Plano de Testes - Sistema API Eventos

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

Disciplina: Qualidade e Teste

Professora: Vânia de Oliveira Neves

Curso: Sistemas de Informação

Semestre: 24.2

**Índice**

1. Introdução

1.1 Estruturação do trabalho

1.1.1 No escopo

1.1.4 Fora do escopo

1.2 Objetivos de Qualidade

1.2.1 API de Eventos

1.2.2 EZ\_LEARNING

1.3 Papéis e Responsabilidades

2. Metodologia de Teste

2.1 Visão Geral

2.2 Fases de Teste

2.3 Completude do Teste

2.4 Critérios de Suspensão e Requisitos de Retomada

2.5 Atividades do projeto, estimativas e cronograma

3. Entregáveis de Teste

4. Necessidades de Recursos e Ambiente

4.1 Ferramentas de Teste

4.2 Ambiente de Teste

5. Termos / Acrônimos

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 e recursos a serem testados, os tipos de teste a serem realizados, os critérios de testes utilizados, o membro responsável pela tarefa, os recursos necessários e o cronograma revisado 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 dos testes. 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.

Papéis e Responsabilidades - API

|  |  |  |
| --- | --- | --- |
| **Módulo** | **Papéis aplicáveis** | **Descrição** |
| EventService | Administrador, cliente | Ambos: Poder consultar o status de uma edição de evento cadastrada no sistema. |
| RatingService | Administrador, cliente | Ambos: Poder consultar as avaliações feitas em um evento. |
| Consultar evento inscrito | Administrador, cliente | Ambos: Poder verificar se está inscrito em um evento. |
| Checar identidade do usuário | Administrador | Verificar se o usuário tem chave válida para operar o sistema |

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 (API).

1.2 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 da classe UserService (API) e MatriculaController (EZ) Análise do sistema das especificações ISOElaboração do Plano de Teste Sugestão de novas ideias de funcionalidades para ter novas implementações de teste. Implementação de novas funcionalidades de serviço do sistema. Apoio na criação da documentação do github. |
| Administrador | Jonathan Silva | Criação de testes unitários da classe EventService (API) e CursoController (EZ) Análise do sistema das especificações ISO. Rodar teste automatizado (Selenium) Sugestão de novas ideias de funcionalidades para ter novas implementações de teste. Implementação de novas funcionalidades de serviço do sistema. Apoio na criação da documentação do github. |
| Administrador | Thales Faria | Criação de testes unitários da classe EventService (API) e MatriculaService (EZ) Análise do sistema das especificações ISO. Rodar teste de integração - Postman Sugestão de novas ideias de funcionalidades para ter novas implementações de teste. Implementação de novas funcionalidades de serviço do sistema. Apoio na criação da documentação do github. |
| Administrador | Lucca França | Criação de testes unitários da classe RatingService (API) e CursoService / ProfessorController (EZ) Análise do sistema das especificações ISO. Sugestão de novas ideias de funcionalidades para ter novas implementações de teste. Implementação de novas funcionalidades de serviço do sistema. Apoio na criação da documentação do github. |

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)** |
| Implementar Testes de Requisitos Funcionais | Todos | 2 h/h / 8h/h |
| Implementar Testes Unitários | Todos | 2 h/h / 8 h/h |
| Executar os testes | Todos | 1 h/h / 4 h/h |
| Criar o relatório de testes | Qualquer Administrador | 2 h/h / 2h/h |
| Entregar os testes | Todos | 1 h/h / 4h/h |
| **TOTAL** |  | **26 h/h** |

3 Entregáveis de Teste

Os entregáveis de teste que serão fornecidos:

* Plano de teste;
* Casos de teste;
* Simuladores de ferramentas de teste;
* 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 (e plugins para testes);
* Eclipse (e plugins para testes);
* Postman;
* Selenium.

5 Termos / Acrônimos

|  |  |
| --- | --- |
| **TERMO / ACRÔNIMO** | **DEFINIÇÃO** |
| API | Application Program Interface |
| IDE | Integrated Development Environment |