**Plano de Teste**

**Eventify:**

*versão 1.1*

*Membros da Equipe: Carlos Eduardo de Almeida Rosa, Alan Francisco, Leandro Leite*

Sumário

[1 - Introdução 4](#__RefHeading___Toc1914_3194431425)

[2 - Requisitos a Testar 4](#__RefHeading___Toc1916_3194431425)

[3 - Tipos de teste 5](#__RefHeading___Toc1918_3194431425)

[3.1 - Métodos da Classe 5](#__RefHeading___Toc1920_3194431425)

[3.2 - Integração dos Componentes 7](#__RefHeading___Toc1922_3194431425)

[4 - Recursos 10](#__RefHeading___Toc1924_3194431425)

[4.1 - Ambiente de teste - Software e Hardware 10](#__RefHeading___Toc1926_3194431425)

[4.2 - Ferramenta de teste 11](#__RefHeading___Toc1928_3194431425)

[5 – Casos de Teste 12](#__RefHeading___Toc1930_3194431425)

[6 - Cronograma 14](#__RefHeading___Toc1932_3194431425)

**Histórico das alterações:**

| **Data** | **Versão** | **Descrição** | **Autor(a)** |
| --- | --- | --- | --- |
| 05/11/2024 | 1.0 | Release incial | Carlos Eduardo de Almeida Rosa |
| 20/11/2024 | 1.1 | Alteração dos casos de testes | Carlos Eduardo de Almeida Rosa |

### 1 - Introdução

Este documento descreve os requisitos a testar, os tipos de testes definidos para cada iteração, os recursos de hardware e software a serem empregados e o cronograma dos testes ao longo do projeto. As seções referentes aos requisitos, recursos e cronograma servem para permitir ao gerente do projeto acompanhar a evolução dos testes.

Com esse documento, você deve:

* Identificar informações de projeto existentes e os componentes de software que devem ser testados.
* Listar os Requisitos a testar.
* 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.

Também é possível apresentar aqui o programa que será testado.

### 2 - Requisitos a Testar

Esta seção deve conter os casos de uso e requisitos não funcionais identificados como objetos dos testes ao longo do desenvolvimento do projeto. Como, em geral, os requisitos a testar são obtidos diretamente dos requisitos do sistema, esta seção é concebida como opcional. Assim sendo, sempre que novos requisitos a testar, que não constem como requisitos do sistema, forem identificados ou, simplesmente, por questões de organização e clareza, esta seção deve ser preenchida. Dependendo das informações disponíveis, essa seção pode começar a ser preenchida já nas primeiras iterações do ciclo de vida a partir do documento de requisitos.

Caso seja necessário, liste aqui os requisitos a testar subdivididos em casos de uso e requisitos não-funcionais.

**Casos de uso:**

| **Identificador do caso de uso** | **Nome do caso de uso** |
| --- | --- |
| id 1 | Deletar evento |
| id 2 | Criar evento |
| Id 3 | Editar evento |
| Id 4 | Visualizar detalhes do evento |
| Id 5 | Login |

**Requisitos não-funcionais:**

| **Identificador do requisito** | **Nome do requisito** |
| --- | --- |
| id 6 | Linguagem Java para implementação Back-end |
| id 7 | Linguagem TypeScript para implementação do Front-end |
| Id 8 | Webdriver.io para teste e2e |
| Id 9 | Junit para teste do Back-end |

### 3 - Tipos de teste

Esta seção deve conter os tipos de testes escolhidos para cada iteração do projeto. Pode-se definir inicialmente apenas os tipos de testes que serão usadas na próxima iteração, mas é possível também já registrar eventuais tipos de teste que se espera utilizar nas demais iterações. Com base no guia de testes, indique os tipos de testes que melhor se adéquam aos requisitos, tipo da aplicação e seus recursos disponíveis e, caso necessário complemente ou forneça mais detalhes da técnica e dos critérios de completude sugeridos no guia para cada tipo de teste indicado.

* Teste de interface de usuário;
* Teste de integração;
* Teste de segurança e controle de acesso;

### 3.1 - Métodos da Classe

Para teste de funcionalidade. Aqui deve-se verificar se cada classe retorna o esperado. Se possível usar teste automatizado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo** | **Deletar um evento. O esperado é retornar para a tela principal** | | | |
| **Técnica:** | **() manual** | | **(x) automática** | |
| **Estágio do teste** | **Integração ( )** | **Sistema (x )** | **Unidade (x)** | **Aceitação ( )** |
| **Abordagem do teste** | **Caixa branca (x)** | | **Caixa preta (x)** | |
| **Responsável(is)** | **Programador(es) ou equipe de testes** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo** | **Criar um evento com sucesso. O esperado é retornar para a tela principal.** | | | |
| **Técnica:** | **() manual** | | **(x) automática** | |
| **Estágio do teste** | **Integração ( )** | **Sistema (x )** | **Unidade (x)** | **Aceitação ( )** |
| **Abordagem do teste** | **Caixa branca (x)** | | **Caixa preta (x)** | |
| **Responsável(is)** | **Programador(es) ou equipe de testes** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo** | **Editar um evento com sucesso. O esperado é retornar para a tela principal.** | | | |
| **Técnica:** | **() manual** | | **(x) automática** | |
| **Estágio do teste** | **Integração ( )** | **Sistema (x )** | **Unidade (x)** | **Aceitação ( )** |
| **Abordagem do teste** | **Caixa branca (x)** | | **Caixa preta (x)** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo** | **Ler um evento com sucesso. O esperado é abrir a tela com as informações relacionadas ao evento.** | | | |
| **Técnica:** | **() manual** | | **(x) automática** | |
| **Estágio do teste** | **Integração ( )** | **Sistema (x )** | **Unidade (x)** | **Aceitação ( )** |
| **Abordagem do teste** | **Caixa branca (x)** | | **Caixa preta (x)** | |

### 3.2 - Integração dos Componentes

Para teste de funcionalidade. Aqui deve-se verificar se as classes e métodos conseguem fazer a integração entre elas para uma sequência de ações do programa. Se possível usar teste automatizado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo** | **Verificar se o fluxo completo de criação de um evento funciona corretamente, desde a interface do usuário (frontend), passando pelo backend (Spring), até o armazenamento no banco de dados PostgreSQL. Isso inclui a criação do evento via interface Angular, a comunicação com a API Spring para salvar o evento, e a verificação de que o evento foi efetivamente armazenado no banco de dados.** | | | |
| **Técnica:** | **(x) manual** | | **(x) automática** | |
| **Estágio do teste** | **Integração (x)** | **Sistema ( )** | **Unidade ( )** | **Aceitação ( )** |
| **Abordagem do teste** | **Caixa branca (x)** | | **Caixa preta (x)** | |
| **Responsável(is)** | **Programador(es) ou equipe de testes** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo** | **Verificar se o fluxo completo de deleção de um evento funciona corretamente, desde a interface do usuário (frontend), passando pelo backend (Spring), até o armazenamento no banco de dados PostgreSQL. Isso inclui a deleção do evento via interface Angular, a comunicação com a API Spring para salvar a deleção do evento, e a verificação de que o evento foi efetivamente deletado do banco de dados.** | | | |
| **Técnica:** | **(x) manual** | | **(x) automática** | |
| **Estágio do teste** | **Integração (x)** | **Sistema ( )** | **Unidade ( )** | **Aceitação ( )** |
| **Abordagem do teste** | **Caixa branca (x)** | | **Caixa preta (x)** | |
| **Responsável(is)** | **Programador(es) ou equipe de testes** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo** | **Verificar se o fluxo completo de edição de um evento funciona corretamente, desde a interface do usuário (frontend), passando pelo backend (Spring), até o armazenamento no banco de dados PostgreSQL. Isso inclui a edição do evento via interface Angular, a comunicação com a API Spring para salvar o evento, e a verificação de que o evento foi efetivamente editado no banco de dados.** | | | |
| **Técnica:** | **(x) manual** | | **(x) automática** | |
| **Estágio do teste** | **Integração (x)** | **Sistema ( )** | **Unidade ( )** | **Aceitação ( )** |
| **Abordagem do teste** | **Caixa branca (x)** | | **Caixa preta (x)** | |
| **Responsável(is)** | **Programador(es) ou equipe de testes** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo** | **Verificar se o fluxo completo de visualização de detalhes de um evento funciona corretamente, desde a interface do usuário (frontend), passando pelo backend (Spring).** | | | |
| **Técnica:** | **(x) manual** | | **(x) automática** | |
| **Estágio do teste** | **Integração (x)** | **Sistema ( )** | **Unidade ( )** | **Aceitação ( )** |
| **Abordagem do teste** | **Caixa branca (x)** | | **Caixa preta (x)** | |
| **Responsável(is)** | **Programador(es) ou equipe de testes** | | | |

### 4 - Recursos

Esta seção deve descrever os recursos humanos, de ambiente de teste (hardware e software) e de ferramentas de automatização de testes necessários para execução dos testes que devem ser descritos nas subseções que seguem.

### 4.1 - Ambiente de teste - Software e Hardware

Descreva aqui o hardware e sua configuração, e o software. Por exemplo, o sistema operacional, browsers, servidor web, etc.

**Hardware:**

* **Máquina de Teste:**
  + **Processador:** Intel Core i5 (ou equivalente) ou superior
  + **Memória RAM:** 8GB ou superior
  + **Armazenamento:** SSD com pelo menos 50GB de espaço livre
  + **Rede:** Conexão à internet para acessar repositórios, APIs externas e o servidor web
  + **Dispositivos de Entrada:** Mouse, teclado e monitor (para testes manuais de UI)

**Software:**

1. **Sistema Operacional:**
   * **Windows 10 ou superior** (64 bits)
2. **Navegadores:**
   * **Google Chrome** (versão mais recente)
3. **Ambiente de Desenvolvimento e Execução:**
   * **Frontend:**
     + **Framework:** **Angular** (versão mais recente)
     + **Node.js** (versão recomendada para a versão do Angular em uso)
     + **npm** (Node Package Manager) para gerenciar dependências
   * **Backend:**
     + **Framework:** **Spring Boot** (versão mais recente compatível com a aplicação)
     + **Java:** **JDK 17** (ou a versão mais recente estável compatível com o Spring Boot)
     + **Servidor de Aplicação:** **Apache Tomcat** (usados pelo Spring Boot para rodar a aplicação web)
4. **Banco de Dados:**
   * **Banco de Dados Relacional:** **PostgreSQL**
     + **Versão:** 15 ou superior
5. **Ferramentas de Desenvolvimento e Testes:**
   * **IDE/Editor de Código:**
     + **IntelliJ IDEA** (para Java) e **Visual Studio Code** (para Typescript)
   * **Ferramentas de Controle de Versão:**
     + **Git** para versionamento de código (GitHub)

### 4.2 - Ferramenta de teste

* + **Ferramenta de Testes de API (Backend):**
    - **Postman**, **Insomnia** ou **Bruno** para testes manuais de APIs RESTful
  + **Framework de Testes do Back-end:**
    - **JUnit** para testes de integração no backend (Spring)
  + **Framework de Testes E2E para o Front-end:**
    - **Webdriver.io** para testes e2e (Angular)

### 5 – Casos de Teste

Modelo de caso de teste: Módulo cadastrar evento

|  |  |
| --- | --- |
| Número do Caso de Teste | CT - 001 |
| Nome | Criar evento |
| Atores | Usuário |
| Requisitos Associados | RF 12 |
| Pré-condições | Usuário cadastrado |
| Entradas | Nome do evento: Final da copa do Brasil Descrição do evento: Venha compartilhar fortes emoções. Data de início: 20/11/2024  Data de fim: 20/11/2024  Clicar em Add address  Clicar em Add payment  Clicar em Conclude |
| Saidas | Retornar para a tela principal |
| Ambientes | Acesso a aplicação usando o browser Chrome |
| Procedimentos | Criar evento - CT - 001 |
| Dependências | Browser Chrome instalado, usuario cadastrado |

|  |  |
| --- | --- |
| Número do Caso de Teste | CT - 002 |
| Nome | Deletar um evento |
| Atores | Usuário |
| Requisitos Associados | RF 02 |
| Pré-condições | Usuário cadastrado e participando de no mínimo um evento |
| Entradas | Clicar sobre algum evento  Clicar em Delete event |
| Saidas | Retornar para a tela principal |
| Ambientes | Acesso a aplicação usando o browser Chrome |
| Procedimentos | Deletar evento - CT - 002 |
| Dependências | Browser Chrome instalado, usuario cadastrado |

|  |  |
| --- | --- |
| Número do Caso de Teste | CT - 003 |
| Nome | Editar evento |
| Atores | Usuário |
| Requisitos Associados | RF 01 |
| Pré-condições | Usuário cadastrado e participando de no mínimo um evento |
| Entradas | Clicar sobre algum evento.  Nome do evento: Nome do evento editado Descrição do evento: Descrição do evento editado Data de início: 20/11/2024  Data de fim: 20/11/2024  Clicar em Add address  Clicar em Add payment  Clicar em Conclude |
| Saidas | Retornar para a tela principal |
| Ambientes | Acesso a aplicação usando o browser Chrome |
| Procedimentos | Editar evento - CT - 003 |
| Dependências | Browser Chrome instalado, usuario cadastrado |

|  |  |
| --- | --- |
| Número do Caso de Teste | CT - 004 |
| Nome | Selecionar evento |
| Atores | Usuário |
| Requisitos Associados | RF 27 |
| Pré-condições | Usuário cadastrado e participando de no mínimo um evento |
| Entradas | Clicar sobre algum evento. |
| Saidas | Abrir a tela com as informações do evento |
| Ambientes | Acesso a aplicação usando o browser Chrome |
| Procedimentos | Selecionar evento - CT - 004 |
| Dependências | Browser Chrome instalado, usuario cadastrado |

|  |  |
| --- | --- |
| Número do Caso de Teste | CT - 005 |
| Nome | Login |
| Atores | Usuário |
| Requisitos Associados | RF 22 |
| Pré-condições | Usuário cadastrado. |
| Entradas | Email: [joao@example.com](mailto:joao@example.com)  Password: password123  Clicar em Sign In |
| Saidas | Abrir a tela principal do usuário |
| Ambientes | Acesso a aplicação usando o browser Chrome |
| Procedimentos | Login - CT - 005 |
| Dependências | Browser Chrome instalado, usuario cadastrado |

### 6 - Cronograma

| **Tipo de teste** | **Duração** | **data de início** | **data de término** |
| --- | --- | --- | --- |
| planejar teste |  | 05/11/2024 | 05/11/2024 |
| projetar teste |  | 19/11/2024 | 20/11/2024 |
| implementar teste |  | 20/11/2024 | 20/11/2024 |
| executar teste |  | 20/11/2024 | 20/11/2024 |
| avaliar teste |  | 20/11/2024 | 20/11/2024 |