



INSTITUTO FEDERAL

Rio Grande do Norte

Campus Natal-Central

Plano de Testes

Desenvolvimento de Sistemas

Professor:

Francisco Sales

Equipe

Ana Célia

Arthur de Melo

Felipe Xavier

Matheus Duarte

Pedro Maure

Yuri Thairony

Julho / 2021

Histórico de Revisões

Data	Versão	Descrição	Autor
11/07/2024	1.0	Teste unitário do model “Ação” implementado	Matheus Duarte
17/07	2.0	Teste estrutural implementado	Matheus Duarte
28/08	3.0	Teste de integração implementado	Matheus Duarte

Índice

1. INTRODUÇÃO	3
1.1 OBJETIVOS	4
1.2 QAJUDA	Error! Bookmark not defined.
1.3 ESCOPO	4
1.4 IDENTIFICAÇÃO DE PROJETO	4
2. REQUISITOS A TESTAR	Error! Bookmark not defined.
2.1 TESTE UNITÁRIO	6
2.2 TESTE ESTRUTURAL	6
2.3 TESTE DE INTEGRAÇÃO	6
3. ESTRATÉGIA DE TESTE	Error! Bookmark not defined.
3.1 TIPOS DE TESTE	7
3.1.1 <i>Teste unitário</i>	
3.1.2 <i>Teste estrutural</i>	
3.1.3 <i>Teste de integração</i>	
4. RECURSOS	9
4.1 TRABALHADORES	9
4.2 SISTEMA	9

1. Introdução

1.1. Objetivos

Esse documento do Plano de Testes do T.I.G. compõe-se dos seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os Requisitos a Testar recomendados (alto nível).
- Recomendar e descrever as estratégias de teste a serem empregadas.
- Identificar os recursos necessários e prover uma estimativa dos esforços de teste.
- Listar os elementos resultantes do projeto de testes.

1.2. QAjuda

O QAjuda é um sistema online destinado a fomentar e organizar ações voluntárias na comunidade. A plataforma permite que indivíduos ou grupos que desejam promover iniciativas de voluntariado registrem suas ações no site. Após o registro, os voluntários interessados podem visualizar as oportunidades e se inscrever para participar das ações. O objetivo é facilitar a mobilização de pessoas para causas comunitárias e promover o engajamento em atividades de voluntariado.

Com interfaces intuitivas e amigáveis para reunir pessoas de variadas idades e perfis sociais.

1.3. Escopo

O sistema do QAjuda passará pelos testes unitário, de integração e estrutural. Os testes unitários e de integração vão lidar com a qualidade funcional, das bases de dados e do controle de acesso; enquanto que os teste estrutural irão lidar com os possíveis erros e bugs relatando onde meus testes unitários não estão sendo eficazes.

Os demais testes não serão implementados devido ao curto espaço de tempo que teremos para implementar, focaremos apenas nesses três para evitar possíveis atrasos e implementação incompleta/prejudicial. Para a execução dos testes serão utilizadas máquinas físicas e virtuais para garantir uma maior cobertura dos nossos clientes.

Os testes mais críticos serão os testes de integração, que compõe a parte mais fundamental do sistema, que será nosso fluxo de tarefa principal:

1. Testaremos o modelo de ação que é o objeto principal do nosso sistema
2. E também a correta atualização do banco de dados para as funções de cadastrar ações, solicitar participação e aceitar participação.

1.4 Identificação de Projeto

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de testes:

Documento	Criado ou Disponível	Recebido ou Revisado
Especificação de Requisitos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Plano de Projeto	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Análise	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Projeto	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Documento de Arquitetura	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Protótipo	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Manual do Usuário	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Lista de Riscos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não

2. Requisitos a testar

A lista abaixo identifica aqueles itens – use cases, requisitos funcionais e não funcionais – que foram identificados como alvos de teste. Essa lista representa o que será testado.

2.1 Teste unitários

- Verifique se os dados de ação estão sendo adicionados como deveriam.
- Verifique se as validações estão sendo executadas como deveriam.
- Verifique se a exibição dos dados está sendo executadas como deveriam.
- Verifique se o tratamento de erros está sendo executados como deveriam.
- Verifique se o relacionamento da ação está conforme o documento ER.

2.2 Teste estrutural (cobertura)

- Verifique se os testes estão abrangendo boa parte dos modelos e views da aplicação.
- Verifique se as condicionais das views estão sendo executadas.
- Verifique se a taxa de cobertura está acima de 70%.

2.3 Teste de integração

- Verifique se os endpoints estão redirecionando para as páginas certas.
- Verifique se os endpoints estão recebendo e tratando os dados de forma correta.
- Verifique se os endpoints estão conseguindo se comunicar com o banco de dados na hora de criar e listar.

3. Estratégia de Teste

3.1 Tipos de Teste

Nota: As transações abaixo se referem às “transações lógicas de negócio”. Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma dada informação.

3.1.1 Teste unitário

Objetivo do Teste:	Garantir que nossos modelos estão sendo criados de forma correta de acordo com o documento ER para que não haja erro na comunicação do banco de dados
Técnica:	<ul style="list-style-type: none">Esta técnica envolve a invocação de métodos que interagem com o banco de dados, utilizando dados de entrada válidos e inválidos para garantir que o código se comporte como esperado. Durante o processo, são empregados objetos simulados (mocks) ou substitutos (stubs) para isolar a unidade de teste de dependências externas, como o banco de dados real.
Critério de Finalização:	Todos os testes foram realizados sem nenhuma corrosão de dados e com os devidos dados inseridos ou exceções lançadas.
Considerações Especiais:	<ul style="list-style-type: none">Inserção de dados deve ser utilizado técnica de valor limite, consiste em inserir dados que estão no limite máximo e mínimo do modelo e inserir dados que estão acima ou abaixo do limite. Isso dá uma visão geral sobre as validações dos dados.

3.1.2 Teste estrutural

Objetivo do Teste:	Garantir a cobertura aceitável do código e possíveis bugs no sistema.
Técnica:	<ul style="list-style-type: none">Garantir que todas as partes necessárias do código-fonte sejam executadas pelo menos uma vez durante o teste.Confirmar que a lógica de decisão e os caminhos de execução estejam corretos e funcionem como esperado.Focar em testar diferentes caminhos e condições de execução dentro da unidade de código.
Critério de Finalização:	<ul style="list-style-type: none">Todos os testes planejados foram executados.Uma taxa de cobertura de todo o código-fonte acima de 70%, devido a arquivos da própria biblioteca que interferem na porcentagem acima.

Considerações Especiais:	Nenhum
--------------------------	--------

3.1.3 Teste de integração

Objetivo do Teste:	<p>Validar Fluxos de Dados:</p> <ul style="list-style-type: none"> • Confirmar que os dados fluem corretamente entre os componentes e sistemas. • Verificar que os dados sejam passados, processados e armazenados corretamente ao longo do sistema, sem perda ou corrupção.
Técnica:	A técnica utilizada será teste de integração do sistema, mas devido ao curto espaço de tempo, não será implementada em todo o sistema, será implementada em apenas dois endpoints principais do sistema.
Critério de Finalização:	O critério de finalização do teste de integração será atingido caso os endpoints façam suas devidas funções sem que haja perda de dados ou corrosão de dados e testar possíveis erros comum dos usuários como o sistema reagiria caso houvesse entrada irregular de dados.
Considerações Especiais:	Nenhum

4. Recursos

Essa seção apresenta os recursos recomendados para o projeto do QAjuda, suas principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

4.1.Trabalhadores

Como nossa equipe é composta por apenas 6 integrantes, foram alocados dois para o design dos testes, execução e validação.

Matheus Duarte - Software Tester e Designer de Testes

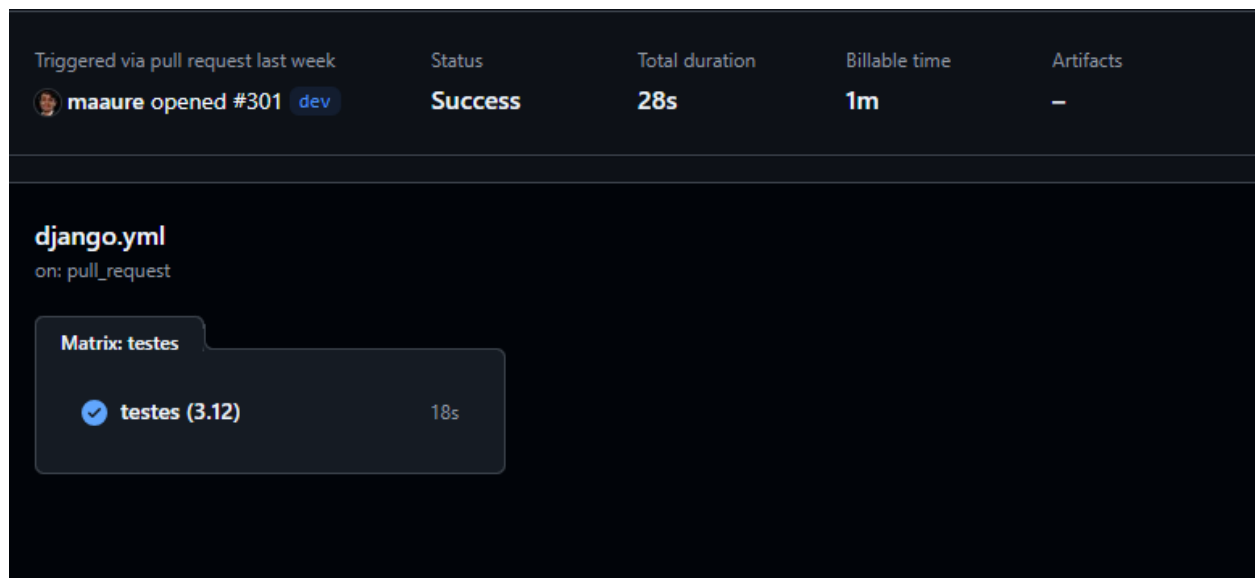
Felipe Xavier - Software Tester

4.2.Sistema


O sistema foi criado, a priori, para a web, sendo assim, para os testes só foram necessários um PC com acesso à internet para simular o acesso de um cliente no sistema.

5. Resultados

5.1. Teste de integração:




The screenshot shows a GitHub Actions workflow run for a Django project. The workflow is named 'django.yml' and is triggered on a pull request. The run was triggered via a pull request last week. The status is 'Success', the total duration is '28s', the billable time is '1m', and there are no artifacts. The workflow file is 'django.yml' and it is triggered on 'pull_request'. The workflow has a matrix named 'testes' with one job named 'testes (3.12)' which passed in 18s.

Triggered via pull request last week	Status	Total duration	Billable time	Artifacts
 maaure opened #301 <code>dev</code>	Success	28s	1m	—

django.yml
on: pull_request

Matrix: testes

Job	Status	Duration
 testes (3.12)	Success	18s

5.2. Teste Unitário:

```
Ran 15 tests in 1.325s  
  
OK  
Destroying test database for alias 'default'...
```

5.3. Teste Estrutural:

Name	Stmts	Miss	Cover
-----	-----	-----	-----
APIQAJuda/__init__.py	0	0	100%
APIQAJuda/admin.py	7	0	100%
APIQAJuda/apps.py	4	0	100%
APIQAJuda/migrations/0001_initial.py	9	0	100%
APIQAJuda/migrations/0002_alter_solicitacaovoluntariado_status.py	4	0	100%
APIQAJuda/migrations/0003_colaborador_acoes.py	4	0	100%
APIQAJuda/migrations/__init__.py	0	0	100%
APIQAJuda/models.py	90	12	87%
APIQAJuda/permissions.py	14	4	71%
APIQAJuda/serializers.py	52	0	100%
APIQAJuda/tests.py	125	3	98%
APIQAJuda/urls.py	11	0	100%
APIQAJuda/views.py	118	50	58%
config/__init__.py	0	0	100%
config/settings.py	29	0	100%
config/urls.py	8	0	100%
manage.py	11	2	82%
-----	-----	-----	-----
TOTAL	486	71	85%