

<Gerador BurnDown > Plano de Teste de <Iteração/Mestre>

Versão <1.0>

Histórico da Revisão

Data	Versão	Descrição	Autor
26/10/2015	1.0	Atualização	Josimar, Kátia
05/11/2015	2.0	Atualização	Micael, Felipe

Índice Analítico

1. Introdução

1.1 Finalidade

1.2 Escopo

1.3 Público-alvo

1.4 Terminologia e Acrônimos do Documento

1.5 Referências

1.6 Estrutura do Documento

2. Missão de Avaliação e Motivação dos Testes

2.1 Fundamentos

2.2 Missão de Avaliação

2.3 Motivadores dos Testes

3. Itens de Teste-Alvo

....

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:

- Identificação dos componentes de software que devem ser testados.
- Enumerar todos os componentes e requisitos para serem testados
- Descrição e recomendação de estratégias para que sejam efetuados os testes.
- Listar os elementos liberados do projeto

1.2 Escopo

Iremos utilizar no Gerador BurnDown testes unitários. O teste unitário irá verificar métodos da classe e irá analisar a performance do mesmo.

1.3 Público-alvo

Profissionais que já trabalham com projetos e desejam conhecer essa técnica ágil para aplicar em seus projetos com características de escopo aberto.

1.4 Terminologia e Acrônimos do Documento

Significado SCRUM	<i>É um processo ágil que permite manter o foco na entrega do maior valor de negócio, no menor tempo possível</i>
<i>Baseline</i>	<i>Conjunto de itens de configuração que conseguiram um estado comprovado de estabilidade.</i>

1.5 Referências

Template de Plano de Gerenciamento de Configuração, 2015.

Plano de Gerenciamento do Projeto (GB), Versão 1.0

1.6 Estrutura do Documento

Este documento contém todos os passos necessários do plano de teste do software Burndown.

Os tópicos estão numerados e detalhados de forma clara e objetiva.

Todas as fases de testes, itens avaliados e resumo dos testes está descrita neste documento.

2. Missão de Avaliação e Motivação dos Testes

Neste documento serão contemplados os principais testes a serem executados no Gerador BurnDown.

O objetivo é encontrar defeitos no Software, para que estes possam ser corrigidos pela equipe de programadores, antes da entrega final, buscando a satisfação do cliente.

2.1 Fundamentos

Considerando que o Software desenvolvido deve obedecer alguns padrões de qualidade, desenvolvimento, confiabilidade e segurança, serão aplicados testes para validar as funcionalidades propostas mapeadas na elaboração da alternativa de solução escolhida.

Os testes serão aplicados para se obter uma qualidade alta do software e evitar retrabalhos.

Serão aplicadas técnicas para encontrar o maior número de erros possíveis e posteriormente solucioná-los.

2.2 Missão de Avaliação

Localizar problemas importantes e avaliar os riscos da qualidade perceptível

2.3 Motivadores dos Testes

Detectos de qualidade, riscos técnicos, de projeto, casos de uso, requisitos funcionais e de risco não funcionais.

3. Itens de Teste-Alvo

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

As principais funções (métodos) do código fonte;

Funcionalidades do BurnDown implementadas;

A execução do sistema em diferentes plataformas.

4. Resumo dos Testes Planejados

Esta seção apresenta os recursos recomendados para o projeto <Burndown>, suas principais responsabilidades e seu conjunto de conhecimentos ou de habilidades.

4.1 Resumo das Inclusões dos Testes

Caso de Teste 1: Dia e início do trabalho.

Caso de Teste 2: Trabalhos terminaram no tempo previsto

Caso de Teste 3: Quantos dias é a duração dos testes

Caso de Teste 4: Testar todas as classes ao mesmo tempo

5. Abordagem dos Testes

Os testes serão executados de forma manual e/ou automatizada.

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

[Forneça uma listagem dos recursos existentes que serão consultados para estimular a identificação e a seleção de testes específicos a serem conduzidos. É fornecido um Catálogo de Idéias de Teste de exemplo na seção de exemplos do RUP.]

5.2 Tipos e Técnicas de Teste

5.2.1 Teste de Integridade de Dados e de Banco de Dados

N.A

5.2.2 Teste de Funcionamento

<i>Experimental a funcionalidade do objetivo do teste, incluindo a navegação, a entrada, o processamento e a recuperação de dados a fim de observar e registrar o comportamento-alvo.</i>	
Técnica:	<i>Testes de caixa branca avaliando os métodos do código do software</i>

Estratégias:	<i>A estratégia consistiu em testar a classes, métodos e parâmetros</i>
Ferramentas Necessárias:	<i>Junit</i>
Critérios de Êxito:	<i>Testes de dia de início, Gerador Fim, Gerador de Quantidade de dias e Gerador</i>
Considerações Especiais:	<i>Alguns parâmetros foram colocados de forma inadequada</i>

5.2.3 Teste de Ciclos de Negócios

N.A

5.2.4 Teste de Interface do Usuário

N.A

5.2.5 Teste de Segurança e de Controle de Acesso

N.A

5.3 Casos de Teste

Caso de Teste 1:

Entrada: 1

Saída: 1

Classe Relacionada: InícioTeste

Teste aprovado visto que o valor de entrada é o mesmo da saída. O objetivo do teste é mostrar se o dia e o início do trabalho.

Caso de Teste 2:

Entrada: 1

Saída: 2

Classe Relacionada: InícioTeste

Teste reprovado visto que o valor de entrada difere do valor de saída.

Caso de Teste 3:

Entrada: 30

Saída: 30

Classe Relacionada: FimTeste

Teste aprovado visto que o valor de entrada é o mesmo da saída. O objetivo do teste é mostrar se os trabalhos terminam no tempo previsto.

Caso de Teste 4:

Entrada: 30

Saída: 31

Classe Relacionada: FimTeste

Teste reprovado visto que o valor de entrada difere do valor de saída

Caso de Teste 5:

Entrada: 30,1

Saída: 29

Classe Relacionada: QuantidadeDeDiasTeste

Teste aprovado, visto que 1 trabalho dura no máximo 30 dias e o resultado esperado teria que ser um valor igual a 29, visto que um trabalho teria no máximo 29 dias. Valor passado como parâmetro seria uma subtração dos valores.

Caso de Teste 6:

Entrada: 30,1

Saída: 28

Classe Relacionada: QuantidadeDeDiasTeste

Teste Reprovado, visto que 1 trabalho durando no máximo 30 dias, teríamos no máximo 29 dias para ser realizado todo o trabalho e a saída padrão foi 29 o que viola o teste.

Caso de Teste 7:

Entrada: FimTeste, InícioTeste, FimTeste

Saída: FimTeste, InícioTeste, FimTeste

Classe Relacionada: AllTests

Teste Aprovado, visto que o principal objetivo é testar todas as classes do projeto.

Caso de Teste 8:

Entrada: FimTeste, InícioTeste, FimTeste

Saída: fimteste, InícioTeste, FimTeste

Classe Relacionada:

Teste Reprovado, visto que o nome da classe passado por parâmetro (fimteste) difere do valor de entrada.

6. Critérios de Entrada e de Saída

6.1 Plano de Teste

6.1.1 Critérios de Entrada de Plano de Teste

Quantidade de dias para implementar a melhoria no projeto

6.1.2 Critérios de Saída de Plano de Teste

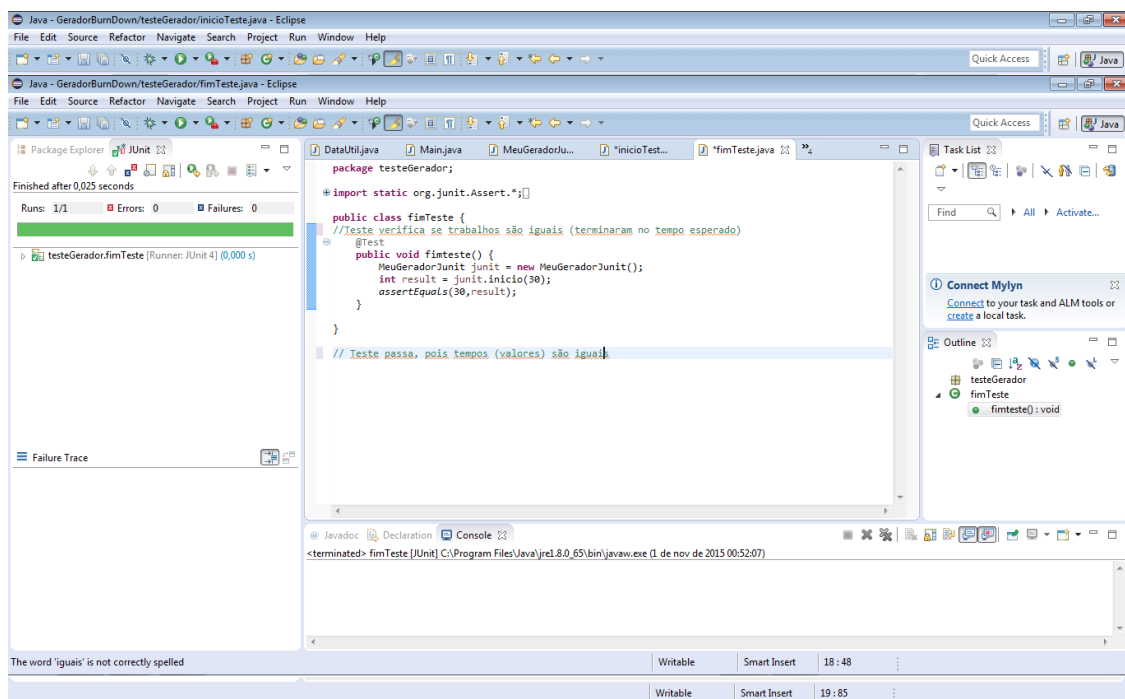
Quantidade de dias esperada na implementação da melhoria do sistema.

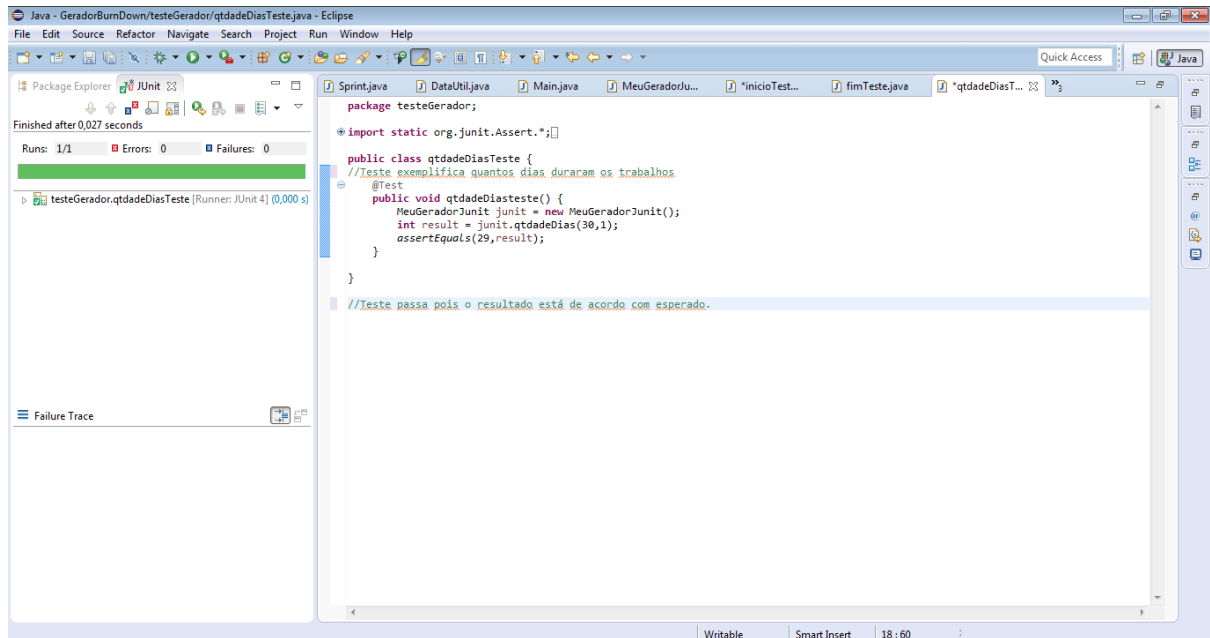
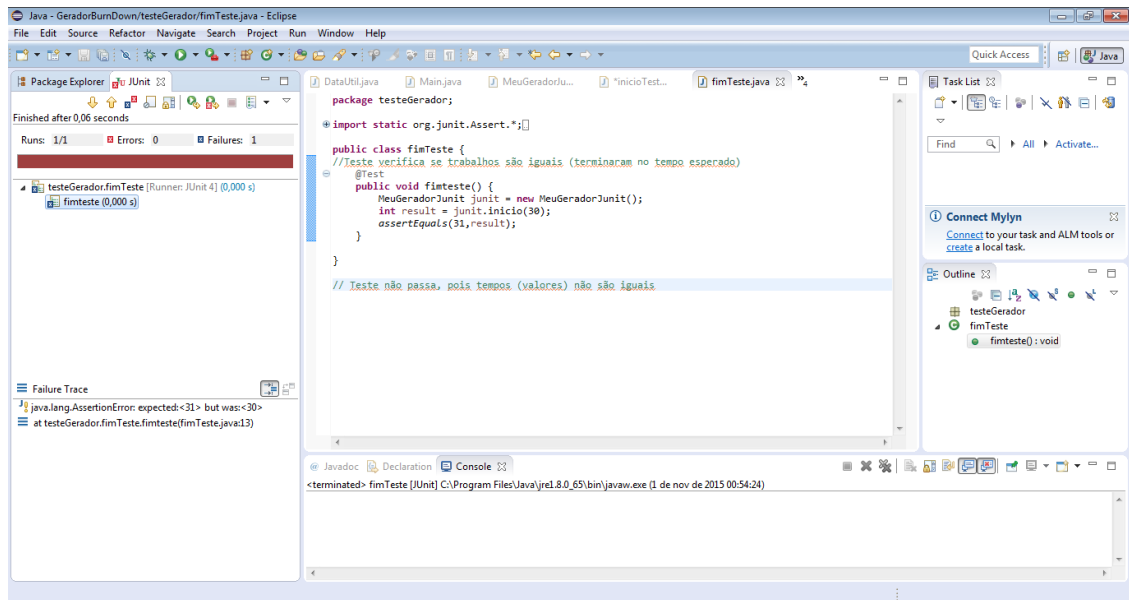
6.1.3 Critérios de Suspensão e de Reinício

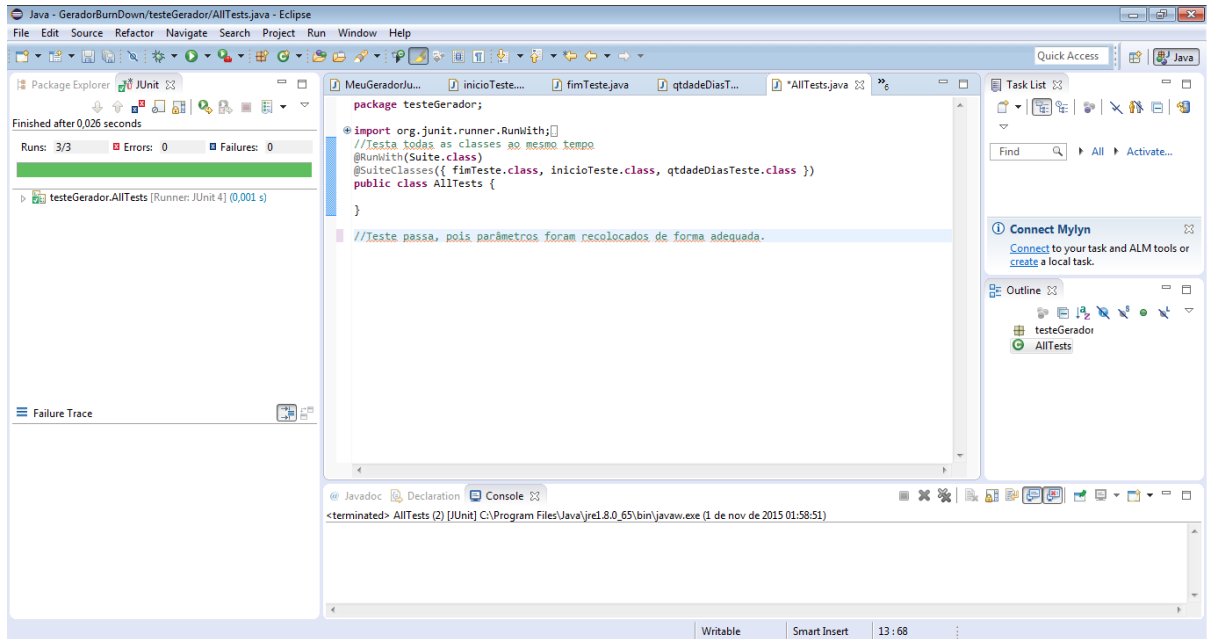
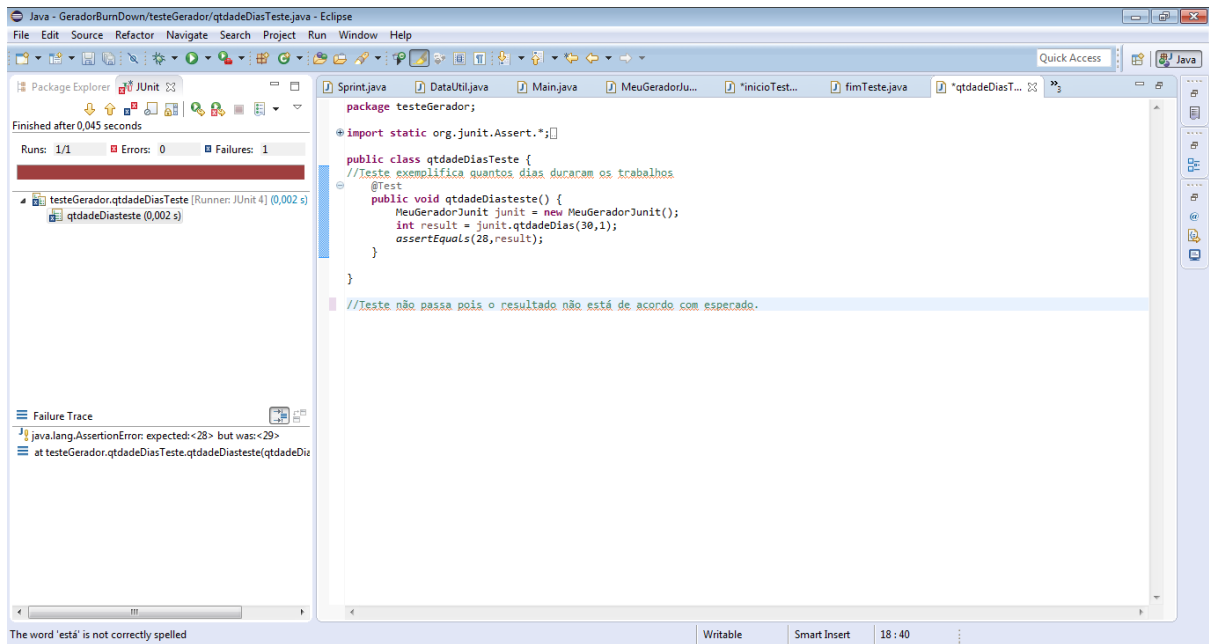
N.A

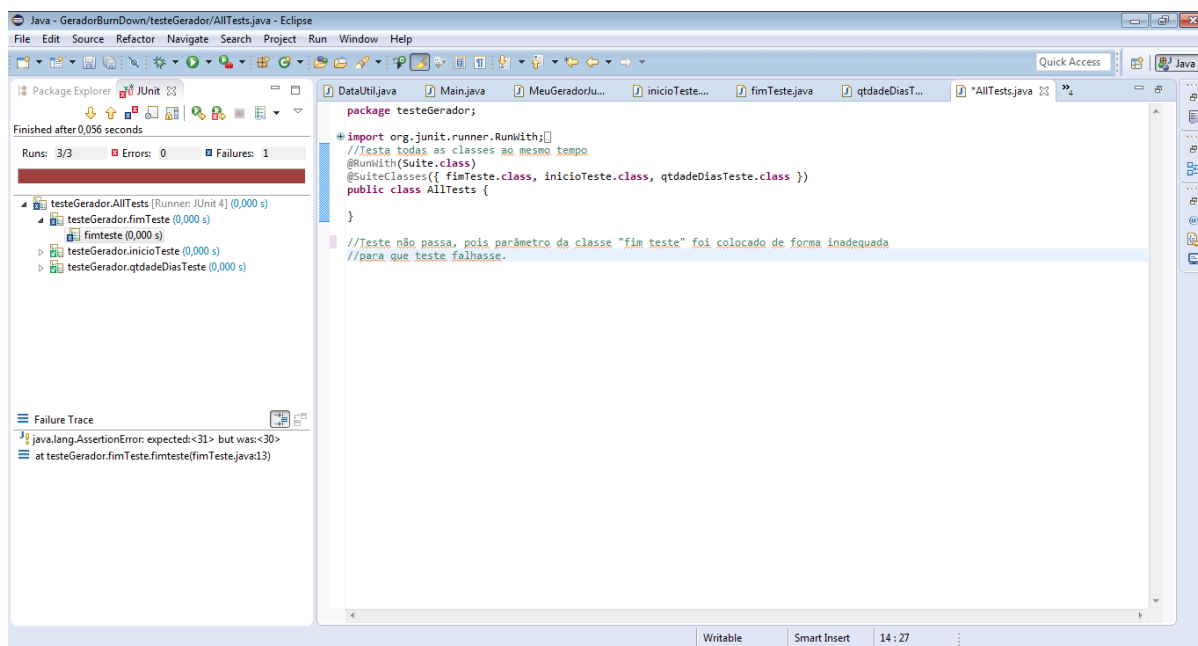
7. Produtos Liberados

7.1 Resultados Detalhados dos Testes









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*.

Versão	Tipo e Outras Observações		
Windows 7		Sistema Operacional	
Java Eclipse		Software de Desenvolvimento	
JUnit		Framework de testes	

9.3 Ferramentas de Produtividade e de Suporte

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

[Observação: Adicione ou exclua itens conforme o necessário.]

Nome da Marca da Ferramenta	Fornecedor ou Desenvolvida Internamente	Versão	
Gerenciamento de Teste	GitHub		

Controle de Defeitos			
Ferramenta ASQ para teste funcional	Java Eclipse - Junit		

9.4 Configurações do Ambiente de Teste

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

Descrição	Implementada na Configuração Física		
Configuração do usuário comum			
Mínima configuração suportada			
Motivada por funções visuais e motoras			
Sistema Operacional Internacional de Dois Bytes			
Instalação de Rede (não cliente)			