FORTESI

Iteração/Mestre Plano de Teste

Versão 1.7

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 23/04/2015 | 1.0 | Introdução a abordagem dos testes | José Lucas |
| 05/05/2015 | 1.1 | Critérios de entrada e saída | Lucas Castanheiro |
| 19/05/2015 | 1.2 | Desenvolvimento do documento | Moisés Novais |
| 22/05/2015 | 1.3 | Tabelas de necessidades e Procedimento | Lucas Castanheiro |
| 24/05/2015 | 1.4 | Critérios de entrada e saída | Raimundo Néto |
| 24/05/2015 | 1.5 | Tópico Estrutura do documento | Lucas Castanheiro |
| 24/05/2015 | 1.6 | Produtos Liberados | Moisés Novais |
| 25/05/2015 | 1.7 | Produtos de trabalhos adicionais | Lucas Castanheiro |

Índice Analítico

1. Introdução 5

1.1 Finalidade 5

1.2 Escopo 5

1.3 Público-alvo 5

1.4 Terminologia e Acrônimos do Documento 6

1.5 Referências 6

1.6 Estrutura do Documento 6

2. Missão de Avaliação e Motivação dos Testes 7

2.1 Informações Detalhadas 7

2.2 Missão de Avaliação 7

2.3 Motivadores dos Testes 8

3. Itens-alvo dos Testes 8

4. Resumo dos Testes Planejados 8\*

4.1 Resumo das Inclusões dos Testes 9

4.2 Resumo dos Outros Candidatos a Possível Inclusão 9

4.3 Resumo das Inclusões dos Testes 9

5. Abordagem dos Testes 10

5.1 Técnicas e Tipos de Teste 11

5.1.1 Teste de Integridade de Dados e de Banco de Dados 11

5.1.2 Teste de Função 11

5.1.3 Teste de Ciclos de Negócios 12

5.1.4 Teste da Interface do Usuário 13

5.1.5 Determinação do Perfil de Desempenho 13

5.1.6 Teste de Carga 14

5.1.7 Teste de Stress 15

5.1.8 Teste de Volume 16

5.1.9 Teste de Segurança e de Controle de Acesso 17

5.1.10 Teste de Tolerância a Falhas e de Recuperação 18

5.1.11 Teste de Configuração 19

5.1.12 Teste de Instalação 20

6. Critérios de Entrada e de Saída 21

6.1 Plano de Teste 21

6.1.1 Critérios de Entrada de Plano de Teste 21

6.1.2 Critérios de Saída de Plano de Teste 22

6.1.3 Critérios de Suspensão e de Reinício 22

6.2 Ciclos de Teste 22

6.2.1 Critérios de Entrada de Ciclo de Teste 22

6.2.2 Critérios de Saída de Ciclo de Teste 22

6.2.3 Término Anormal do Ciclo de Teste 22

7. Produtos Liberados 22

7.1 Sumários de Avaliação de Testes 22

7.2 Relatórios da Cobertura de Teste 22

7.3 Relatórios da Qualidade Perceptível 23

7.4 Registros de Incidentes e Solicitações de Mudança 23

7.5 Conjunto de Testes de Regressão e Scripts de Teste de Suporte 24

7.6 Produtos de Trabalho Adicionais 24

7.6.1 Resultados Detalhados dos Testes 24

7.6.2 Scripts de Teste Funcionais Automatizados Adicionais 24

7.6.3 Guia de Teste 24

7.6.4 Matrizes de Rastreabilidade 24

8. Fluxo de Trabalho de Teste 25

9. Necessidades Ambientais 25

9.1 Hardware Básico do Sistema 25

9.2 Elementos de Software Básicos do Ambiente de Teste 25

9.3 Ferramentas de Produtividade e de Suporte 25

9.4 Configurações do Ambiente de Teste 26

10. Responsabilidades, Perfil da Equipe e Necessidades de Treinamento 26

10.1 Pessoas e Papéis 26

10.2 Perfil da Equipe e Necessidades de Treinamento 29

11. Marcos da Iteração 29

12. Riscos, Dependências, Suposições e Restrições 30

13. Procedimentos e Processos de Gerenciamento 31

13.1 Medição e Avaliação da Extensão do Teste 31

13.2 Avaliação dos Produtos Liberados deste Plano de Teste 31

13.3 Relato de Problemas, Seleção de Pessoas para Resolvê-los e Busca de Soluções 31

13.4 Gerenciamento de Ciclos de Teste 31

13.5 Estratégias de Rastreabilidade 31

13.6 Aprovação e Encerramento 32

Plano de Teste

# Introdução

Documento de teste do sistema FORTESI, um sistema de controle é processo jurídico. É tem como objetivo a apresentação de teste de componentes do sistema para serem avaliados e reavaliados.

## Finalidade

A finalidade do Plano de Teste de Iteração é reunir todas as informações necessárias ao planejamento e ao controle do esforço de teste referente a uma iteração específica. Ele descreve a abordagem dada ao teste do software e é o plano de nível superior gerado e usado pelos gerentes para coordenar o esforço de teste.

Este plano de testereferente ao FORTESI suporta os seguintes objetivos:

* Identificar os itens do sistema FORTESI que serão avaliados nesta fase.
* Identificar a motivação e ideias que são abrangidas nessa fase de teste.
* Detalhar qual são os métodos de testes usados.
* Identificar quais recursos vai ser utilizado e mostrar uma estimativa dos esforços de teste.
* Listar os elementos liberados do projeto de teste.

## 1.2 Escopo

Os Níveis de teste escolhido é o de Unidade é Integração realizada a parti da necessidade de teste de cada bloco do sistema.

* Teste de funcionalidade: composto pelas classes login, Agenda, Busca de processos e o CRUD das informações/processos.
* Teste de usabilidade: Treinamento de usuário, acesso a funcionalidades e consistência na interface.
* Teste de confiabilidade: Segurança do banco de dados e verificação de componentes do sistema.
* Teste de desempenho: buscar informações do banco de dados e armazenamento de dados.
* Teste de suportabilidade:

## 1.3 Público-alvo

Este documento de teste tem como destinatário principal aqueles relacionados à fase de requisitos, desenvolvimento, gerenciador do projeto e o próprio teste.

Este documento de teste é indicado a qualquer pessoa que esteja envolvido em qualquer parte de um ciclo de vida de um software, já que o teste esta envolvido desde a fase de requisitos que é o pontapé inicial de um software. E os testes finais para ver os objetivos concluídos.

O software é basicamente desenvolvido em três etapas essenciais inicial, meio e final onde o teste tem quem participar de todos, ou seja, o teste é importantíssimo para o desenvolvimento de um software.

## 1.4 Terminologia e Acrônimos do Documento

CRUD – termo inglês Create, read, update, delete que indica criar, ler atualizar e deletar informações de um determinado sistema.

FORTESI – Sistema a ser desenvolvido.

## Referências

Maurício Aniche. unidade, integração ou sistema? Qual teste fazer?.2014. Disponível em: <http://blog.caelum.com.br/unidade-integracao-ou-sistema-qual-teste-fazer/>. Acesso em 23 abril. 2015.

FUNPAR. Conceitos : Tipos de teste.2014. Disponível em: <http://www.funpar.ufpr.br:8080/rup/process/workflow/test/co_tytst.htm#Top>. Acesso em 01 abril 2015

Camila tamico. VANT-EC-SAME Plano de Teste. Disponível em: <http://camilatamico.xpg.uol.com.br/art_11.htm>. Acesso em 01 abril 2015

Requisitos para o Processo de Testes – Abordagens, Estágios e Tipos de Testes. Disponível em:

<http://webcache.googleusercontent.com/search?q=cache:NXYSomP0Y04J:www.cin.ufpe.br/~tds/meso/Abordagens,%2520estrategias%2520e%2520tipos%2520de%2520testes.doc+&cd=1&hl=pt-BR&ct=clnk&gl=br>. Acesso em 3 abril 2015

Instituto de computação. Disponível em: <http://www.ic.unicamp.br/~ranido/mc626/FasesTestes.pdf>. Acessado em 20 abril 2015.

## Estrutura do Documento

O documento de está representado da segunda forma:

1. Missão de avaliação dos testes: Apresentará os métodos de avaliação e objetivo do plano do teste.

2. Item-alvos dos testes: Apresentas os Pontos, áreas e funcionalidades que serão específicos do sistema.

3. Resumo da Inclusão dos testes: Apresenta os métodos inclusos no teste, havendo mudanças e se estas mudanças estão sendo eficazes através de probabilidades.

4. Abordagens dos testes: Monstra as formas que o teste está sendo aplicado e quais as possibilidades de melhoria.

5. Critérios de entrada e saída: Apresenta as entradas e os critérios das informações iniciais para a aplicação do plano de teste.

6. Produtos liberados: Apresenta uma pequena lista de artefatos e fatores importantes no plano de teste apresentado.

7. Fluxo de trabalho de teste: Apresenta as áreas do software que estão sendo verificadas e quais os setores.

8. Necessidades ambientais: Apresenta os requisitos primários para o funcionamento do sistema.

9. Responsabilidades, perfil da equipe e necessidades de treinamento: Apresenta o papel de cada membro e suas reponsabilidades com uma tabela ilustrativa.

10. Marcos da iteração: Apresenta as datas das iterações no plano de teste.

11. Riscos, dependências, suposições e restrições: Apresentas os riscos significativos para a execução do plano de teste.

12. Procedimentos e processos de gerenciamento: Apresenta os métodos de gerenciamento e monitoramento do gerente do plano.

# 2.0 Missão de Avaliação e Motivação dos Testes

A missão do teste é exatamente verificar se ainda há algum defeito no software, verificar se objetivo foi concluído e deixar da maneira que o usuário deseja. A motivação é melhorar ainda mais o que foi feito ate o momento para uma maior satisfação do cliente.

## 2.1 Informações Detalhadas

Considerando que o teste unitário e de sistema que são os utilizados pelos FORTESI obedeça a um padrão de qualidade, desenvolvimento, que traga uma maior comodidade e segurança para os usuários e então ser validado todos os artefatos testados.

Com correção e complementação do que falta ou não ao software para que ambas as partes equipe de desenvolvimento e cliente chegue a um resultado final satisfatório.

## 2.2 Missão de Avaliação

Localizar problemas importantes e verificar os riscos de qualidade perceptível, Riscos de projeto e cumprimento de determinações de processo.

* Cumprir as determinações do processo;
* Advertir sobre os riscos de projeto percebidos;
* Localizar números de erros possíveis;
* Advertir sobre a qualidade do produto e os envolvidos.

## 2.3 Motivadores dos Testes

Motivação principal do teste é detectar os seguintes itens:

* Risco de qualidade
* Risco de projeto
* Falhas ou erros suspeitos
* Solicitações de mudanças

# 3. Itens-alvo dos Testes

A listagem abaixo identifica os itens que foram identificados como alvo dos testes. Essa lista representa os itens que serão testados.

* Métodos (Função) do código fonte;
* Execução do sistema em diversos computadores;
* Funcionalidades implementadas;
* Tempo de resposta do sistema;
* Acesso simultâneo.

Métodos (Funções) do código fonte.

|  |  |
| --- | --- |
| Objetivo do tipo de teste: | Verificar se os métodos estão funcionando normalmente e se há alguma possibilidade de redução de código. |
| Função: | É a realização de alguma tarefa dentro do sistema quando o mesmo é chamado. |

Execução do sistema em diversos computadores.

|  |  |
| --- | --- |
| Objetivo do tipo de teste: | Verificar se há alguma incompatibilidade no sistema. |
| Função: | Testar o sistema em qualquer computador com sistema operacional Windows. |

Funcionalidades implementadas.

|  |  |
| --- | --- |
| Objetivo do tipo de teste: | Verificação por partes de cada funcionalidade implementada. |
| Função: | Evitar que acumule muito código sem ser testado. |

Tempo de resposta do sistema.

|  |  |
| --- | --- |
| Objetivo do tipo de teste: | Avaliar se não há anormalidade com software em execução. |
| Função: | Que o sistema não tenha nenhum problema para abrir alguma funcionalidade. |

Acesso simultâneo.

|  |  |
| --- | --- |
| Objetivo do tipo de teste: | Verificar qual a reação do sistema quando duas pessoas entram com o mesmo login no sistema. |
| Função: | Apenas uma pessoa por login no sistema. |

## 4. Resumo das Inclusões dos Testes

* Verificar se o método está funcionando normalmente e se há alguma possibilidade de redução de código.
* Verificar se há alguma incompatibilidade no sistema.
* Verificação por partes de cada funcionalidade implementada.
* Avaliar se não há anormalidade com software em execução.
* Verificar qual a reação do sistema quando duas pessoas entram com o mesmo login no sistema.

## 4.1 Resumos dos Outros Candidatos a Possível Inclusão

* Integridade das informações armazenadas.
* Disponibilidade das informações do banco de dados.

São informações que são úteis, porém os testes só podem ser feito no release final do sistema é um processo que necessita do sistema completo.

## 4.2 Resumos das Inclusões dos Testes

* Verificação de componentes: Desnecessário já que o sistema terá testes unitários de métodos de cada componente.

# 5. Abordagem dos Testes

|  |
| --- |
| Métodos (Funções) do código fonte |
| Descrição: Analisa a estrutura interna do sistema (código fonte) e faz verificação nos métodos implementados. |
| Objetivo (s): diminuir volume de código do sistema, deixando o código mais organizado. |

|  |
| --- |
| Execução do sistema em diversos computadores |
| Descrição: Comportamento do sistema em ambiente diferente do que ele foi desenvolvido. |
| Objetivo (s): Detectar possíveis incompatibilidades no sistema. |

|  |
| --- |
| Funcionalidades implementadas |
| Descrição: Analise de como essas funcionalidades vão se comportar no sistema. |
| Objetivo (s): Verificar se há alguma falha nessas funcionalidades. |

|  |
| --- |
| Tempo de resposta do sistema |
| Descrição: Verifica se o sistema está agindo dentro da normalidade sem qualquer falha em tempo de resposta ao usuário. |
| Objetivo (s): Evitar lentidão no sistema. |

|  |
| --- |
| Acesso simultâneo |
| Descrição: Reação do sistema ao tentarem usar o mesmo login simultaneamente já que o permitido é apenas um. |
| Objetivo (s): permite que apenas o primeiro usuário a usar o login tenha acesso ao sistema. |

## Técnicas e Tipos de Teste

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

|  |  |
| --- | --- |
| Objetivo da Técnica: | O objetivo técnico é verificar se os dados armazenados estão logicamente organizados. |
| Técnica: | Use database login  Select \* from login  Select \* from usuário  Select \* from senha |
| Estratégias: | Se o select \* from usuário e select \* from senha não retornar nenhum valor há possiblidade de haver algum problema com a logica no banco de dados. |
| Ferramentas Necessárias: | A técnica exige a seguinte ferramenta:   * Banco de dados MYSQL. |
| Critérios de Êxito: | As instruções de banco de dados usadas acimas serão capazes de retornar se o banco de dados foi mal projetado ou se ele vai retorna algo para o desenvolvedor indicando que há uma conexão com banco de dados. |
| Considerações Especiais: | Deverão ser usados bancos de dados pequenos ou de tamanho mínimo (com um número limitado de registros) para aumentar a visibilidade de quaisquer eventos não aceitáveis. |

### **5.1.2 Teste de Função**

|  |  |
| --- | --- |
| Objetivo da Técnica: | Analisar as funcionalidades dos objetivos de testes incluídos na interação do usuário navegação, a entrada e processamento além de recuperação de dados. |
| Técnica: | Ao visualizar o caso de uso será possível detectar as funcionalidades do sistema independentes de usar dados validos ou inválidos. |
| Estratégias: | Para ter êxito nessa fase é necessário que o testador possa avaliar todas as partes do sistema e as telas para o usuário em todos os momentos. Para que seja necessário que nenhuma informação seja perdida. |
| Ferramentas Necessárias: | A ser definida. |
| Critérios de Êxito: | Todos os principais cenários de caso de uso  Todos os principais recursos. |
| Considerações Especiais: | Nenhuma. |

### **5.1.3 Teste de Ciclos de Negócios**

|  |  |
| --- | --- |
| Objetivo da Técnica: | O objetivo é verificar por meio de simulação ao longo do tempo, se o sistema corresponderá há as alterações realizadas pelos os usuários para identificar falhas e correções necessárias envolvendo o ciclo de negócio. |
| Técnica: | O teste simulará vários ciclos de negócios executando o seguinte:   * Os testes destinados a verificar o funcionamento do objetivo do teste serão modificados para aumentar o número de vezes que cada função é executada, a fim de simular vários usuários diferentes ao longo de um período de tempo especificado para uma análise de todas as funcionalidades sob circunstâncias reais de uso. * Todas as funções que ocorrerem segundo uma programação periódica será executadas ou iniciadas no momento adequado. |
| Estratégias: | A ser definida. |
| Ferramentas Necessárias: | Documentação do sistema e algumas funcionalidades do sistema funcionando.  Ferramenta especifica a ser definida. |
| Critérios de Êxito: | A técnica suporta o teste de todos os ciclos de negócios essenciais. |
| Considerações Especiais: | É necessário um modelo de negócios para identificar requisitos e procedimentos de teste adequados. |

### **5.1.4 Teste da Interface do Usuário**

|  |  |
| --- | --- |
| Objetivo da Técnica: | Verificar se todos os componentes das telas estão agindo de maneira correta aceitando comando do mouse garantindo que o usuário tenha acesso a todas as funcionalidades expostas nas telas.  Os componentes e características das janelas poderão ser experimentados como, por exemplo, Menus, Tamanhos, Posição, Estado é foco. |
| Técnica: | Clicando com o mouse para fazer reações nos botões.  Chamadas de outro formulário ou telas. |
| Estratégias: | O uso de uma tela para interação com o usuário especialmente para teste para ver a reação do usuário. |
| Ferramentas Necessárias: | Interface do sistema.  Ferramenta especifica a ser definida. |
| Critérios de Êxito: | A técnica suporta o teste de cada tela existente no sistema. |
| Considerações Especiais: | Nenhuma. |

### **5.1.5 Determinação do Perfil de Desempenho**

|  |  |
| --- | --- |
| Objetivo da Técnica: | * Cargas maiores que a capacidade do banco de dados. * Precisão do FORTESI que é atingida quando chega a sua carga máxima. * Qual o tempo de resposta para o usuário com diferentes condições de cargas. |
| Técnica: | Modifique os arquivos de dados a fim de aumentar o número de transações. |
| Estratégias: | Os testes deverão ser executados em uma máquina (o melhor é avaliar o desempenho de um único usuário, uma única transação) e deverão ser repetidos em outras maquinas com diferentes ambientes de trabalhos. |
| Ferramentas Necessárias: | * Uma ferramenta para a determinação do perfil de desempenho do aplicativo como, por exemplo, o Rational Quantify. * Ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.). * Ferramentas de restrição de recursos. |
| Critérios de Êxito: | A técnica suporta o teste de:   * Uma única transação ou um único usuário: a emulação bem-sucedida dos scripts de transação sem que ocorra nenhuma falha devido a problemas de implementação do teste. * Várias transações ou vários usuários: a emulação bem-sucedida da carga de trabalho sem que ocorra nenhuma falha devido a problemas de implementação do teste. |
| Considerações Especiais: | O método que pode ser usado para executar esse teste é:   * Usar um cliente físico, executando scripts de teste, para inserir carga no sistema. * Os bancos de dados usados para o Teste de Determinação de Perfil de Desempenho deverão ter um tamanho real ou deverão ser dimensionados igualmente em escala. |

### **5.1.6 Teste de Carga**

|  |  |
| --- | --- |
| Objetivo da Técnica: | Verifica o tempo de resposta do sistema para casos de negócios ou transações sob carga de trabalho. |
| Técnica: | Modificar os arquivos de dados a fim de aumentar o número de transações a fim de aumentar o número de vezes que cada transação ocorre.  As Cargas de Trabalho devem representar cargas Médias assim como cargas de Pico.  As Cargas de Trabalho devem ser executadas com diferentes Configurações do Ambiente de Teste. |
| Estratégias: | Os testes feitos nos dados para aumentar as transações serão feitos diretamente no sistema para evitar contato com a base de dados e evitar futuros erros. |
| Ferramentas Necessárias: | * Banco de dados * Sistema FORTESI |
| Critérios de Êxito: | Conclusão de teste realizada com êxito dentro do tempo de resposta estabelecido e sem nenhum defeito detectável. |
| Considerações Especiais: | O teste de carga deverá ser executado em uma máquina dedicada ou em um período de tempo dedicado. Isso permitirá o controle total e a medição exata.  Os bancos de dados usados para o teste de carga deverão ter um tamanho real ou deverão ser dimensionados igualmente em escala. |

### **5.1.7 Teste de Stress**

|  |  |
| --- | --- |
| Objetivo da Técnica: | * Pouca ou nenhuma memória disponível (memória RAM e espaço de armazenamento persistente); * Vários usuários executando a mesma operação de logar em um mesmo usuário ou senha; * Conjunto ou volume de alguma operação que geram “sobrecarga”. |
| Técnica: | Para testar recursos limitados citados à cima, os testes deverão ser executados em uma única máquina, e a memória RAM e o espaço de armazenamento persistente deverá ser reduzido ou limitado.  Para os testes de stress restantes, deverão ser usados vários clientes, executando-se de um mesmo teste ou testes complementares a fim de produzir o conjunto de transações para avaliar o desenvolvimento do sistema de acordo com o ocorrido. |
| Estratégias: | Redução de memória e avaliação do desempenho do sistema de acordo com essa necessidade. Execução do sistema com mais de um usuário testando a entrada no sistema com a mesma identificação. |
| Ferramentas Necessárias: | A ser definido. |
| Critérios de Êxito: | Todos os testes planejados são executados para alcançar os limites impostos do sistema especificado. |
| Considerações Especiais: | O armazenamento persistente usado para o sistema deverá ser reduzido temporariamente, a fim de restringir o espaço disponível para que o banco de dados se desenvolva.  Sincronize o acesso simultâneo dos clientes aos mesmos registros ou contas de dados. |

### **5.1.8 Teste de Volume**

|  |  |
| --- | --- |
| Objetivo da Técnica: | * O número de cliente tentando entrar no sistema com o mesmo usuário e senha * Se tamanho máximo do banco de dados alcançado. |
| Técnica: | Deverão ser usados vários clientes, executando o mesmo processo de entrar no sistema com o mesmo usuário e senha para verificar a reação do sistema. A fim de produzir o conjunto ou o volume de transações no pior caso durante um longo período de tempo.  Será disponibilizado o tamanho máximo do banco de dados e o testador deverá armazena o limite de dados possível no banco para observar a reação do sistema. |
| Estratégias: | Backup do banco de dados para realizar os testes necessários. |
| Ferramentas Necessárias: | A ser definido. |
| Critérios de Êxito: | A técnica suporta o teste de Emulação de Volume. É possível emular, de forma eficaz, grandes quantidades de usuários, dados ou outros aspectos do sistema utilizados em volume e poderá ser capturada uma observação sobre as mudanças de estado do sistema durante o teste de volume. |
| Considerações Especiais: | Que período de tempo seria considerado aceitável para as condições de elevado volume. |

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

|  |  |
| --- | --- |
| Objetivo da Técnica: | Testar acessar o sistema alternando entre usuário e senha inválidos, Segurança do aplicativo envolvendo dados do sistema. |
| Técnica: | Exibir mensagem de erro de usuário ou senha através de autenticação. Testes de integridade de dados |
| Estratégias: | Tentar entrar no sistema com usuários e senhas inválidos além de um verificar se o sistema está buscando corretamente os dados armazenados no banco. |
| Ferramentas Necessárias: | * Backup banco de dados * Firewall do Windows |
| Critérios de Êxito: | Para cada tipo de usuário os dados sejam disponíveis e todas as transações funcionem como esperado. |
| Considerações Especiais: | O acesso ao sistema deve ser revisado com o desenvolvedor por ser a parte que dará o acesso total ao sistema |

### **5.1.10 Teste de Tolerância a Falhas e de Recuperação**

|  |  |
| --- | --- |
| Objetivo da Técnica: | * Os seguintes tipos de condições estão incluídos no teste para observar e registrar o comportamento-alvo após a recuperação: * Interrupção da energia para o cliente; * Ciclos incompletos (processos de filtragem de dados interrompidos, processos de sincronização de dados interrompidos); * Ponteiros ou chaves de banco de dados inválidas; * Elementos de dados inválidos ou corrompidos no banco de dados. |
| Técnica: | * Interrupção da energia para o cliente: desligue o PC; * Ciclos incompletos: Interromper o sistema na hora filtrar dados ou organiza-los.   Depois que as condições acima ou as condições simuladas tiverem sido alcançadas, as transações adicionais deverão ser executadas e, quando o estado desse segundo ponto do teste for atingido, os procedimentos de recuperação deverão ser disparados.  O teste de ciclos incompletos utiliza a mesma técnica descrita acima, exceto pelos processos de banco de dados propriamente ditos, que deverão ser anulados ou prematuramente encerrados.  O teste das condições a seguir exige que seja atingido um estado conhecido do banco de dados.  Vários campos, ponteiros e chaves de banco de dados deverão ser corrompidos manualmente e diretamente no banco de dados (através das ferramentas de banco de dados). |
| Estratégias: | Acesso direto com o sistema e o banco de dados e finalizar o processo do aplicativo em tempo de execução em momento de filtragem de dados. |
| Ferramentas Necessárias: | Cópia do banco de dados para teste;  Ferramenta de backup a definir. |
| Critérios de Êxito: | Em todos os testes citados acima o banco de dados e o sistema deverá após a conclusão do processo de recuperação retoma a um estado conhecido. |
| Considerações Especiais: | O teste de recuperação é altamente invasivo. Os procedimentos para desconectar cabos (simular perda de energia) talvez não sejam desejáveis ou viáveis. Poderão ser necessários métodos alternativos como, por exemplo, ferramentas de software de diagnóstico;  Serão necessários Recursos dos Sistemas e Bancos de Dados;  Esses testes deverão ser executados em uma máquina isolada. |

### **5.1.11 Teste de Configuração**

|  |  |
| --- | --- |
| Objetivo da Técnica: | Verificar e validar se o aplicativo está funcionando normalmente na máquina do usuário. |
| Técnica: | * Abra e feche vários softwares relacionados que não sejam o objetivo do teste (por exemplo, os aplicativos Microsoft Excel e Word) como parte do teste, ou antes, do início do teste; * Execute as transações selecionadas para simular atores interagindo com softwares que sejam o objetivo do teste e com os que não sejam o objetivo do teste. |
| Estratégias: | Projetar os testes de configurações para avaliar diversas categorias de usuários, para determinar as prováveis configurações que vão ser encontradas. |
| Ferramentas Necessárias: | * Restaurador e reprodutor de imagem da configuração básica * Ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.). |
| Critérios de Êxito: | A cada teste realizado as transações são concluídas com êxito e sem falhas. |
| Considerações Especiais: | * Que software, que não seja o objetivo do teste, é necessário, está disponível e é acessível na área de trabalho? * Quais os aplicativos normalmente usados? * Que dados estão em execução nos aplicativos. |

### **5.1.12 Teste de Instalação**

|  |  |
| --- | --- |
| Objetivo da Técnica: | * Nova instalação: uma nova máquina, em que nunca foi instalado anteriormente o FORTESI; * Atualização: uma máquina em que foi instalado anteriormente o FORTESI, na mesma versão; * Atualização: uma máquina em que foi instalado anteriormente o FORTESI, em uma versão mais antiga. |
| Técnica: | * Desenvolva scripts automatizados ou manuais para validar a condição da máquina-alvo; * Inicie ou execute a instalação; * Utilizar um subconjunto predeterminado para executar as transações. |
| Estratégias: | Experimentar a instalação do objetivo do teste em cada configuração de hardware exigida nas condições. |
| Ferramentas Necessárias: | A técnica exige as seguintes ferramentas:   * Restaurador e reprodutor de imagem da configuração básica. * Ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.). |
| Critérios de Êxito: | A técnica suporta o teste da instalação do produto desenvolvido em uma ou mais configurações de instalação. |
| Considerações Especiais: | Que transações de devem serem selecionadas para constituir um teste que comprove que o software FORTESI foi instalado com êxito e que não está faltando nenhum componente de software principal. |

# 6. Critérios de Entrada e de Saída

## 6.1 Planos de Teste

### **6.1.1 Critérios de Entrada de Plano de Teste**

O plano de teste será iniciado após o fim da implementação dos requisitos especificados no documento de requisitos.

Ao final de cada implementação, serão realizados os testes referentes aos requisitos implementados.

* Segurança
* Tempo de resposta
* IHC
* Criptografia
* Armazenamento de dados

### **6.1.2 Critérios de Saída de Plano de Teste**

Os testes de uns requisitos serão finalizados e concluídos após uma bateria de teste e verificação sobre a presença de erros ou bugs ou em caso de erros que impossibilite a continuidade dos testes referentes.

Ao fim de todos os testes dos requisitos, caso não encontre problemas (Erros), a execução do plano de teste será finalizada, ou pelo contrário será suspensa até a resolução do problema.

* Segurança: Feita através de login e senha pessoal e intransferível;
* Tempo de Resposta: Chamada entre um formulário e outro;
* IHC: Transparência na interface em relação às funcionalidades que podem ser executadas;
* Criptográfica: Tipo de encriptação de dados e seu nível de segurança;
* Armazenamento de Dados: Banco usado, tamanho de armazenamento e limite acesso.

### **6.1.3 Critérios de Suspensão e de Reinício**

Na existência de algum problema de desenvolvimento que impossibilite a continuidade da execução dos testes dos requisitos, o plano de teste do requisito será suspenso até que o problema seja resolvido.

Após a resolução do problema, os testes serão novamente executados a fim de validar o requisito.

Caso se encontre algum problema e seja possível continuar, os testes dos requisitos continuarão sendo executados.

Ao executar os testes de um requisito e não encontrar erros, os testes do requisito serão dados como concluídos.

## 6.2 Ciclos de Teste

### **6.2.1 Critérios de Entrada de Ciclo de Teste**

### **6.2.2 Critérios de Saída de Ciclo de Teste**

Os testes devem passar por todos os ciclos, sendo que, nos últimos testes, os possíveis defeitos e exceções terão que estar corrigidos.

### **6.2.3 Término Anormal do Ciclo de Teste**

Só ocorrera suspensão caso haja uma falha no planejamento de plano de testes.

# 7. Produtos Liberados

## Sumários de Avaliação de Testes

A ser definido.

## Relatórios da Cobertura de Teste

Será feito um formulário para cada teste executado. Isso deverá incluir o Nome do Teste, o Caso de Uso ou a Especificação Suplementar com qual o teste encontra-se relacionado, a Data do Teste, o Nome do Testador, e os Resultados do Teste.

## Relatórios da Qualidade Perceptível

Os relatórios serão feitos a cada teste executado, neles terá informações e resumos dos testes feitos, e a qualidade dos testes, O software responsável por reportar a qualidade perceptível do produto será o Rational. A qualidade dos testes será medida através cincos parâmetros:

* **Alcance:** deve ser capaz de lidar com várias tecnologias. A maioria dos aplicativos modernos contém vários idiomas e sistemas que são ligados entre si de forma complexa.
* **Profundidade:** deve ser capaz de gerar mapas completos e detalhados da arquitetura do aplicativo do Graphical User Interface (GUI), ferramenta de captura, processamento e análise de imagem, para o banco de dados. Sem essa detalhada arquitetura, seria impossível obter contextualização da aplicação.
* **Tornar o conhecimento explícito de engenharia de software:** deve ser capaz de verificar a aplicação inteira contra centenas de padrões de implementação que codificam as melhores práticas de engenharia.
* **Métricas acionáveis:** as métricas de qualidade não devem apenas informar, mas também orientar sobre como realizar a melhoria da qualidade do software, mostrando o que fazer primeiro, como fazê-lo, próximos passos etc.
* **Automatização:** finalmente, deve ser capaz de realizar todos os pontos descritos acima de forma automatizada. Nenhum profissional ou equipe pode fazer essa tarefa, muito menos fazê-la em um curto espaço de tempo.

## Registros de Incidentes e Solicitações de Mudança

Para todos os incidentes localizados deverá ser gerado um relatório de problema de software com as seguintes informações:

* Título do item que gerou o problema;
* Versão do item;
* Prioridade (Crítico, Urgente, Rotina);
* Descrição do problema;
* Solução recomendada;
* Nome do testador;
* Para as solicitações de mudança do software será gerado um relatório com as seguintes informações:
* Título do item de software a ser modificado;
* Versão do item, prioridade (crítico/urgente/rotina);
* Descrição da alteração.

## Conjunto de Testes de Regressão e Scripts de Teste de Suporte

Serão elaborados scripts de teste para os tipos de teste definidos. Esses scripts de teste são instruções, passo a passo, que permitem a execução de um teste. Eles podem assumir a forma de instruções de texto documentadas e executadas manualmente, ou de instruções que podem ser lidas pelo computador para ativar a execução automática do teste.

## Produtos de Trabalho Adicionais

Os Produtos de trabalhos são identificados de acordo com o nível de utilização da ferramenta e sua iteração.

* Tela de login, que é a iteração inicial com o sistema.
* Armazenamento interno cliente.
* Tipo de Criptografia.
* Baixo nível de informática requerido.

### **Resultados Detalhados dos Testes**

A ser definido.

### **7.6.2** **Scripts de Teste Funcionais Automatizados Adicionais**

Os scripts de teste serão elaborados para os tipos de teste definidos, Sendo instruções de um passo a passo que permitem a execução de um teste.

As instruções de texto serão documentadas e executadas manualmente, podendo ser lidas pelo computador para ativar a execução automática do teste que são Produtos de Trabalho Adicionais.

### **7.6.3** **Guia de Teste**

A ser definido.

### **7.6.4 Matrizes de Rastreabilidade**

Após a demonstração de todos os testes, assim haverá necessidade de se desenvolver matrizes de rastreabilidade a fim de minimizar os defeitos que existirem.

# 8. Fluxo de Trabalho de Teste

Teste unitário é um dos meios de teste utilizado pela a equipe SOFTECH.

* Métodos ou função.
* Interface: Verificar parâmetros de entradas e saídas.
* Estrutura de dados: Verifica integridade dos dados armazenados.
* Condições de limites: Verifica se aquele modulo ou build opera normalmente nos limites estabelecidos.
* Revelar falhas de sistemas.
* Verifica se o sistema foi corretamente construído.

# 9. Necessidades Ambientais

# 9.1 Hardwares Básicos do Sistema

Os conjuntos de tabelas a seguir apresentam os recursos do sistema necessários ao esforço de teste descrito neste Plano de Teste.

| **Recursos do Sistema** | | |
| --- | --- | --- |
| **Recurso** | **Quantidade** | **Nome e Tipo** |
| Servidor de Banco de Dados | Um | A ser definido |
| Nome do Banco de Dados | Um | MYSQL |
| PCs de Teste Cliente | Dois | A ser definido |
| Repositório de Teste | Um | A ser definido |
| PCs de Desenvolvimento de Teste | Dois | Notbook Dell |

## 9.2 Elementos de Software Básicos do Ambiente de Teste

São necessários os seguintes elementos de software básicos no ambiente de teste deste *Plano de Teste*.

| **Nome do Elemento de Software** | **Versão** | **Tipo e Outras Observações** |
| --- | --- | --- |
| Windows | 7,8 | Sistema Operacional |
| Anti-virus | A ser definido | Software de Detecção e Recuperação de Vírus |

## 9.3 Ferramentas de Produtividade e de Suporte

Serão utilizadas as seguintes ferramentas para suportar o processo de teste deste *Plano de Teste*.

| **Categoria ou Tipo de Ferramenta** | **Nome da Marca da Ferramenta** | **Fornecedor ou Desenvolvida Internamente** | **Versão** |
| --- | --- | --- | --- |
| Gerenciamento de Teste | Rational | IBM Rational Unified Process | A ser definido |
| Controle de Defeitos | Rational | IBM Rational Unified Process | A ser definido |
| Ferramenta para teste de desempenho | Rational | IBM Rational Unified Process | A ser definido |
| Ferramentas DBMS | MYSQL | Oracle corporation | 5.7.0 |

## 9.4 Configurações do Ambiente de Teste

Devem ser fornecidas e suportadas as seguintes Configurações de Ambiente de Teste para este projeto.

| **Nome da Configuração** | **Descrição** | **Implementada na Configuração Física** |
| --- | --- | --- |
| Configuração do usuário comum |  |  |
| Mínima configuração suportada |  |  |

# 10. Responsabilidades, Perfil da Equipe e Necessidades de Treinamento.

## 10.1 Pessoas e Papéis

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

| **Recursos Humanos** | | |
| --- | --- | --- |
| Gerente de Testes | Gerenciar todo o plano de teste fazendo o acompanhamento evolutivo. | Supervisiona o gerenciamento.  Estas são as responsabilidades:   * Combinar missão * Identificar motivadores * Adquirir recursos apropriados * Apresentar relatórios de gerenciamento * Defender os interesses do teste * Avaliar a eficiência do esforço de teste |
| Analista de Teste | Analisa os sucessos do plano de teste e o rumo do mesmo. | Identifica e define os testes específicos a serem conduzidos.  Estas são as responsabilidades:   * Identificar ideias de teste * Definir detalhes dos testes * Determinar os resultados dos testes * Documentar solicitações de mudança * Avaliar a qualidade do produto |
| Designer de Teste | Responsável por ordenar os tipos de testes usadas e quais os fundamentais. | Define a abordagem técnica referente à implementação do esforço de teste.  Estas são as responsabilidades:   * Definir a abordagem dos testes * Verificar as técnicas de teste * Definir os elementos de testabilidade |
| Testador | Responsável pela execução dos testes e verificação e detecção de erros. | Implementa e executa os testes.  Estas são as responsabilidades:   * Implementar os testes e os conjuntos de testes * Executar os conjuntos de testes * Registrar os resultados * Analisar as falhas dos testes e possibilitar a recuperação posterior * Documentar incidentes |
| Administrador do Banco de Dados, Gerente do Banco de Dados | Gerencia o fluxo de dados interno no banco. | Assegura o gerenciamento e a manutenção dos recursos e do ambiente dos dados de teste (banco de dados).  Estas são as responsabilidades:   * Suportar a administração dos dados de teste e das plataformas de teste (banco de dados) |
| Designer | Desenvolvedor da interface de interação. | Identifica e define as operações, os atributos e as associações das classes de teste.  Estas são as responsabilidades:  Define as classes de teste necessárias para suportar os requisitos de testabilidade conforme definido pela equipe de teste. |
| Implementador | Responsável por gerir as informações adicionais no software. | Implementa as classes de teste e os pacotes de teste e efetua testes de unidade nos mesmos.  Estas são as responsabilidades:  Cria os componentes de teste necessários para suportar os requisitos de testabilidade conforme definido pelo designer. |

## 10.2 Perfil da Equipe e Necessidades de Treinamento

Esta seção resume como abordar o perfil da equipe e o treinamento dos profissionais que ocuparão os papéis de teste no projeto.

Esta seção resume como abordar o perfil da equipe e o treinamento dos profissionais que ocuparão os papéis de teste no projeto.

O perfil da equipe recomenda-se que seja em auto nível de interação entre os membros da equipe responsável pelo software, com o intuito de chegar ao mesmo objetivo que é um sistema opto a gerenciar processos, sem haver qualquer tipo de problema seja de incompatibilidade, Desempenho, Componentes dentre outros.

Para o projeto devem ser listadas as necessidades por grau de dificuldade que serão demandadas. E verificar o real desempenho de cada indivíduo na hora do desenvolvimento das tarefas, levando em conta que dependendo do feedback do responsável poderá ocorre o treinamento.

Independente da necessidade de treinamento ou não deve haver um planejamento em torno do treinamento para evitar treinamentos em momentos inadequados. E acionar a equipe de treino exatamente no momento onde é necessária ajuda. Além de planejar uma ajuda da equipe externa a teste que é Administradores de Sistema, Administradores de Banco de Dados e Desenvolvedores, que são profissionais necessários para viabilizar o esforço de teste.

# 11. Marcos da Iteração

| **Marco** | **Data de Início Planejada** | **Data de Início Real** | **Data de Término Planejada** | **Data de Término Real** |
| --- | --- | --- | --- | --- |
| Plano de Iteração combinado | 04/04/2015 | 05/04/2015 | 06/04/2015 | 08/04/2015 |
| Início da iteração | 09/04/2015 | 11/04/2015 | 14/04/2015 | 16/04/2015 |
| Liberação do primeiro build para teste | 20/04/2015 | 19/04/2015 | 25/04/2015 | 24/04/2015 |
| Aceitação do primeiro build para teste | 26/04/2015 | 28/04/2015 | 30/04/2015 | 02/05/2015 |
| Término do ciclo de teste do primeiro build | 04/05/2015 | 04/05/2015 | 07/05/2015 | 10/05/2015 |
| Revisão da Avaliação de Iteração | 11/05/2015 | 11/05/2015 | 13/05/2015 | 13/05/2015 |
| Término da iteração | 15/05/2015 | 15/05/2015 | 18/05/2015 | 18/05/2015 |

# 12. Riscos, Dependências, Suposições e Restrições

| **Risco** | **Estratégia de Diminuição** | **Contingência (O risco se concretizou)** |
| --- | --- | --- |
| As necessidades de o usuário não serem atendidas. | O testador Moises Urias através do doc. De requisitos especificará e diminuirá os riscos.  O Administrador José Lucas entenderá os requisitos necessários com clareza para minimizar o risco. | O testador Moises Urias através do doc. De requisitos especificará e diminuirá os riscos.  O Administrador José Lucas entenderá os requisitos necessários com clareza para minimizar o risco. |
| Queda de energia acarretando perda de dados | O advogado apresentará um equipamento chamado nobreak para evitar a perda de informações devido ao desligamento do PC com a queda de energia.  . | Não há plano de contingência para esta situação. |
| O banco de dados necessitar de uma atualização. | O Administrador José Lucas tentará assegurar que o Banco de Dados seja atualizado regularmente conforme exigido pelo testador José Lucas. | Restaurar os dados e reiniciar  Limpar o Banco de Dados |

| **Dependência entre** | **Impacto Potencial da Dependência** | **Proprietários** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

| **Suposição a ser comprovada** | **Impacto se a suposição for incorreta** | **Proprietários** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

| **Restrição** | **Impacto da restrição no esforço de teste** | **Proprietários** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# 13. Procedimentos e Processos de Gerenciamento

Paralisação da execução do plano de teste após a verificação de um erro ou fator não humano que impossibilite a continuidade, corrigir os erros e informar via relatório o que foi o problema, qual seu impacto, o que foi resolvido e quais as ferramentas usadas após esse processo de paralisação, verificação e solução do erro ou problema.

## 13.1 Medição e Avaliação da Extensão do Teste

Verificação do andamento do plano de teste quais os progressos, problemas enfrentados e a situação atual em cada setor, avaliando o nível de eficácia em cada solução de erro.

## 13.2 Avaliações dos Produtos Liberados deste Plano de Teste

Relatórios contendo os processos usados no plano de teste e cada decisão tomada detalhadamente, mostrando os pontos que impactam na estrutura e funcionalidade do software.

## 13.3 Relatos de Problemas, Seleção de Pessoas para Resolvê-los e Busca de Soluções

Os problemas e erros encontrados serão relatados através de relatórios especificando o ponto de que tipo de significância tem para a arquitetura do software.

Os testadores após fazerem este relatório os levarão para uma reunião entre gerente e testadores que junto articularão uma solução para um determinado erro.

## 13.4 Gerenciamentos de Ciclos de Teste

O gerente de projeto é responsável por gerenciar as atividades efetuadas pelos testadores e definir as iterações da fase de testes, que será de acordo com as iterações da implementação do sistema.

## Estratégias de Rastreabilidade

Reflita sobre estratégias de rastreabilidade adequadas referentes à:

* Cobertura de Teste em relação às Especificações — possibilita a medição da extensão do teste.
* Motivações de Teste — possibilitam a avaliação da relevância dos testes a fim de ajudar a determinar se eles deverão ser mantidos ou não.
* Elementos de Design de Software — possibilitam o rastreamento das mudanças de design subsequentes que exigirão que os testes sejam executados novamente, ou seja, cancelados.
* Solicitações de Mudança Resultantes — fazem com que os testes que descobriram a necessidade da mudança sejam identificados e executados novamente para verificar se a solicitação de mudança foi efetuada com êxito.

## 13.6 Aprovação e Encerramento

Ao fim dos testes o sistema deverá corresponder às necessidades e planejamento inicial, mesmo após os testes ter ocorrido qualquer falha na aplicação deverá ser corrigida e automaticamente refazer os testes.

Os testes são realizados por:

Lucas Castanheiro – Gerente de teste;

José Lucas – Testador.