**Conectando Python a um banco de dados Mysql**

Universidade Católica de Pelotas

Guilherme Moura Baccarin

**Introdução**

Este projeto tem como objetivo a captação e armazenamento de informações importantes de um computador através da conexão da linguagem de programação Python e o uso de um bancp de dados Mysql. O projeto prevê a leitura de infomações como, percentual de uso de memória RAM, percentual livre do disco, para futuras manipulações dos mesmos, com inúmeras possíves aplicações nos dias atuais.

**Sumário**

[**Introdução** 2](#_Toc51711074)

[**Bibliotecas** 4](#_Toc51711075)

[**Dados obtidos** 5](#_Toc51711076)

[**Implementação** 6](#_Toc51711077)

[**Estrutura do banco** 9](#_Toc51711078)

[**Consultas no banco** 10](#_Toc51711079)

[**Conclusão** 11](#_Toc51711080)

**Bibliotecas**

* **SQLAlchemy** -SQLAlchemy é um [**framework**](https://www.treinaweb.com.br/blog/para-que-serve-um-framework/) de mapeamento objeto-relacional SQL (ORM) , [**ORM**](https://www.treinaweb.com.br/blog/o-que-e-orm/) trata-se de um Framework que visa auxiliar na redução da impedância, realizando todas as conversões necessárias entre o modelo relacional e o modelo orientado a objetos de maneira automática.
* **Platform** - Usado para acessar os dados da plataforma subjacente, como hardware, sistema operacional e informações de versão do interpretador.
* **Os** – Um dos principais módulos em Python, traz informações sobre o sistema operacional.
* **Time (Utilizada)** - Usada para possibilitar um intervalo entre as leituras.
* **Psutil (Utilizada)** – Usada para recuperar informações sobre processos em execução e utilização sistema.
* **Mysql.connector (Utilizada)** – Permite que programas Python acessem bancos de dados MySQL.

**Dados obtidos**

* **Valor usado de disco** – Importante manter um controle sob a utilizalção do disco, caso aconteça uma invasão de um usuário não desejado e esteja usando a máquina para fins não desejados.
* **Valor de disco livre** - Importante manter um controle sob a utilizalção do disco para eventuais necessidades de um upgrade de máquina.
* **Valor de disco livre percentual** – Valor mais visível da utilização do disco.
* **Valor de swap percentual** – Importante se ter conhecimento da quantidade de swap que a máquina está utilizando, pois caso esteja sendo usado um valor alto, deixará a máquina mais lenta.
* **Valor de memória RAM livre** – Um valor muito usado para medir a velocidade de processamento da máquina.
* **Valor de memória RAM livre percentual** – Uma forma mais palpável de visualizar o uso de memória RAM.
* **Data de leitura** – Indispensável saber quando foi feita a leitura das informações.

**Implementação**

* **Conexão com o banco de dados.**

1. connection = mysql.connector.connect(
2. #Informações da base de dados
3. host='localhost', database='BASE', user='root',
4. password='SENHA')
5. if connection.is\_connected(): #Caso funcione a conexão
6. db\_Info = connection.get\_server\_info() #Info do db
7. print("Conectado a MySQL Server versão ", db\_Info)
8. cursor = connection.cursor()
9. cursor.execute("select database();") #Retorna nome da base
10. record = cursor.fetchone()
11. print("Base conectada: ", record)

* **Informa qual disco**

1. discoG = psutil.disk\_usage('G://')
2. discoC = psutil.disk\_usage('C://')
3. discoE = psutil.disk\_usage('E://')

* **Swap**

1. swap = psutil.swap\_memory()

* **RAM**

1. ram = psutil.virtual\_memory()

* **Insert**

1. #Texto base do insert na base de dados
2. insert = """insert into leitura (id\_disco,valor\_usado,valor\_livre,valor\_livre\_percentual,
3. swap\_percentual,ram\_livre,ram\_livre\_percentual)
4. values (%s,%s,%s,%s,%s,%s,%s);"""

* **Leituras**
  + **Disco G**

1. valor\_usadoG = round(discoG.used / (1024.0 \*\* 3), 2)
2. valor\_livreG = round(discoG.free / (1024.0 \*\* 3), 2)
3. valor\_livre\_percentualG = discoG.percent
   * **Disco C**
4. valor\_usadoC = round(discoC.used / (1024.0 \*\* 3), 2)
5. valor\_livreC = round(discoC.free / (1024.0 \*\* 3), 2)
6. valor\_livre\_percentualC = discoC.percent
   * **Disco E**
7. valor\_usadoE = round(discoE.used / (1024.0 \*\* 3), 2)
8. valor\_livreE = round(discoE.free / (1024.0 \*\* 3), 2)
9. valor\_livre\_percentualE = discoE.percent
   * **Swap**
10. swap\_percentual = round(swap.percent, 2)
    * **RAM**
11. ram\_livre = round(ram.percent, 2)
12. ram\_livre\_percentual = round(ram.available \* 100 / ram.total, 2)

* **Executa**

1. #Executando os inserts na base de dados
2. cursor.execute(insert, (id\_disco\_G,valor\_usadoG,valor\_livreG,valor\_livre\_percentualG,swap\_percentual,ram\_livre,ram\_livre\_percentual))
3. cursor.execute(insert, (id\_disco\_C,valor\_usadoC,valor\_livreC,valor\_livre\_percentualC,swap\_percentual,ram\_livre,ram\_livre\_percentual))
4. cursor.execute(insert, (id\_disco\_E,valor\_usadoE,valor\_livreE,valor\_livre\_percentualE,swap\_percentual,ram\_livre,ram\_livre\_percentual))

* **Commit**

1. #Envia o insert para o banco de dados
2. connection.commit()

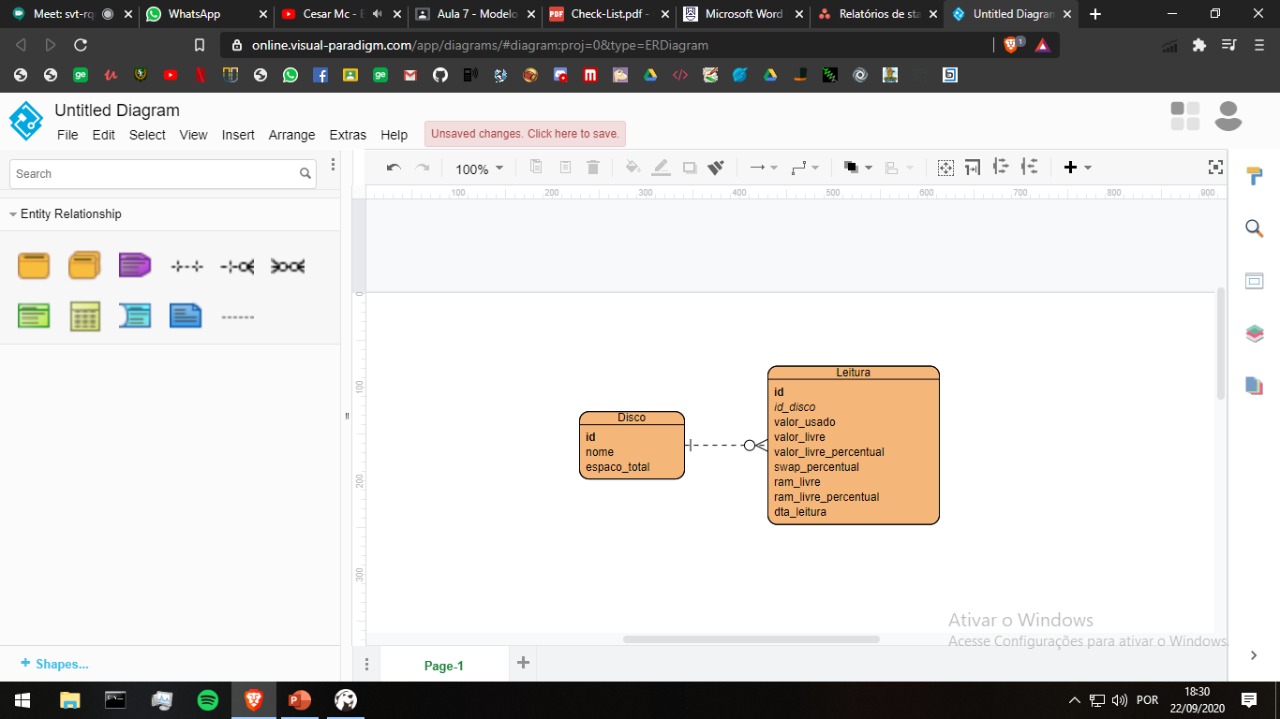
* **Sleep**

1. time.sleep(50)

**Estrutura do banco**

* + **SQL**

1. **create** **table** disco (
2. id **int** **unique** auto\_increment **primary** **key**,
3. nome **varchar**(100) **not** **null**,
4. espaco\_total **float** **not** **null**
5. );
6. **create** **table** leitura(
7. id **int** **unique** auto\_increment **primary** **key**,
8. id\_disco **int** **not** **null**,
9. valor\_usado **float** **not** **null**,
10. valor\_livre **float** **not** **null**,
11. valor\_livre\_percentual **float** **not** **null**,
12. swap\_percentual **float** **not** **null**,
13. ram\_livre **float** **not** **null**,
14. ram\_livre\_percentual **float** **not** **null**,
15. dta\_leitura **TIMESTAMP** **NOT** **NULL** **DEFAULT** **CURRENT\_TIMESTAMP**,
16. **FOREIGN** **KEY** (id\_disco) **REFERENCES** disco(id)
17. );
    * **Entidade-Relacionamento**



**Consultas no banco**

* + -- Quando que a utilização da CPU foi menor que 20% e maior que 80%?
    1. **select** \* **from** leitura **where** valor\_livre\_percentual > 20 **and** valor\_livre\_percentual < 80;
  + -- Qual foi o máximo e o mínimo de memória utilizado nos últimos três dias?
    - * maximo
        1. **select** \* **from** leitura **where** dta\_leitura **between** STR\_TO\_DATE( "20/09/2020", "%d/%m/%Y" ) **AND**
        2. STR\_TO\_DATE( "23/09/2020", "%d/%m/%Y" )
        3. **order** **by** valor\_livre\_percentual **desc** **limit** 1;
      * minimo

1. **select** \* **from** leitura **where** dta\_leitura **between** STR\_TO\_DATE( "20/09/2020", "%d/%m/%Y" ) **AND**

STR\_TO\_DATE( "23/09/2020", "%d/%m/%Y" )

1. **order** **by** valor\_livre\_percentual **limit** 1;
   * -- Quando a CPU obteve a frequência mínima e máxima?
     + - -- maximo
2. **select** \* **from** leitura
3. **order** **by** valor\_livre\_percentual **desc** **limit** 1;
4. -- minimo
5. **select** \* **from** leitura
6. **order** **by** valor\_livre\_percentual **limit** 1;
   * -- Considerando o monitoramento por 5 dias, qual a frequência média da CPU?
7. **select** id\_disco,round(**avg**(valor\_usado),2) media
8. **from** leitura **where** dta\_leitura **between**
9. STR\_TO\_DATE( "18/09/2020", "%d/%m/%Y" ) **AND**
10. STR\_TO\_DATE( "23/09/2020", "%d/%m/%Y" )
11. **group** **by** id\_disco;

**Conclusão**

Por meio desse projeto, foi possível visualizar que a implementação de uma conexão da linguagem de programação python e um banco de dados mysql é uma tarefa simples mas muito poderosa, pois a usabilidade da mesma é enorme.