Plano de Teste

Meraki – E-Commerce

Versão 1.0

Informação do Documento de Requisitos

|  |  |
| --- | --- |
| Título do documento | Documento de Teste – Meraki (E-Commerce) |
| Autor | Vitor Keidi Sakassegawa |
| Comentários |  |
| Nome do arquivo | PlanoTeste.docx |

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 09/05/2016 | 1.0 | Documento de plano de testes para a 1ª iteração: PlanoTeste.docx | Vitor Keidi Sakassegawa |

Conteúdo

[Plano de teste 4](#_Toc450573678)

[1. Introdução 4](#_Toc450573679)

[1.1 Objetivos 4](#_Toc450573680)

[1.2 Contexto do projeto 4](#_Toc450573681)

[1.3 Escopo 4](#_Toc450573682)

[1.4 Identificação do projeto 5](#_Toc450573683)

[2. Requisitos para teste 6](#_Toc450573684)

[2.1. Teste Unitários 6](#_Toc450573685)

[3. Estratégia de teste 6](#_Toc450573686)

[3.1. Teste Unitário 6](#_Toc450573687)

[3.2. Ferramentas de teste 6](#_Toc450573688)

[4. Recursos 7](#_Toc450573689)

[4.1. Regras 7](#_Toc450573690)

[4.2. Sistema 7](#_Toc450573691)

# 

# Plano de teste

|  |
| --- |
| 1. Introdução |

|  |
| --- |
| Objetivos |

Este documento reúne as instruções para planejamento dos testes a serem aplicados ao final da 1ª iteração do projeto da Loja Virtual - Meraki a fim de validar por completo os artefatos produzidos nesse período.

O planejamento comporta o modelo de testes a ser utilizado. Um procedimento de testes estabelece as instruções a serem seguidas para realizar os casos de testes, ou seja, trata-se de um manual de testes. Para isso, é necessário levar em conta os seguintes pontos: as condições iniciais para se testar um determinado requisito do sistema, o ponto de partida do teste, as ações a serem realizadas, os resultados esperados e, por fim, os critérios para aprovação.

Além de avaliar se o sistema está funcionando corretamente (objetivando a procura por erros), os testes servem também para verificar como o sistema se comporta durante a execução de uma tarefa, podendo avaliar a performance do mesmo, por exemplo.

|  |
| --- |
| Contexto do projeto |

O projeto representa uma loja virtual de produtos esportivos, no qual tem como objetivo facilitar a compra de itens de esportes, utilizando o browser.

O projeto segue a arquitetural MVC (Modelo, Visão e Controle) utiliza a linguagem Java e o Banco de Dados PostgreSQL

|  |
| --- |
| Escopo |

O plano de testes deve documentar os casos de teste, as ações e os procedimentos e parâmetros utilizados nos testes. Devem ser testados desde os casos mais comuns até situações para as quais o sistema não está programado (o que fornece uma boa ideia das limitações do sistema).

Vale ressaltar que nenhum teste é definitivo e prova que o programa está correto, no entanto, a prática deles é indispensável para consolidar um bom projeto. Durante o desenvolvimento do sistema, os modelos de testes podem ser revistos, criando-se assim novas versões de testes que contenham as anteriores e levem em conta novas funcionalidades do sistema.

1 – Teste unitário

Os testes unitários foram realizados durante a implementação da camada de dados e negócios para cada funcionalidade criada. Testes foram escritos para ficarem documentados e serem utilizados sempre que necessários, no entanto, essa atividade foi realizada manualmente, isto é, sem o auxílio de ferramentas para automatizar esses testes unitários, mas com sucesso atingido.

|  |
| --- |
| Identificação do projeto |

A tabela abaixo identifica a documentação utilizada para produzir o plano de teste:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Documento** | **Criado ou reutilizado** | **Recebido ou revisado** | **Autor ou fonte** | **Notas** |
| Especificação de requisitos | Criado | Revisado | Vitor Sakassegawa | Todos os documentos foram revisados pois houveram modificações na arquitetura do sistema e no documento de requisitos |
| Requisitos funcionais | Criado | Revisado | Vitor Sakassegawa | Todos os documentos foram revisados pois houveram modificações na arquitetura do sistema e no documento de requisitos |
| Relatório de casos de uso | Criado | Revisado | Vitor Sakassegawa | Todos os documentos foram revisados pois houveram modificações na arquitetura do sistema e no documento de requisitos |
| Documento de Arquitetura de Software | Criado | Revisado | Vitor Sakassegawa | Todos os documentos foram revisados pois houveram modificações na arquitetura do sistema e no documento de requisitos |
| Documento de Visão do Software | Criado | Revisado | Vitor Sakassegawa | Todos os documentos foram revisados pois houveram modificações na arquitetura do sistema e no documento de requisitos |
| Protótipo | Criado | Revisado | Vitor Sakassegawa | Todos os documentos foram revisados pois houveram modificações na arquitetura do sistema e no documento de requisitos |
| Regra de Negócio | Criado | Revisado | Vitor Sakassegawa | Todos os documentos foram revisados pois houveram modificações na arquitetura do sistema e no documento de requisitos |

|  |
| --- |
| 2. Requisitos para teste |

A lista abaixo identifica aqueles itens – casos de uso, requisitos funcionais e requisitos não funcionais – que foram identificados como alvos para os testes. Esta lista representa o que será testado.

|  |
| --- |
| 2.1. Teste Unitários |

Etapa de identificação possíveis falhas que não foram identificados no decorrer do desenvolvimento, onde é examinado o comportamento de cada processo, estipulado pelas variadas entradas.

* Garantia de que um usuário só possa efetuar uma compra quando autenticado no sistema, através de seu logín e senha.
* Garantia que cada usuário posso acessar apenas a sua conta através de seu logín e senha.
* Garantia de que o cliente possa cancelar o seu pedido quando permitido.
* Garantia de que a compra seja efetuada com sucesso.
* Garantia de que os processes de cadastramento, alteração, consulta e exclusão sejam feitos de forma correta.
* Garantia de que os campos obrigatórios sejam preenchidos.

|  |
| --- |
| 3. Estratégia de teste |

A estratégia de teste apresenta a abordagem recomendada para o teste do software. O tópico passado, requisitos para teste, descreveu o que será testado – esse tópico descreve como isso será feito.

|  |
| --- |
| 3.1. Teste Unitário |

Foram testadas as principais funcionalidades em relação aos requisitos da Loja de Virtual – Meraki, utilizando as seguintes estratégias:

|  |  |
| --- | --- |
| Objetivo do teste: | Verificar possíveis ocorrências de erros em cada ação solicitada no sistema |
| Estratégia estabelecida: | Utilização das interfaces pré-desenvolvidas e inserção dos dados manualmente a fim de identificar algum erro de validação, persistência e navegação de páginas. |
| Critérios de teste de validação | Inserção de dados fictícios e reais nos devidos campos de cada página, com o objetivo de analisar a resposta de cada requisição. |
| Critérios de Sucesso | Analise da resposta obtida em cada requisição e observar do que foi persistido no banco, quando necessário. |

|  |
| --- |
| 3.2. Ferramentas de teste |

O sistema será testado manualmente utilizando a própria interface estabelecida no desenvolvimento, em uma máquina pessoal.

|  |
| --- |
| 4. Recursos |

Esta seção apresenta os recursos humanos e do sistema recomendados para o projeto da Loja de Virtual – Meraki, suas principais funcionalidades, e seu conjunto de conhecimento ou de habilidade.

|  |
| --- |
| 4.1. Regras |

A tabela abaixo mostra as suspostas alocação para equipe de testes.

|  |  |  |
| --- | --- | --- |
| Recursos Humanos | | |
| Trabalhador | Recursos Mínimos Recomendados | Responsabilidades Específicas ou Comentários |
| Desenvolvedor | Vitor Sakassegawa | Realiza supervisão gerencial.  Responsabilidades:   * Provê direcionamento técnico * Adquire recursos apropriados * Fornece relatórios de gerenciamento |
| Designer de Teste | Vitor Sakassegawa | Identifica, prioriza e implementa o caso de teste.  Responsabilidades:   * Confeccionar o plano de teste * Gerar o modelo de teste |
| Analista de Teste | Vitor Sakassegawa | Executa os testes.  Responsabilidades:   * Executas os testes * Registra dos resultados * Corrige os erros * Modifica o documento de requisitos |
| Administrador do Banco de Dados | Vitor Sakassegawa | Assegura o ambiente de dados.  Responsabilidades:   * Testar a persistência dos dados |
| Designer | Vitor Sakassegawa | Identifica e define as operações, atributos, e associações das classes de teste.  Responsabilidades:   * Identificar e definir as classes de teste * Identificar e definir os pacotes de teste |

|  |
| --- |
| 4.2. Sistema |

A seguinte tabela determinou os recursos de sistema para o projeto testado.

|  |  |
| --- | --- |
| Recursos do Sistema | |
| Recursos | Nome / Tipo |
| Banco de Dados  Java  Container Web | PostgreSQL  JDK 8  Tomcat 8 |