

Swagger

Descrição Técnica do Backend

O backend do projeto é estruturado de forma modular, com separação clara de responsabilidades entre rotas, serviços, middlewares e utilitários. A aplicação é desenvolvida utilizando **Node.js** com **Express** e toda a API está documentada de forma interativa utilizando o **Swagger UI**, acessível pelo endereço:

```
“ http://localhost:3008/api-docs/#/ ”
```

Lembre-se de ter o projeto rodando no docker para funcionar!

Estrutura de Diretórios

```
/api/  
├─ src/  
│ └─ docs/          # [] Configurações e definição dos endpoints no Swagger  
│   └─ swagger.ts  
│  
│ └─ middlewares/   # ⚙ Middlewares diversos, incluindo o multer para uploads  
│   └─ multer.ts  
│  
│ └─ router/        # [] Definição de todas as rotas do sistema organizadas por domínio  
│   └─ avaliador/  
│   └─ corregedor/  
│   └─ defensor/  
│   └─ usuario/  
│   └─ router.ts    # Roteador principal que importa e agrupa os módulos acima  
│  
└─ services/        # [] Regras de negócio e funcionalidades para cada tipo de usuário
```

```
| | └─ avaliador/
| | └─ corregedor/
| | └─ defensor/
| | └─ usuario/
|
| └─ utils/      # Utilitários de suporte à aplicação
| | └─ opt/      # Envio de OTPs e recuperação de senha
| | └─ pdf/      # Operações de manipulação e leitura de PDFs
| | └─ validations/ # Validações como token, CPF, e campos diversos
| | └─ ...       # Outros utilitários adicionais
|
| └─ app.tsx     # Ponto de entrada da aplicação e inicialização dos serviços
```

Documentação Interativa com Swagger

A API conta com documentação completa utilizando **Swagger**, onde é possível:

- Visualizar **todos os endpoints disponíveis**
- Ver os **parâmetros esperados** por cada rota
- Testar requisições diretamente pela interface
- Visualizar exemplos de **respostas de sucesso e erro**

Exemplo de rotas documentadas:

- **POST /api/defensor/registrar/triagem**
Registra uma triagem com múltiplos arquivos PDF, processo e protocolo.
- **POST /api/avaliador/pecas/avaliar**
Permite que um avaliador avalie peças de um defensor.
- **GET /api/usuario/email**
Retorna as informações do email mascarado.
- **POST /api/corregedor/pecas/aprovar**
Permite ao corregedor aprovar ou reprovar uma triagem enviada.

Essas e outras rotas estão organizadas por grupos lógicos (Defensor, Avaliador, Corregedor, Usuário) dentro do Swagger UI.

Upload de Arquivos com Multer

A aplicação utiliza o middleware **Multer** para lidar com uploads de arquivos (como PDFs), salvando-os em diretórios específicos com nomes padronizados conforme o campo da requisição.

A lógica para isso está implementada em:

```
src/middlewares/multer.ts
```

Esse middleware identifica dinamicamente o campo do arquivo (`fileRaf`, `fileProcesso_1`, etc.) e direciona o salvamento para a pasta correta, como por exemplo:

- `documentos/raf/`
- `documentos/processos/`
- `documentos/recursos/`
- `documentos/avaliacoes/`
- `documentos/relatorios`

Revision #8

Created 19 May 2025 14:38:05 by Acua

Updated 22 October 2025 15:56:54 by GuilhermeBDias