**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**Graduação em Sistemas de Informação**

**Cristovão Olegário de Castro**

**Diogo Lourenço Pereira**

**Franco Birro Godinho**

**Samara Luiza da Silva**

**Trabalho Final - Fundamentos de Testes**

**Belo Horizonte**

**2017**

**MASTER TEST PLAN**

1. **Identificador do Plano de Testes MM-MTP-1.0**
2. **Referências**

Não Especificadas/Identificadas.

1. **Introdução**

Este é o plano de testes do projeto MadMotors para a disciplina de Fundamentos de Testes de Software do 5° Período de Sistemas de Informação da PUC Minas. Esse plano de testes irá conter os itens de vários módulos do projeto, mas principalmente as classes do modelo MVC (Model,Control e Views) o qual foi estabelecido pelos desenvolvedores (alunos) como a melhor abordagem para a estruturação do sistema. O foco desse Plano de Testes é garantir que o sistema forneça dados fidedignos, permitindo melhoras futuras e aumentos na aquisição de dados e nível de detalhamento. (E é claro, encontrar F A L H A S)

Esse projeto terá quatro camadas de testes, os testes de Unidade, testes de Integração, testes de Sistema e testes Aceitação. Os detalhes de cada camada de testes serão especificados na devida secção de cada tipo de teste e com seus planos específicos.

1. **Itens a serem testados**

A seguir temos uma lista dos itens que deverão ser testados (métodos?/classes?):

* Classes de Modelo pacote, versão 1.0
* Classes de Controle pacote, versão 1.0
* Classes de Views, pacote, versão 1.0

1. **Questões de Risco no Software**

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.(coloquei só pra poder dar uma quebra de linha no item 7)

1. **Características a serem testadas**

A seguir consta uma lista das áreas do sistema que estarão em foco durante os testes:

* Conexões e Aquisição de dados do Banco de Dados;
* Ligações com o Json;
* Telas;
* JavaScript?

1. **Características que não serão testadas**

A seguir consta uma lista das áreas do sistema que não estarão em foco durante os testes:

* A estabelecer
* A estabelecer
* A estabelecer

1. **Abordagem**

**8.1 - Leveis dos Testes:**

Os testes do sistema vão consistir em leveis de testes de Unidade, testes de Sistema e Integração (juntos) e testes de Aceitação. É esperado que haja pelo menos um pessoa independente para realizar testes de sistema/integração. A maioria dos testes será feita pela equipe de desenvolvimento.

**Testes de Unidade –** Os testes serão realizados pelo desenvolvedor, depois pelos outros membros da equipe e então poderam aprovados pela equipe. A prova da realização dos testes (lista de casos de testes, dados de saida, prints de dados, informação da falha) deverá ser fornecida pelo desenvolvedor para a equipe antes dos testes serem repassados para a equipe testar. A ferramenta de testes que será utilizada será o phpUnit.

**Testes de Sistema/Integração –** Os testes serão realizados pelo desenvolvedor, depois pelos outros membros da equipe e então poderam aprovados pela equipe. Os códigos entrarão nessa fazer de testes após as correções de dos seus defeitos na parte de testes de unidade.

**Testes de Aceitação -** Os testes serão realizados pelos alunos da matéria durante a apresentação e então poderam aprovados pela sala e pelo professor.

**8.2 – Gerencia de Configuração/ Controle de Versão:**

--Falar sobre gerência de Configuração

Todas as mudanças, aprimoramentos e quaisquer outras mudanças fazem com que seja necessário que lidemos com o controle de alterações. O software que deverá ser utilizado será o Git com um servidor remoto no GitHub. Qualquer modificações necessárias nos códigos poderão ser identificadas durante o desenvolvimento dos testes facilitando o controle das alterações para a equipe.

**8.3 – Ferramentas de Teste:**

As ferramentas de teste que serão utilizadas estão listadas abaixo:

* + phpUnit – Para a realização de testes de Unidade;
  + Selenium, HTTPerf, WAPT, WebLoad,Jmeter – Para a realização dos testes de integração, sistema, aceitação, desempenho/carga/stress;

**8.4 – Medidas e Métricas:**

As informações abaixo serão coletadas pela equipe durante os testes de Unidade:

* + Defeitos por módulo e gravidade;
  + Origem do defeito (Requisito, Código ou Design);
  + Tempo gasto na resolução do defeito.

1. **Aprovação do Item/ Critérios de Falha**
2. **Suspensão do Critério e Resumo de Requisitos**
3. **Entregas de Testes**

* Plano de testes de Aceitação
* Plano de testes de Sistema/Integração
* Plano de testes de Unidade/ Documentação
* Protótipos das telas
* Mock-ups
* Relatório de Defeitos
* Auditoria e relatório de status de configuração

1. **Tarefas de Testes Remanescentes**

|  |  |  |
| --- | --- | --- |
| Tarefa | Designada a | Status |
| Criar Plano de Teste de Aceitação | Equipe |  |
| Criar Plano de Teste de Sistema/Integração | Equipe |  |
| Definir Testes de Unidade e os Procedimentos | Equipe |  |

1. **Responsabilidades**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | C | D | F | P | S |
| Documentação e execução dos testes de Aceitação | X | X | X | X | X |
| Documentação e execução dos testes de Sistema/Integração | X | X | X | X | X |
| Documentação e execução dos testes de Unidade | X | X | X | X | X |
| Documentação e execução dos testes de Compatibilidade | X | X | X | X | X |
| Documentação e aplicação dos conceitos de gerência de configuração e controle de versões | X | X | X | X | X |

Toda a equipe de desenvolvimento irá participar e terá responsabilidade na documentação e na execução dos testes.

1. **Cronograma**

**FAZER TUDO NO SÁBADO.**

1. **Planejamento de Riscos e Contingências**
2. **Aprovações**