

Criptografando a senha com Hash-Code

```
import os
from http.server import SimpleHTTPRequestHandler
import socketserver
#from urllib.parse import parse_qs
from urllib.parse import urlparse, parse_qs
import hashlib # Importe a biblioteca hashlib para usar a criptografia SHA256
```



Criptografando a senha com Hash-Code – na função `usuario_existente`

```
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:  
            if line.strip():  
                stored_login, stored_senha_hash, stored_nome = line.strip().split(';')  
                if login == stored_login:  
                    print("chequei aqui significando que localizei o login informado")  
                    print("senha: " + senha)  
                    print(" senha_armazenada: " + senha)  
  
                    #----- TRECHO HASH  
                    # Hash a senha fornecida pelo usuário  
                    senha_hash = hashlib.sha256(senha.encode('utf-8')).hexdigest()  
                    # Verifique se os hashes coincidem  
                    return senha_hash == stored_senha_hash  
  
                    #----- TRECHO HASH  
  
                    #return senha == stored_senha  
  
    return False
```

Criptografando a senha com Hash-Code – em `do_POST`

```
def do_POST(self):
    # Verifica se a rota é "/enviar_login"
    if self.path == '/enviar_login':
        .
        .
        .
        if self.usuario_existente(login, senha):
            .
            .
            .
        else:
            # Verifica se o login já existe no arquivo
            if any(line.startswith(f"{login};") for line in open('dados_login.txt', 'r', encoding='utf-8')):
                .
                .
                .
            else:
                #----- Inserir Aqui
                #----- Fim

                

```

                ###Trecho Hash
                # Adiciona o novo usuário ao arquivo
                self.adicionar_usuario(login, senha, nome='None')
                ###Trecho Hash

                # Redireciona o cliente para a rota "/cadastro" com os dados de login e senha
                self.send_response(302)
                self.send_header('Location', f'/cadastro?login={login}&senha={senha}')
                self.end_headers()

                return # Adicionando um return para evitar a execução do restante do código

```


```

Criptografando a senha com Hash-Code – criando a função `adicionar_cadastro`

```
> def usuario_existente(self, login, senha):...  
  
# --- Trecho HASH  
def adicionar_usuario(self, login, senha, nome):  
    # Hash a senha fornecida pelo usuário  
    senha_hash = hashlib.sha256(senha.encode('utf-8')).hexdigest()  
    # Adiciona o novo usuário ao arquivo  
    with open('dados_login.txt', 'a', encoding='utf-8') as file:  
        file.write(f"{login};{senha_hash};{nome}\n")  
# --- Trecho HASH  
  
> def remover_ultima_linha(self, arquivo):...
```

← Código novo

# Construindo uma Aplicação

Criptografando a senha com Hash-Code – em **do\_POST** na rota **/confirmar\_cadastro**

```
elif self.path.startswith('/confirmar_cadastro'):  
  
    # Obtém o comprimento do corpo da requisição  
    content_length = int(self.headers['Content-Length'])  
    # Lê o corpo da requisição  
    body = self.rfile.read(content_length).decode('utf-8')  
    # Parseia os dados do formulário  
    form_data = parse_qs(body, keep_blank_values=True)  
  
    #query_params = parse_qs(urlparse(self.path).query)  
    login = form_data.get('login', [''])[0]  
    senha = form_data.get('senha', [''])[0]  
    nome = form_data.get('nome', [''])[0]  
  
    # Hash a senha fornecida pelo usuário  
    senha_hash = hashlib.sha256(senha.encode('utf-8')).hexdigest()  
  
    print("nome: " + nome)  
  
    # Verifica se o usuário já existe  
    if self.usuario_existente(login, senha):  
  
        # Atualiza o arquivo com o nome, se a senha estiver correta  
        with open('dados_login.txt', 'r', encoding='utf-8') as file:  
            lines = file.readlines()  
  
        with open('dados_login.txt', 'w', encoding='utf-8') as file:  
  
            for line in lines:  
  
                stored_login, stored_senha, stored_nome = line.strip().split(';')  
                if login == stored_login and senha_hash == stored_senha:  
                    line = f"{login};{senha_hash};{nome}\n"  
  
            file.write(line)
```

Criptografando a senha com Hash-Code – arquivo com dados criptografados

Senha: 123456 →

Senha: 123456 →

Senha: 123457 →

```
fulanodasilva@gmail.com;123456;Fulano da Silva  
beltrano@gmail.com;8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92;Beltrano da Silva  
ciclano@gmail.com;8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92;Ciclano da Silva  
senhanova@gmail.com;54b688a517f7654563a6c64d945a3670880a4c602ec67a065bbebbcd2b22edd5;Senha Nova
```

