

Documentação Técnica - Resume.AI

Versão: 1.0

Status: Em desenvolvimento

1. Visão Geral do Sistema

O Resume.AI é uma aplicação web desenvolvida em Django cujo objetivo principal é permitir que usuários façam upload de PDFs e recebam resumos automáticos, mantendo um histórico organizado de seus arquivos processados.

2. Arquitetura do Sistema

O sistema segue o padrão MVC/MVT do Django, com separação clara entre modelos, views e templates. Também utiliza autenticação nativa do Django e componentes adicionais como envio de e-mails, validação de usuário e histórico de arquivos.

3. Principais Funcionalidades

- Cadastro e login de usuários (autenticação padrão do Django).
- Alteração de dados pessoais.
- Alteração de senha utilizando senha atual
- Exclusão de conta com confirmação.
- Envio de arquivos PDF e geração automática de resumos.
- Histórico de PDFs resumidos.

4. Modelos (Models)

4.1 User (Django Default)

Utiliza o modelo de usuário padrão do Django (`get_user_model`).

4.2 PDFHistory

Armazena resumos gerados dos Uploads dos PDFs. Inclui informações como nome do arquivo, conteúdo resumido.

5. Views

Cadastro

Criação de novos usuários.

Login

Autenticação com Django.

Logout

Finaliza sessão e exibe página de agradecimento.

Alteração de Dados

Atualiza nome, e-mail ou telefone.

Alteração de Senha

Dois modos: senha atual ou código por e-mail (Em desenvolvimento).

Exclusão de Conta

Exige confirmação antes de deletar o usuário.

Histórico de PDFs

Lista arquivos já enviados e resumidos.

6. Templates

O projeto utiliza Bootstrap via CDN e organiza templates dentro de core/templates/. Entre os principais templates estão: alterar_senha.html, confirmar_exclusao.html, conta_excluida.html, configuracoes.html, e telas de login/cadastro.

7. Fluxo de Alteração de Senha

O fluxo permite alteração via senha atual:

1. No modo 'senha', o sistema valida a senha atual.
2. A senha é atualizada e o usuário é redirecionado.

8. Segurança

- Senhas nunca são exibidas nem armazenadas em texto puro.
- Resumo de PDFs sempre associado ao usuário logado.
- Operações críticas exigem login obrigatório.

9. Testes Automatizados

Os testes podem ser executados com:

`python manage.py test`

Incluem:

- Criação de usuários
- Login/logout
- Alteração de senha
- Fluxo de exclusão de conta

10. Status do Projeto

⚠ *Este projeto ainda está em desenvolvimento e não está 100% finalizado.*