

Nome: Lucas Gomes da Silva

1- O programa entra com as credenciais e o método de onde quer pegar os dados, sendo um sqlite ou por csv, faz a verificação das credenciais e se tudo estiver tudo de acordo printa na tela as informações de vendas.

2-Ocorreu um erro de credenciais inválidas, o arquivo credencias3.txt, não possui senha para acesso;

3-Ocorreu um erro de credenciais inválidas, o arquivo credencias6.txt, está passando um algoritmo como csv, mas está passando um dos parâmetros como db;

4- Ocorreu um erro, onde não foi encontrado o banco, pois o banco dados 3 não possui dados;

#### 5- Resposta desafio 5

```
def carregar_credenciais(arquivo):
    credenciais = {}
    try:
        with open(arquivo) as file:
            linhas = (line.rstrip() for line in file)
            linhas = (line for line in linhas if line)

            for line in linhas:
                if "#" not in line:
                    key, valor = line.split('==>')
                    valor = valor.replace('\n', '')
                    credenciais[key] = valor
    except FileNotFoundError:
        raise FileNotFoundError('Arquivo não encontrado: ' + arquivo)
    return credenciais
```

## 6- Resposta Desafio 6

```
class Estrategia_SQLite(Estrategia):
    def execute(self, dados):
        lista_registros = []
        teste = []
        db = dados['db']
        with closing(sqlite3.connect(db)) as conn:
            cursor = conn.cursor()
            cursor.execute("SELECT * FROM vendas;")
            for linha in cursor.fetchall():
                lista_registros.append(linha)
        #Modificação Desafio 6
        for item in lista_registros:
            teste.append((item[-2], item[-1]))
        return teste
        #Finaliza Modificação 6
```

## 7- Desafio 7

```
class Estrategia_CSV(Estrategia):
    def execute(self, dados):
        teste=[]
        arquivo = dados['arquivo']
        with open(arquivo, newline='\n') as csvfile:
            reader = csv.DictReader(csvfile)
            for line in reader:
                lista_registros = []
                lista_registros2 = []
                lista_registros.append(line['total'])
                lista_registros2.append(line['vendido_em'])
                teste.append(dict(zip(lista_registros2,
lista_registros)))
        return teste
```