Construindo uma Aplicação

Verificando se Login e senha estão corretos

```python
import os
from http.server import SimpleHTTPRequestHandler
import socketserver
from urllib.parse import parse_qs

# Esta versão manin8 verifica se usuário já existe e se senha informada está correta
# Em caso de novo usuario, ou seja, um login que não está na base, ele é cadastrado na base e recebe uma mensgaem de boas vinadas


2 usages
class MyHandler(SimpleHTTPRequestHandler):
    1 usage (1 dynamic)
    def list_directory(self, path):
        try:
            # Tenta abrir o arquivo idx.html
            f = open(os.path.join(path, 'idx.html'), 'r')
            # Se existir, envia o conteúdo do arquivo
            self.send_response(200)
            self.send_header("Content-type", "text/html")
            self.end_headers()
            self.wfile.write(f.read().encode('utf-8'))
            f.close()
            return None
        except FileNotFoundError:
            pass

        return super().list_directory(path)
```

Construindo uma Aplicação

Verificando se Login e senha estão corretos

```
26
27      def do_GET(self):
28          if self.path == '/login':
29              # Tenta abrir o arquivo login.html
30              try:
31                  with open(os.path.join(os.getcwd(), 'login.html'), 'r') as login_file:
32                      content = login_file.read()
33                  self.send_response(200)
34                  self.send_header("Content-type", "text/html")
35                  self.end_headers()
36                  self.wfile.write(content.encode('utf-8'))
37              except FileNotFoundError:
38                  self.send_error(404, "File not found")
39
40          elif self.path == '/login_failed':...
56
57          else:
58              # Se não for a rota "/login", continua com o comportamento padrão
59              super().do_GET()
```

Construindo uma Aplicação

Verificando se Login e senha estão corretos

```
40      elif self.path == '/login_failed':
41          # Responde ao cliente com a mensagem de login/senha incorreta
42          self.send_response(200)
43          self.send_header("Content-type", "text/html; charset=utf-8")
44          self.end_headers()
45
46          # Lê o conteúdo da página login.html
47          with open(os.path.join(os.getcwd(), 'login.html'), 'r', encoding='utf-8') as login_
48              content = login_file.read()
49
50          # Adiciona a mensagem de erro no conteúdo da página
51          mensagem = "Login e/ou senha incorreta. Tente novamente."
52          content = content.replace('<!-- Mensagem de erro será inserida aqui -->',
53                                    f'<div class="error-message">{mensagem}</div>')
54          # Envia o conteúdo modificado para o cliente
55          self.wfile.write(content.encode('utf-8'))
56
57      else:
```

Construindo uma Aplicação

Verificando se pagina de login.html para exibir erro de Login

```
53  <body>
54
55      <div class="login-container">
56          <div class="error-message">
57              <!-- Mensagem de erro será inserida aqui -->
58          </div>
59          <h2>Login</h2>
60          <form action="/enviar_login" method="post">
61              <label for="email">Email:</label>
62              <input type="email" id="email" name="email" required>
63
64              <label for="senha">Senha:</label>
65              <input type="password" id="senha" name="senha" required>
66
67              <button type="submit">Enviar</button>
68          </form>
69      </div>
70
71  </body>
```

Construindo uma Aplicação

Verificando se Login e senha estão corretos

```
1 usage
def usuario_existente(self, login, senha):
    # Verifica se o login já existe no arquivo

    with open('dados_login.txt', 'r', encoding='utf-8') as file:
        for line in file:
            stored_login, stored_senha = line.strip().split(';')

            if login == stored_login:
                print ("cheguei aqui significando que localizei o login informado")
                print ("senha: " + senha)
                print(" senha_armazenada: " + senha)
                return senha == stored_senha
    return False
```

Construindo uma Aplicação

Verificando se Login e senha estão corretos

```
74
75      def do_POST(self):
76          # Verifica se a rota é "/enviar_login"
77          if self.path == '/enviar_login':
78              # Obtém o comprimento do corpo da requisição
79              content_length = int(self.headers['Content-Length'])
80              # Lê o corpo da requisição
81              body = self.rfile.read(content_length).decode('utf-8')
82              # Parseia os dados do formulário
83              form_data = parse_qs(body, keep_blank_values=True)
84
85              # Exibe os dados no terminal
86              print("Dados do formulário:")
87              print("Email:", form_data.get('email', [''])[0])
88              print("Senha:", form_data.get('senha', [''])[0])
89
90              # Verifica se o usuário já existe
91              login = form_data.get('email', [''])[0]
92              senha = form_data.get('senha', [''])[0]
93
94              if self.usuario_existente(login, senha):
95                  # Responde ao cliente indicando que o usuário logou com sucesso
96                  self.send_response(200)
97                  self.send_header("Content-type", "text/html; charset=utf-8")
98                  self.end_headers()
99                  mensagem = f"Usuário {login} logado com sucesso!!!"
100                 self.wfile.write(mensagem.encode('utf-8'))
```

Verificando se Login e senha estão corretos

```
94              if self.usuario_existente(login, senha):
95                  # Responde ao cliente indicando que o usuário logou com sucesso
96                  self.send_response(200)
97                  self.send_header("Content-type", "text/html; charset=utf-8")
98                  self.end_headers()
99                  mensagem = f"Usuário {login} logado com sucesso!!!"
100                 self.wfile.write(mensagem.encode('utf-8'))
101             else:
102                 # Verifica se o login já existe no arquivo
103                 if any(line.startswith(f"{login};") for line in
104                        open('dados_login.txt', 'r', encoding='utf-8')):
105                     # Redireciona o cliente para a rota "/login_failed"
106                     self.send_response(302)
107                     self.send_header('Location', '/login_failed')
108                     self.end_headers()
109                     return  # Adicionando um return para evitar a execução do restante do código
110                 else:
111                     # Adiciona o novo usuário ao arquivo
112                     with open('dados_login.txt', 'a', encoding='utf-8') as file:
113                         file.write(f"{login};{senha}\n")
114                     # Responde ao cliente com a mensagem de boas-vindas
115                     self.send_response(200)
116                     self.send_header("Content-type", "text/html; charset=utf-8")
117                     self.end_headers()
118                     mensagem = f"Olá {login}, seja bem-vindo! Percebemos que você é novo por aqui."
119                     self.wfile.write(mensagem.encode('utf-8'))
120         else:
```

Verificando se Login e senha estão corretos

```
                    return  # Adicionando um return para evitar a execução do restante do código
                else:
                    # Adiciona o novo usuário ao arquivo
                    with open('dados_login.txt', 'a', encoding='utf-8') as file:
                        file.write(f"{login};{senha}\n")
                    # Responde ao cliente com a mensagem de boas-vindas
                    self.send_response(200)
                    self.send_header("Content-type", "text/html; charset=utf-8")
                    self.end_headers()
                    mensagem = f"Olá {login}, seja bem-vindo! Percebemos que você é novo por aqui."
                    self.wfile.write(mensagem.encode('utf-8'))
            else:
                # Se não for a rota "/enviar_login", continua com o comportamento padrão
                super(MyHandler, self).do_POST()


# Define o IP e a porta a serem utilizados
endereco_ip = "0.0.0.0"
porta = 8000

# Cria um servidor na porta e IP especificados
with socketserver.TCPServer((endereco_ip, porta), MyHandler) as httpd:
    print(f"Servidor iniciado em {endereco_ip}:{porta}")
    # Mantém o servidor em execução
    httpd.serve_forever()

```