

**Pró-Reitoria Acadêmica  
Curso de Ciência da Computação  
Trabalho de Conclusão de Curso**

**Autor: Eduardo Edson  
Hygor Fellipe  
Orientador: Mario Braga**

## Sumário

Introdução.....	2
Instalação das dependências do sistema .....	2
Baixar o projeto.....	3
Criar Banco de Dados .....	3
Instalação da Virtualenv .....	4
Dependências do Projeto .....	5
Acessando o Sistema .....	6

## Introdução

Este manual foi feito para a instalação do sistema Prontuário Único UCB feito como projeto final do curso de Ciência da Computação. Nesta instalação será usado um sistema operacional Linux, mais especificamente [Ubuntu 17.10](#), podendo ocorrer divergências em outras distribuições e sistemas operacionais.

## Instalação das dependências do sistema

Abra um terminal com o comando **Ctrl+Alt+T** e cole os seguintes comandos:

```
sudo apt-get install python-pip python-dev python3-dev libpq-dev postgresql postgresql-contrib npm curl git vim pgadmin3
```

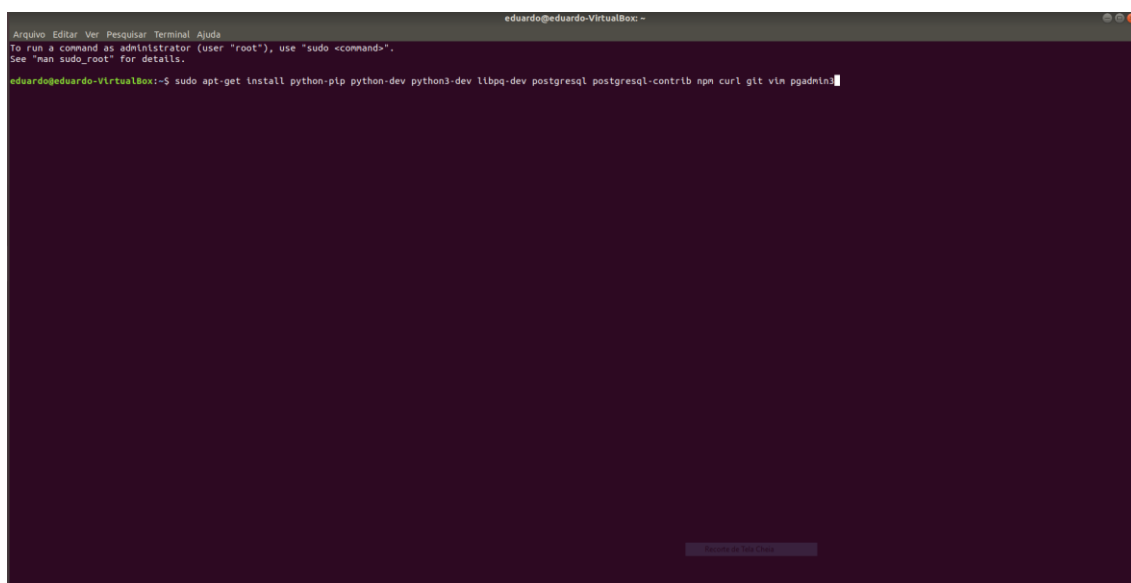


Figura 1

Após concluir a execução deste comando, você terá que colar os seguintes comandos e esperar suas execuções:

```
sudo curl -sL https://deb.nodesource.com/setup_7.x | sudo -E bash -
```

```
sudo apt-get install -y nodejs
```

```
sudo apt-get install -y build-essential
```

```
sudo npm install -g bower
```

Após concluir todas essas instalações, as dependências do sistema operacional foram atendidas.

## Baixar o projeto

O projeto se encontra disponível no GitHub, então para poder baixá-lo é necessário rodar o seguinte comando no terminal:

```
git clone https://github.com/eduardoedson/TCC.git
```

## Criar Banco de Dados

Primeiro precisamos entrar no terminal do postgresql usando o comando:

```
sudo su - postgres
```

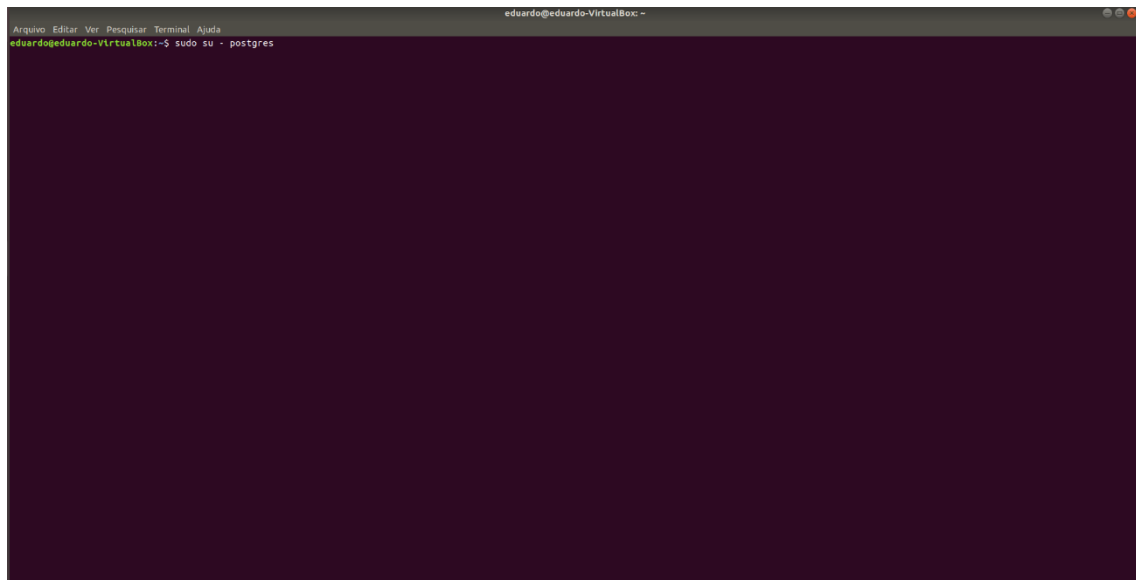


Figura 2

Agora vamos criar o banco com o seguinte comando:

```
createdb prontuario
```

Após esse comando, vamos entrar no psql com o seguinte comando e criar o usuário e suas permissões:

```
psql
```

```
CREATE USER root WITH PASSWORD 'root';
```

```
GRANT ALL PRIVILEGES ON DATABASE prontuario TO root;
```

Para sair do psql e do postgresql use os seguintes comandos respectivamente:

```
\q
```

```
exit
```

## Instalação da Virtualenv

A virtualenv será usada para instalar as dependências específicas do projeto sem prejudicar as instalações do sistema operacional.

Para instalar use o seguinte comando no terminal:

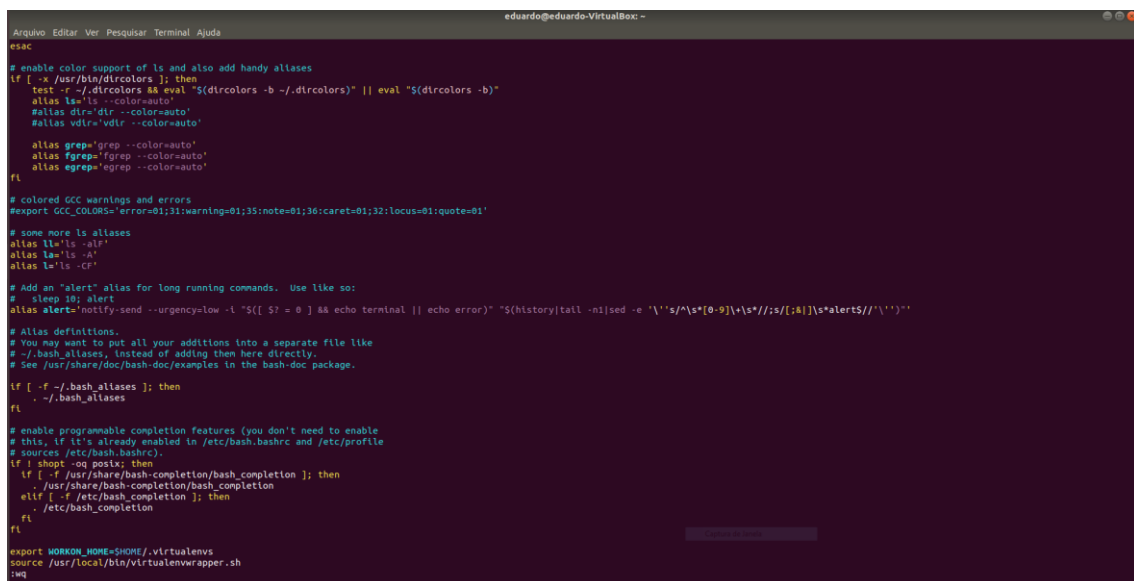
```
sudo pip install virtualenvwrapper
```

Após instalar, devemos configurar o bashrc. Para abri-lo usaremos o seguinte comando:

```
sudo vim ~/.bashrc
```

No final do arquivo iremos adicionar essas duas novas linhas:

```
export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```



```
eduardo@eduardo-VirtualBox: ~
$ cat ~/.bashrc
# If not running in sudo, enable color support of ls and also add handy aliases
if [ "$EUID" -ne 0 ] ; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    alias dir='dir --color=auto'
    alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -lF'
alias la='ls -A'
alias ls='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&]\s*$alert$//'\'' )"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -o posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```

Figura 3

Após adicionar as linhas, iremos apertar **ESC** e digitar **:wq** para fechar e salvar o arquivo.

Agora temos que recarregar o arquivo com o seguinte comando:

```
source ~/.bashrc
```

Com o seguinte comando iremos criar e executar uma virtualenv configurada para usar python3:

```
mkvirtualenv --python=/usr/bin/python3 prontuario
```

O `(prontuario)` (Figura 4) escrito no terminal indica que você está dentro da virtualenv, que já foi executada, por ser a primeira vez. Porém nas outras vezes que precisar rodar o sistema, você terá que usar o comando `workon prontuario` para ligar a virtualenv.

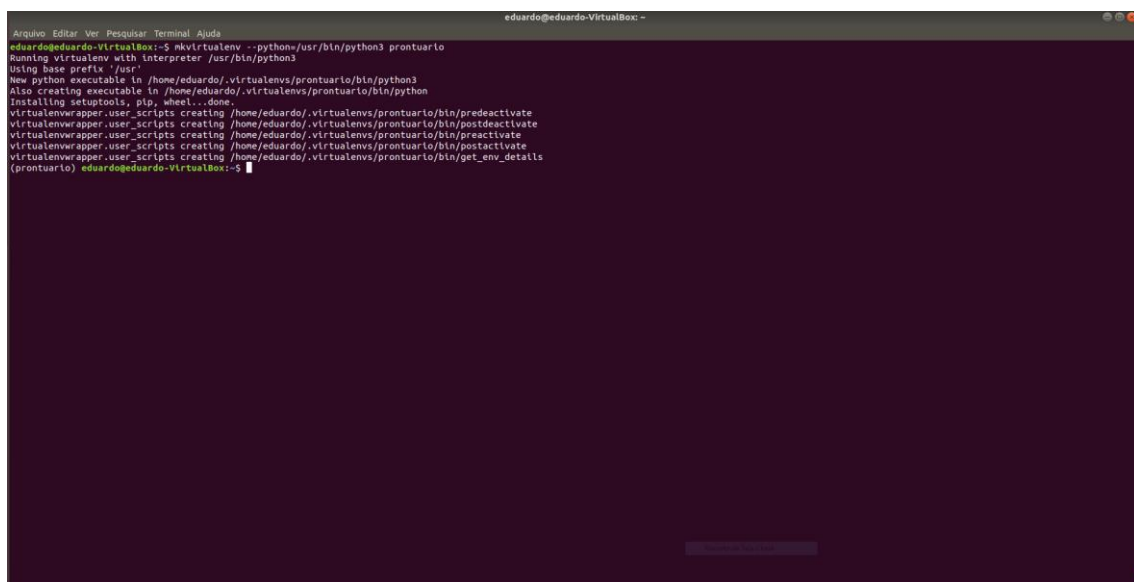
A terminal window titled 'eduardo@eduardo-VirtualBox: -' shows the execution of the command 'mkvirtualenv --python=/usr/bin/python3 prontuario'. The output displays the process of creating a new Python virtual environment, including installing setuptools, pip, and wheel, and creating wrapper scripts for predeactivate, deactivate, preactivate, and activate. The prompt changes from 'eduardo@eduardo-VirtualBox:~\$' to '(prontuario) eduardo@eduardo-VirtualBox:~\$', indicating the environment is active.

Figura 4

## Dependências do Projeto

Agora que já estamos dentro da virtualenv precisamos entrar pasta do projeto que foi clonado nos passos anteriores:

```
cd /TCC/django_project
```

Agora iremos instalar as dependências específicas para o projeto com os seguintes comandos:

```
pip install -r requirements.txt
```

```
./manage.py bower install
```

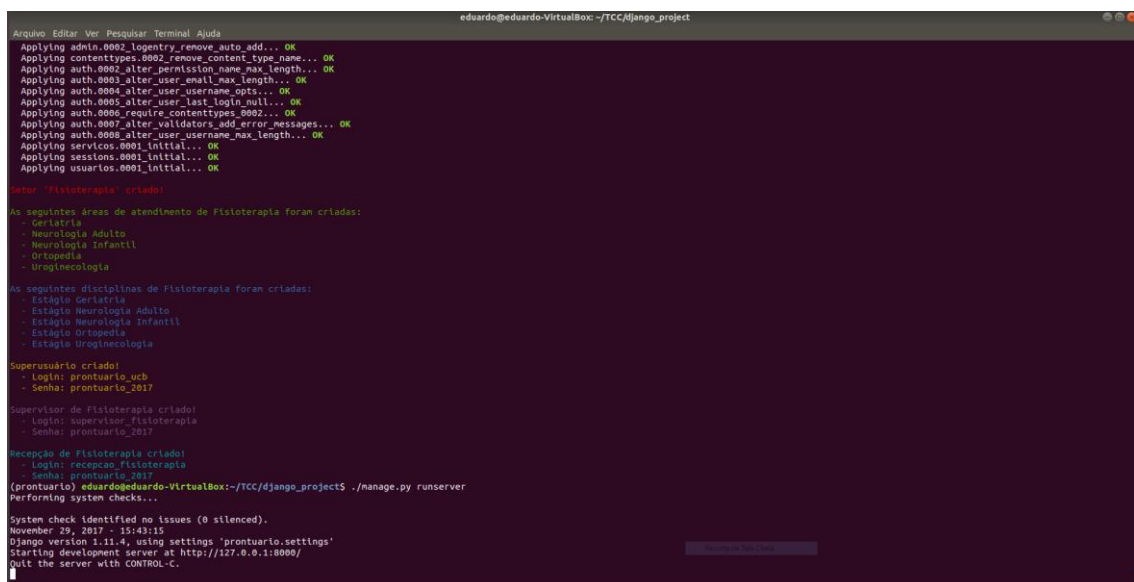
Caso existam opções na hora de instalação, qualquer uma pode ser selecionada.

Com isso terminamos todas as instalações, faltando somente criar as tabelas do banco de dados com o seguinte comando:

```
./manage.py migrate
```

Agora só precisamos rodar o sistema para poder acessá-lo no navegador:

```
./manage.py runserver
```



```
eduardo@eduardo-VirtualBox: ~/TCC/django_project
Arquivo Editor Ver Pesquisar Terminal Ajuda
Applying admin.0002_logentry_remove_auto_add... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying services.0001_initial... OK
Applying sessions.0001_initial... OK
Applying usuarios.0001_initial... OK

Instar 'Fisioterapia' criado!

As seguintes áreas de atendimento de Fisioterapia foram criadas:
- Geriatria
- Neurologia Adulto
- Neurologia Infantil
- Ortopedia
- Urologia

As seguintes disciplinas de Fisioterapia foram criadas:
- Estágio Geriatria
- Estágio Neurologia Adulto
- Estágio Neurologia Infantil
- Estágio Ortopedia
- Estágio Urologia

Superusuário criado!
- Login: prontuario_uch
- Senha: prontuario_2017

Supervisor de Fisioterapia criado!
- Login: supervisor.fisioterapia
- Senha: prontuario_2017

Recepção de Fisioterapia criado!
- Login: rececao.fisioterapia
- Senha: prontuario_2017

(prontuario) eduardo@eduardo-VirtualBox:~/TCC/django_project$ ./manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
November 29, 2017 - 15:43:15
Django version 1.11.4, using settings 'prontuario.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Figura 5

Após todos esses passos, o sistema foi instalado e está rodando.

## Acessando o Sistema

Para acessar o sistema, basta você ir no seu navegador e acessar a url <http://localhost:8000>.

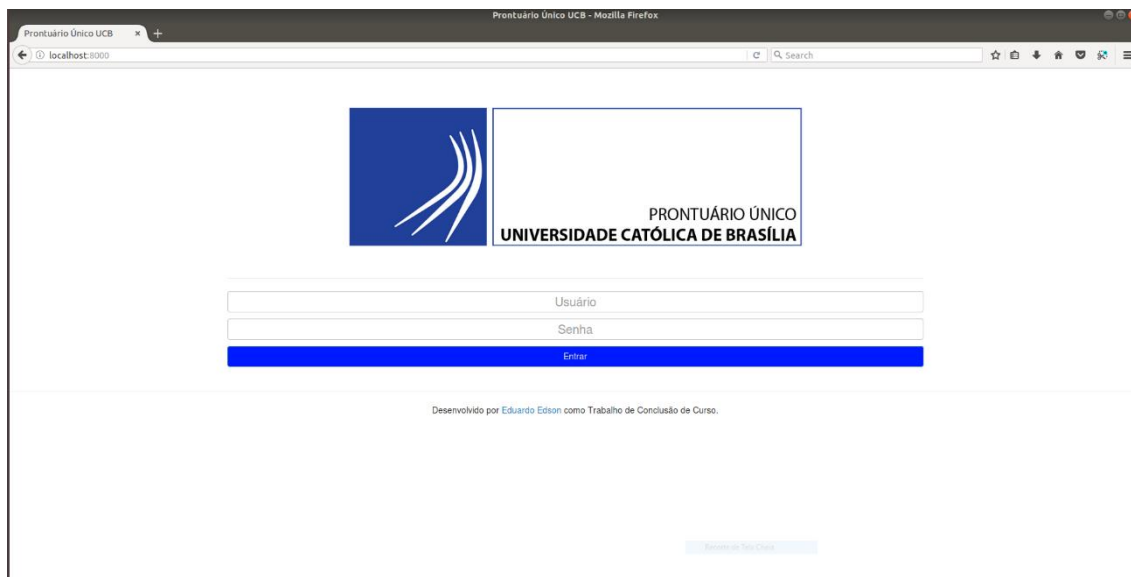


Figura 6

Durante a instalação foram criados usuários padrões para que você possa ter acesso:

Usuário	Senha	Tipo
prontuário_ucb	prontuário_2017	Superusuário
supervisor_fisioterapia	prontuário_2017	Supervisor
recepção_fisioterapia	prontuário_2017	Recepcionista

## Acessando o sistema outras vezes

Para acessar o sistema outras vezes iremos usar 3 comandos:

```
workon prontuario
```

```
Cd TCC/django_project
```

```
./manage.py runserver
```

E agora é só acessar a url <http://localhost:8000> no navegador.