

# Documentação - Cadastro de Usuários (Zenith IA)

## Objetivo da Etapa

Implementar o sistema de cadastro de usuários no projeto Zenith IA, utilizando Flask, conexão MySQL e segurança de senha com hash. O cadastro recebe nome, email e senha, e salva essas informações de forma segura no banco de dados.

## .env (variáveis de ambiente)

Arquivo responsável por armazenar as configurações sensíveis, como dados de acesso ao banco.

Exemplo:

```
SECRET_KEY=zenithsupersecreto123
```

```
DB_HOST=localhost
```

```
DB_USER=root
```

```
DB_PASSWORD=
```

```
DB_NAME=zenith_db
```

## database/connection.py

Arquivo que realiza a conexão com o banco de dados MySQL, utilizando as variáveis definidas no .env. Utiliza a biblioteca mysql-connector-python e dotenv para carregar as configurações.

Incluimos também um print para debug do usuário utilizado na conexão.

## app/routes.py (rota /registro)

Arquivo que define a rota POST /registro. Ele recebe JSON com nome, email e senha, valida os dados, gera um hash seguro da senha com werkzeug.security, e salva o usuário no banco.

O bloco try/except/finally garante o fechamento seguro da conexão e tratamento de erros. Caso tudo ocorra bem, retorna a mensagem de sucesso em JSON.

## app/\_\_init\_\_.py

Inicializa o app Flask e registra as rotas. A função create\_app() monta a aplicação e adiciona o blueprint main, onde estão definidas as rotas como /registro.

## run.py

Arquivo principal que executa o Flask. Importa a função create\_app e executa o app localmente com debug=True, escutando na porta 5000 (localhost:5000).

## Documentação - Cadastro de Usuários (Zenith IA)

### Teste com Thunder Client (VS Code)

O Thunder Client foi usado para enviar uma requisição POST para `http://localhost:5000/registro` com um JSON contendo nome, email e senha. Após corrigir o nome do arquivo `.env` e garantir a conexão com o banco, o retorno foi a mensagem de sucesso.