

Acessando BD pela aplicação

1 – Instalar biblioteca Python para conexão com BD MySQL

Acesse o terminal de seu ambiente IDE (PyCharm ou VSCode) e execute a o comando que segue:

```
pip install mysql-connector-python
```

2 - Criar conexão com o BD no WorkBench

3 – Testar Script de conexão com o BD

```
# antes, executar
# pip install mysql-connector-python
import mysql.connector

# Conecta ao servidor MySQL
conexao = mysql.connector.connect(
    host="localhost",
    user="root",
    password="senai@mange"
)

# Cria um cursor para executar comandos SQL
cursor = conexao.cursor()

# Cria um banco de dados chamado 'teste' se ele ainda não existir
cursor.execute("CREATE DATABASE IF NOT EXISTS PWBE")

# Seleciona o banco de dados 'pwbe'
cursor.execute("USE PWBE")

# Cria uma tabela chamada 'tabela_1' com os campos 'id' e 'nome'
cursor.execute("CREATE TABLE IF NOT EXISTS tabela_pwbe (id INT AUTO_INCREMENT PRIMARY KEY, nome VARCHAR(255))")

# Fecha o cursor e a conexão
cursor.close()
conexao.close()

print("Banco de dados e tabela criados com sucesso.")
```

Acessando BD pela aplicação

4 – Criar arquivo database.py

```
import mysql.connector

2 usages
def conectar():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="senai@mange",
        database="pwbe_escola"
    )
```

5 – Incluindo o novo arquivo database.py na importação

```
from database import conectar # Importa a função conectar no Banco de dados

conexao = conectar()

1 usage
class MyHandler(SimpleHTTPRequestHandler):
```

6 – Alterando a função usuario_existente para consultar BD

```
def usuario_existente(self, login, senha):  
  
    cursor = conexao.cursor()  
    cursor.execute("SELECT senha FROM dados_login WHERE login = %s", (login,))  
    resultado = cursor.fetchone()  
    cursor.close()  
  
    if resultado:  
        senha_hash = hashlib.sha256(senha.encode('utf-8')).hexdigest()  
        return senha_hash == resultado[0]  
  
    return False
```

7 – Alterando a função adicionar_usuario para usar BD

```
def adicionar_usuario(self, login, senha, nome):  
    cursor = conexao.cursor()  
  
    senha_hash = hashlib.sha256(senha.encode('utf-8')).hexdigest()  
    cursor.execute("INSERT INTO dados_login (login, senha, nome) VALUES (%s, %s, %s)", (login, senha_hash, nome))  
  
    conexao.commit()  
  
    cursor.close()
```

8 – Alterando a função da rota `/enviar_login` em `do_POST` para usar BD

```
def do_POST(self):
    if self.path == '/enviar_login':
        content_length = int(self.headers['Content-Length'])
        body = self.rfile.read(content_length).decode('utf-8')
        form_data = parse_qs(body, keep_blank_values=True)

        login = form_data.get('email', [''])[0]
        senha = form_data.get('senha', [''])[0]

        if self.usuario_existente(login, senha):
            self.send_response(200)
            self.send_header(keyword="Content-type", value="text/html; charset=utf-8")
            self.end_headers()
            mensagem = f"Usuário {login} logado com sucesso!!!"
            self.wfile.write(mensagem.encode('utf-8'))
        else:
            #Verifica se usuário já está cadastrado. Caso não esteja foi caso de login errado
            cursor = conexao.cursor()
            cursor.execute("SELECT login FROM dados_login WHERE login = %s", (login,))
            resultado = cursor.fetchone()

            if resultado:
                self.send_response(302)
                self.send_header(keyword="Location", value="/login_failed")
                self.end_headers()
                cursor.close()
                return
            else:
                self.send_response(302)
                self.send_header(keyword="Location", value=f"/cadastro?login={login}&senha={senha}")
                self.end_headers()
                cursor.close()
                return
```

Acessando BD pela aplicação

9 – Alterando a função da rota `/confirmar_cadastro` em `do_POST` para usar BD

```
elif self.path.startswith('/confirmar_cadastro'):
    content_length = int(self.headers['Content-Length'])
    body = self.rfile.read(content_length).decode('utf-8')
    form_data = parse_qs(body, keep_blank_values=True)

    login = form_data.get('login', [''])[0]
    senha = form_data.get('senha', [''])[0]
    nome = form_data.get('nome', [''])[0]

    self.adicionar_usuario(login, senha, nome)

    with open(os.path.join(os.getcwd(), 'msg_sucesso.html'), 'rb') as file:
        content = file.read().decode('utf-8')

    content = content.replace(_old: '{login}', login)
    content = content.replace(_old: '{nome}', nome)

    self.send_response(200)
    self.send_header(keyword: "Content-type", value: "text/html; charset=utf-8")
    self.end_headers()
    self.wfile.write(content.encode('utf-8'))
```