**MODELO DE PLANO DE TESTE JAE**

**001**

**1 IDENTIFICADOR DO PLANO DE TESTE**

JAE-001

**2 REFERÊNCIAS**

-SAMPLE MASTER TEST PLAN TEST PLAN IDENTIFIER RS-MTP01.3 - Systeme Evolutif Limited.

-MODELO DE PLANO DE TESTE IEEE 829 - Prof. Lori Machado Filho.

**3 INTRODUÇÃO**

Este plano de testes foi elaborado como parte da matéria de Projeto Integrador II, com a finalidade de verificar e validar o software a ser desenvolvido, permitindo verificar erros, defeitos e falhas em dito software com o objetivo de deixá-lo funcional.

**4 ITENS DE TESTE (FUNÇÕES)**

Todas as partes do projeto desenvolvidas pela equipe de desenvolvimento do software devem ser testadas.

• Testar a conectividade do banco de dados com o azure entre as máquinas.

• A segurança do banco de dados e do website deverão ser testadas.

• O frontend e o backend dos componentes desenvolvidos deverão ser testados.

• O frontend e o backend deverão ser testados individualmente.

• A integração do website com o host deve ser testada.

**5 PROBLEMAS DE RISCO DE SOFTWARE**

\*A aplicação a ser desenvolvida é uma aplicação web, então ela possui vulnerabilidades de segurança acentuadas.

\* A equipe não possui experiência com javascript, logo é possível que algumas funções não sejam implementadas corretamente.

\*A conexão com o banco de dados deve ser estável e segura.

**6 CARACTERÍSTICAS A TESTAR**

A escala para a classificação do risco de cada recurso é (A - Alto, M - Médio, B - Baixo).

1. Funcionalidade do banco de dados, testando se é possível ler e escrever no banco sem problemas, bem como sua conectividade e segurança. (M)

2. A funcionalidade da front-end do website, verificando que ele apresenta uma funcionalidade correta da interface gráfica, de todos os menus e que sua aparência está de acordo com o planejado.(M)

3. A funcionalidade da back-end, observando um funcionamento da lógica interna da programação do software.(A)

4. A segurança do banco de dados e do website.(M)

5. A implementação do website, testando que o mesmo está no ar e funcionando. (A)

**7 CARACTERÍSTICAS QUE NÃO DEVEM SER TESTADAS**

Os seguintes testes deverão ser realizados:

\*Teste de integração e de segurança do banco de dados.

\*Teste unitário do controle de estoque, do backend e do frontend separadamente.

\*Teste unitário do controle de cadastro, do backend e do frontend separadamente.

\*Teste unitário do controle de orçamento, do backend e do frontend separadamente.

\*Teste unitário do controle de agendamento, do backend e do frontend separadamente.

\*Teste de segurança do login do website.

\*Teste de integração do host do website.

**8 ABORDAGEM (ESTRATÉGIA)**

Como a equipe de desenvolvimento consiste de dois desenvolvedores, o desenvolvedor que não construiu o componente será a pessoa responsável por testá-lo. Tal pessoa irá testar os aspectos de uso do componente tentando “quebrar” o software e procurando e documentando defeitos, erros e falhas para uma correção futura, caso tais eventos não impeçam o uso do software para testes. Caso o uso do sistema seja impedido por tais eventos, o testador irá alertar o resto da equipe para que o software possa ser reparado o mais rápido possível, e então mais testes devem ser feitos. Os testes de integração irão utilizar uma lógica contínua com testes de “fumaça” por serem bem compatíveis com desenvolvimento ágil.

O encarregado do teste deverá tentar acessar o componente de maneira imprópria, tentando burlar a sua segurança. Bem como os testes de integração, os testes de segurança serão feitos pelo desenvolvedor que não construiu o componente.

Os testes de unidade irão ocorrer com a criação de código para testar a funcionalidade de certos componentes do projeto. Esses testes têm a finalidade de examinar a funcionalidade interna do código dos componentes do projeto de maneira isolada. Observando os seus printouts e defeitos nos dados gerados pelo software. Como os outros testes, o encarregado dos testes de unidade será o desenvolvedor que não trabalhou no seu desenvolvimento.

Caso ambos desenvolvedores trabalhem em um componente, ambos irão testá-lo.

**9 CRITÉRIOS DE APROVAÇÃO/FALHA DO ITEM**

Os testes são concluídos com aprovação quando:

• Todos os casos de teste concluídos nos testes de unidade sem ou com mínimos defeitos, com tanto que tais defeitos sejam triviais ou insignificantes.

• No caso dos testes de segurança os testes terminam quando for determinado que é impossível ou improvável que ocorram brechas de segurança no componente.

• Caso o teste de integração conclua sem ou com mínimos defeitos, com tanto que tais defeitos sejam triviais ou insignificantes.

Os testes são concluídos com falha e devem ser reparados e re-testados quando:

•Defeitos impedem a continuação do teste.

•Caso um teste de segurança falhe.

•Caso um ou mais defeitos significativos que impeçam o funcionamento, aparência e ou resultado se apresentem em um teste de unidade.

•Caso uma funcionalidade falhe ou apresente defeitos significativos no teste de integração.

**10 CRITÉRIOS DE SUSPENSÃO E REQUISITOS DE RETOMA**

Testes deverão ser suspensos caso o número de erros observados torne a continuação do teste redundante, visto que grandes correções e mudanças deverão ser feitas. Os testes devem ser retomados após tais correções.

**11 ENTREGAS DE TESTE**

Após a conclusão de um teste as seguintes documentações devem ser entregues pelo desenvolvedor encarregado dos testes:

• Ferramentas e suas saídas.

• Geradores estáticos e dinâmicos.

• Logs de erros e logs de execução.

• Relatórios de problemas e ações corretivas.

**12 TAREFAS DE TESTE RESTANTES**

O grupo está presentemente contemplando desenvolver um sistema para gerar notas fiscais reais invés de apenas gerar notas fiscais de exemplo para demonstrar a funcionalidade do software. Caso este componente seja realmente desenvolvido ele precisará ser submetido a testes.

**13 NECESSIDADES AMBIENTAIS**

Não existem requisitos especiais para este plano de teste.

**14 NECESSIDADES DE PESSOAL E TREINAMENTO**

A equipe de desenvolvimento consiste de dois desenvolvedores. Os componentes do projeto serão testados pelo desenvolvedor que não trabalhou nele, caso ambos desenvolvedores trabalhem em um componente, ambos irão fazer os testes de tal componente.

**15 RESPONSABILIDADES**

O desenvolvedor realizando o teste estará encarregado de tomar as decisões críticas de tal teste, incluindo a sua definição de riscos. No entanto, testes realizados por um desenvolvedor serão observados pelo outro desenvolvedor, que irá determinar se o teste falhou ou foi aprovado.

**16 CRONOGRAMA**

O cronograma dos testes está agendado no scrum board. Com o teste do banco de dados ocorrendo imediatamente após a definição do mesmo, os outros testes serão feitos na última etapa do desenvolvimento. O tempo alocado para os testes é entre 4 e 5 horas.

**17 PLANEJAMENTO DE RISCOS E CONTINGÊNCIAS**

• O time é novato em relação ao scrum e o javascript, logo atrasos na entrega do software são um risco.

• Atrasos no treinamento sobre o aplicativo e/ou ferramentas.

• Alterações nos requisitos ou projetos originais podem ocorrer dependendo do andamento do projeto.

**18 APROVAÇÕES**

O desenvolvedor que não trabalhou no componente será quem determina se os testes estão prontos ou não, caso ambos os desenvolvedores trabalhem no teste, ambos terão que chegar a um consenso.

**19 GLOSSÁRIO**

Nenhum termo no glossário.