

Atividade 08

Aluno: Arthur Lima dos Santos

Prints do pwd:

```
#####
#                               #
#      WARNING!!!!             #
# This is a sandbox environment. Using personal credentials   #
# is HIGHLY! discouraged. Any consequences of doing so are    #
# completely the user's responsibilities.                       #
#                                                               #
# The PWD team.                                                 #
#####
[node1] (local) root@192.168.0.23 ~
$ docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=root -e MYSQL_DATABASE=BD_AVIOES -p 3306:3306 -d mysql:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
43759093d4f6: Pull complete
d255dceb9ed5: Pull complete
23d22e42ea50: Pull complete
431b106548a3: Pull complete
2be0d473cadf: Pull complete
```

Ativar o Windows

Acesse Configurações para ativar o Windows.

```
Status: Downloaded newer image for mysql:latest
d5f3b53164d012d74ae9bd486fc81519e825c6500c41bb52fe18951731df3846
[node1] (local) root@192.168.0.23 ~
$ docker exec -it mysql-container mysql -uroot -proot
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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.
mysql> USE BD_AVIOES
```

Ativar o Windows

Acesse Configurações para ativar o Windows.

```
mysql> USE BD_AVIOES
Database changed
mysql> CREATE TABLE TB_CLIENTES (
  -> id INT NOT NULL PRIMARY KEY,
  -> customerName VARCHAR(200) DEFAULT NULL,
  -> contactLastName VARCHAR(100) DEFAULT NULL,
  -> contactFirstName VARCHAR(100) DEFAULT NULL,
  -> phone VARCHAR(20) DEFAULT NULL,
  -> addressLine1 VARCHAR(40) DEFAULT NULL,
  -> addressLine2 VARCHAR(40) DEFAULT NULL,
  -> city VARCHAR(80) DEFAULT NULL,
  -> state VARCHAR(80) DEFAULT NULL,
  -> country VARCHAR(80) DEFAULT NULL,
  -> postalCode VARCHAR(80) DEFAULT NULL,
  -> salesRepEmployeeNumber VARCHAR(40) DEFAULT NULL,
  -> creditLimit INT DEFAULT NULL );
Query OK, 0 rows affected (0.02 sec)
```

Ativar o Windows

Acesse Configurações para ativar o Windows.

```
mysql> CREATE TABLE TB_DETALHES_PEDIDOS (
  -> id_orderNumber INT NOT NULL,
  -> id_productCode INT NOT NULL,
  -> quantityOrdered INT DEFAULT NULL,
  -> priceEach INT DEFAULT NULL,
  -> orderLineNumber VARCHAR(20) DEFAULT NULL);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE TB_EMPREGADOS (
  -> id INT NOT NULL,
  -> lastName VARCHAR(80) DEFAULT NULL,
  -> firstName VARCHAR(20) DEFAULT NULL,
  -> extension VARCHAR(40) DEFAULT NULL,
  -> email VARCHAR(80) DEFAULT NULL,
  -> officeCode VARCHAR(20) DEFAULT NULL,
  -> reportsTo VARCHAR(20) DEFAULT NULL,
  -> jobTittle VARCHAR(40) DEFAULT NULL );
Query OK, 0 rows affected (0.02 sec)
```

Ativar o Windows

Acesse Configurações para ativar o Windows.

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE TB_ESCRITORIOS (
  -> id INT NOT NULL,
  -> city VARCHAR(100) DEFAULT NULL,
  -> state VARCHAR(100) DEFAULT NULL,
  -> country VARCHAR(100) DEFAULT NULL,
  -> phone VARCHAR(20) DEFAULT NULL,
  -> addressLine1 VARCHAR(40) DEFAULT NULL,
  -> addressLine2 VARCHAR(40) DEFAULT NULL,
  -> postalCode VARCHAR(40) DEFAULT NULL,
  -> territory VARCHAR(60) DEFAULT NULL );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE TB_LINHAS_DE_PRODUTOS (
  -> id INT NOT NULL,
  -> htmlDescription TEXT,
  -> image VARCHAR(100) DEFAULT NULL,
```

Ativar o Windows

Acesse Configurações para ativar o Windows.

```

-> textDescription VARCHAR(100) DEFAULT NULL );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE TB_PAGAMENTOS (
-> id INT NOT NULL,
-> id_customerNumber VARCHAR(100) DEFAULT NULL,
-> paymentDate VARCHAR(20) DEFAULT NULL,
-> amount INT DEFAULT NULL );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE TB_PEDIDOS (
-> id INT NOT NULL,
-> id_customerNumber VARCHAR(100) DEFAULT NULL,
-> orderDate VARCHAR(20) DEFAULT NULL,
-> requiredDate VARCHAR(20) DEFAULT NULL,
-> shippedDate VARCHAR(20) DEFAULT NULL,
-> status VARCHAR(40) DEFAULT NULL,
-> comments TEXT );

```

Ativar o Windows

Acesse Configurações para ativar o Windows.

```

-> shippedDate VARCHAR(20) DEFAULT NULL,
-> status VARCHAR(40) DEFAULT NULL,
-> comments TEXT );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE TB_PRODUTOS (
-> id INT NOT NULL,
-> productName VARCHAR(100) DEFAULT NULL,
-> id_productLine INT DEFAULT NULL,
-> productScale VARCHAR(100) DEFAULT NULL,
-> productVendor VARCHAR(100) DEFAULT NULL,
-> productDescription TEXT,
-> quantityInStock INT DEFAULT NULL,
-> buyPrice INT DEFAULT NULL,
-> MSRP INT DEFAULT NULL );
Query OK, 0 rows affected (0.02 sec)

mysql> 

```

Ativar o Windows

Acesse Configurações para ativar o Windows.

Fazendo a integração com python:

```

# The PWD team.
#####
[node2] (local) root@192.168.0.22 ~
$ python -m venv myenv
[node2] (local) root@192.168.0.22 ~
$ source myenv/bin/activate
(myenv) [node2] (local) root@192.168.0.22 ~
$ pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.2.0-py2.py3-none-any.whl.metadata (6.0 kB)
  Downloading mysql_connector_python-9.2.0-py2.py3-none-any.whl (398 kB)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.2.0

[notice] A new release of pip is available: 24.2 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
(myenv) [node2] (local) root@192.168.0.22 ~
$ 

```

Ativar o Windows

Acesse Configurações para ativar o Windows.

Script.py (VS Code):

```

1 import mysql.connector
2 from mysql.connector import Error
3
4 def create_connection():
5     """Cria uma conexão com o banco de dados MySQL."""
6     connection = None
7     try:
8         connection = mysql.connector.connect(
9             host='192.168.0.18',
10             port='3306',
11             user='root',
12             password='root',
13             database='BD_AVIOES'
14         )
15         print("Conexão com o MySQL bem-sucedida")
16     except Error as e:
17         print(f"Erro '{e}' ocorreu")
18
19     return connection
20
21 def create_cliente(connection, id, customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, country, postalCode):
22     """Insere um novo cliente na tabela TB_CLIENTES."""
23     cursor = connection.cursor()
24     query = "INSERT INTO TB_CLIENTES (id, customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, country, postalCode) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"
25     cursor.execute(query, (id, customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, country, postalCode))
26     connection.commit()
27     print("Cliente adicionado com sucesso")
28
29 def read_clientes(connection):
30     """Listar todos os clientes da tabela TB_CLIENTES."""
31     cursor = connection.cursor()

```

Ativar o Windows
Acesse Configurações para ativar

```

29 def read_clientes(connection):
30     cursor = connection.cursor()
31     cursor.execute("SELECT * FROM TB_CLIENTES")
32     clientes = cursor.fetchall()
33     for cliente in clientes:
34         print(cliente)
35
36
37 def update_cliente(connection, id, customerName, phone):
38     """Atualiza um cliente existente na tabela TB_CLIENTES."""
39     cursor = connection.cursor()
40     query = "UPDATE TB_CLIENTES SET customerName = %s, phone = %s WHERE id = %s"
41     cursor.execute(query, (customerName, phone, id))
42     connection.commit()
43     print("Cliente atualizado com sucesso")
44
45 def delete_cliente(connection, id):
46     """Deleta um cliente da tabela TB_CLIENTES."""
47     cursor = connection.cursor()
48     query = "DELETE FROM TB_CLIENTES WHERE id = %s"
49     cursor.execute(query, (id,))
50     connection.commit()
51     print("Cliente deletado com sucesso")
52
53 def create_detalhe_pedido(connection, id_orderNumber, id_productCode, quantityOrdered, priceEach, orderLineNumber):
54     """Insere um novo detalhe pedido na tabela TB_DETALHES_PEDIDOS."""
55     cursor = connection.cursor()
56     query = "INSERT INTO TB_DETALHES_PEDIDOS (id_orderNumber, id_productCode, quantityOrdered, priceEach, orderLineNumber) VALUES (%s, %s, %s, %s, %s)"
57     cursor.execute(query, (id_orderNumber, id_productCode, quantityOrdered, priceEach, orderLineNumber))
58     connection.commit()
59     print("Detalhe pedido adicionado com sucesso")
60

```

Ativar o Windows
Acesse Configurações para ativar

```

60
61 def read_detalhe_pedido(connection):
62     """Listar todos os detalhes da tabela TB_DETALHES_PEDIDOS."""
63     cursor = connection.cursor()
64     cursor.execute("SELECT * FROM TB_DETALHES_PEDIDOS")
65     detalhes = cursor.fetchall()
66     for detalhe in detalhes:
67         print(detalhe)
68
69 def update_detalhe_pedido(connection, id_orderNumber, priceEach, quantityOrdered):
70     """Atualiza um cliente existente na tabela TB_CLIENTES."""
71     cursor = connection.cursor()
72     query = "UPDATE TB_DETALHES_PEDIDOS SET priceEach = %s, quantityOrdered = %s WHERE id_orderNumber = %s"
73     cursor.execute(query, (priceEach, quantityOrdered, id_orderNumber))
74     connection.commit()
75     print("Pedido atualizado com sucesso")
76
77 def delete_detalhe_pedido(connection, id_orderNumber):
78     """Deleta um cliente da tabela TB_DETALHES_PEDIDOS."""
79     cursor = connection.cursor()
80     query = "DELETE FROM TB_DETALHES_PEDIDOS WHERE id_orderNumber = %s"
81     cursor.execute(query, (id_orderNumber,))
82     connection.commit()
83     print("Detalhe pedido deletado com sucesso")
84
85
86 def create_empregados(connection, id, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle):
87     """Insere um novo detalhe pedido na tabela TB_EMPREGADOS."""
88     cursor = connection.cursor()
89     query = "INSERT INTO TB_EMPREGADOS (id, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
90     cursor.execute(query, (id, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle))

```

Ativar o Windows
Acesse Configurações para ativar

```

91     connection.commit()
92     print("Empregado adicionado com sucesso")
93
94 def read_empregados(connection):
95     """Listar todos os detalhes da tabela TB_EMPREGADOS."""
96     cursor = connection.cursor()
97     cursor.execute("SELECT * FROM TB_EMPREGADOS")
98     empregados = cursor.fetchall()
99     for empregado in empregados:
100         print(empregado)
101
102 def update_empregado(connection, id, firstName, lastName):
103     """Atualiza um cliente existente na tabela TB_EMPREGADOS."""
104     cursor = connection.cursor()
105     query = "UPDATE TB_EMPREGADOS SET firstName = %s, lastName = %s WHERE id = %s"
106     cursor.execute(query, (firstName, lastName, id))
107     connection.commit()
108     print("Empregado atualizado com sucesso")
109
110 def delete_empregado(connection, id):
111     """Deleta um cliente da tabela TB_EMPREGADOS."""
112     cursor = connection.cursor()
113     query = "DELETE FROM TB_EMPREGADOS WHERE id = %s"
114     cursor.execute(query, (id,))
115     connection.commit()
116     print("Empregado deletado com sucesso")
117
118 def create_escritorios(connection, id, city, state, country, phone, addressLine1, addressLine2, postalCode, territory):
119     """Insere um novo detalhe pedido na tabela TB_ESCRITORIOS."""

```

Ativar o Windows
Acesse Configurações para ativar


```

120     cursor = connection.cursor()
121     query = "INSERT INTO TB_ESCRITORIOS (id, city,state,country,phone,addressLine1,addressLine2, postalCode, territory) VALUES (%s, %s, %s, %s,%s,%s,%s,%s,%s)"
122     cursor.execute(query, (id, city,state,country,phone, addressLine1, addressLine2, postalCode, territory))
123     connection.commit()
124     print("Escritorio adicionado com sucesso")
125
126 def read_escritorios(connection):
127     """Retorna todos os detalhes da tabela TB_ESCRITORIOS."""
128     cursor = connection.cursor()
129     cursor.execute("SELECT * FROM TB_ESCRITORIOS")
130     escritorios = cursor.fetchall()
131     for escritorio in escritorios:
132         print(escritorio)
133
134 def update_escritorios(connection, id, city, phone):
135     """Atualiza um cliente existente na tabela TB_ESCRITORIOS."""
136     cursor = connection.cursor()
137     query = "UPDATE TB_ESCRITORIOS SET city = %s, phone = %s WHERE id = %s"
138     cursor.execute(query, (city, phone, id))
139     connection.commit()
140     print("Escritorio atualizado com sucesso")
141
142 def delete_escritorios(connection, id):
143     """Deleta um cliente da tabela TB_ESCRITORIOS."""
144     cursor = connection.cursor()
145     query = "DELETE FROM TB_ESCRITORIOS WHERE id = %s"
146     cursor.execute(query, (id,))
147     connection.commit()
148     print("Escritorio deletado com sucesso")
149

```

Ativar o Windows
Acesse Configurações para ativar

```

150 def create_linhas_produto(connection, id, htmlDescription,image,textDescription):
151     """Insere um novo detalhe pedido na tabela TB_LINHAS_DE_PRODUTO."""
152     cursor = connection.cursor()
153     query = "INSERT INTO TB_LINHAS_DE_PRODUTO (id, htmlDescription,image,textDescription) VALUES (%s, %s, %s, %s)"
154     cursor.execute(query, (id, htmlDescription,image,textDescription))
155     connection.commit()
156     print("linhas de produto adicionado com sucesso")
157
158 def read_linhas_produto(connection):
159     """Retorna todos os detalhes da tabela TB_LINHAS_DE_PRODUTO."""
160     cursor = connection.cursor()
161     cursor.execute("SELECT * FROM TB_LINHAS_DE_PRODUTO")
162     linhas = cursor.fetchall()
163     for linha in linhas:
164         print(linha)
165
166 def update_linhas_produto(connection, id, htmlDescription, image):
167     """Atualiza um cliente existente na tabela TB_LINHAS_DE_PRODUTO."""
168     cursor = connection.cursor()
169     query = "UPDATE TB_LINHAS_DE_PRODUTO SET htmlDescription = %s, image = %s WHERE id = %s"
170     cursor.execute(query, (htmlDescription, image, id))
171     connection.commit()
172     print("Linhas atualizado com sucesso")
173
174 def delete_linhas_produto(connection, id):
175     """Deleta um cliente da tabela TB_LINHAS_DE_PRODUTO."""
176     cursor = connection.cursor()
177     query = "DELETE FROM TB_LINHAS_DE_PRODUTO WHERE id = %s"
178     cursor.execute(query, (id,))
179     connection.commit()

```

Ativar o Windows
Acesse Configurações para ativar

```

179     connection.commit()
180     print("Linhas deletado com sucesso")
181
182 def create_pagamentos(connection, id, id_costumerNumber, paymentDate, amount):
183     """Insere um novo detalhe pedido na tabela TB_PAGAMENTOS."""
184     cursor = connection.cursor()
185     query = "INSERT INTO TB_PAGAMENTOS (id, id_costumerNumber, paymentDate, amount) VALUES (%s, %s, %s, %s)"
186     cursor.execute(query, (id, id_costumerNumber, paymentDate, amount))
187     connection.commit()
188     print("pagamentos de produto adicionado com sucesso")
189
190 def read_pagamentos(connection):
191     """Listar todos os detalhes da tabela TB_PAGAMENTOS."""
192     cursor = connection.cursor()
193     cursor.execute("SELECT * FROM TB_PAGAMENTOS")
194     pagamentos = cursor.fetchall()
195     for pagamento in pagamentos:
196         print(pagamento)
197
198 def update_pagamentos(connection, id, paymentDate, amount):
199     """Atualiza um cliente existente na tabela TB_PAGAMENTOS."""
200     cursor = connection.cursor()
201     query = "UPDATE TB_PAGAMENTOS SET paymentDate = %s, amount = %s WHERE id = %s"
202     cursor.execute(query, (paymentDate, amount, id))
203     connection.commit()
204     print("pagamento atualizado com sucesso")
205
206 def delete_pagamentos(connection, id):
207     """Deleta um cliente da tabela TB_PAGAMENTOS."""
208     cursor = connection.cursor()

```

Ativar o Windows
Acesse Configurações para ativar

```

211     connection.commit()
212     print("pagamento deletado com sucesso")
213
214 def create_pedidos(connection, id, id_costumerNumber, orderDate, requiredDate, shippedDate, status, comments):
215     """Insere um novo detalhe pedido na tabela TB_PEDIDOS."""
216     cursor = connection.cursor()
217     query = "INSERT INTO TB_PEDIDOS (id, id_costumerNumber, orderDate, requiredDate, shippedDate, status, comments) VALUES (%s, %s, %s, %s, %s, %s, %s)"
218     cursor.execute(query, (id, id_costumerNumber, orderDate, requiredDate, shippedDate, status, comments))
219     connection.commit()
220     print("pedido de produto adicionado com sucesso")
221
222 def read_pedidos(connection):
223     """Listar todos os detalhes da tabela TB_PEDIDOS."""
224     cursor = connection.cursor()
225     cursor.execute("SELECT * FROM TB_PEDIDOS")
226     pedidos = cursor.fetchall()
227     for pedido in pedidos:
228         print(pedido)
229
230 def update_pedidos(connection, id, orderDate, status):
231     """Atualiza um cliente existente na tabela TB_PEDIDOS."""
232     cursor = connection.cursor()
233     query = "UPDATE TB_PEDIDOS SET orderDate = %s, status = %s WHERE id = %s"
234     cursor.execute(query, (orderDate, status, id))
235     connection.commit()
236     print("pedido atualizado com sucesso")
237
238 def delete_pedidos(connection, id):
239     """Deleta um cliente da tabela TB_PEDIDOS."""
240     cursor = connection.cursor()

```

Ativar o Windows
Acesse Configurações para ativar

```

240     cursor = connection.cursor()
241     query = "DELETE FROM TB_PEDIDOS WHERE id = %s"
242     cursor.execute(query, (id,))
243     connection.commit()
244     print("pedido deletado com sucesso")
245
246 def create_produtos(connection, id, productName, id_productLine, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP):
247     """Insere um novo detalhe pedido na tabela TB_PRODUTOS."""
248     cursor = connection.cursor()
249     query = "INSERT INTO TB_PRODUTOS (id, productName, id_productLine, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP)"
250     cursor.execute(query, (id, productName, id_productLine, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP))
251     connection.commit()
252     print("produto adicionado com sucesso")
253
254 def read_produtos(connection):
255     """Lê todos os detalhes da tabela TB_PRODUTOS."""
256     cursor = connection.cursor()
257     cursor.execute("SELECT * FROM TB_PRODUTOS")
258     produtos = cursor.fetchall()
259     for produto in produtos:
260         print(produto)
261
262 def update_produtos(connection, id, productName, buyPrice):
263     """Atualiza um cliente existente na tabela TB_PRODUTOS."""
264     cursor = connection.cursor()
265     query = "UPDATE TB_PRODUTOS SET productName = %s, buyPrice = %s WHERE id = %s"
266     cursor.execute(query, (productName, buyPrice, id))
267     connection.commit()
268     print("produto atualizado com sucesso")
269

```

Ativar o Windows
Acesse Configurações para ativar

```

270 def delete_produtos(connection, id):
271     """Deleta um cliente da tabela TB_PRODUTOS."""
272     cursor = connection.cursor()
273     query = "DELETE FROM TB_PRODUTOS WHERE id = %s"
274     cursor.execute(query, (id,))
275     connection.commit()
276     print("produto deletado com sucesso")
277
278
279 def main():
280     connection = create_connection()
281     if connection is None:
282         return
283
284     # Exemplo de uso das funções CRUD
285     create_cliente(connection, 1, 'Empresa A', 'Santos', 'Arthur', '8599999999', 'Rua 6', 'Casa 300', 'Fortaleza', 'CE', 'Brazil', '12243-987', N
286     read_clientes(connection)
287     update_cliente(connection, 1, 'Empresa A & CIA', '77777777')
288     read_clientes(connection)
289     delete_cliente(connection, 1)
290     read_clientes(connection)
291
292     #Crud em TB_DETALHES_PEDIDOS
293     create_detalhe_pedido(connection, 1, 23, 10, 5, '6')
294     read_detalhe_pedido(connection)
295     update_detalhe_pedido(connection, 1, 22, 63)
296     read_detalhe_pedido(connection)
297     delete_detalhe_pedido(connection, 1)
298     read_detalhe_pedido(connection)
299
300

```

Ativar o Windows
Acesse Configurações para ativar


```

#Crud em TB_EMPREGADOS
create_empregados(connection, 1, 'Santos', 'Arthur', '2014', 'arthursantos@hotmail.com', '12345', 'None', 'Gerente')
read_empregados(connection)
update_empregado(connection, 1, 'Arthur', 'dos Santos')
read_empregados(connection)
delete_empregado(connection, 1)
read_empregados(connection)

#Crud em TB_ESCRITORIOS
create_escritorios(connection, 1, 'Xique-xique', 'Bahia', 'Bazil', '8512345678', 'Rua Xique', 'Bairro Xique', '62670000', 'Cidade')
read_escritorios(connection)
update_escritorios(connection, 1, 'Fortaleza', '2222')
read_escritorios(connection)
delete_escritorios(connection, 1)
read_escritorios(connection)

#Crud em TB_LINHAS_DE_PRODUTO
create_linhas_produto(connection, 1, '<p>Aviao preto</p>', 'Aviao preto.png', 'Aviao preto')
read_linhas_produto(connection)
update_linhas_produto(connection, 1, 'Nova descricao', 'Motor')
read_linhas_produto(connection)
delete_linhas_produto(connection, 1)
read_linhas_produto(connection)

#Crud em TB_PAGAMENTOS
create_pagamentos(connection, 1, 100, '26/02/2025', 10)
read_pagamentos(connection)
update_pagamentos(connection, 1, '27/02/2025', 25)
read_pagamentos(connection)

```

Ativar o Windows
Acesse Configurações para ativa

```

324 #Crud em TB_PAGAMENTOS
325 create_pagamentos(connection, 1, 100, '26/02/2025', 10)
326 read_pagamentos(connection)
327 update_pagamentos(connection, 1, '27/02/2025', 25)
328 read_pagamentos(connection)
329 delete_pagamentos(connection, 1)
330 read_pagamentos(connection)
331
332 #Crud em TB_PEDIDOS
333 create_pedidos(connection, 1, '1234156', '10/01/2026', '25/01/2025', '26/02/2025', 'A caminho', 'Fragil')
334 read_pedidos(connection)
335 update_pedidos(connection, 1, '2/02/2025', 'De boa')
336 read_pedidos(connection)
337 delete_pedidos(connection, 1)
338 read_pedidos(connection)
339
340 #Crud em TB_PRODUTOS
341 create_produtos(connection, 1, 'Headset', 40, '1:30', 'Bom Som', 'Headset de qualidade', 2, 100, 120)
342 read_produtos(connection)
343 update_produtos(connection, 1, 'Head', 150)
344 read_produtos(connection)
345 delete_produtos(connection, 1)
346 read_produtos(connection)
347
348 connection.close()
349
350 if __name__ == "__main__":
351     main()

```

Ativar o Windows
Acesse Configurações para ativa