Gerador BurnDown
Plano de Teste de <Iteração/Mestre>

Versão <1.0>

Histórico da Revisão

Data	Versão	Descrição	Autor
13/11/2014	1.0	Programa tem como propósito gerar um gráfico para acompanhamento da produção das estórias de usuários	Diego Valder Wander

Índice Analítico

Plano de Teste de <Iteração/Mestre>

1. Introdução

1.1 Finalidade

A finalidade do Plano de Teste de Iteração é reunir todas as informações necessárias para planejar e controlar o 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 documento *Plano de Teste* referente ao Gerador BurnDown suporta os seguintes objetivos:

- Os itens a serem inspecionados pelos testes consistem nas classes, métodos, variáveis e estruturas de controles que constituem o programa.

- O objetivo de se efetuar os testes dessas estruturas é o de certificar que elas funcionarão corretamente.

- Utilizaremos, predominantemente testes do tipo caixa branca.

. - Para se efetuar os testes utilizaremos a IDE Eclipse e o framework JUnit

1.2 Escopo

Os teste serão basicamente os de unidade, com foco principal nos métodos. Os demais tipos de testes não serão realizados por fugir ao escopo do nosso trabalho neste momento.

1.3 Público-alvo

Como se trata de um trabalho escolar o público alvo principal consiste basicamente nos professores e alunos.

1.4 Terminologia e Acrônimos do Documento

Não possui.

1.5 Referências

Não possui.

2. Missão de Avaliação e Motivação dos Testes

Temos como propósito cobrir a maior parte do código do programa com testes de unidade, de modo a ampliar a garantia de se conseguir a menor quantidade de falhas possível.

2.1 Fundamentos

A razões para se fazer os testes do programas são diversas, a começar pela maior confiabilidade quanto ao funcionamento do software. Quando se negligência esta valiosa etapa a probabilidade do programa ir para o ambiente de produção apresentando uma gama maior de defeitos cresce. Nosso foco consiste em se fazer testes de unidades que visão garantir que cada peça do código esteja funcionando corretamente e, quando não se está, corrigi-la. O plano de testes, bem como o conjunto dos testes em si, foi projetado para se efetuar os testes do sistema que tem como funcionalidade gerar os gráficos de BurnDown. O programa é divido num conjunto de classes que, por sua vez serão testadas separadamente, bem como os membros que as constitui. O programa foi desenvolvido por uma aluna do curso de sistemas da informação ao longo do seu trabalho de conclusão do curso.

2.2 Missão de Avaliação

- localizar o maior número de erros possível
- localizar problemas importantes
- avaliar os riscos da qualidade perceptível
- informar sobre os riscos perceptíveis do projeto
- informar sobre a qualidade do produto
- satisfazer os envolvidos
- informar sobre os testes

- cumprir as determinações do processo

- e assim por diante

2.3 Motivadores dos Testes

Neste caso específico o principal elemento motivador dos testes consiste na possibilidade de aprendizado ao se efetuá-los. De modo geral a motivação principal e fundamental dos testes encontra-se na possibilidade de se detectar erros o mais cedo possível e se corrigi-los precocemente sem causar maiores danos ou prejuízos ao projeto ao se faze-lo no ambiente de produção.

3. Itens de Teste-Alvo

A listagem abaixo identifica os itens de software, de hardware e elementos de suporte do produto que foram identificados como objetivos dos testes. Esta lista representa os itens que serão testados.

Basicamente os itens as serem testados no nosso projeto consiste nos elementos do código em si tais como:

- variáveis
- métodos
- classes
- o software em si.

4. Resumo dos Testes Planejados

4.1 Resumo das Inclusões dos Testes

- Basicamente os testes que serão executados serão os testes de unidade e um teste caixa preta, visando este demonstrar o funcionamento do sistema em sua integralidade.

5. Abordagem dos Testes

A abordagem para a elaboração dos testes consiste em se compreender o modo como o código foi construído e, a partir daí, se projetar casos de testes específicos

que sejam eficientes para o propósito de se testar o código da melhor maneira possível reduzindo-se a probabilidade dele conter erros. Como o trabalho será realizado em grupo cada integrante ficará incumbido de testar algumas classes do código com seus respectivos membros constitutivos. É importante deixar claro que dois desafios se apresentam na realização desta tarefa, o primeiro é o de compreender o código do sistema e o segundo é o de se projetar os testes incluindo aqui sua implementação. A ferramenta a ser utilizada será a IDE do Eclipse com o JUnit integrado.

5.1 Catálogos Iniciais de Idéias de Teste e Outras Fontes de Referência

Nosso catálogo de idéias iniciais será extraído da internet na consulta de sites especializados. O site do Junit, bem como o próprio pacote que contêm o plugin traz exemplos de elaboração de testes diversos.

5.2 Tipos e Técnicas de Teste

5.2.1 Teste de Integridade de Dados e de Banco de Dados

Nosso programa não possui um banco de dados que possa e\ou precisa ser testado, ele não trabalha com um, portanto este teste não será executado.

Objetivo da Técnica:	*[Experimentar processos e métodos de acesso a banco d da UI para que você possa observar e registrar comport incorretos ou a existência de dados corrompidos.]*
Técnica:	*[Dispare cada processo e método de acesso a banco de dados válidos e inválidos ou solicitações de dados em co* *Inspecione o banco de dados para assegurar que os dad conforme o planejado e que todos os eventos de banco d forma adequada, ou revise os dados retornados para ass corretos foram recuperados pelas razões corretas.]*
Estratégias:	*[Descreva uma ou mais estratégias que podem ser usad observar, de forma precisa, os resultados do teste. A est elementos do método através do qual a observação pode características dos resultados específicos que indicam u falha do teste. O ideal é que as estratégias sejam autove que os testes automatizados façam uma avaliação inicia teste. No entanto, tenha atenção para reduzir os riscos i*

	determinação automática dos resultados.]
Ferramentas Necessárias:	*[A técnica exige as seguintes ferramentas:* *Ferramenta de Automação de Scripts de Teste* *restaurador e reprodutor de imagem da configuração ba* *ferramentas de backup e de recuperação* *ferramentas de monitoramento de instalação (registro, a memória etc)* *ferramentas e utilitários SQL de banco de dados* *ferramentas de geração de dados]*
Critérios de Êxito:	*[A técnica suporta o teste de todos os principais processo a banco de dados.]*
Considerações Especiais:	*[Os testes poderão necessitar de drivers ou de um ambie desenvolvimento DBMS para inserir ou modificar dados de dados.* *Os processos deverão ser disparados manualmente.* *Deverão ser usados bancos de dados pequenos ou de tam número limitado de registros) para aumentar a visibilida eventos não aceitáveis.]*

5.2.2 Teste de Funcionamento

[O teste de funcionamento do objetivo do teste deve concentrar-se em todos os requisitos de teste que possam ser diretamente associados a casos de uso ou funções e regras de negócios. Esse teste tem por fim verificar a adequada aceitação, processamento e recuperação dos dados, e a implementação apropriada das regras de negócios. Esse tipo de teste baseia-se em técnicas de caixa preta; ou seja, verificar o aplicativo e seus processos internos interagindo com o aplicativo através da Interface Gráfica do Usuário (GUI) e analisar a saída ou os resultados. A tabela a seguir identifica um resumo do teste recomendado para cada aplicativo.]

Objetivo da Técnica:	Verificar características referentes a funcionalidade do sistema
Técnica:	Executar o programa e entrar com dados válidos (No nosso caso apenas confe adequada e retorno esperado corresponde à expectativa). Verificar se erros ocorreram e se as mensagens de erro correspondente

	emitidas.
Estratégias:	A principal estratégia para verificar se erros foram gerados ou não consiste em observar e analisar diretamente os resultados da execução dos testes. No nosso caso, ao executar o sistema, ele precisa gerar um gráfico. Os testes de unidade devem ser av com o auxílio do JUnit.
Ferramentas Necessárias:	IDE Eclipse com o JUnit.
Critérios de Êxito:	A técnica suporta o teste de: Todos aspectos do sistema.
Considerações Especiais:	É preciso ficar atento ao processo de referênciar as bibliotecas que envolvem o sistema.

5.2.3 Teste de Ciclos de Negócios

[O Teste de Ciclos de Negócios deverá emular as atividades executadas no <Nome do Projeto> ao longo do tempo. Deverá ser identificado um período como, por exemplo, um ano, e deverão ser executadas as transações e atividades que ocorreriam durante esse período de um ano. Isso incluirá todos os ciclos diários, semanais e mensais, assim como os eventos que mudam com as datas como, por exemplo, lembretes.]

Objetivo da Técnica:	Experimentar processos de segundo plano e do objetivo do teste de acordo com as programações e os modelos de negócios necessários, a fim de observar e registrar o comportamento-alvo.
Técnica:	O teste incluirá o uso de casos válidos e inválidos para verificar se: - Os resultados esperados ocorrerão quando forem usados dados válidos. - As mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos. - Cada regra de negócio será aplicada de forma adequada.
Estratégias:	Os teste serão executados na IDE Eclipse com auxílio do

	framework JUnit. A chave para o sucesso ou fracasso do processo de elaboração dos testes consiste em projetá-los adequadamente bem como utilizar o framework JUnit de modo adequado.
Ferramentas Necessárias:	IDE Eclipse com framework JUnit.
Critérios de Êxito:	Execução dos testes com retorno de mensagem de erro em caso de falha.
Considerações Especiais:	Não há.

5.2.4 Teste de Interface do Usuário

O sistema não disponibiliza uma interface com o usuário. Quando o sistema é executado aparece apenas o gráfico.

Objetivo da Técnica:	*[Experimentar o seguinte para observar e registrar a co padrões e o comportamento-alvo:* *A navegação pelo objetivo do teste para verificar se refl funções de negócios, incluindo a navegação janela a jane o uso de métodos de acesso (teclas de tabulação, movime aceleradoras).* *As características e os objetos das janelas poderão ser e por exemplo, menus, tamanho, posição, estado e foco.]*
Técnica:	*[Crie ou modifique testes para cada janela a fim de veri adequada e os estados de objeto apropriados para cada aplicativo.]*
Estratégias:	*[Descreva uma ou mais estratégias que podem ser usad observar, de forma precisa, os resultados do teste. A est elementos do método através do qual a observação pode características dos resultados específicos que indicam u falha do teste. O ideal é que as estratégias sejam autove que os testes automatizados façam uma avaliação inicia teste. No entanto, tenha atenção para reduzir os riscos i determinação automática dos resultados.]*

Ferramentas Necessárias:	*[A técnica necessita da Ferramenta de Automação de Sc*
Critérios de Êxito:	*[A técnica suporta o teste de cada tela ou janela principa pelo usuário final.]*
Considerações Especiais:	*[Nem todas as propriedades referentes a objetos person poderão ser acessadas.]*

5.2.5 Teste de Segurança e de Controle de Acesso

Este teste não será executado porque não há necessidade de fazê-lo.

Objetivo da Técnica:	*[Experimentar o objetivo do teste nas seguintes condiçõ registrar o comportamento-alvo:* *Segurança no nível do aplicativo: um ator poderá acess ou os dados para o quais seu tipo de usuário tenha rece* *Segurança no nível do sistema: somente os atores com a aplicativos têm permissão para acessá-los].*
Técnica:	*[Segurança no nível do aplicativo: Identifique e liste ca funções ou os dados para os quais cada tipo tem permis* • *Crie testes para cada tipo de usuário e verifique cada perm transações específicas para cada tipo de usuário.* • *Modifique o tipo de usuário e execute novamente os testes p usuários. Em cada caso, verifique se as funções ou dad corretamente disponíveis ou se têm seu acesso negado.* *Acesso no nível do sistema: Consulte Considerações Esp*
Estratégias:	*[Descreva uma ou mais estratégias que podem ser usad observar, de forma precisa, os resultados do teste. A est elementos do método através do qual a observação pode características dos resultados específicos que indicam u falha do teste. O ideal é que as estratégias sejam autove que os testes automatizados façam uma avaliação inicia teste. No entanto, tenha atenção para reduzir os riscos i determinação automática dos resultados.]*
Ferramentas Necessárias:	*[A técnica exige as seguintes ferramentas:* *Ferramenta de Automação de Scripts de Teste* *Ferramentas de investigação e contra a violação da seg*

	Ferramentas de Administração da Segurança do Sistema
Critérios de Êxito:	*[A técnica suporta o teste das funções apropriadas. É po dados afetados pelas configurações de segurança sejam tipo de ator conhecido.]*
Considerações Especiais:	*[O acesso ao sistema deverá ser revisto ou discutido con sistemas ou de rede adequado. Talvez esse teste não seja poderá ser uma das funções da administração de sistema*

5.3 Casos de Teste

Os casos de testes que serão executados consistem, inicialmente, nos testes de unidades que serão armazenados em outro ambiente.

6. Critérios de Entrada e de Saída

6.1 Plano de Teste

6.1.1 Critérios de Entrada de Plano de Teste

O critério a ser utilizado consiste na disponibilidade dos participantes do grupo para dar início a execução do plano de teste.

6.1.2 Critérios de Saída de Plano de Teste

O critério utilizado para determinar a conclusão ou término da execução do plano de teste consiste em se testar todas as classes do sistema.

6.1.3 Critérios de Suspensão e de Reinício

A disponibilidade dos membros da equipe bem como a proximidade da data para a entrega consistem em critérios determinantes quanto a suspensão e\ou reinício dos testes.

7. Produtos Liberados

[Nesta seção, liste os vários artefatos que serão criados pelo esforço de teste e que serão produtos liberados úteis aos vários envolvidos do esforço de teste. Não liste todos os produtos do trabalho; liste apenas os que propiciam benefícios diretos tangíveis aos envolvidos e os que permitem medir o êxito do esforço de teste.]

7.1 Sumários de Avaliação de Testes

Os sumários são organizados numa ordem lógica de acordo com os passos que o esforço de teste exige. O seu conteúdo consiste em descrever de modo resumido o conjunto de itens que compõem os documentos que fazem parte dos artefatos de testes bem como os critérios para sua execução.

7.2 Geração de Relatórios sobre Cobertura de Teste

[Forneça um breve resumo da forma e do conteúdo dos relatórios usados para medir a extensão do teste e indique com que freqüência eles serão gerados Forneça uma indicação referente ao método e às ferramentas usados para registrar, medir e reportar a extensão do teste.]

O relatório adotado para se medir a cobertura dos testes.

7.3 Registros de Incidentes e Solicitações de Mudança

[Forneça um breve resumo do método e das ferramentas usados para registrar, rastrear e gerenciar incidentes dos testes, as solicitações de mudança associadas e seus status.]

Não adotaremos uma ferramenta específica para isto.

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

Não ocorrerão.

7.5 Produtos de Trabalho Adicionais

7.5.1 Resultados Detalhados dos Testes

Não serão usados.

7.5.2 Scripts de Teste Funcionais Automatizados Adicionais

Não serão usados.

7.5.3 Guia de Teste

Não serão usados.

7.5.4 Matrizes de Rastreabilidade

Não serão usadas.

9. Necessidades Ambientais

9.1 Hardware Básico do Sistema

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

Recurso	Quantidade	Nome e Tipo
Servidor de Banco de Dados	0	
Rede ou Sub-rede	0	Não incluso no projeto.
	0	Não incluso no projeto.
Nome do Servidor Nome do Banco de Dados	0	Não incluso no projeto.
PCs de Teste Cliente		PC qualquer.
Inclua requisitos de configuração especiais		Não existem.
Repositório de Teste		GitHub
Rede ou Sub-rede		Não incluso no projeto.
Nome do Servidor		Não incluso no projeto.
PCs de Desenvolvimento de Teste		Não incluso no projeto.

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 Observaç
NT Workstation	Não há necessidade.	Sistema Operacional
Windows 2000	Qualquer	Sistema Operacional
Internet Explorer	Não há necessidade.	Navegador da Internet
Netscape Navigator	Não há necessidade.	Navegador da Internet
Microsoft Outlook	Não há necessidade.	Software Cliente de E-Mail
Network Associates McAffee Virus Checker	Não há necessidade.	Software de Detecção e Recuperação de V

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
Gerenciamento de Teste	JUnit	
Controle de Defeitos	Não há necessidade.	
Ferramenta ASQ para teste funcional	Não há necessidade.	
Ferramenta ASQ para teste de desempenho	Não há necessidade.	
Gerador de Perfil ou Monitor de Cobertura de Teste	JUnit	
Gerenciamento de Projeto	Não há necessidade.	
Ferramentas DBMS	Não há necessidade.	

9.4 Configurações do Ambiente de Teste

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

Obs.: Não há necessidade de se fazer configurações especiais.

Nome da Configuração	Descrição	Implementada na Configuração Física
onfiguração do usuário comum		
ínima configuração suportada		
otivada por funções visuais e motoras		
stema Operacional Internacional de Dois Bytes		
stalação de Rede (não cliente)		