

Alunos:

Gabriela Rocha Passotto - RA: 2454351

Gustavo Bueno de Carvalho - RA: 2454378

Marcos Vinicius Bueno Prestes - RA: 2465760

Requisitos Funcionais

RF01 - O sistema deve permitir que os diretores do projeto ELLP façam login para acessar as funcionalidades do sistema.

RF02 - O sistema deve permitir o cadastro de alunos voluntários com os seguintes dados:

- Nome
- Idade
- Telefone
- CPF
- RG
- Escola/Turma
- Nome de um responsável
- Telefone do responsável
- Email do responsável
- CPF do responsável
- Endereço

RF03 - O sistema deve listar os voluntários cadastrados e permitir a busca por nomes.

RF04 - O sistema deve permitir a edição e exclusão dos dados dos alunos voluntários.

RF05 - O sistema deve permitir a geração de certificados para os alunos voluntários.

RF06 - O sistema deve permitir a pesquisa pelo nome do voluntário na aba de geração de certificados.

RF07 - O sistema deve permitir que a carga horária (em horas) seja inserida antes de gerar o certificado.

RF08 - O sistema deve utilizar um modelo de certificado padrão a ser utilizado na geração do PDF.

Estratégia para os testes automatizados

1. Tecnologias Utilizadas

- Framework de Testes: Jest (Front-end e Back-end)
- Cobertura de Testes: Jest Coverage

2. Tipos de Testes Planejados

2.1. Testes de Unidade

- Escopo: Testar funções e métodos individuais
- Ferramentas: Jest

2.2. Testes de Integração

- Escopo: Testar interação entre múltiplos módulos
- Ferramentas: Jest + Supertest (para APIs)

3. Organização dos Testes

- Todos os testes serão armazenados na pasta tests/ ou junto dos componentes/módulos.
- Nomenclatura dos arquivos: nomeDoArquivo.test.ts / .test.tsx
- Scripts de execução definidos no package.json de cada projeto (front e back)

4. Execução e Automatização

- Comandos locais: npm run test
- Scripts para o package.json

```
"scripts": {  
  "test": "jest",  
  "test:watch": "jest --watch",  
  "test:coverage": "jest --coverage"  
}
```

5. Considerações

- Cada membro é responsável por escrever testes para os módulos que desenvolver.
- Todos os requisitos funcionais do sistema devem ter testes de unidade e/ou integração.
- O foco será garantir confiabilidade sem criar testes excessivamente frágeis ou complexos.