

Copilot

Para criar um sistema de login com usuário e senha no Flask e garantir que apenas usuários logados acessem as URLs do projeto, você pode seguir os passos abaixo:

1. **Crie um modelo de usuário** no arquivo `modelos.py` para armazenar informações de usuário e senha.
2. **Utilize Flask-Login** para gerenciar sessões de usuário.
3. **Proteja as rotas** com um decorador que verifica se o usuário está logado.

Aqui está um exemplo de como você pode implementar isso:

modelos.py

```
from werkzeug.security import generate_password_hash, check_password_hash
from flask_login import UserMixin
from app import db, login_manager

class Usuario(UserMixin, db.Model):
    __tablename__ = 'usuarios'
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(64), unique=True, index=True)
    password_hash = db.Column(db.String(128))

    def set_password(self, password):
        self.password_hash = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password_hash, password)

@login_manager.user_loader
def load_user(id):
    return Usuario.query.get(int(id))
```

init.py (no diretório app)

Inicialize o Flask-Login e configure o gerenciador de login.

```
from flask_login import LoginManager
# ... outras importações ...

login_manager = LoginManager()
login_manager.login_view = 'auth.login'

def create_app(config_name):
    # ... configuração do app ...
    login_manager.init_app(app)
    # ... mais inicializações ...
```

auth.py (dentro do diretório de módulos)

Crie as rotas de autenticação.

```
from flask import render_template, redirect, url_for, request, flash
from flask_login import login_user, logout_user, login_required
from . import auth
from ..modelos import Usuario

@auth.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        usuario = Usuario.query.filter_by(username=request.form['username']).first()
        if usuario is not None and usuario.check_password(request.form['password']):
            login_user(usuario)
            return redirect(url_for('main.index'))
        flash('Usuário ou senha inválidos.')
```

```
        return render_template('login.html')

@auth.route('/logout')
@login_required
def logout():
    logout_user()
    return redirect(url_for('main.index'))
```

visoes.py (em cada módulo)

Proteja as rotas com o decorador @login_required.

```
from flask_login import login_required
# ... outras importações ...

@clientes.route('/')
@login_required
def listar_clientes():
    # ... lógica para listar clientes ...
```

Com essa implementação, quando um usuário tentar acessar uma rota protegida, ele será redirecionado para a página de login se não estiver autenticado. Após o login bem-sucedido, o usuário será redirecionado de volta à rota original que tentou acessar.

Lembre-se de que você precisará instalar o Flask-Login e inicializar o LoginManager, além de configurar o modelo de usuário e as rotas de autenticação. Se precisar de mais ajuda ou tiver outras perguntas, estou aqui para ajudar. Boa sorte com o seu projeto! 🚀