Criação do banco de dados e tabelas:

```
############################################################
#                      WARNING!!!!                        #
# This is a sandbox environment. Using personal credentials  #
# is HIGHLY! discouraged. Any consequences of doing so are   #
# completely the user's responsibilites.                  #
#                                                          #
# 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
f56a22f949f9: Pull complete
277ab5f6ddde: Pull complete
df1ba1ac457a: Pull complete
cc9646b08259: Pull complete
893b018337e2: Pull complete
Digest: sha256:146682692a3aa409eae7b7dc6a30f637c6cb49b6ca901c2cd160becc81127d3b
Status: Downloaded newer image for mysql:latest
4cfa605cf1dd19415f7219632bcca7b718daed9dbbc033525113c486c433adec
[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.
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)
[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
Database changed
mysql> CREATE TABLE `TB_CLIENTES` (
    ->   `id` varchar(80) NOT NULL PRIMARY KEY,
    ->   `customerName` varchar(100) DEFAULT NULL,
    ->   `contactLastName` varchar(60) DEFAULT NULL,
    ->   `contactFirstName` varchar(40) 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,
    ->   `postalCode` varchar(60) DEFAULT NULL,
    ->   `country` varchar(80) DEFAULT NULL,
    ->   `salesRepEmployeeNumber` varchar(40) DEFAULT NULL,
    ->   `creditLimit` int DEFAULT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)

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(40) DEFAULT NULL
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE `TB_EMPREGADOS` (
    ->   `id` int NOT NULL,
    ->   `lastName` varchar(80) DEFAULT NULL,
```

```
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(40) DEFAULT NULL
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE `TB_EMPREGADOS` (
    ->   `id` int NOT NULL,
    ->   `lastName` varchar(80) DEFAULT NULL,
    ->   `firstName` varchar(40) DEFAULT NULL,
    ->   `extension` varchar(20) DEFAULT NULL,
    ->   `email` varchar(60) DEFAULT NULL,
    ->   `officeCode` varchar(40) DEFAULT NULL,
    ->   `reportsTo` varchar(40) DEFAULT NULL,
    ->   `jobTittle` varchar(40) DEFAULT NULL
    -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE `TB_ESCRITORIOS` (
    ->   `id` int NOT NULL,
    ->   `city` varchar(100) DEFAULT NULL,
    ->   `phone` varchar(40) DEFAULT NULL,
    ->   `addresLine1` varchar(60) DEFAULT NULL,
    ->   `addresLine2` varchar(60) DEFAULT NULL,
    ->   `state` varchar(80) DEFAULT NULL,
    ->   `country` varchar(80) DEFAULT NULL,
    ->   `postalCode` varchar(40) DEFAULT NULL,
    ->   `territory` varchar(40) DEFAULT NULL
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> CREATE TABLE `TB_LINHAS_DE_PRODUTO` (
    ->   `id` int NOT NULL,
    ->   `htmlDescription` text,
```

Criando ambiente virtual em python (segunda instância) e o script:

```
###########################################################
#                        WARNING!!!!                      #
# This is a sandbox environment. Using personal credentials  #
# is HIGHLY! discouraged. Any consequences of doing so are   #
# completely the user's responsibilites.                     #
#                                                            #
# 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 ~
$ vi app.py
(myenv) [node2] (local) root@192.168.0.22 ~
$ python app.py
Conexão com o MySQL bem-sucedida
Cliente adicionado com sucesso
('1', 'Empresa XYZ', 'Silva', 'João', '99999999', 'Rua A', 'Apto 1', 'São Paulo', 'SP', '12345-678', 'Brasil', None, 50000)
Cliente atualizado com sucesso
('1', 'Empresa XYZ Ltda', 'Silva', 'João', '77777777', 'Rua A', 'Apto 1', 'São Paulo', 'SP', '12345-678', 'Brasil', None, 50000)
Cliente deletado com sucesso
Detalhe pedido adicionado com sucesso
(1, 1, 2, 20, '2')
Pedido atualizado com sucesso
(1, 1, 33, 1, '2')
Detalhe pedido deletado com sucesso
```

```
Cliente deletado com sucesso
Detalhe pedido adicionado com sucesso
(1, 1, 2, 20, '2')
Pedido atualizado com sucesso
(1, 1, 33, 1, '2')
Detalhe pedido deletado com sucesso
Empregado adicionado com sucesso
(1, 'Lívia', 'Ana', '9999', 'ana.candido09@aluno.ifce.edu.br', '11111', 'None', 'Dono')
Empregado atualizado com sucesso
(1, 'Carneiro', 'Candido', '9999', 'ana.candido09@aluno.ifce.edu.br', '11111', 'None', 'Dono')
Empregado deletado com sucesso
Escritorio adicionado com sucesso
(1, 'Cascavel', '23232', 'Rio Novo', '213', 'CE', 'BR', '62850000', 'Cidade')
Escritorio atualizado com sucesso
(1, 'Fortaleza', '2222', 'Rio Novo', '213', 'CE', 'BR', '62850000', 'Cidade')
Escritorio deletado com sucesso
linhas de produto adicionado com sucesso
(1, '<p>Uno Branco</p>', 'Uno Branco.png', 'Uno Branco')
Linhas atualizado com sucesso
(1, 'Nova desc', 'Tecnology', 'Uno Branco')
Linhas deletado com sucesso
pagamentos de produto adicionado com sucesso
(1, '1', '1/01/2025', 20)
pagamento atualizado com sucesso
(1, '1', '2/02/2025', 30)
pagamento deletado com sucesso
pedido de produto adicionado com sucesso
(1, '1/01/2025', '20/01/2026', '2/02/2025', 'indo', 'Urgente', '1')
pedido atualizado com sucesso
(1, '2/02/2025', '20/01/2026', '2/02/2025', 'De boa', 'Urgente', '1')
pedido deletado com sucesso
produto adicionado com sucesso
(1, 'Mouse', 1, '1:10', 'Razer', 'Mouse gamer', 1, 20, 150)
produto atualizado com sucesso
(1, 'Musse', 1, '1:10', 'Razer', 'Mouse gamer', 1, 3000, 150)
produto deletado com sucesso
(myenv) [node2] (local) root@192.168.0.22 ~
$
```

Vi app.py:

```python
 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='192.168.0.23',
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, postalCode, country, salesRepEmpl
oyeeNumber, creditLimit):
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, postalCode, country, sales
RepEmployeeNumber, creditLimit) VALUES (%s, %s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
25     cursor.execute(query, (id, customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, postalCode, country, salesRepEmployee
Number, creditLimit))
26     connection.commit()
27     print("Cliente adicionado com sucesso")
28
29 def read_clientes(connection):
30     """Lê todos os clientes da tabela TB_CLIENTES."""
31     cursor = connection.cursor()
32     cursor.execute("SELECT * FROM TB_CLIENTES")
33     clientes = cursor.fetchall()
34     for cliente in clientes:
```
"app.py" 351L, 15121B   1,1   Top

```python
34     for cliente in clientes:
35         print(cliente)
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
61 def read_detalhe_pedido(connection):
62     """Lê 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."""
```
                                                                                                                                34,1   10%

```python
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))
91     connection.commit()
92     print("Empregado adicionado com sucesso")
93
94 def read_empregados(connection):
95     """Lê 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))
```
                                                                                                                               106,1   21%

```python
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,phone, addresLine1, addresLine2, state, country, postalCode, territory):
119     """Insere um novo detalhe pedido na tabela TB_ESCRITORIOS."""
120     cursor = connection.cursor()
121     query = "INSERT INTO TB_ESCRITORIOS (id, city,phone,addresLine1,addresLine2, state, country, postalCode, territory) VALUES (%s, %s, %s, %s,%s,%s, %s, %s, %s)"
122     cursor.execute(query, (id, city,phone,addresLine1,addresLine2, state, country, postalCode, territory))
123     connection.commit()
124     print("Escritorio adicionado com sucesso")
125
126 def read_escritorios(connection):
127     """Lê 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):
                                                                    111,1        33%
```

```python
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
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     """Lê 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,))
                                                                    178,1        44%
```

```python
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     """Lê 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()
209     query = "DELETE FROM TB_PAGAMENTOS WHERE id = %s"
210     cursor.execute(query, (id,))
211     connection.commit()
212     print("pagamento deletado com sucesso")
213
214 def create_pedidos(connection, id, orderDate,requiredDate,shippedDate, status, comments, id_costumerNumber):
215     """Insere um novo detalhe pedido na tabela TB_PEDIDOS."""
                                                                    215,1        56%
```

```python
215        """Insere um novo detalhe pedido na tabela TB_PEDIDOS."""
216        cursor = connection.cursor()
217        query = "INSERT INTO TB_PEDIDOS (id, orderDate,requiredDate,shippedDate, status, comments, id_costumerNumber) VALUES (%s, %s, %s, %s, %s, %s, %s)"
218        cursor.execute(query, (id, orderDate,requiredDate,shippedDate, status, comments, id_costumerNumber))
219        connection.commit()
220        print("pedido de produto adicionado com sucesso")
221
222 def read_pedidos(connection):
223        """Lê 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()
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) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)"
250        cursor.execute(query, (id, productName,id_productLine,productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP))
                                                                                        250,1          67%
```

```python
286        read_clientes(connection)
287        update_cliente(connection, 1, 'Empresa XYZ Ltda', '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, 1, 2, 20, '2')
294        read_detalhe_pedido(connection)
295        update_detalhe_pedido(connection, 1, 1, 33)
296        read_detalhe_pedido(connection)
297        delete_detalhe_pedido(connection, 1)
298        read_detalhe_pedido(connection)
299
300        #Crud em TB_EMPREGADOS
301        create_empregados(connection, 1, 'Livia', 'Ana', '9999', 'ana.candido09@aluno.ifce.edu.br', '11111', 'None', 'Dono')
302        read_empregados(connection)
303        update_empregado(connection, 1, 'Candido', 'Carneiro')
304        read_empregados(connection)
305        delete_empregado(connection, 1)
306        read_empregados(connection)
307
308        #Crud em TB_ESCRITORIOS
309        create_escritorios(connection, 1, 'Cascavel', '23232', 'Rio Novo', '213', 'CE', 'BR', '62850000', 'Cidade')
310        read_escritorios(connection)
311        update_escritorios(connection, 1, 'Fortaleza', '2222')
312        read_escritorios(connection)
313        delete_escritorios(connection, 1)
314        read_escritorios(connection)
315
316        #Crud em TB_LINHAS_DE_PRODUTO
317        create_linhas_produto(connection, 1, '<p>Uno Branco</p>', 'Uno Branco.png', 'Uno Branco')
318        read_linhas_produto(connection)
319        update_linhas_produto(connection, 1, 'Nova desc', 'Tecnology')
320        read_linhas_produto(connection)
321        delete_linhas_produto(connection, 1)
322        read_linhas_produto(connection)
                                                                                        286,1          90%
```

```python
315
316        #Crud em TB_LINHAS_DE_PRODUTO
317        create_linhas_produto(connection, 1, '<p>Uno Branco</p>', 'Uno Branco.png', 'Uno Branco')
318        read_linhas_produto(connection)
319        update_linhas_produto(connection, 1, 'Nova desc', 'Tecnology')
320        read_linhas_produto(connection)
321        delete_linhas_produto(connection, 1)
322        read_linhas_produto(connection)
323
324        #Crud em TB_PAGAMENTOS
325        create_pagamentos(connection, 1, 1, '1/01/2025', 20)
326        read_pagamentos(connection)
327        update_pagamentos(connection, 1, '2/02/2025', 30)
328        read_pagamentos(connection)
329        delete_pagamentos(connection, 1)
330        read_pagamentos(connection)
331
332        #Crud em TB_PEDIDOS
333        create_pedidos(connection, 1, '1/01/2025', '20/01/2026','2/02/2025', 'indo', 'Urgente', 1)
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, 'Mouse', 1,'1:10', 'Razer', 'Mouse gamer', 1, 20, 150)
342        read_produtos(connection)
343        update_produtos(connection, 1, 'Musse', 3000)
344        read_produtos(connection)
345        delete_produtos(connection, 1)
346        read_produtos(connection)
347
348        connection.close()
349
350 if __name__ == "__main__":
351        main()
                                                                                        351,1          Bot
```