Plano de Testes

**Cliente: *<Ivna Valença>***

Projeto: <My Money>

Versão: <0>.<1>

Equipe:

Eudes Tarquino

Fernando Nilo

Joanna Maressith

Wallace Roger

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 21/11/2013 | 1.0 | Release Inicial | Joanna Maressith |
| 03/10/2013 | 0.4 | Identificador do plano de teste, Objetivos, Escopo, Escopo Negativo,Nível na sequência de teste, Requisitos de suspensão e retomada, Matriz de rastreabilidade, Responsabilidades, Necessidade treinamento da equipe Cobertura dos testes | Joanna Maressith |
| 31/10/2013 | 0.3 | Teste da Interface do Usuário , Teste de Funcionalidade , Teste do Banco de Dados, Teste , Teste do Ciclo de Negócios , Teste da Interface do Usuário ,Teste de Performance , Teste de Carga ,Teste de Integridade de Dados e do Banco de Dados. | Eudes Tarquino |
| 21/11/2013 | 0.2 | Teste de Stress, Teste de carga, teste segurança e controle de acesso | Fernando Nilo |
| 09/11/2013 | 0.1 | Teste de Falha/Recuperação, Teste de Instalação, ferramentas e cronograma. | Wallace Rooger |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice

1. Introdução 4

1.1 Identificador do plano de teste 4

1.2 Objetivos 4

1.3 O Sistema <Projeto> 4

1.4 Escopo 4

1.5 Escopo Negativo: 4

1.6 Referências 4

1.7 Nível na sequência de teste. 4

2. REQUISITOS A TESTAR 5

2.1 Teste do Banco de Dados 5

2.2 Teste Funcional 5

2.3 Teste do Ciclo de Negócios 5

2.4 Teste da Interface do Usuário 5

2.5 Teste de Performance 5

2.6 Teste de Carga 5

2.7 Teste de Stress 5

2.8 Teste de Segurança e de Controle de Acesso 5

2.9 Teste de Falha/Recuperação 5

2.10 Teste de Instalação 5

3. Estratégia de Teste 6

3.1 Tipos de Teste 6

**3.1.1** **Teste de Integridade de Dados e do Banco de Dados** 6

**3.1.2** **Teste de Funcionalidade** 6

**3.1.3** **Teste da Interface do Usuário** 6

**3.1.4** **Teste de Performance** 6

**3.1.5** **Teste de Carga** 7

**3.1.6** **Teste de Segurança e Controle de Acesso** 7

**3.1.7** **Teste de Instalação** 7

3.2 Ferramentas 7

3.3 Riscos 8

4. Requisitos de suspensão e retomada 9

5. Matriz de rastreabilidade 10

6. Responsabilidades 11

7. Necessidade treinamento da equipe 12

8. Cobertura dos testes 13

9. Cronograma 14

# Introdução

## Identificador do plano de teste

M.M-projeto-my-money

## Objetivos

Identificar informações de projeto existentes e os componentes de software que devem ser testados.

Listar os Requisitos a Testar.

Recomendar e descrever as estratégias de teste a serem empregadas.

Identificar os recursos necessários e prover uma estimativa dos esforços de teste.

Listar os elementos resultantes do projeto de testes.

## O Sistema <My Money >

O MyMoney é uma aplicação a ser desenvolvida em uma plataforma open source que será capaz de registrar as transações financeiras pessoais de um indivíduo. Suas funções de gerenciamento financeiro permitirão aos usuários cadastrar lançamentos de despesas e receitas, realizar o fechamento dos períodos de transação (mensal e anual), gerar relatórios de fluxo de caixa, receber notificações via e-mail e manter as informações básicas sobre os investimentos pessoais.

## Escopo

O M.M passará pelos testes com a qualidade funcional, das bases de dados, interface gráfica e do controle de acesso enquanto que os testes de sistema tratarão as questões de performance.

Para a execução dos testes serão utilizadas máquinas o mais idênticas possível, em termos de hardware, àquelas que serão implantadas, a fim de garantir a previsibilidade de performance e compatibilidade.

Os testes mais críticos serão os testes de banco de dados, que compõe a maior parte do sistema e os de performance:

Testaremos o tempo de resposta para operações que envolvam dados multimídia

E também a correta atualização do banco de dados para as funções de cadastrar, atualizar e remover.

## Escopo Negativo:

Não será testado requisitos não-funcionais, testes de configuração, teste de instalação e especificamente as telas pois ao testar as demais funcionalidades elas serão testadas.

**1.4 Identificação do Projeto**

|  |  |  |
| --- | --- | --- |
| **Documento** | **Criado ou Disponível** | **Recebido ou Revisado** |
| Especificação de Requisitos | ◼ Sim 🞏 Não | ◼ Sim 🞏 Não |
| Plano de Projeto | ◼ Sim 🞏 Não | ◼ Sim 🞏 Não |
| Modelo de Análise | 🞏 Sim ◼ Não | 🞏 Sim ◼ Não |
| Modelo de Projeto | 🞏 Sim ◼ Não | 🞏 Sim ◼ Não |
| Documento de Arquitetura | 🞏 Sim ◼ Não | 🞏 Sim ◼ Não |
| Protótipo | 🞏 Sim ◼ Não | 🞏 Sim ◼ Não |
| Manual do Usuário | 🞏 Sim ◼ Não | 🞏 Sim ◼ Não |
| Lista de Riscos | 🞏 Sim ◼ Não | 🞏 Sim ◼ Não |

## Referências

Requisito do MyMoney.

## Nível na sequência de teste.

## Teste do Banco de Dados

## Teste Funcional

## Teste do Ciclo de Negócios

## Teste da Interface do Usuário

## Teste de Performance

Teste de Carga

## Teste de Segurança e de Controle de Acesso

Teste de Falha/Recuperação

# REQUISITOS A TESTAR

## Teste do Banco de Dados

* Verificar se o sistema está guardando todos possíveis dados cadastrais do usuário.
* Verificar se o sistema está guardando todos os dados do balanço mensal.
* Verificar se o sistema está salvando todos os dados do balanço anual.
* Verifica se espaço no banco de dados é suportável para total aproveitamento de 1 usuário.
* Verifique que as informações especificas de cada usuário podem ser acompanhadas.
* Verifique que as informações sobre o sistema possam ser cadastradas, removidas e atualizadas pelo administrador do sistema.
* Verifique que as informações úteis cadastradas possam ser consultadas
* Verifique que o sistema é capaz de buscar e manter atualizadas as informações do usuário no sistema.

## Teste Funcional

* Verificar se o programa permite o fechamento do mês corrente, bloqueando a entrada de novos lançamentos.
* Verificar se o programa realiza o fechamento do ano corrente bloqueando a entrada de novos lançamentos nos meses que compõem o ano.
* Verificar se o programa gera um relatório de fluxo.
* Verificar se o programa permite gerar um relatório de fluxo de caixa podendo ser impresso.
* Verificar se o programa gera um balanço mensais das despesas / receitas de usuários, consolidando o total de despesas e receitas por tipo de lançamento e rubrica. Consolida o saldo dos meses do ano corrente.
* Verificar se o programa permite o envio diário de relatórios de fluxo de caixa através do e-mail do usuário. Quando os níveis de despesas estourarem o orçamento a aplicação deverá enviar uma mensagem SMS para o celular do usuário.
* Verificar se o programa permite lançar e consultar os investimentos pessoais do usuário. Os tipos de investimento controlados serão inicialmente apenas os de fundos de investimento e aposentadoria.
* Verificar se as tabelas básicas de apoio ao sistema, inicialmente, as tabelas mantidas serão as de Ano Corrente, Tipo de Fundo, Tipo de Lançamento, Rubricas e Parâmetros. A tabela de Parâmetros deverá conter as informações mínimas para configuração do sistema (conexão ao banco de dados, perfil do usuário e parâmetros gerais).
* Verificar se o programa permite a inclusão de um orçamento mensal, com a previsão de receitas e despesas.
* Verificar se o programa permite o lançamento de uma transação de receita ou despesa na base de dados.
* O sistema deverá acontecer através de um controle de login e senha.
* Verificar se no primeiro acesso ao sistema o login e senha deverão ser obrigatoriamente cadastrados.
* Verifica se futuramente o sistema permiti a utilização de certificados digitais para a autenticação.
* Verificar se o sistema enviara uma nova senha para o e-mail do usuário, quando o mesmo esquecer

## Teste do Ciclo de Negócios

* . Verificar se todos os campos da janela do login estão alinhados.

## Teste da Interface do Usuário

* Verificar se todos os campos da janela do login estão alinhados.

## Teste de Performance

* Verificar se o tempo resposta do sistema não ultrapassa 5 segundos, contados entre o tempo que o usuário confirma o cadastro e volta do controle da aplicação para o mesmo.

## Teste de Carga

* Verificar a resposta do sistema com 100 usuários.
* Verificar se o sistema suporta no máximo 100 conexões simultânea.

## Teste de Stress

* Verificar se o tempo resposta do sistema não ultrapassa 5 segundos, contados entre o tempo que o usuário confirma o cadastro e volta do controle da aplicação para o mesmo.

## Teste de Segurança e de Controle de Acesso

* Verificar que os usuários do sistema podem acessar apenas as funcionalidades e dados associados ao seu próprio tipo de usuário.
* Verificar que além do administrador, ninguém mais pode inserir, atualizar ou remover dados do sistema.
* Verificar que usuários não cadastrados não podem acessar informações restritas aos cadastrados.
* Verificar se todas as informações sensíveis à aplicação, como login e senha, por exemplo, deverão ser armazenadas criptografadas.
* Valida senha cujo tenha 8 caracteres no mínimo com obrigatoriedade de utilização de letras e números.
* Verifica se o sistema não haverá inicialmente controle de perfis dos usuários.
* Verificar se o sistema enviara uma nova senha para o e-mail do usuário, quando o mesmo esquecer
* Verificar que a atualização do sistema pode ser feita apenas a partir da rede interna do sistema

## Teste de Falha/Recuperação

* Sistema devera salva última atualizações do usuário caso o sistema caia.
* Caso o sistema apresente alguma falha no seu procedimento, o sistema devera exibir mensagens reportando o erro ao usuário especificamente onde o erro está ocorrendo.
* Caso o sistema aja uma falha na conexão do sistema ele devera salva todas atualizações feitas no sistema.

## Teste de Instalação

* Verifica se o unistall do sistema está exatamente excluindo todos seus arquivos do pc do usuário.
* Verifique que a instalação do sistema ocorre normalmente em todas as máquinas.
* Verifique que qualquer terminal do sistema do sistema é capaz de rodar o TIG normalmente.
* Verifique que o sistema é capaz de obter e atualizar as informações úteis a que se propõe a disponibilizar.
* Verifique que o espaço disponível em disco para informações deve ser capaz de armazenar todos os dados/atualizações que forem cadastrados.

# Estratégia de Teste

## Tipos de Teste

### **Teste de Integridade de Dados e do Banco de Dados**

|  |  |  |
| --- | --- | --- |
| Objetivo do Teste: | Garantir que os métodos e processos de acesso ao banco de dados funcionam apropriadamente e sem corrupção dos dados. |  |
| Técnica: | * Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos.   Inspecionar o banco de dados para garantir que os dados foram migrados como pretendido, que todos os eventos do banco de dados ocorreram apropriadamente, ou revisar os dados retornados para garantir que os dados corretos foram recuperados pelas razões corretas. |  |
| Critério de Finalização: | Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados. |  |
| Considerações Especiais: | * O teste podem necessitar de um ambiente de desenvolvimento para inserir ou modificar os dados diretamente nas bases de dados. * Processos devem ser informados manualmente. |  |

### **Teste de Funcionalidade**

|  |  |  |
| --- | --- | --- |
| Objetivo do Teste: | Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento, e recuperação. |  |
| Técnica: | Executar cada caso de uso, fluxo de caso de uso, 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.   Cada regra de negócio é aplicada apropriadamente. |  |
| Critério de Finalização: | * Todos os testes planejados foram executados.   Todos os defeitos identificados foram tratados. |  |
| Considerações Especiais: | Nada consta. |  |

### **Teste da Interface do Usuário**

|  |  |  |
| --- | --- | --- |
| Objetivo do Teste: | Verificar o seguinte:  Verificar se o usuário recebe os dados do servidor e realiza a operação para cada lançamento de forma prática e eficiente. |  |
| Técnica: | Manter as tabelas do Ano Corrente, Tipo de Fundo, Tipo de Lançamento, Rubricas e Parâmetros. A tabela de Parâmetros deverá conter as informações mínimas para configuração do sistema (conexão ao banco de dados, perfil do usuário e parâmetros gerais). |  |
| Critério de Finalização: | É verificado que cada janela permanece consistente com a versão de comparação ou dentro de padrões aceitáveis.  A tabela de Parâmetros deverá conter as informações mínimas para configuração do sistema. |  |
| Considerações Especiais: | Verificar se todas as tabelas estão de acordo com a interface do sistema. |  |

### **Teste de Performance**

|  |  |  |
| --- | --- | --- |
| Objetivo do Teste: | Verificar que os comportamentos de performance para as transações designadas ou funções de negócio sob as seguintes condições:   * O sistema deverá ser exemplar mesmo com 100 usuários utilizando o sistema (MM). * O sistema deverá ter uma certa diminuição do desempenho com mais de 100 usuários utilizando o sistema (MM). |  |
| Técnica: | Deverão estar conectados 100 usuários utilizando o sistema (MM). |  |
| Critério de Finalização: | O sistema deverá atender a uma performance exemplar mesmo com os 100 usuários utilizando o sistema. |  |
| Considerações Especiais: | Nada consta. |  |

### **Teste de Carga**

|  |  |
| --- | --- |
| Objetivo do Teste: | Verifique o tempo de resposta para as transações designadas ou casos de negócio sob condições variantes de carga de trabalho. |
| Técnica: | Queda de energia, interrupção da conexão. |
| Critério de Finalização: | O sistema deverá recuperar todas as informações e cadastradas antes da interrupção. |
| Considerações Especiais: | Nada consta. |

### **Teste de Segurança e Controle de Acesso**

|  |  |
| --- | --- |
| Objetivo do Teste: | Segurança do Nível de Aplicação: Verifique que um usuário pode acessar apenas aquelas funções ou dados para os quais o seu tipo de usuário tem permissão.  Segurança do Nível de Sistema: Verifique que apenas aqueles usuários com acesso ao sistema e aplicações têm permissão de acessá-los. |
| Técnica: | * Segurança do Nível de Aplicação: Identifique e liste cada tipo de usuário e as funções ou dados para os quais cada tipo tem permissão. * Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicos para cada tipo de usuário. * Modifique o tipo de usuário e repita os testes para os mesmos usuários. Em cada caso, verifique que funções ou dados adicionais estão corretamente disponíveis ou negados. |
| Critério de Finalização: | Para cada tipo de usuário conhecido as funções ou dados apropriados estão disponíveis, e todas as transações funcionam como esperado e rodam nos Testes de Função anteriores. |
| Considerações Especiais: | O Acesso ao sistema deve ser revisado ou discutido com o administrador de rede ou de sistema apropriado. Esse teste pode não ser necessário já que ele pode ser uma função da administração da rede ou sistema. |

### **Teste de Instalação**

|  |  |
| --- | --- |
| Objetivo do Teste: | Verifique que os alvos de teste instalam apropriadamente em cada configuração de hardware necessária sobre as seguintes condições:   * Uma nova instalação, em um nova máquina, que nunca fora anteriormente instalada com o (MM). * Atualização, numa máquina onde o (MM) já fora previamente instalado, para a mesma versão * Excluir todos seus arquivos caso o usuário apite pela opção Unistall |
| Técnica: | Manualmente ou desenvolva scripts automatizados, para validar a condição da máquina alvo – novo TIG nunca instalado; TIG na mesma versão ou versão mais velha já instalada.  Começar ou executar a instalação  Usando um subgrupo predeterminado de scripts de teste de funções, rode as transações.  Executar aplicativo com o nome “unistall” dentro da pasta onde o sistema foi instalado. |
| Critério de Finalização: | As transações do (MM)executam de forma bem sucedida, sem falha.  Todos arquivos da pasta do (MM) deverão ser completamente apagados |
| Considerações Especiais: | Nada consta. |

## Ferramentas

As seguintes ferramentas serão empregadas para esse projeto:

|  |  |  |
| --- | --- | --- |
|  | Ferramenta | Vendedor |
| Gerenciamento de Teste | Não possui | Não possui |
| Projeto de Teste | Excel | Não possui |
| Gerenciamento de incidentes | Não possui | Não possui |
| Gerenciamento de Projeto | GitHub | Não possui |
| Ferramentas do SGBD | Não possui | Não possui |
| Servidor WEB | Não possui | Não possui |
| Browser | Não possui | Não possui |
| Desenvolvimento | Não possui | Não possui |

## Riscos

|  |  |  |
| --- | --- | --- |
| Risco | Mitigação | Contingência |
| Falha na conexão | Contrato com 2 empresas diferentes. | Novo contrato em caráter emergencial. |
| Falha com banco de dados | Backup | Migração e interligação. |
|  |  |  |
|  |  |  |
|  |  |  |

# Requisitos de suspensão e retomada

* Suspensão:

1. Em caso de falta de luz
2. Problema com hardware
3. Falha com banco de dados

* Retomada:

1. Salvar as últimas alterações na aplicação
2. Salvar relatórios dos últimos teste.

# Matriz de rastreabilidade

Para todos os teste é necessário estar autenticado na aplicação.

# Responsabilidades

|  |  |  |
| --- | --- | --- |
| Recursos Humanos | | |
| Trabalhador | Recursos Mínimos Recomendados | Responsabilidades Específicas ou Comentários |
| Gerente de Teste  Gerente do Projeto de Teste  Analista de Teste | Joanna Maressith | Fornece suporte gerencial.  Identifica, prioriza, e implementa os casos de teste.  Responsabilidades:   * Provê direcionamento técnico * Fornece relatórios de gerenciamento * Gera o plano de teste * Cria o modelo de teste * Documentar solicitações de mudança |
| Analista de Teste  Testador | Eudes Tarquino | Executa os testes.  Responsabilidades:   * Gera o plano de teste * Executar os testes * Registrar os resultados * Cria o modelo de teste |
| Administrador do Sistema de Teste  Testador | Wallace Rooger | Executa os testes.  Responsabilidades:   * Gera o plano de teste * Executar os testes * Registrar os resultados |
| Testador | Fernando Nilo | Executa os testes.  Responsabilidades:   * Gera o plano de teste * Executar os testes * Registrar os resultados * Reestabelecer-se dos erros * Administrar o sistema de gerenciamento teste |

# Necessidade treinamento da equipe

* Treinamento da equipe com a ferramenta GITHUB, que é o repositório do projeto.

# Cobertura dos testes

Os casos de teste foram baseados nos 28 casos de uso. Testando as diversas funcionalidades do mesmo.

# Cronograma

|  |  |  |
| --- | --- | --- |
| **Fluxo de teste** | **Data de Início** | **Data de Término** |
| Cronograma | 09/10/2013 | 10/10/2013 |
| Planejar Teste | 10/10/2013 | 31/10/2013 |
| Projetar Teste | 31/10/2013 | 07/11/2013 |
| Avaliar Teste | 09/11/2013 | 13/11/2013 |
| Versão final do projeto | 03/10/2013 | 14/11/2013 |