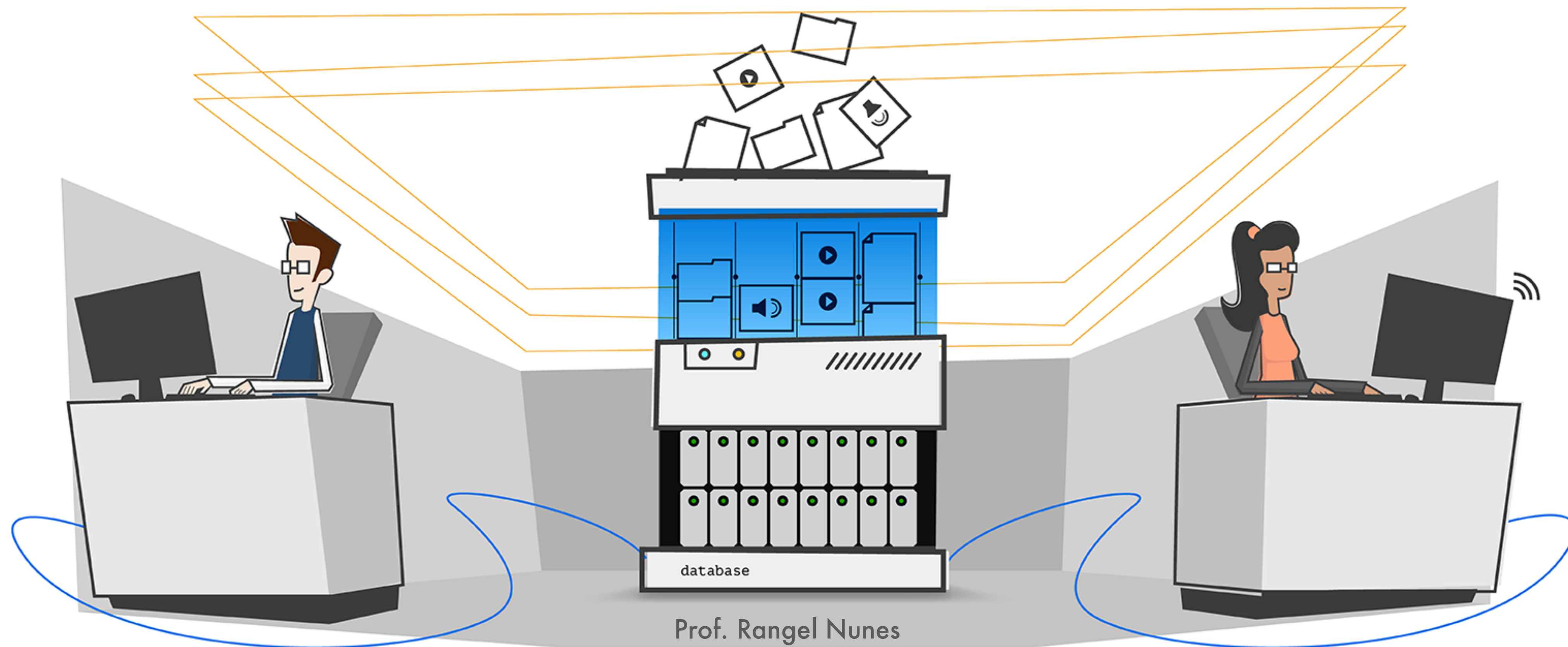


APLICAÇÃO COM ACESSO A BANCO DE DADOS



Prof. Rangel Nunes

PYTHON + POSTGRESQL



Prof. Rangel Nunes

PSYCOPG2

Psycopg2 é o adaptador de banco de dados PostgreSQL mais popular para a linguagem de programação Python



<https://www.psycopg.org/docs/install.html#quick-install>

CONFIGURAÇÃO DO AMBIENTE VIRTUAL E INSTALAÇÃO DO PSYCOPG2

Criando um ambiente virtual



```
python -m venv nome_do_ambiente
```

Para o nosso projeto



```
python -m venv .venv
```

CONFIGURAÇÃO DO AMBIENTE VIRTUAL E INSTALAÇÃO DO PSYCOPG2

Ativando o ambiente virtual no **Linux** ou **Mac**



```
source meu_ambiente/bin/activate
```

Ativando o ambiente virtual no **Windows**



```
.\meu_ambiente\Scripts\activate
```

Ativando o ambiente do nosso projeto

Ativando o ambiente virtual no **Linux** ou **Mac**



```
source .venv/bin/activate
```

Ativando o ambiente virtual no **Windows**



```
.\.venv\Scripts\activate
```

CONFIGURAÇÃO DO AMBIENTE VIRTUAL E INSTALAÇÃO DO PSYCOPG2

Instalando o módulo *Psycpg2*



```
pip install psycpg2-binary
```

GERENCIAMENTO DAS DEPENDÊNCIAS

Congelando o ambiente virtual



```
pip freeze > requirements.txt
```

Quando precisar instalar as dependências



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


IMPLEMENTANDO A CONEXÃO PYTHON + POSTGRESQL COM PARÂMETROS DE CONEXÃO NO CÓDIGO

database.py

```
import psycopg2 ←  
  
parametros = {  
    "host": "localhost",  
    "dbname": "aula_python_postgresql",  
    "user": "postgres",  
    "password": "pgsql"  
}
```

<https://youtu.be/9ftyzmYmyJI>


IMPLEMENTANDO A CONEXÃO PYTHON + POSTGRESQL

MÉTODO CONECTA_BD

Método *conecta_bd()*

```
• • •  
  
# database.py  
  
def conecta_bd( ):  
    conexao = None  
    try:  
        conexao = psycopg2.connect(**parametros)  
        print('conexao realizada com sucesso!')  
    except Exception as erro:  
        print('[ERRO]:', erro)  
    return conexao
```

TABELA QUE SERÁ ACESSADA

curso	
 id_curso	int
nome	varchar(30)
sigla	char(4)
turno	enum

CRIANDO A TABELA CURSO NO POSTGRESQL: DIRETO NO SGDB



```
create table if not exists curso(  
    id_curso serial primary key,  
    nome varchar(60) not null unique,  
    sigla char(5) not null,  
    turno varchar(40) check(turno in ('manhã', 'tarde', 'noite'))  
);
```

CRIANDO A TABELA CURSO NO POSTGRESQL: VIA DATABASE.PY



```
# database.py
```

```
CRIA_TABELA_CURSO = """
    create table if not exists curso(
        id_curso serial primary key,
        nome varchar(60) not null unique,
        sigla char(5) not null,
        turno varchar(40) check(turno in ('manhã', 'tarde', 'noite'))
    );
"""
```

CRIANDO A TABELA CURSO NO POSTGRESQL: VIA DATABASE.PY



```
# database.py

def cria_tabelas(conexao):
    try:
        with conexao:
            with conexao.cursor() as cursor:
                cursor.execute(CRIA_TABELA_CURSO)
                print('Tabela criada com sucesso!')
                conexao.commit()
    except Exception as erro:
        print('[ERRO] - ', erro)
```

CRIANDO A TABELA **CURSO** NO POSTGRESQL: VIA DATABASE.PY, COM ARQUIVO SQL

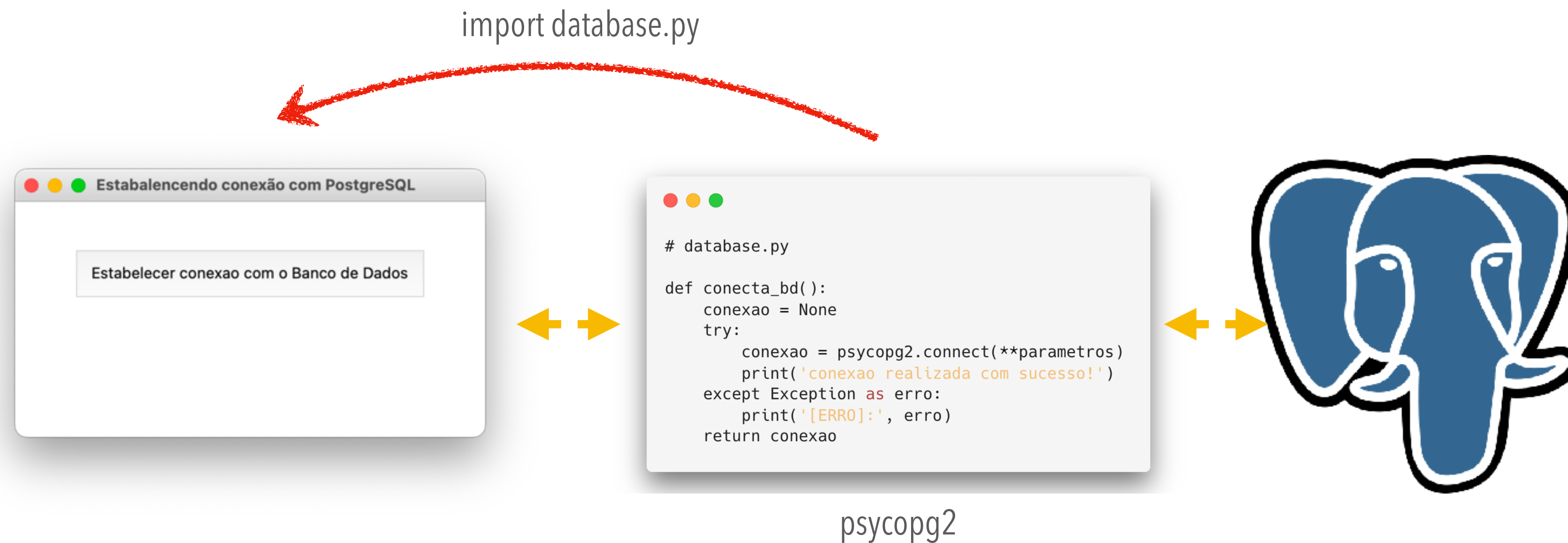


```
# database.py

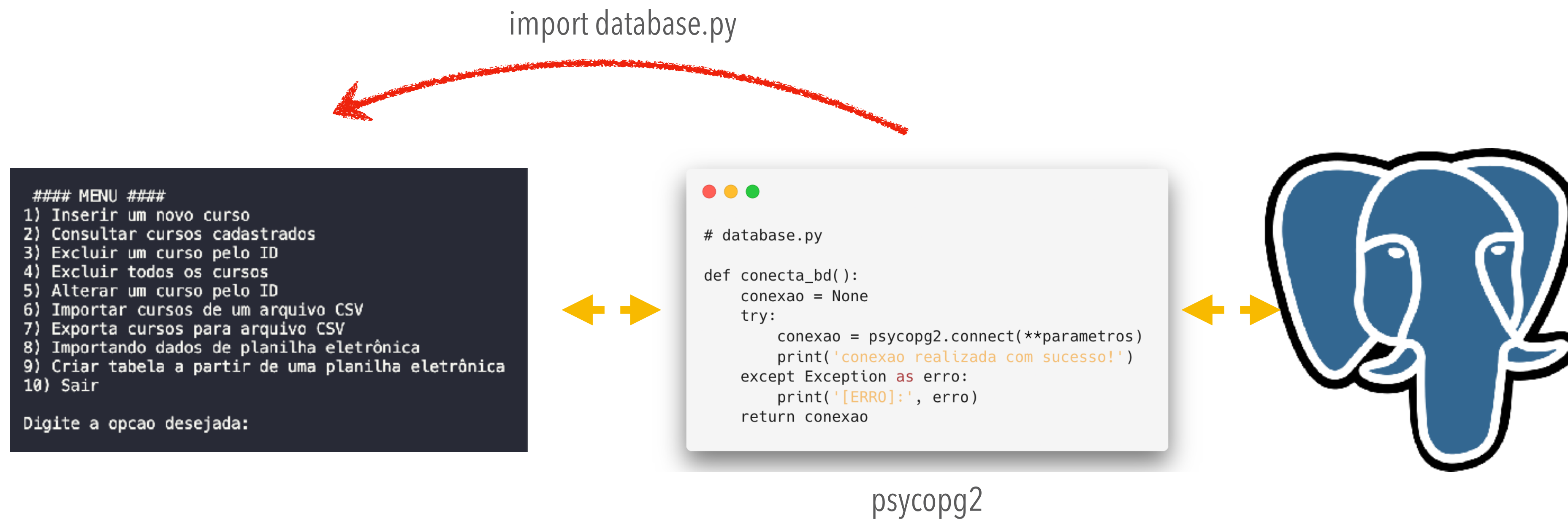
arquivo_sql = open('script_criacao_tabelas.sql', 'r')

def cria_tabelas(conexao, arquivo_sql):
    try:
        with conexao:
            with conexao.cursor() as cursor:
                cursor.execute(arquivo_sql.read())
                print( 'Tabela criada com sucesso!')
                conexao.commit()
    except Exception as erro:
        print(' [ERRO] - ', erro)
```

CRIANDO UMA APLICAÇÃO EM PYTHON PARA CONECTAR-SE AO BANCO DE DADOS



CRIANDO UMA APLICAÇÃO EM PYTHON PARA CONECTAR-SE AO BANCO DE DADOS



CRIANDO UMA APLICAÇÃO EM PYTHON PARA CONECTAR-SE AO BANCO DE DADOS



```
# main.py
```

```
import database as db
```

```
MENU_DE_OPCOES = """ ##### MENU #####
```

```
1) Inserir um novo curso
2) Consultar cursos cadastrados
3) Excluir um curso pelo ID
4) Excluir todos os cursos
5) Alterar um curso pelo ID
6) Importar cursos de um arquivo CSV
7) Sair
"""
```

CAPTURANDO A OPÇÃO DESEJADA

```

# main.py
# import os
if __name__ == '__main__':
    while(True):
        os.system('clear')
        print(MENU_DE_OPCOES)
        opcao = None

        try:
            opcao = int(input('Digite a opcao desejada: '))
            if opcao == 1:
                pass
            elif opcao == 2:
                pass
            ...
            elif opcao == 7:
                print('Até logo!')
                if conexao:
                    conexao.close()
                break
            else:
                print('[ATENÇÃO]: Digite opções entre 1 e 5')
        except:
            print('[ERRO]: Opção inválida! Tente novamente.')
        input('Digite ENTER tecla para continuar...')
```