Grupo BUGBUSTERS: Marcio Brito, Jonathan Silva, Thales Soares, Lucca França

Disciplina: Qualidade e Teste Professora: Vânia de Oliveira Neves

Curso: Sistemas de informação Semestre: 23.1

**Trabalho Prático - Plano de Testes - Sistema API Eventos**

**Registro de Mudanças**

**Versão** 1.0 **Data** 25/01/25

**Descrição:** Versão inicial do documento, especificando quais testes serão utilizados, quais ferramentas, e requisitos funcionais e não funcionais.

**Índice**

**<ajustar na versão final>**

**1 Introdução**

O Plano de Teste é projetado para descrever o escopo, abordagem, recursos e cronograma de todas as atividades de teste do projeto Sistema API Eventos. O plano identifica os itens a serem testados, os recursos a serem testados, os tipos de teste a serem realizados, o membro responsável pelo teste, os recursos e o cronograma necessários para conclusão dos testes e os riscos associados ao plano.

**1.1 Estruturação do trabalho**

Neste trecho, apresentaremos o processo de desenvolvimento do trabalho, destacando como ele foi estruturado para garantir a transmissão clara e completa das informações necessárias à compreensão do processo de produção do produto. Serão abordadas as metodologias aplicadas para a organização eficiente das tarefas, além dos requisitos funcionais e não funcionais do sistema.

**1.1.1 No escopo**

Todos do Sistema API Evento / EZ-Learning que foram definidos nas especificações de requisitos de software precisam ser testados.

* + 1. - Papéis e Responsabilidades - API

|  |  |  |
| --- | --- | --- |
| Módulo | Papéis aplicáveis | Descrição |
| Consultar Status da edição. | Administrador, cliente | Ambos: Poder consultar o status de uma edição de evento cadastrada no sistema |
| Consultar valiações feitas em um evento. | Administrador, cliente | Ambos: Poder consultar as avaliações feitas em um evento |
| Consultar ingressos comprados por um cliente. | Administrador, cliente | Ambos: Poder consultar os ingressos que um usuário possui |
| Calcular média de avaliações | Administrador, cliente | Ambos: Poder calcular a média de avaliações de um evento específico |
| Cancelar a compra de um ingresso | Administrador, cliente | Ambos: Poder cancelar um ingresso comprado. |
| Verificar cargo de um usuário | Administrador | Administrador: Poder verificar um usuário para saber se possui a chave de Administrador ou não. |

* 1. .3– EZ-LEARNING

|  |  |  |
| --- | --- | --- |
| Nome do Módulo | Papéis aplicáveis | Descrição |
| MatriculaController | Aluno, Professor, Administrador | Aluno - um aluno pode aplicar em varios cursos, ele apenas tem acesso a informação dos cursos em que está matriculado e as suas informações pessoais;  Professor - um professor pode ver a lista de alunos matriculados nos cursos que leciona; ele não tem acesso a lista de alunos de outros cursos em que ele não é professor;  Administrador - um adm tem acesso a todas listas de alunos e professores vinculados aos cursos, ele pode excluir turmas, alunos e professores dos cursos disponiveis. |
| ProfessorController | Usuário, Administrador | Usuário - um usuário pode criar um professor, ver a lista existente de professores, incluir detalhes do professor criado, editar os dados do professor criado.  Administrador - um adm além dos direitos de usuário consegue alterar e excluir todos os professores. |
| CursoController | Aluno, Professor, Administrador | Aluno - um aluno pode visualizar os cursos em que está matriculado e acessar detalhes dos cursos, desde que esteja logado e autorizado. Não possui permissão para adicionar ou remover cursos.  Professor - um professor pode visualizar e gerenciar os cursos que leciona, adicionar informações aos cursos e acompanhar a lista de alunos matriculados nos seus cursos. O professor não pode acessar cursos de outros professores.  Administrador - um administrador tem acesso total à gestão dos cursos, podendo adicionar, editar, excluir cursos e gerenciar professores e alunos associados a cada curso. Este papel possui controle completo sobre as informações dos cursos disponíveis no sistema. |
| MatriculaService | Aluno | Responsável pela lógica de negócio das inscrições dos alunos nos cursos. Aluno seleciona um curso desejado e se matricula nele (lidando com exceções como matrícula duplicada ou curso inexistente |
|  |  |  |

**1.1.4 Fora do escopo**

- Interfaces de hardware (ambos);

- Interfaces de software (ambos);

- Lógica de banco de dados (API);

- Interfaces de comunicação (API);

- Segurança e desempenho do site (EZ-LEARNING;

- Front End de um cliente (EZ-LEARNING).

* 1. **Objetivos de Qualidade**

**1.2.1 – API de Eventos**

Os objetivos do teste são verificar a funcionalidade da API de Eventos, o projeto deve testar o fluxo de cadastro de eventos, edições, ingressos e chaves dos usuários, para garantir que todas essas operações possam funcionar normalmente.

1.2.2 – EZ\_LEARNING

Os objetivos são verificar os testes funcionais da plataforma (cadastro de alunos, cadastro de professores, inclusão de alunos nas turmas, etc) e garantir que essas funcionalidades estão devidamente operacionais.

**1.3 Papéis e Responsabilidades**

|  |  |  |
| --- | --- | --- |
| Papel | Membro | Tarefas |
| Administrador | Marcio Brito | Criação de testes unitários das classes de serviço.  Análise do sistema das especificações ISO.  Sugestão de novas idéias de funcionalidades para ter novas  implementações de teste.  Implementação de novas funcionalidades de serviço do sistema.  Apoio na criação da documentação do Readme.md do github. |
| Administrador | Jonathan Silva |  |
| Administrador | Thales Faria |  |
| Administrador | Lucca França | Criação de testes unitários das classes de serviço.  Sugestão de novas idéias de funcionalidades para ter novas  implementações de teste.  Implementação de novas funcionalidades de serviço do sistema.  Documentação do Readme.md do github.  Criação da Pipeline do Projeto. |

**2 Metodologia de Teste**

**2.1 Visão Geral**

Utilizamos as técnicas de testes demonstradas em sala de aula para executar as tarefas definidas para este trabalho.

**2.2 Fases de Teste**

Teste de unidade (métodos das classes de serviço são testados individualmente)

utilizando JUnit - API e EZ-LEARNING)

● Teste de integração (Teste de requisições utilizando Postman) - API;

● Teste de sistema: teste do sistema utilizando Selenium – EZ-LEARNING;

● Teste de mutação utilizando Pit Runner - API;

● Teste Funcional – EZ-LEARNING.

**2.3 Completude do Teste**

● A taxa de 100% de execução dos testes é obrigatória, a menos que um motivo

claro seja fornecido.

● Mínimo de 80% de cobertura feita.

● A taxa de 80% de aprovação dos testes é obrigatória.

**2.4 Critérios de Suspensão e Requisitos de Retomada**

Se os membros da equipe relatarem que há 40% dos casos de teste com falha, suspenda o teste até que a equipe de desenvolvimento corrija todos os casos com falha.

2.5 Atividades do projeto, estimativas e cronograma

|  |  |  |
| --- | --- | --- |
| Tarefa | Membros | Estimativa de esforço (por pessoa / total) |
| Criar testes utilizando JMeter | Todos | 6 h/h / 24h/h |
| Executar os testes | Todos | 1 h/h / 4 h/h |
| Criar o relatório de testes | Qquer Adm | 2 h/h / 2h/h |
| Entregar os testes | Todos | 1 h/h / 4h/h |
| Implementação de Pipeline de CI/CD | ??? |  |
| Implementar Testes Unitários |  | Há redundância?? |
| Implementar Testes de Requisitos Funcionais |  | Há redundância?? |
| TOTAL |  |  |

**3 Entregáveis de Teste**

Os entregáveis de teste que serão fornecidos:

1. Plano de teste.

2. Casos de teste.

3. Relatório de erros.

4. Simuladores de ferramentas de teste.

5. Dados de teste.

**4 Necessidades de Recursos e Ambiente**

**4.1 Ferramentas de Teste**

- Necessário um servidor de banco de dados com MySQL instalado e um servidor

Web com Apache instalado - API;

Necessário uma ferramenta para teste de requisições (Postman, por exemplo) - API;

Necessário uma IDE para conseguir abrir o código (para o API utilizamos o IntelliJ, para o EZ-LEARNING utilizamos o Eclipse;

Necessário a ferramenta Selenium para realização dos testes automatizados – EZ- LEARNING.

**4.2 Ambiente de Teste**

Os seguintes softwares são necessários, além de softwares específicos do cliente.

- Windows 8 ou superior

- Office 2013 ou superior

- IntelliJ Community Edition

- Eclipse

- Postman

- Selenium

- ???

**5 Termos / Acrônimos**

**TERMO /**

**ACRÔNIMO**

**DEFINIÇÃO**

API *Application Program Interface*

IDE *Integrated Development Environment*

CI/CD *Continuous Integration / Continuous Delivery*