

documentação sonda linux

Mariana Gomes Costa (mariana.gomes@ee.ufcg.edu.br)

October 2020

1 Linux

Nesta seção, o simulador será instalado em um sistema operacional Linux. Novamente, é apenas uma das formas de instalação do SONDA. Este guia deve ser útil, portanto, para Ubuntu e Debian podendo talvez ser aplicados para qualquer outro terminal que utilize o pacote Aptitude (apt-get)

- Para a instalação, foi usado o ubuntu 18.10;
- Novamente, é necessário obter alguma IDE para linguagem *python* de sua preferência.
 - Para este exemplo está sendo instalada a IDE PyCharm;
 - Se o usuário já tiver uma IDE de preferência ignorar os próximos dois passos.
- Para a instalação de programas e pacotes, estaremos utilizando o terminal;
- Digite a seguinte linha de comando no terminal para a instalação da IDE PyCharm Community:
 - `sudo snap install pycharm-community --classic`
- Após a IDE ser instalada, é necessário instalar os programas para a GUI e para o banco de dados MySQL. Para isso, digite no terminal as linhas de código abaixo:
 - `sudo apt update`
 - * Essa linha de comando se refere à atualização do pacote de instalações;
 - `sudo apt install mysql-server`
 - * Essa linha irá instalar o MySQL;
 - Para testar o funcionamento do MySQL, basta digitar no terminal:
 - * `sudo systemctl status mysql`

Figura 1: Ouput da linha de comando acima

```
[sudo] senha para mariana:
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: e
   Active: active (running) since Tue 2020-07-14 18:52:52 -03; 21min ago
     Main PID: 2339 (mysqld)
       Tasks: 28 (limit: 2328)
      CGroup: /system.slice/mysql.service
              └─2339 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pi

jul 14 18:52:51 mariana-VirtualBox systemd[1]: Starting MySQL Community Server.
jul 14 18:52:52 mariana-VirtualBox systemd[1]: Started MySQL Community Server.
```

- * Na figura acima, podemos perceber que o MySQL está ativo e em execução.
- Se o MySQL não está executando, execute a seguinte linha de código:
 - * `sudo systemctl start mysql`
- Para uma verificação adicional, você pode tentar se conectar ao banco de dados utilizando a ferramenta `mysqladmin`, que é um cliente que lhe permite executar comandos administrativos. Por exemplo, este comando diz para conectar como `root` (`-u root`), solicitar uma senha (`-p`), e retornar a versão
 - * `sudo mysqladmin -p -u root version`

Figura 2: Ouput do comando: `sudo mysqladmin -p -u root version`

```
mysqladmin Ver 8.42 Distrib 5.7.30, for Linux on x86_64
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.

Server version          5.7.30-0ubuntu0.18.04.1
Protocol version        10
Connection               Localhost via UNIX socket
UNIX socket              /var/run/mysqld/mysqld.sock
Uptime:                  24 min 56 sec

Threads: 1 Questions: 6 Slow queries: 0 Opens: 106 Flush tables: 1 Open ta
bles: 99 Queries per second avg: 0.004
```

- Após esses passos, a versão mais básica do banco de dados já deve ter sido instalada no seu sistema operacional. O passo seguinte é instalar o PyQT5 para a execução da interface gráfica, para isso digite a seguinte linha de comando no seu terminal:
 - `sudo apt-get install python3-pyqt5`

Após a instalação dos programas necessários, o próximo passo é criar o banco de dados 'sonda' no MySQL.

- Primeiramente, é necessário se conectar ao banco de dados. no terminal, digite a seguinte linha de código:

- `mysql -u root -p`
- o item `-u` indica o usuário e `-p` a senha, `'root'` e a senha nula são padrões nas instalações do banco de dados.
- **IMPORTANTE:** Caso tenha aparecido o ERROR 1968 como o da figura abaixo, digite as seguintes linhas de comando no terminal:

Figura 3: ERROR 1968

```
mariana@mariana-VirtualBox:~$ mysql -u root -p
Enter password:
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
```

* `sudo mysql -u root`

Figura 4: Output do comando `sudo mysql -u root`

```
mariana@mariana-VirtualBox:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
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.
```

* `USE mysql;`

Figura 5: Output do comando `USE mysql`

```
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
```

* `UPDATE user SET plugin='mysql_native_password' WHERE User='root';`

Figura 6: Ouput do comando acima

```
mysql> UPDATE user SET plugin='mysql_native_password' WHERE User='root';  
Query OK, 1 row affected (0.09 sec)  
Rows matched: 1 Changed: 1 Warnings: 0
```

* FLUSH PRIVILEGES;

Figura 7: Ouput da linha de comando acima

```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.04 sec)
```

* exit;

* service mysql restart

Figura 8: Ouput da linha de comando acima

```
mysql> exit;  
Bye  
mariana@mariana-VirtualBox:~$ service mysql restart
```

- Após essa correção, já deve ser possível executar a linha de comando `mysql -u root -p` sem nenhum problema, como é mostrado na figura a seguir.

Figura 9: Ouput do comando `mysql -u root -p` após correção de erro

```
mariana@mariana-VirtualBox:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
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.
```

- Agora conectados ao mysql, podemos ver as bases de dados instaladas previamente a partir do comando:
 - `show databases;`
- Para criar uma nova base de dados 'sonda', basta digitar no terminal o seguinte comando:
 - `create database sonda;`
 - **IMPORTANTE:** lembre-se de NÃO alterar o nome para a base de dados, já que o nome 'sonda' é utilizado no código.
- Para checar se foi realmente criado, basta verificar com o comando `show databases;` novamente
- Para criar tabelas, é necessário seguir os comandos abaixo:
 - `use sonda;`

Figura 10: Ouput do comando `use sonda;`

```
mysql> use sonda;
Database changed
```

- Após esse comando, basta criar a table a partir do comando abaixo da mesma forma que foi descrita no sistema operacional Windows. **IMPORTANTE:** lembre-se de NÃO alterar o nome para a tabela, já que o nome 'simulacao' é utilizado no código.
- **CORREÇÃO:** distamplificador DOUBLE (5,2) e não distamplificador INT(5) como está exibido na figura.
- Da mesma forma que a tabela simulacao foi criada, basta repetir o procedimento para criar a tabela de simulação bloqueios. **IMPOR- TANTE:** lembre-se de NÃO alterar o nome para a tabela, já que o nome 'simulacao' é utilizado no código.
- Para verificar a criação da tabela, utilizamos o comando:

Figura 11: Criando tabelas no terminal Ubuntu

```
mysql> create table simulacao(topologia VARCHAR(30),
-> rede VARCHAR(4),
-> fibra INT(3),
-> modos INT(2),
-> nucleo INT(2),
-> mintrafego INT(4),
-> maxtrafego INT(4),
-> passo INT(4),
-> distamplificador INT(5),
-> chamadas INT(11),
-> probabilidade VARCHAR(255)
-> );
Query OK, 0 rows affected (0.09 sec)
```

Figura 12: Criando tabelas no terminal Ubuntu

```
create table simulacao_bloqueios(topologia VARCHAR(30),
rede VARCHAR(4),
fibra INT(3),
modos INT(2),
nucleo INT(2),
mintrafego INT(4),
maxtrafego INT(4),
passo INT(4),
distamplificador DOUBLE(5,2),
bloqueios INT(11),
probabilidade VARCHAR(255)
);|
```

* show tables;

Figura 13: *show tables;*

```
mysql> show tables;
+-----+
| Tables_in_sonda |
+-----+
| simulacao        |
+-----+
1 row in set (0.00 sec)
```

- Para descrição da tabela 'simulacao' criada, utilizamos o comando:
* describe simulacao;

Figura 14: *describe simulacao;*

```
mysql> describe simulacao;
```

Field	Type	Null	Key	Default	Extra
topologia	varchar(30)	YES		NULL	
rede	varchar(4)	YES		NULL	
fibra	int(3)	YES		NULL	
modos	int(2)	YES		NULL	
nucleo	int(2)	YES		NULL	
mintrafego	int(4)	YES		NULL	
maxtrafego	int(4)	YES		NULL	
passo	int(4)	YES		NULL	
distamplificador	int(5)	YES		NULL	
chamadas	int(11)	YES		NULL	
probabilidade	varchar(255)	YES		NULL	

11 rows in set (0.01 sec)

- Também é possível criar a base de dados e tabelas a partir do phpMyAdmin, assim como foi feito no *Windows*. Este passo é algo totalmente adicional, como foi dito antes nesse documento, a phpMyAdmin é uma forma de visualizar e gerenciar o seu banco de dados a partir do navegador. Para isso, siga os comandos a seguir:
 - * `sudo apt update`
 - * `sudo apt install phpmyadmin php-mbstring php-gettext`
 - * Para a seleção do servidor, escolha `apache2`
 - * **IMPORTANTE:** Caso o servidor `apache2` não esteja selecionado aperte as teclas `SPACE`, `TAB` e, então, `ENTER` para selecionar `apache`
 - * **IMPORTANTE:** Selecione `Sim` quando indagado quanto a se usará o `dbconfig-common` para configurar o banco de dados
 - * **IMPORTANTE:** A senha *default* para o MySQL é nula, portanto quando a senha for solicitada basta apertar `ENTER`
 - * `sudo phpenmod mbstring`
 - * **IMPORTANTE:** O comando acima está habilitando explicitamente a extensão do PHP, mas a instalação adiciona, automaticamente, o arquivo de configuração do `apache2` `phpMyAdmin` no diretório `/etc/apache2/conf-enabled/`.
 - * `sudo systemctl restart apache2`
 - * O acesso é feito a partir do seu navegador: `http://seu_domínio.ou_IP/phpmyadmin`
 - * A criação da tabela é feita da mesma forma descrita para *Windows* a partir da Figura 22 desse documento.

Instalado todos os requisitos necessários, instalaremos o GitHub para termos acesso ao código. Para isso, siga os passos descritos a seguir:

- `sudo apt-get install git`
- `git clone https://github.com/helderufcg/SONDA`

Com isso, será instalado o git no computador e no comando seguinte, será feita uma cópia do projeto para a pasta escolhida com as últimas modificações no github.

- Outra opção é acessar o site `https://github.com/helderufcg/SONDA` a partir do navegador e baixar manualmente o arquivo.

Após esses procedimentos, basta abrir a IDE e procurar a pasta onde o SONDA está e começar a simular.

- **IMPORTANTE:** Não esquecer de instalar os seguintes pacotes:
 - `matplotlib` (para a visualização dos gráficos)
 - `pyqt5` (para a execução da interface gráfica)

- numpy
- networkx
- mysql.connector
- dill
- NÃO esquecer de verificar a versão do interpretador que está na IDE, pois o SONDA foi construído em Python3.
- IMPORTANTE: Elimine a linha `port="3308"` no código quando for executar em Linux, pois essa linha só foi adicionada no código quando o *WampServer* também instalou o banco de dados MariaDB. O teste de conexão com o banco de dados pode ser feito da seguinte forma:

Figura 15: Teste de conexão

```
import mysql.connector

banco = mysql.connector.connect(
    host="localhost",
    user="root",
    password=""
)

print(banco)
```

- Se não ocorrer nenhum erro, é porque o código está devidamente conectado ao banco de dados;
- **ATUALIZAÇÃO 22/07/2020:**
 - Para a instalação do pacote 'MySQLdb' execute a seguinte linha de comando no terminal:
 - * `sudo apt-get install libmysqlclient-dev`
 - Em seguida, instale o pacote:
 - * `mysqlclient`
 - Nesse passo podem existir erros específicos para cada máquina que podem impedir a execução da primeira linha de comando. Para isso, são necessários métodos específicos de correção de erro para cada máquina. Veja [AQUI](#) um dos erros comuns e alguns métodos para solucionar o problema.
- **ATUALIZAÇÃO 31/08/2020:**
 - Criação do banco de dados para a simulação 3:

Figura 16: *simulacao_porcentagem*

```
create table simulacao_porcentagem(topologia VARCHAR(30),
                                   rede VARCHAR(4),
                                   fibra INT(3),
                                   modos INT(2),
                                   nucleo INT(2),
                                   mintrafego INT(4),
                                   maxtrafego INT(4),
                                   passo INT(4),
                                   distamplificador DOUBLE(5,2),
                                   chamadas INT(11),
                                   cargatrafego INT(8),
                                   probabilidade VARCHAR(255));
```

- Criação do banco de dados para a simulação 4:

Figura 17: *simulacao_ber*

```
create table simulacao_ber(topologia VARCHAR(30),
                           rede VARCHAR(4),
                           fibra INT(3),
                           modos INT(2),
                           nucleo INT(2),
                           mintrafego INT(4),
                           maxtrafego INT(4),
                           passo INT(4),
                           distamplificador DOUBLE(5,2),
                           chamadas INT(11),
                           probabilidade VARCHAR(255));
```

- **ATUALIZAÇÃO 24/02/2021:**

- instalação do prettytable para a exibição dos resultados fora da interface gráfica:
 - * pip install prettytable