




# Aula11 (21/11/2020) pygresql ou psycopg

## Materis:

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/2cd144de-d68c-4a88-a561-728923fa6550/aula11-pygresql\\_e-psycpg.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/2cd144de-d68c-4a88-a561-728923fa6550/aula11-pygresql_e-psycpg.pdf)

## Links:

### Aprofundamento no Python -

 [https://docs.google.com/presentation/d/1TuZKeH7Ri9t-gy3iaU5hj385Q\\_sdkMI2fyMEpCCmgqE/edit#slide=id.ga7397483fc\\_0\\_100](https://docs.google.com/presentation/d/1TuZKeH7Ri9t-gy3iaU5hj385Q_sdkMI2fyMEpCCmgqE/edit#slide=id.ga7397483fc_0_100)



### First Steps with PyGreSQL - PyGreSQL 5.2.1 documentation


In this small tutorial we show you the basic operations you can perform with both flavors of the PyGreSQL interface. Please choose your flavor: Before doing anything else, it's necessary to create a database connection. You can omit one or even all parameters if you want to use their default

 <https://pygresql.org/contents/tutorial.html>



### [Python] Arquivo com exemplo de conexão com banco de dados utilizando a biblioteca PyGreSQL

Instantly share code, notes, and snippets. [Python] Arquivo com exemplo de conexão com banco de dados utilizando a biblioteca PyGreSQL You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session.

 <https://gist.github.com/liniribeiro/1caff2e0cf31ac98c90f6bec0a3be789>

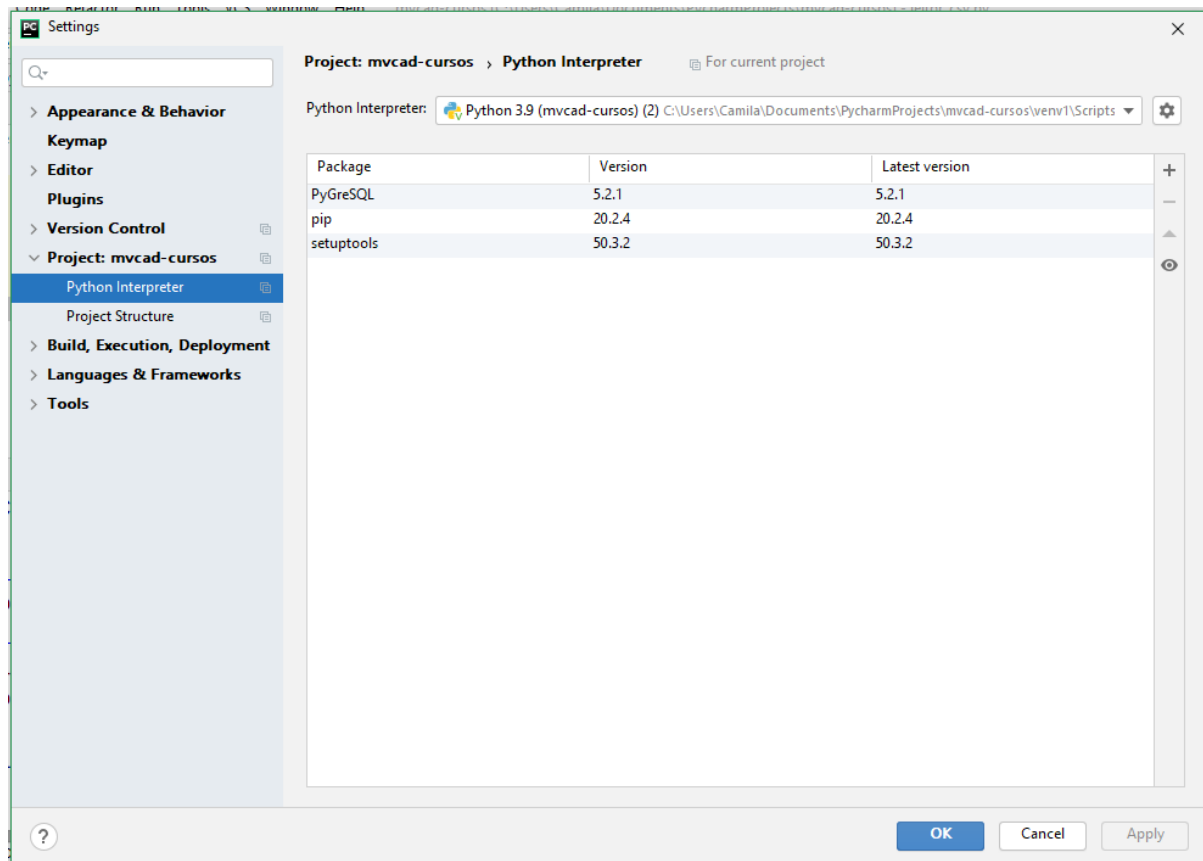


## Gravação da aula:

### Aula 11 - PyGreSQL e Psycopg.mkv

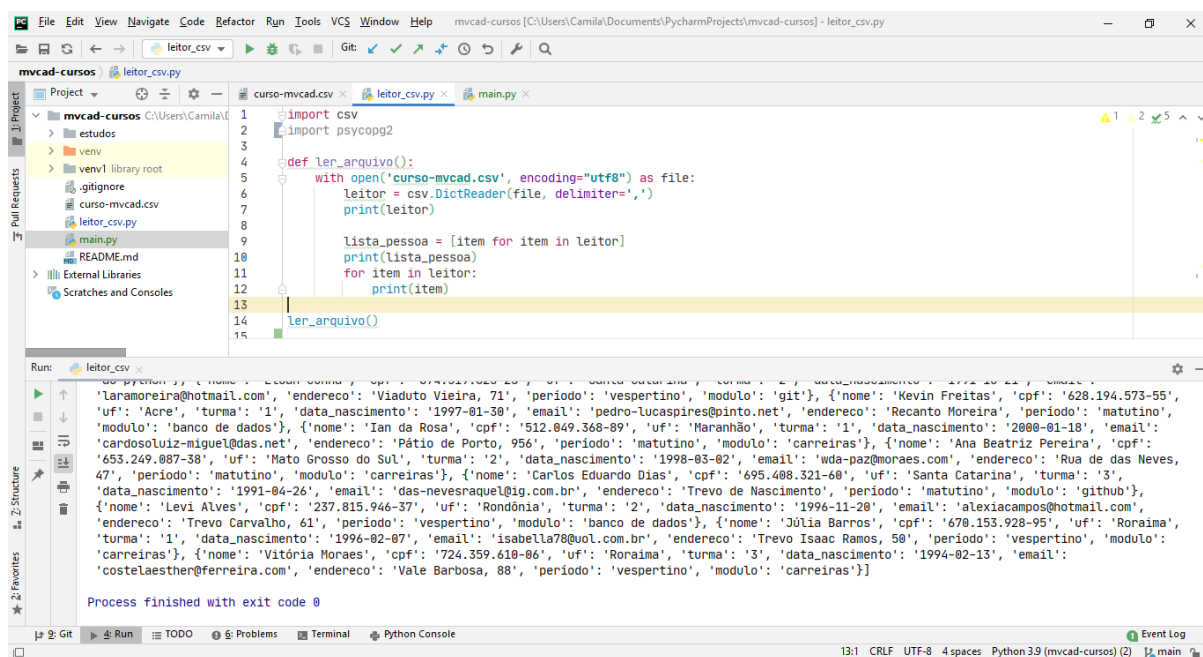
 <https://drive.google.com/file/d/1VDYjzNKByTyKPUGKfy132T1LAjdMLICc/view?usp=sharing>

No interpretador precisar ter instalado os seguintes programas:



Passo 1 - Importar a conexão com o banco abaixo de **import csv**.

- Se ele funcionar quer dizer que l está sincronizado com o banco.
- Caso apresente algum erro e importante consultar.



## PG ADMIN

Acessar o pgadmin e criar um nova database e uma nova tabela:

```
CREATE DATABASE cursos_mvcd;  
  
CREATE TABLE pessoa (  
    id_pessoa serial PRIMARY KEY,  
    nome varchar(200) NOT NULL,  
    endereco varchar(255),  
    cpf char(11),  
    estado varchar(50),  
    turma varchar(100),  
    periodo varchar(50),  
    modulo varchar(100)
```

## PYCHARM

### 1 - psycopg

- [psycopg\\_connection.py](#)

```
import psycopg2  
  
conn = psycopg2.connect("dbname=curso_mvcd user=postgres password=1234 host=localhost")  
  
conn.autocommit = True  
  
cursor = conn.cursor()
```

- [psycopg\\_main.py](#)

```
import csv  
  
from pessoa_psycopg import insere_pessoa, retorna_pessoas  
  
def ler_arquivo():  
    with open('curso-mvcd.csv', encoding="utf8") as file:  
        leitor = csv.DictReader(file, delimiter=',')  
  
        # List Comprehension  
        lista_pessoas = [item for item in leitor]  
        #print(lista_pessoas)  
  
        #for item in lista_pessoas:  
        #    print(item)  
  
        #cursor.execute("select * from pessoa ")  
        #cursor.execute("select relname from pgclass where relkin='r' and relname !~ '^(pg|sql_)';")  
        #print(cursor.fetchall())  
  
        #print(leitor)  
        #lista_pessoa = [item for item in leitor]  
        #print(lista_pessoa)  
        #for item in leitor:  
        #    print(item)  
  
#ler_arquivo()  
  
pessoa = {  
    'nome': "Leticia Maia",  
    'endereco': "Vila Nova",  
    'cpf': '43288899945',  
    'estado': "Santa Catarina",  
    'turma': "MVCAD Python 1",  
    'periodo': "matutino",  
    'modulo': "MVCAD"
```

```

}

insere_pessoa(pessoa)
print(retorna_pessoas())

```

- [psycpg\\_pessoa.py](#)

```

from psycpg_connection import cursor

def insere_pessoa(dados_pessoa):

    cursor.execute("INSERT INTO pessoa (nome, endereco, cpf, estado,"
                    "turma, periodo, modulo)"
                    "VALUES (%s, %s, %s, %s, %s, %s, %s)",
                    (
                        dados_pessoa['nome'],
                        dados_pessoa['endereco'],
                        dados_pessoa['cpf'],
                        dados_pessoa['estado'],
                        dados_pessoa['turma'],
                        dados_pessoa['periodo'],
                        dados_pessoa['modulo']
                    )
                )

def retorna_pessoas():
    cursor.execute("SELECT * FROM pessoa")
    return cursor.fetchall()

```

## 2 - pygresql

- [pygresql\\_connection.py](#)

```

from pg import DB
db = DB(dbname='mvcad-cursos', host='localhost', port=5432, user='postgres', passwd='postgres')

```

- [pygresql\\_main.py](#)

```

import csv

from pessoa_pygres import insere_pessoa, retorna_pessoas

def ler_arquivo():
    with open('curso-mvcad.csv', encoding="utf8") as file:
        leitor = csv.DictReader(file, delimiter=',')

        # List Comprehension
        lista_pessoas = [item for item in leitor]
        #print(lista_pessoas)

        #for item in lista_pessoas:
        #    print(item)

#ler_arquivo()

pessoa = {
    'nome': "Priscila",
    'endereco': "Vila Nova",
    'cpf': '01234567899',
    'estado': "Santa Catarina",
    'turma': "MVCAD Python 1",
}

```

```
        'periodo': "matutino",
        'modulo': "MVCAD"
    }

    insere_pessoa(pessoa)
    print(retorna_pessoas())
```

- [pygresql\\_pessoa.py](#)

```
from pygres_connection import db

def insere_pessoa(dados_pessoa):

    db.insert("pessoa",
              nome=dados_pessoa['nome'],
              endereco=dados_pessoa['endereco'],
              cpf=dados_pessoa['cpf'],
              estado=dados_pessoa['estado'],
              turma=dados_pessoa['turma'],
              periodo=dados_pessoa['periodo'],
              modulo=dados_pessoa['modulo'])

def retorna_pessoas():

    return db.query("SELECT * FROM pessoa")
```