**Implementações e Testes do servidor Flask**

Testes com SQLMAP:

🛡️ RELATÓRIO DE SEGURANÇA – SQL INJECTION NA ROTA /consulta

🔴 ANTES DA CORREÇÃO (VULNERÁVEL)

Código vulnerável:

query = f"SELECT nome, profissao, salario FROM pessoas WHERE nome LIKE '%{termo}%'"

c.execute(query)

Problemas identificados:

- Concatenação direta da entrada do usuário (termo) na query.

- Total ausência de parametrização.

- A aplicação não valida o tamanho ou formato da entrada.

- Sanitização ausente, expondo o HTML a injeção de scripts (XSS).

Testes realizados com sqlmap:

sqlmap -u "http://127.0.0.1:5000/consulta" --data="termo=teste" --batch --level=5 --risk=3 --flush-session

Resultados:

- Parâmetro vulnerável: termo (POST)

- Banco de dados identificado: SQLite

- Técnicas exploradas com sucesso:

• Boolean-based blind:

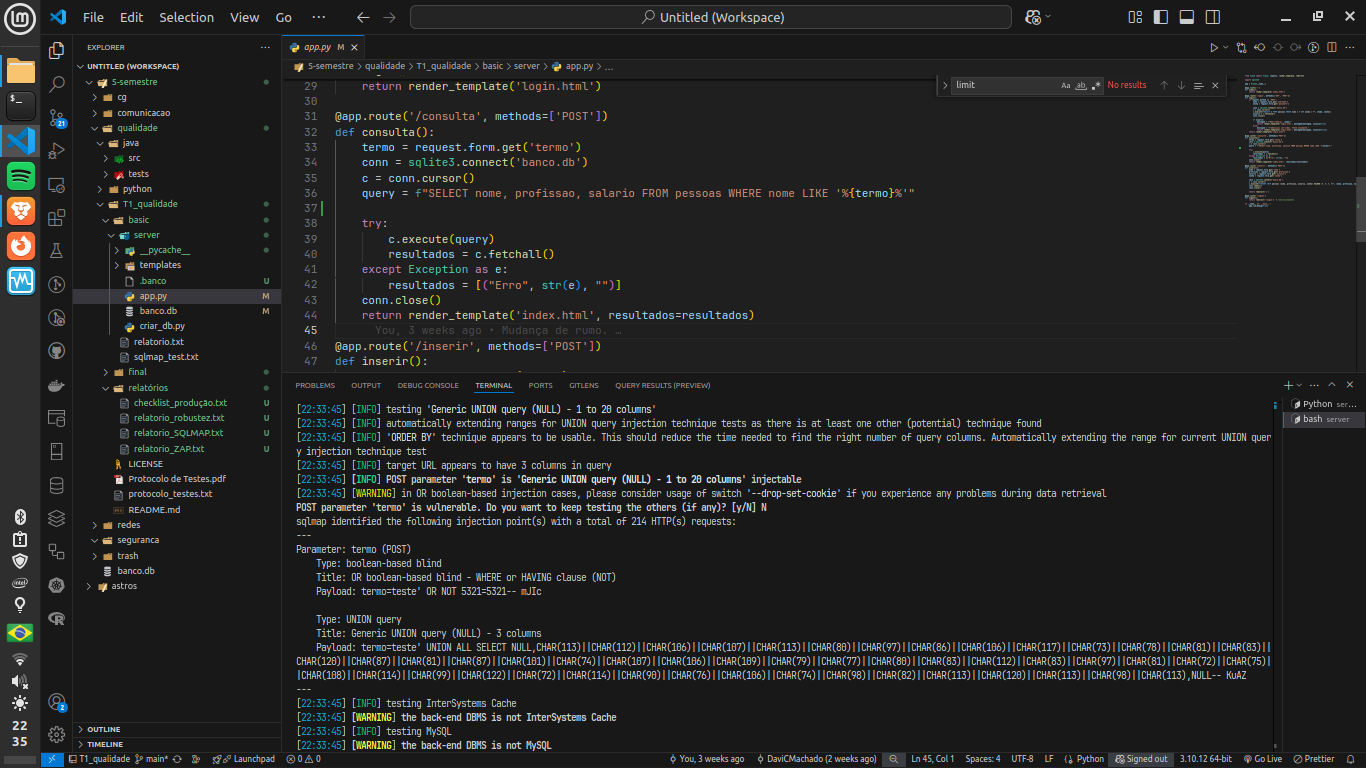
termo=-1984' OR 4854=4854--

• Time-based blind:

termo=teste' OR 6220=LIKE(CHAR(...),UPPER(HEX(RANDOMBLOB(...))))--

• UNION-based:

termo=teste' UNION ALL SELECT NULL,NULL,CHAR(...)--



✅ DEPOIS DA CORREÇÃO (SEGURO)

Código corrigido:

c.execute("SELECT nome, profissao, salario FROM pessoas WHERE nome LIKE ?", ('%' + termo + '%',))

Medidas implementadas:

1. Parametrização segura da query (uso de ? com tupla).

2. Validação da entrada do usuário (vazia ou muito longa).

3. Sanitização com html.escape() antes de exibir no HTML.

4. Tratamento de exceções com try/except para evitar mensagens sensíveis.

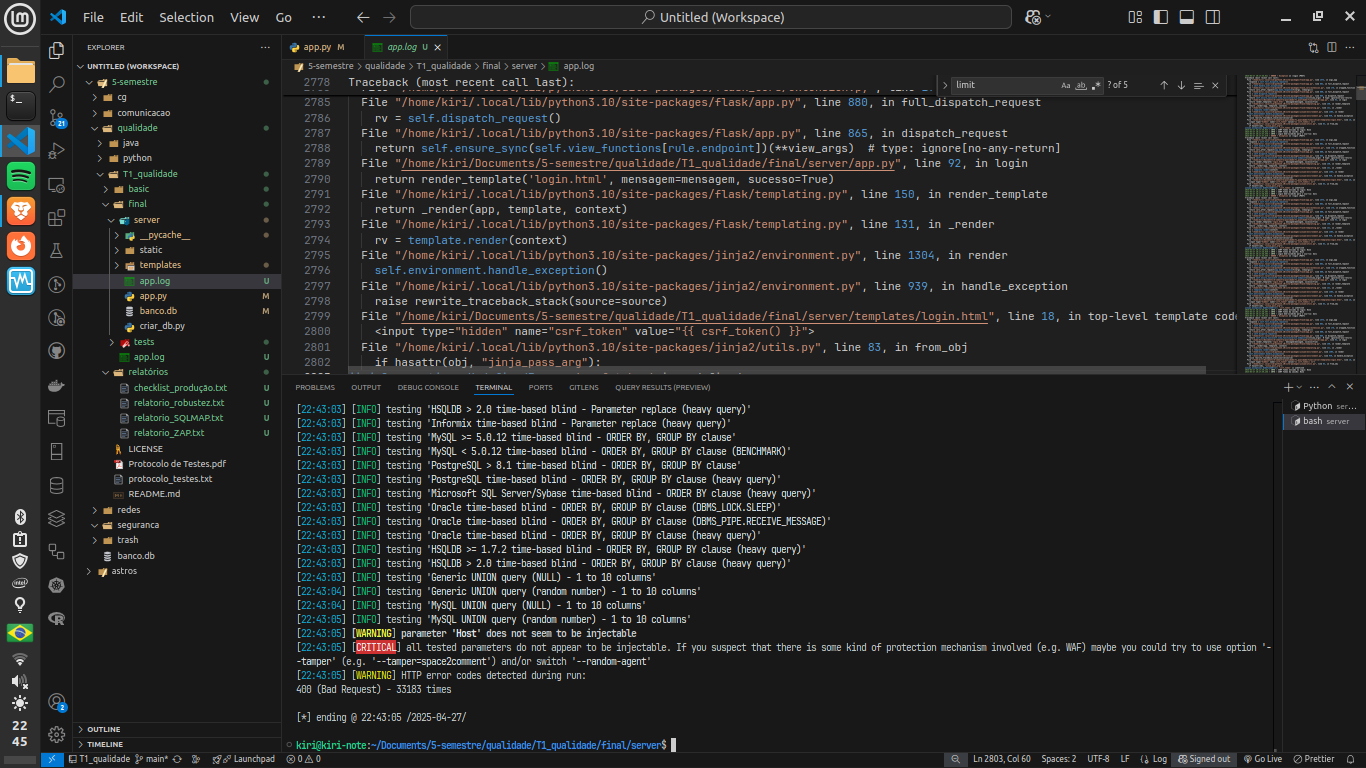
5. Log de acessos para auditoria futura.

Reexecução dos testes com sqlmap:

Resultados obtidos:

[WARNING] POST parameter 'termo' does not seem to be injectable

[CRITICAL] all tested parameters do not appear to be injectable.



📌 CONCLUSÃO

- A vulnerabilidade inicial permitia extração e manipulação arbitrária de dados via injeção SQL.

- Após as correções, a aplicação passou a seguir boas práticas de segurança:

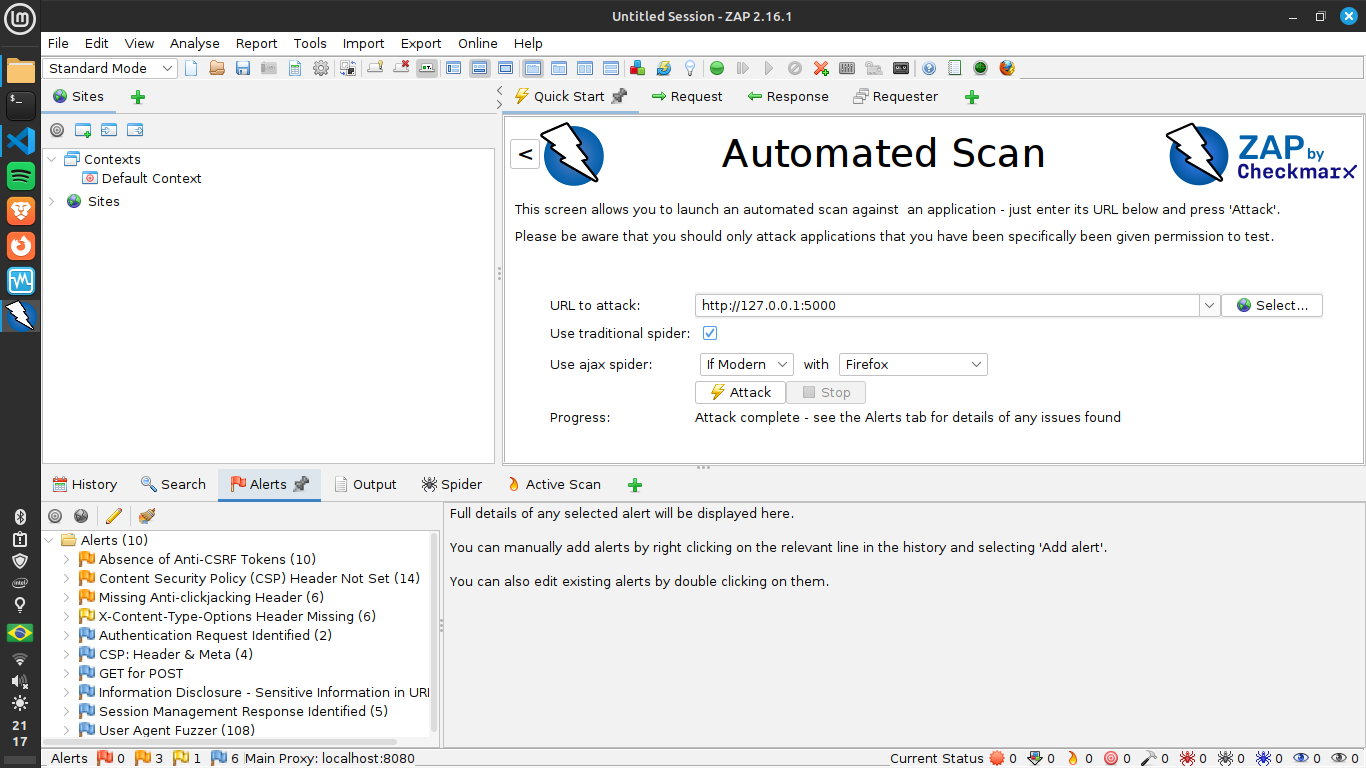
• Uso de parâmetros no SQL.

• Validação e sanitização adequadas.

- O sistema se tornou resistente às técnicas utilizadas pelo sqlmap, como boolean, time-based e union.

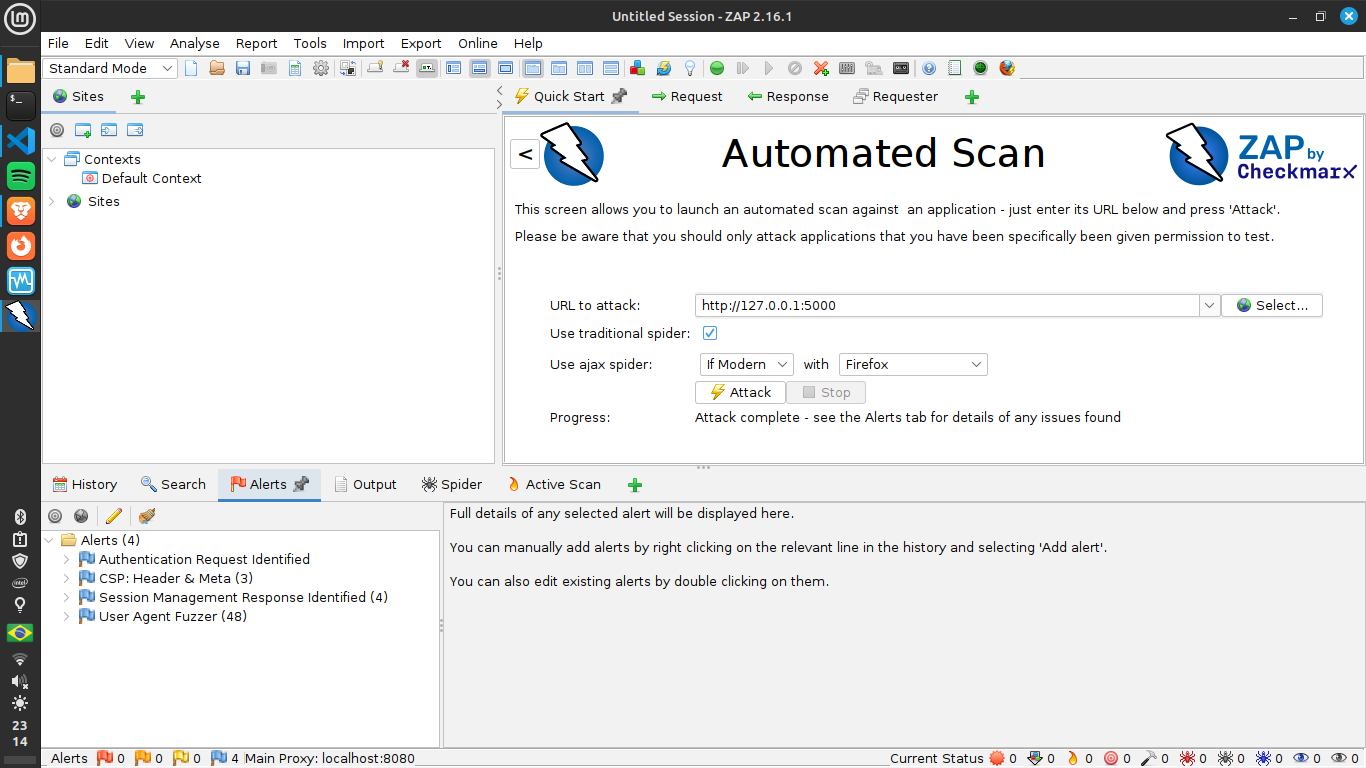
Testes com OWASP ZAP

Antes da implementação de CSP, CSRF e Cookies de sessão:



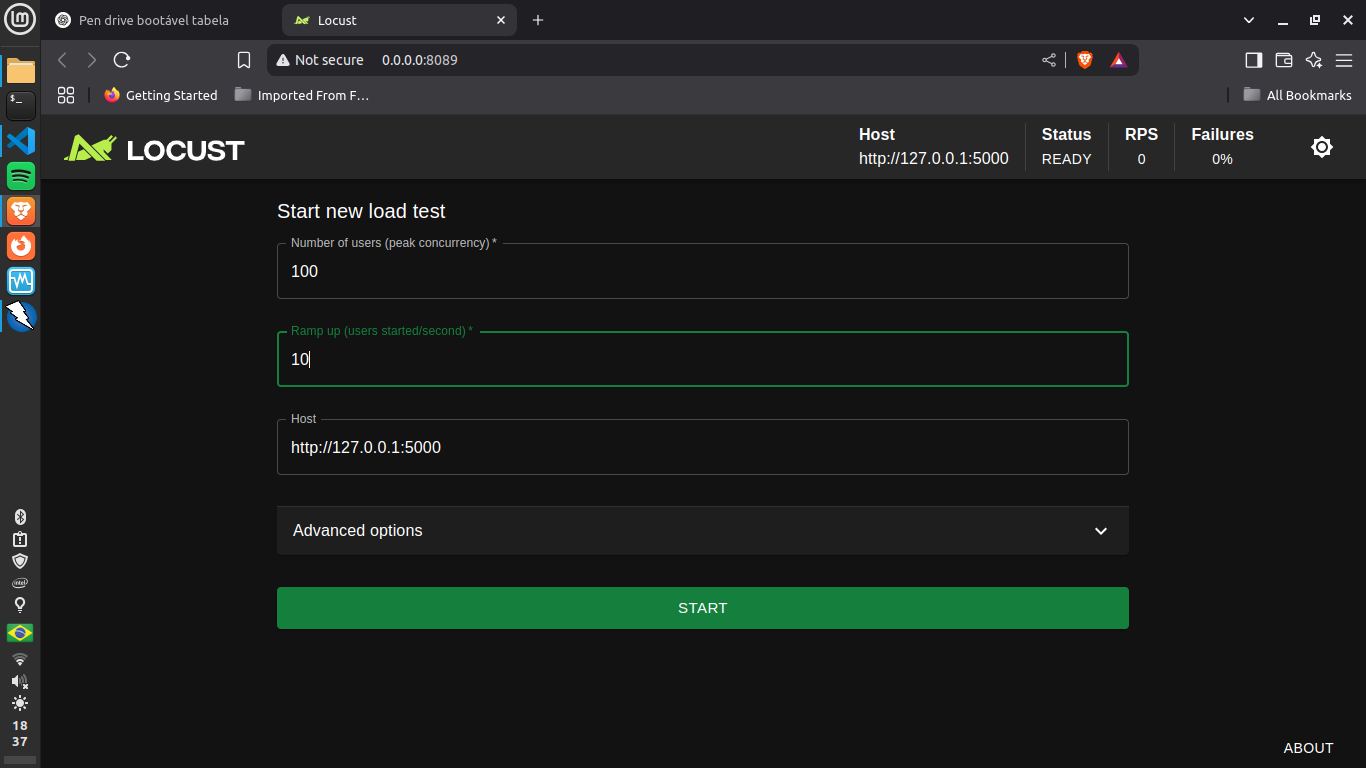
4 Alertas de segurança, sendo 3 de risco médio, mostrando que o servidor estava suscetível a ataques de XSS e Clickjacking.

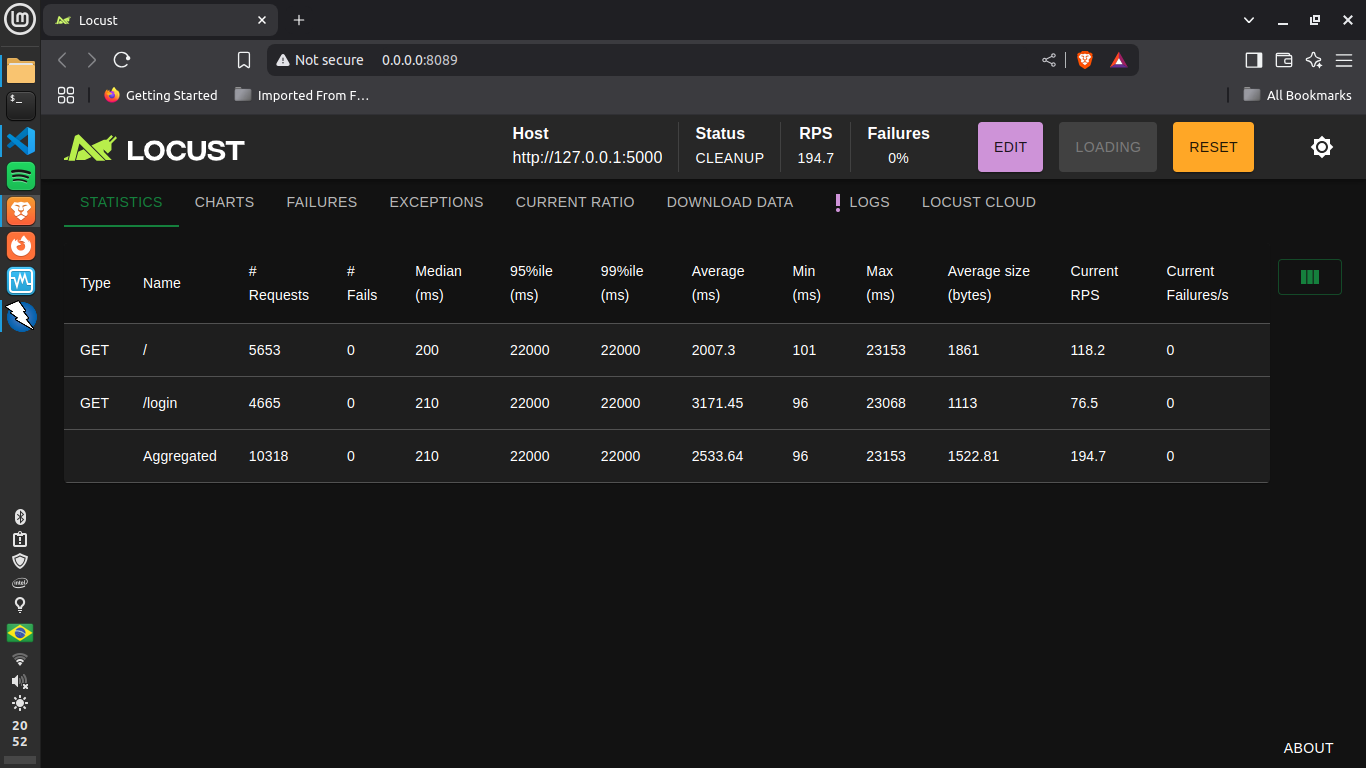
Após implementar medidas de segurança:



Nenhum alerta de risco, apenas alertas informacionais.

Testes de Carga (Resiliência)





Usando a ferramenta LOCUST testamos os seguintes casos:

a) Testes de Carga (Load Testing)

Teste 1

MAX de usuarios: 100

Incremento de Usuarios por segundo: 10

Tempo entre cada requisição: 3-6 seg

Resultados:

Média | Mínimo | Máximo

26 ms | 4 ms | 113 ms

Zero falhas

Teste 2

MAX de usuarios: 200

Incremento de Usuarios por segundo: 20

Tempo entre cada requisição: 3-6 seg

Resultados:

Média | Mínimo | Máximo

45 ms | 4 ms | 269 ms

Zero falhas

Teste 3

MAX de usuarios: 500

Incremento de Usuarios por segundo: 50

Tempo entre cada requisição: 3-6 seg

Resultados:

Média | Mínimo | Máximo

93 ms | 4 ms | 611 ms

Zero falhas

b) Testes de Resiliência (Stress Testing)

Teste 1

MAX de usuarios: 1000

Incremento de Usuarios por segundo: 100

Tempo entre cada requisição: 1-3 seg

Resultados:

Média | Mínimo | Máximo

2527 ms | 96 ms | 23152 ms

Zero falhas

Teste 2

MAX de usuarios: 500

Incremento de Usuarios por segundo: 0 (já começa com 500 users)

Tempo entre cada requisição: 3-6 seg

Resultados:

Média | Mínimo | Máximo

121 ms | 4 ms | 4139 ms

Zero falhas

Conclusão

Os testes realizados demonstraram que o servidor é capaz de lidar

com altas quantidades de requisições simultâneas sem falhas,

mantendo uma boa performance mesmo em condições de carga extrema.

A média de tempo de resposta foi adequada em todos os testes,

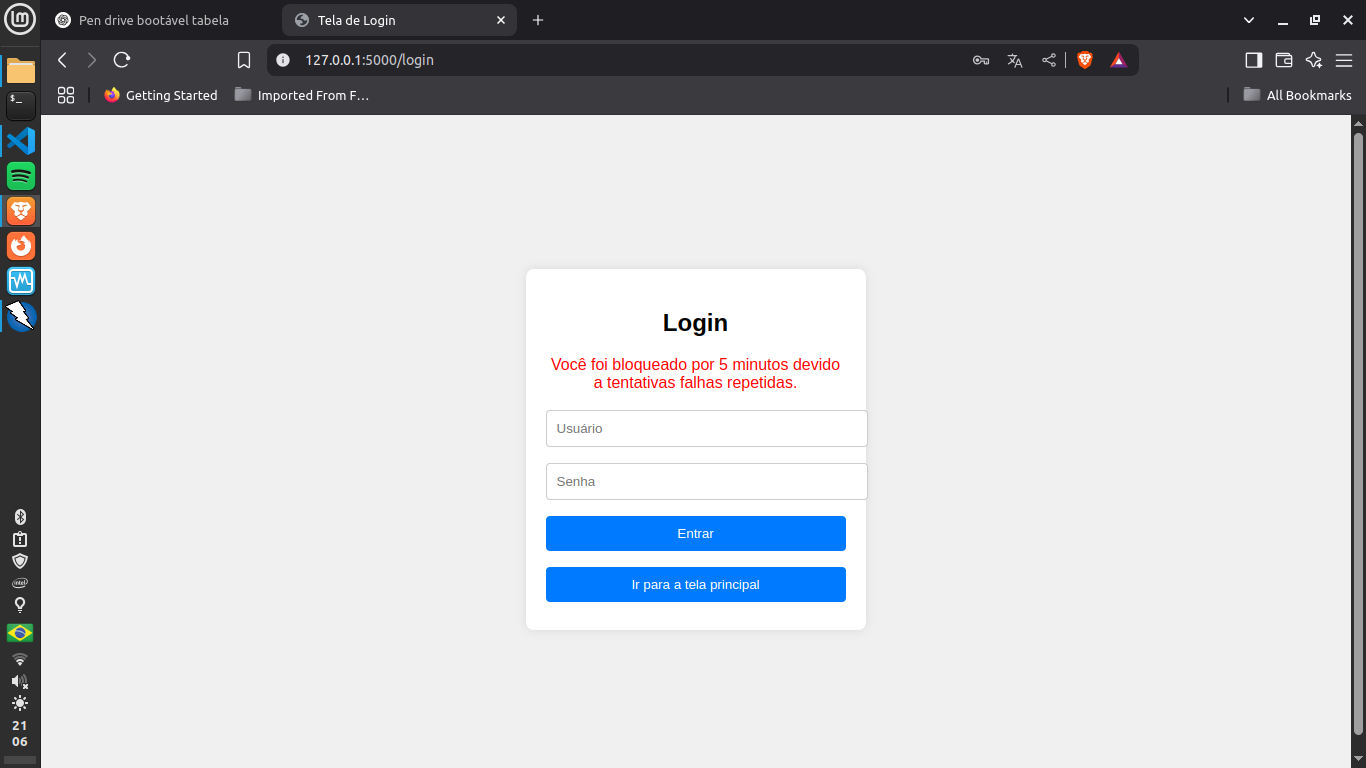
com uma variação aceitável no máximo de requisições por segundo.

Outras implementações:

O sistema possui um limitador de tentativas de login, limitando cada sessão a 5 tentativas por minuto, baseado no IP do usuário.

Um mesmo usuário pode errar até 3 vezes consecutivas.

Se passar disso, o usuário sofre time-out de 5 minutos.



O sistema também possui um sistema de log que registra acessos aos recursos do servidor por parte dos usuários e também erros e exceções.

