**HiFood**

Documento de Teste de Software

Recife, 21 de julho de 2014

Conteúdo

1. Introdução 2

1.1 Objetivos 2

1.2 Público Alvo 2

1.3 Organização do documento 3

2. Recursos necessários para os testes 3

3. Estratégias e técnicas de teste 3

4. Casos de teste 3

Bibliografia 5

Anexos 5

# Introdução

Este é o documento de teste de software do projeto HiFood – Sistema de Gerenciamento de Fast Food.

## Objetivos

O objetivo deste documento é descrever o planejamento geral das atividades de teste do projeto HiFood – Sistema de Gerenciamento de Fast Food, bem como critérios de aceitação dos artefatos a serem testados. A seguir, serão identificados os casos de teste a serem utilizados e descritas as estratégias a serem utilizadas.

## Público Alvo

Esse documento se destina principalmente aqueles relacionados às fases de requisitos, desenvolvimento, gerenciamento da qualidade e testes.

Os testes estão presentes desde a concepção até a transição dentro do processo, portanto, o presente documento é indicado a todos os envolvidos no ciclo de vida do software.

## Organização do documento

Este documento foi organizado em cinco sessões. A primeira se trata da introdução, contendo uma breve descrição do documento. A segunda sessão descreve recursos necessários para os testes. A terceira sessão descreve as estratégias e técnicas de teste. A quarta sessão descreve os casos de testes e a forma na qual devem ser testados. E, por fim, é na quinta sessão que o documento é formalmente aprovado.

# Recursos necessários para os testes

Os recursos necessários para a execução dos testes, incluem:

* Hardware – Um computador com configuração igual ou superior a: memória RAM de 2GB, HD com 160 GB livres, Core 2 Duo 2.0, Placa de Rede Ethernet 10/100.
* Software – Windows XP ou superior. Navegador Internet Explorer 6.0 ou Mozilla Firefox 3.6 (ou superiores).
* Recursos humanos – Dois analistas de teste. Responsáveis por definir, especificar e realizar os testes específicos a serem conduzidos. Dessa forma, avaliando a qualidade do produto.

# Estratégias e técnicas de teste

**Testes Unitários –** Ostestes de unidade de um código tem como objetivo detectar erros ou defeitos em certo nível de classe do sistema. Se trata de um teste Funcional, de Caixa Branca.

**Testes de Integração** – Depois de implementados, os módulos do sistema devem ser integrados. Os testes de integração consistem basicamente num conjunto dos testes unitários cumulativos construídos para testar especificamente o funcionamento de um componente individualmente, após a integração de componentes e/ou subsistemas. Nesta integração, podem ser empregadas ferramentas de automação. Assim como o Teste Unitário, este também é um teste de Caixa Branca.

**Tipos de Teste**

#### Teste de Banco de Dados – TBD

* Verificar se as informações do ator podem ser fornecidas e recuperadas.
* Verificar se o conteúdo e as categorias podem ser inseridos e exibidos.
* Verificar se os perfis do usuário e as informações de conta podem ser fornecidos e exibidos.
* Verificar se as informações de uso específicas do usuário são rastreadas.

1. **Teste Funcional - TF**

* Verificar se os usuários conseguem ver as informações sobre as quais solicitaram mensagens.
* Verificar se as mensagens são endereçadas aos usuários quando o conteúdo chega.
* Verificar se a inserção de conteúdo automático está funcionando.
* Verificar se a aprovação do editor provoca a inserção de conteúdo não-automático.
* Verificar se o conteúdo obsoleto foi excluído.
* Verificar se os relatórios são exportados devidamente em formato desejado do projeto.

#### Teste de Configuração - TC

* Verificar a operação usando o Microsoft Internet Explorer 7 ou superior e Mozilla Firefox 3.5 ou superior, conforme Plano de Implantação.

# Casos de teste

Nesta seção serão descritos os casos de teste a serem utilizados, bem como a forma que deverão ser testados (procedimento de teste).

Esta seção será ser realimentada com os resultados obtidos dos testes. Segue abaixo o modelo do quadro a ser preenchido para cada caso de teste:

|  |  |
| --- | --- |
| CASO DE TESTE | |
| Código | Identificador do caso de teste. |
| Finalidade | Definição dos objetivos do caso de teste (opcional). |
| Entradas | Valores das entradas para o caso de teste, ou identificação dos arquivos a serem utilizados como entrada. |
| Resultados esperados | Resultados que são aceitos como corretos para o caso de teste. |
| Dependências | Identificadores dos casos de teste que devem ser executados antes deste. |
| Procedimentos de teste | |
| Preparação | Descrever as tarefas a serem realizadas antes do início da execução dos casos de teste como, por exemplo, preparação de ambiente. |
| Inicialização | Descrever as tarefas necessárias para iniciar a execução dos casos de teste. Ex.:   1. Executar programa receptor de arquivos PRA001 no servidor. 2. Executar programa transmissor de arquivos PTA001 nos clientes. |
| Execução | Descrever as tarefas a serem realizadas durante a execução dos casos de teste, como, por exemplo, a intervenção de um operador. Descrever também como os testes serão acompanhados e, se for o caso, como medidas serão observadas e registradas (ex.: tempo de resposta, número de transações por segundo, etc.). |
| Reorganização | Descrever as tarefas a serem realizadas após o término normal da execução dos casos de teste. Ex.: recuperação do ambiente original. |
| Interrupção | Descrever as tarefas a serem executadas em caso de interrupção não planejada dos testes. |
| Recursos Específicos | Software, hardware, pessoal ou outro recurso necessário à execução desse procedimento de teste especificamente (opcional). |
| AVALIAÇÃO DO CASO DE TESTE | |
| Testador responsável |  |
| Período de teste |  |
| Resultados obtidos |  |
| Observações do testador |  |

# Aprovação Formal

# Bibliografia

*[1] Ronaldo Lopes de Oliveira,* ***DTS – Modelo de Documento de Teste de Software*, *Versão 1.0, agosto de 2005****.*

*Meta-Modelo usado como guia para definição do formato e conteúdo deste documento.*