remove the loop. This
feature will be removed in version 2.11. Deprecation warnings can be disabled by setting
```

# Processo de implementação

Pela primeira vez conseguimos mudar a senha do root no mysql

```
TASK [mysql : Cria um usuario de banco para o WordPress] *****
[WARNING]: Module did not set no_log for update_password
changed: [127.0.0.1]

PLAY RECAP *****
127.0.0.1          : ok=5    changed=3    unreachable=0    failed=0    skipped=0
red=0
```

# Processo de implementação

Após esse `playbook.yml`, basta você mover os arquivos da aplicação para o site e criar as tabelas no `mysql` para rodar a aplicação novamente

[illegible]

# Processo de implementação

Na segunda ele já reclama porque estamos tentando conectar ao mysql com o root sem uma senha

```
TASK [mysql : Cria um usuario de banco para o WordPress] *****
[WARNING]: Module did not set no_log for update_password
fatal: [127.0.0.1]: FAILED! => {"changed": false, "msg": "unable to connect to database, check login_user and
login_password are correct or /root/.my.cnf has the credentials. Exception message: (1045, \"Access denied f
or user 'root'@'localhost' (using password: NO)\")"}

PLAY RECAP *****
127.0.0.1      : ok=4    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    igno
red=0
```



# Processo de implementação

Acessando o mysql com a senha criada

```
analista@ubuntu:~/timesheet-ansible$ sudo mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.30-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
```

# Comandos para inserir no Mysql

- `exec("CREATE DATABASE IF NOT EXISTS timesheet");`
- `exec("use timesheet");`
- `exec("CREATE TABLE IF NOT EXISTS resources (  
id INTEGER PRIMARY KEY AUTO_INCREMENT,  
name VARCHAR(200))")`
- `exec("CREATE TABLE IF NOT EXISTS events (  
id INTEGER PRIMARY KEY AUTO_INCREMENT,  
name TEXT,  
resource_id INTEGER,  
start DATETIME,  
end DATETIME)")`;
- `exec "INSERT INTO resources (name) VALUES ("nome do funcionário)";`

# Processo de implementação

Agora vamos upar no github o nosso ansible, criamos um novo repositório


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


---

Owner

Repository name \*

 J-Pedro ▾


/

assessment-virtualizacao 


Great repository names are short and memorable. Need inspiration? How about [fictional-telegram](#)?

Description (optional)

---

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.


---

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

---

Create repository

# Processo de implementação

## Configurando o git no linux

```
analista@ubuntu:~/timesheet-ansible$ git --version
git version 2.17.1
analista@ubuntu:~/timesheet-ansible$ git config --global user.name J-Pedro
analista@ubuntu:~/timesheet-ansible$ git config --global user.name j.pedroo2000@hotmail.com
analista@ubuntu:~/timesheet-ansible$
```

# Processo de implementação

Dando um “git init”

```
analista@ubuntu:~/timesheet-ansible$ git init  
Initialized empty Git repository in /home/analista/timesheet-ansible/.git/
```

Dando um “git add” para adicionarmos todo o arquivo do ansible

```
analista@ubuntu:~/timesheet-ansible$ git add .
```

# Processo de implementação

Agora vamos realizar o nosso primeiro commit

```
analista@ubuntu:~/timesheet-ansible$ git commit -m "commitando arquivos"
[master 45ff4db] commitando arquivos
31 files changed, 448 insertions(+), 1 deletion(-)
delete mode 100644 README.md
create mode 100644 README.rdoc
create mode 100644 hosts
create mode 100644 playbook.yml
create mode 100644 roles/mysql/.travis.yml
create mode 100644 roles/mysql/README.md
create mode 100644 roles/mysql/defaults/main.yml
create mode 100644 roles/mysql/handlers/main.yml
create mode 100644 roles/mysql/meta/main.yml
create mode 100644 roles/mysql/tasks/main.yml
create mode 100644 roles/mysql/tests/inventory
create mode 100644 roles/mysql/tests/test.yml
create mode 100644 roles/mysql/vars/main.yml
create mode 100644 roles/php/.travis.yml
create mode 100644 roles/php/README.md
create mode 100644 roles/php/defaults/main.yml
create mode 100644 roles/php/handlers/main.yml
create mode 100644 roles/php/meta/main.yml
create mode 100644 roles/php/tasks/main.yml
```

# Processo de implementação

Use os 2 comandos utilizado abaixo para subir os arquivos para o repositório

```
analista@ubuntu:~/timesheet-ansible$ git remote add origin https://github.com/J-Pedro/assessment-virtualizaca
o
analista@ubuntu:~/timesheet-ansible$ git push -u origin master
Username for 'https://github.com': J-Pedro
Password for 'https://J-Pedro@github.com':
Counting objects: 44, done.
Compressing objects: 100% (21/21), done.
Writing objects: 100% (44/44), 4.88 KiB | 554.00 KiB/s, done.
Total 44 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/J-Pedro/assessment-virtualizacao
 6ca82f0..45ff4db master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
analista@ubuntu:~/timesheet-ansible$
```

## Capítulo 4 - Resultados



# Considerações finais

Prazos - O prazo do projeto foi ultrapassado, pois no começo do projeto foi decidido utilizar docker, porém os desenvolvedores decidiram remover o docker e utilizar somente a instalação simples.

Recursos planejados - Os recursos planejados foram atendidos com sucesso

Funcionalidades previstas - Funcionando corretamente

Melhorias futuras - Criar um botão para que possa remover a atividade descrita, sem que seja necessário acessar o banco para isso e instalar o vcenter ao ESXI



“E lembre-se: você é seu próprio general. Então, tome agora a iniciativa, planeje e marche decidido para a vitória.” – Sun Tzu. (A arte da guerra)