A Fome não Pode Esperar

Plano de Testes

Versão 1.1

Histórico de Revisão

Data	**Versão**	**Descrição**	**Autor**
30/10/2019	<1.0>	Release Inicial	Antonio Luciano Ferreira Neto
31/10/2019	<1.1>	Finalização e revisão	Antonio Luciano Ferreira Neto

Sumário

Plano de Testes

Introdução

Este documento relaciona os casos de uso a serem testados, os estágios de testes, método de qualificação, detalhamento dos tipos de testes, alvos de testes, a estratégia adotada para a execução dos testes, os recursos humanos necessários, bem como os produtos que serão gerados.

Escopo

O projeto A fome não pode esperar deverá ser submetido a testes de unidade (Caixa branca) usando o framework JUnit, funcional, usabilidade e aceitação.
Os testes de unidade irão avaliar isoladamente a interface gráfica, e todos os outros componentes internos do projeto como o código.
Os testes funcionais que testa os requisitos funcionais e os casos de uso.
Os testes de usabilidade e aceitação que apresenta o produto final ao cliente para validação e últimos ajustes.

Estágios de Teste

Definem o momento do ciclo de vida do software em que são realizados testes por pessoas diferentes daquelas que o programaram. Entretanto, considerando a divisão das tarefas de teste em quatro níveis relacionados ao escopo do software, estão previstos para o projeto **A Fome não Pode Esperar** os seguintes estágios de teste:

- Teste de Integração: são realizados para verificar basicamente se as unidades testadas de forma individual executam corretamente quando colocadas juntas, isto é, quando integradas. Os testes são realizados pelo Analista de Testes.
- Teste de Sistema: são realizados pelo Analista de Testes, visando a execução do sistema, dentro de um ambiente operacional controlado, para validar a exatidão e perfeição na execução de suas funções.
- Teste de Aceitação ou Homologação: são os testes finais de execução do sistema, realizados pelos usuários, visando verificar se a solução atende aos objetivos do negócio e a seus requisitos, no que diz respeito à funcionalidade e usabilidade, antes da utilização no ambiente de produção.

Tipos de Testes

Seguem abaixo os tipos de testes a serem aplicados ao projeto **A Fome não Pode Esperar:**

- **Funcional:** grupos de testes que avaliam se o que foi especificado foi implementado.
- **Integridade de dados:** verificar se os dados do sistema foram incluídos, alterados, excluídos e pesquisados corretamente no banco de dados. Além de validar conteúdos de campos.
- **Performance:** mede e avalia o tempo de resposta de cada transação dos requisitos sensíveis ao tempo.
- **Usabilidade:** verificam o nível de facilidade de uso do software pelos usuários.
- **Acessibilidade**: verifica se a interface do usuário fornece o acesso apropriado às funções do sistema e a navegação adequada. Além disso, estes testes garantem que os objetos dentro da interface do usuário funcionem de acordo com os padrões definidos pelo cliente.

Recursos necessários

Recursos Humanos

O analista de testes do projeto ficou incumbido da realização deste com a homologação do cliente nos testes funcionais, usabilidade e aceitação. Como estamos em constante desenvolvimento ainda nossos testes são baseados em TDD e com a homologação do cliente.

Recursos Computacionais

Os recursos computacionais para a realização dos testes foi utilizada uma máquina com as configurações mais próximas o possível das máquinas que serão utilizadas pelo usuário final, tentando assim, simular o ambiente final em que o programa será executado.

Estações de Trabalho	**Descrição**
Notebook HP G42	*Windows 8.1 PRO, RAM 3GB, Processador Intel Core i3 CPU M350 @ 2.27 GHz.*

Riscos e Restrições

Nessa primeira parte não tem nenhum risco a ser mencionado, mesmo com o teste do framework não sendo aprovado, cabe só ao desenvolvedor corrigir os erros do código. Quanto às restrições, por ser um projeto ainda em construção, diversos outros testes não foram feitos, mas não vão ser descartados. Irão ser feitos posteriormente quando as funcionalidades forem sendo atreladas ao projeto e logo após os testes, é feita a atualização deste documento

Produtos Gerados

Teste Funcional

Objetivo do Teste:	Garantir que as funcionalidades do sistema, especificadas nos casos de usos, gerando os resultados esperados.
Técnica:	Executar cada caso de uso funcional através de seu fluxo principal e dário, usando dados válidos e inválidos, para verificar o seguinte: ● Os resultados esperados ocorrem quando dados válidos são usados. ● As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados.
Critério de zação:	● Todos os testes planejados foram executados. ● Todos os defeitos identificados foram tratados.
Considerações ciais:	Nenhum

Teste de caixa branca

Objetivo do Teste:	Previne o aparecimento de “BUG’S” oriundos de códigos mal escritos junto com situações de sucesso e de falha
Técnica:	Foram testados os códigos junto com suas classes na a IDE de desenvolvimento o framework utilizado foi o JUnit ado como biblioteca do NetBeans.
Critério de zação:	● Todos os testes planejados foram executados. ● Todos os defeitos identificados foram relatados e anexados para que os desenvolvedores possam corrigir o código.
Considerações ciais:	Nenhum

Teste de usabilidade e acessibilidade

Objetivo do Teste:	● Verificar se a navegação através dos alvos de teste reflete as funções e os requisitos do negócio apropriadamente. ● Objetos e características da janela, tais como menus, tamanho, posição, estado e foco conformam-se aos padrões. ● Homologar junto com o cliente, as duas telas funcionais criadas.
Técnica:	● Criar ou modificar os testes para cada janela para verificar a

	navegação e os estados de objeto apropriados para cada janela e objetos da aplicação. ● Observar grupos de usuários usando a interface, analisando a taxa de aprendizado dos mesmos com o sistema e a aceitação da interface pelos usuários.
Critério de zação:	● É verificado que cada janela permanece consistente com a versão de comparação ou dentro de padrões aceitáveis. ● É verificado que o usuário consegue usar a interface sem precisar de treinamento e a considera agradável.
Considerações ciais:	Nem todas as propriedades para objetos foram criadas, mas foram testadas saíram satisfatórias ao usuário.

Referências

#	Nome	Data/versão
url	https://www.devmedia.com.br/junit-tutorial/1432	*26/10/2019*
doc	Projeto: Gerenciador de Serviços Automotivos (GSA)	Versão: 1.1cits
doc	Documento modelo para testes by https://www.cits.br/	29/10/2019

Anexos

Resultado do teste JUnit: