## 

## *Projeto final da turma*

## *avançada do curso:*

## *Bugou! QA tá on!*



## ***SQUAD: Alone S***

## *Lucila Y. M. Beger*

Mmm

Luic

## 

## 

## 

## 

## 

## ***Plano de Teste - Instituto Joga Junto***

**1. Introdução**

Este plano de teste detalha as atividades a serem realizadas pela equipe de QA para garantir a qualidade do software do Instituto Joga Junto. O objetivo principal é garantir que o software atenda aos requisitos definidos pelo cliente e funcione de acordo com as expectativas dos usuários.

**2.** **Arquitetura**

O site é desenvolvido com Node JS no backend, que está hospedado no EC2, e react no frontend, hospedado na AWS Amplify, que utilizam o protocolo REST para a comunicação.

**3. Escopo do Teste**

O escopo deste plano de testes abrange todo o software do Instituto Joga Junto, incluindo:

* Módulo de Cadastro de Usuário
* Módulo de Gerenciamento de Produtos
* Módulo de Estoque
* Módulo de Pedidos
* Módulo de Relatórios
* Interface Web
* API

**4. Referências**

* Documentação do Produto
* Swagger da API
* Ambiente de Teste
* Plano de Teste Detalhado - Cadastro de Usuário

**5. Abordagem de Teste**

A abordagem de teste para este projeto será baseada nos seguintes princípios:

* Teste baseado em requisitos: Os testes serão baseados nos requisitos definidos na documentação do produto.
* Teste funcional: O foco principal será testar as funcionalidades do sistema, garantindo que elas funcionem conforme o esperado.
* Teste manual e automatizado: Uma combinação de testes manuais e automatizados será utilizada para garantir a cobertura completa dos requisitos.
* Testes em diferentes navegadores e dispositivos: Os testes serão realizados em diferentes navegadores web (Chrome, Firefox, Edge) e dispositivos (desktops, smartphones) para garantir a compatibilidade.
* Testes de internacionalização: Os testes serão realizados em diferentes idiomas para garantir que o sistema funcione corretamente para usuários de todo o mundo.

**6. Ferramentas e ambiente de Teste**

* Ferramentas de teste:
  + Selenium
  + Python
  + Behave
  + Postman
  + Computador com Windows 10
  + Navegadores Chrome, Firefox, Edge e Opera
  + Smartphone com Android 8 ou superior
* Ambiente de teste:
  + <https://projetofinal.jogajuntoinstituto.org/>
  + https://apipf.jogajuntoinstituto.org/swagger/

**7. Classificação de Bugs**

Os bugs encontrados durante os testes serão classificados de acordo com a seguinte severidade:

* 1-Crítico/Blocker: Bug que impede o uso do sistema ou causa perda de dados.
* 2-Grave: Bug que causa problemas graves na usabilidade do sistema ou impede a execução de algumas funcionalidades importantes.
* 3-Médio: Bug que causa problemas na usabilidade do sistema ou impede a execução de algumas funcionalidades menos importantes.
* 4-Baixo: Bug de problemas estéticos no sistema ou erros ortográficos.

**8. Priorização de Bugs**

Os bugs serão priorizados de acordo com a seguinte critério:

* Prioridade 1 - Alta: Bugs de severidade 1 ou 2 que afetam funcionalidades críticas do sistema.
* Prioridade 2 - Média: Bugs de severidade 2 ou 3 que afetam funcionalidades importantes do sistema.
* Prioridade 3 - Baixa: Bugs de severidade 3 ou 4 que não afetam funcionalidades críticas do sistema.

**9. Cenários de Teste**

Os cenários de teste serão criados para cada módulo do software, com base nos requisitos definidos na documentação do produto e incluirão os seguintes elementos:

* Descrição do cenário: Uma breve descrição do que o cenário de teste irá testar.
* Passos: Os passos que o usuário irá realizar para executar o cenário de teste.
* Resultado esperado: O resultado que o sistema deve apresentar para que o cenário de teste seja considerado bem-sucedido.
* Resultado obtido: O resultado que o sistema realmente apresentou durante a execução do cenário de teste.
* Passou/Falhou: Indica se o cenário de teste foi bem-sucedido ou falhou.
* Observações: Quaisquer observações relevantes sobre o cenário de teste.

Para visualizar a tabela com os cenários de teste, é só acessar o link abaixo.

[CENÁRIOS DE TESTE - TABELA](https://docs.google.com/spreadsheets/d/1cV9SVY0UtJ8v--iEro_4OUYARURqmKWZK5d_GrGu1Kc/edit?usp=sharing)

**11. Critérios de Aprovação**

O software do Instituto Joga Junto precisa atender aos seguintes critérios:

* Todos os requisitos funcionais e não-funcionais definidos na documentação do produto foram testados com sucesso.
* Não foram encontrados defeitos críticos que impeçam o uso do sistema.
* A performance do sistema está dentro dos limites aceitáveis.
* A usabilidade do sistema é satisfatória para os usuários.
* O sistema está compatível com diferentes navegadores web e dispositivos.
* O sistema funciona corretamente em diferentes idiomas.

**12. Riscos**

Os principais riscos associados a este projeto de teste são:

* Falta de tempo para a execução completa dos testes.
* Mudanças nos requisitos do produto durante o processo de teste.
* Dificuldades na integração com outros sistemas.

**13. Métricas de Teste**

A equipe de QA irá coletar e analisar as seguintes métricas durante a execução dos testes:

* **Cobertura de testes:** A porcentagem de requisitos funcionais do software que foram testados.
* **Defeitos por módulo:** O número de defeitos encontrados em cada módulo do software.
* **Defeitos por severidade:** O número de defeitos encontrados em cada nível de severidade (crítico, alto, médio, baixo).
* **Tempo médio de resolução de defeitos:** O tempo médio que a equipe de desenvolvimento leva para corrigir os defeitos encontrados.
* **Taxa de aprovação dos testes:** A porcentagem de cenários de teste que foram executados com sucesso.

As métricas coletadas serão utilizadas para:

* Monitorar o progresso da equipe de teste.
* Identificar áreas do software que precisam de mais testes.
* Priorizar a correção de defeitos.
* Avaliar a qualidade geral do software.