



**Universidade
Católica de Brasília**

**Pró-Reitoria Acadêmica
Escola de Educação, Tecnologia e Comunicação
Curso de Ciência da Computação
Trabalho de Conclusão de Curso**

Manual de Instalação Prontuário Único UCB

**Autores: Eduardo Edson Batista Cordeiro Alves
Hygor Fellipe
Orientador: Mario de Oliveira Braga Filho
Co-Orientadora: Samara Haddah Simões Machado**

**Brasília - DF
2018**

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	6
Acessando o Sistema	7

Introdução

Este manual foi feito para a instalação do sistema Prontuário Único UCB, produzido como projeto final do curso de Ciência da Computação. Nessa 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
```

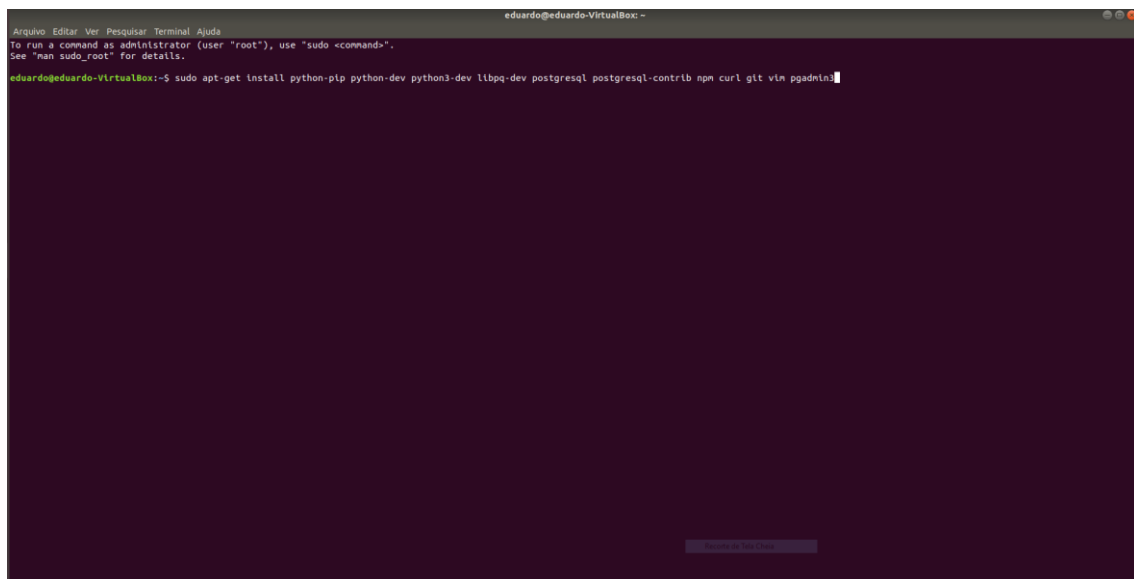


Figura 1

Após concluir a execução desse comando, 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
```

Ao finalizar todas as instalações, as dependências do sistema operacional foram atendidas.

Baixar o projeto

O projeto encontra-se disponível no GitHub, 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, será preciso entrar no terminal do postgresql usando o comando:

```
sudo su - postgres
```

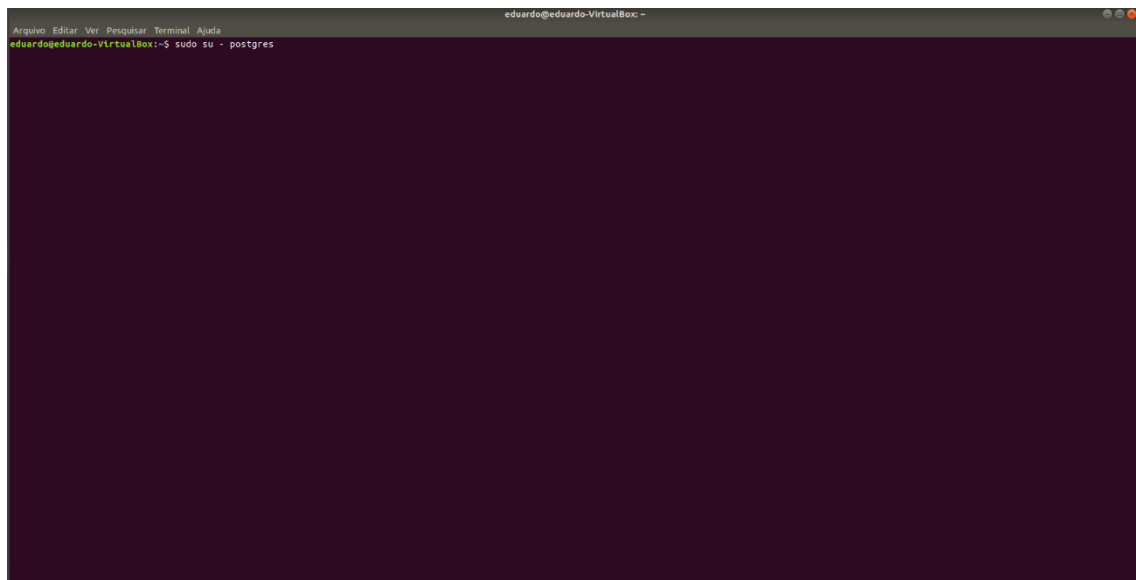


Figura 2

Agora vamos criar o banco com o seguinte comando:

```
createdb prontuario
```

Após esse comando, é necessário 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
```

Em seguida, deverá configurar o bashrc. Para abri-lo, usaremos o seguinte comando:

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

No final do arquivo, será adicionado essas duas novas linhas:

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

```
eduardo@eduardo-VirtualBox: ~
$ cat
# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; 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 "${ S? = 0 }" && echo terminal || echo error)' "${history/tail -n1}sed -e '\s/\s*[0-9]\{1,\}/:z/[!&]]\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 -oq 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

:wq
```

Figura 3

Depois de adicionar as linhas, iremos apertar **ESC** e digitas **:wq** para fechar e salvar o arquivo.

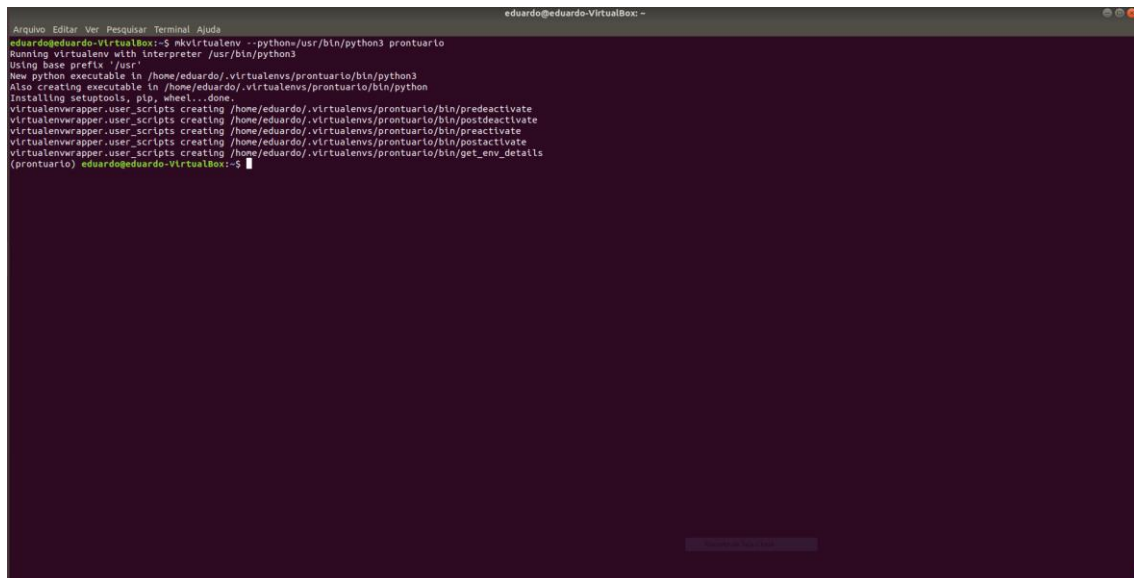
Nesse momento é preciso recarregar o arquivo com o seguinte comando:

```
source ~/.bashrc
```

Com o seguinte comando, será criada e executada 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 rodar o sistema, terá que usar o comando **workon prontuario** para ligar a virtualenv.



```
eduardo@eduardo-VirtualBox: ~$ mkvirtualenv --python=/usr/bin/python3 prontuario
Running virtualenv with interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/eduardo/.virtualenvs/prontuario/bin/python3
Also creating executable in /home/eduardo/.virtualenvs/prontuario/bin/python
Installing setuptools, pip, wheel...done.
virtualenvwrapper.user_scripts creating /home/eduardo/.virtualenvs/prontuario/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/eduardo/.virtualenvs/prontuario/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/eduardo/.virtualenvs/prontuario/bin/preactivate
virtualenvwrapper.user_scripts creating /home/eduardo/.virtualenvs/prontuario/bin/postactivate
virtualenvwrapper.user_scripts creating /home/eduardo/.virtualenvs/prontuario/bin/get_env_details
(prontuario) eduardo@eduardo-VirtualBox: ~$
```

Figura 4

Dependências do Projeto

A partir deste momento, estando dentro da virtualenv, será preciso entrar pasta do projeto que foi clonado nos passos anteriores:

```
cd /TCC/django_project
```

Neste momento será preciso instalar as dependências específicas do 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 a instalação está completa, faltando somente criar as tabelas do banco de dados com o seguinte comando:

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

Para rodar o sistema para poder acessá-lo no navegador:

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

```
Arquivo Editar Ver Pesquisar Terminal Ajuda
eduardo@eduardo-VirtualBox: ~/TCC/django_project

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_is_superuser_opts... 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

Instalar 'Fisioterapia' criado!

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

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

Superusuário criado!
- Login: prontuario_uch
- Senha: prontuario_2017

Supervisor de Fisioterapia criado!
- Login: supervisor_fisioterapia
- Senha: prontuario_2017

Recepção de Fisioterapia criado!
- Login: recepcao_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 acessar a url <http://localhost:8000>.

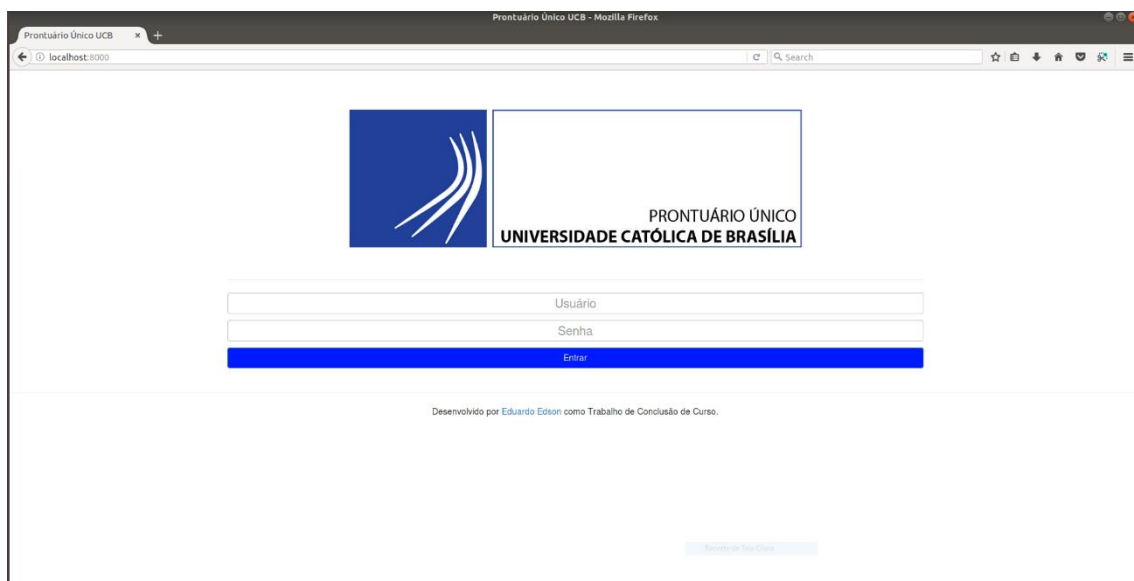


Figura 6

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, pra finalizar, é só acessar a url <http://localhost:8000> no navegador.