

GERADOR BURNDOWN

Plano de Testes

Desenvolvimento de Sistemas

Professor:

Maria Augusta

Equipe:

Daniel Parreira
Geanderson Esteves

Outubro / 2015

Histórico de Revisões

Data	Versão	Descrição	Autor
<31/out/15>	<1.0>	Release Inicial	Daniel e Geanderson

Índice

1. INTRODUÇÃO	4
1.1 OBJETIVOS	4
1.2 O TOURIST INFORMATION GUIDE	4
1.3 ESCOPO	4
1.4 IDENTIFICAÇÃO DE PROJETO.....	5
2. REQUISITOS A TESTAR.....	6
3. ESTRATÉGIA DE TESTE	7
3.1 TIPOS DE TESTE	7
3.2 FERRAMENTAS.....	7

1. Introdução

1.1 Objetivos

Esse documento do Plano de Testes do T.I.G. compõe-se dos seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os Requisitos a Testar recomendados (alto nível).
- 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.

1.2 O Tourist Information Guide

O GeradorBurndown é um sistema que tem por objetivo gerar gráficos de burndown de horas e pontos para um determinado projeto.

Para entender o seu funcionamento, é necessário um mínimo de conhecimento sobre Scrum. Alguns dos conceitos fundamentais:

Sprint

No Scrum, uma sprint é caracterizada por um período de tempo em que é realizada parte do desenvolvimento do projeto. Normalmente, esse período é fixo, possuindo sempre 10, 15, ou 20 dias, por exemplo.

Estória

Uma estória de usuário é geralmente um item do product backlog que precisa ser desenvolvida durante determinada sprint. Cada estória possui um tempo estimado e um numero de pontos que representa o seu tamanho, ou o esforço necessário para concluí-la.

Gráfico de Burndown

O gráfico de burndown é uma das mais importantes ferramentas presentes em um projeto de Scrum. Seu objetivo é informar quanto tempo da sprint já foi consumido, e quanto falta para termina-la. No exemplo abaixo, temos uma sprint de 7 dias, com um total de 140 horas a serem consumidas. A linha azul representa qual deve ser a “descida” ideal do gráfico para que o sprint seja concluído conforme o planejado. A vermelha representa o que está acontecendo na prática no dia a dia desse projeto. Nesse caso, como a linha real está abaixo da ideal, percebemos que estamos concluindo de maneira mais rápida, ou seja, o sprint será encerrado antes do tempo previsto.

1.3 Escopo

O gerador Burndown passará pelos testes unitário, de integração e de sistema. Os testes unitários e de integração vão lidar 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.

1.4 Identificação de Projeto

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de testes:

Documento	Criado ou Disponível	Recebido ou Revisado
Especificação de Requisitos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Plano de Projeto	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Análise	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Projeto	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Documento de Arquitetura	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Protótipo	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Manual do Usuário	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Lista de Riscos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não


























2. Requisitos a Testar

A lista abaixo identifica aqueles itens – use cases, requisitos funcionais e não funcionais – que foram identificados como alvos de teste. Essa lista representa o que será testado. Foi priorizado a cobertura de todas as classes do projeto.

- Teste Quantidade de Horas Consumidas
- Teste Consumo Diário Ideal
- Teste Total de Horas
- Teste Quantidade de Horas Restantes
- Teste Data
- Teste Tempo Gasto
- Teste Estória
- Teste do Main
- Teste dos Itens
- Teste Burndown Horas
- Teste Burndown Pontos
- Teste Dias
- Teste Data de Fim
- Teste Data de Início
- Teste Itens Histórico
- Teste Tempo Estimado

3. Estratégia de Teste

3.1 Tipos de Teste

Coverage 				
Backlog (1-Nov-2015 6:46:40 PM)				
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ Backlog	 99.8 %	1,970	4	1,974
▼ src	 99.8 %	1,970	4	1,974
▼ br.com.tcc.util	 99.0 %	414	4	418
▶ DataUtilTest.java	 94.5 %	69	4	73
▶ DataUtil.java	 100.0 %	51	0	51
▶ Main.java	 100.0 %	286	0	286
▶ MainTest.java	 100.0 %	8	0	8
▼ br.com.tcc.bo	 100.0 %	891	0	891
▶ BurndownBO.java	 100.0 %	311	0	311
▶ BurndownBOTest.java	 100.0 %	284	0	284
▶ LineChart.java	 100.0 %	184	0	184
▶ SprintBO.java	 100.0 %	52	0	52
▶ SprintBOTest.java	 100.0 %	60	0	60
▼ br.com.tcc.model	 100.0 %	665	0	665
▶ Burndown.java	 100.0 %	31	0	31
▶ BurndownTest.java	 100.0 %	61	0	61
▶ Estoria.java	 100.0 %	53	0	53
▶ EstoriaTest.java	 100.0 %	80	0	80
▶ Historico.java	 100.0 %	10	0	10
▶ HistoricoTest.java	 100.0 %	84	0	84
▶ ItemHistorico.java	 100.0 %	24	0	24
▶ ItemHistoricoTest.java	 100.0 %	51	0	51
▶ Sprint.java	 100.0 %	45	0	45
▶ SprintTest.java	 100.0 %	226	0	226

3.2 Ferramentas

As seguintes ferramentas serão empregadas para esse projeto:

	Ferramenta	Vendedor
Gerenciamento de Teste	EclEmma Java Code Coverage	Eclipse
Projeto de Teste	EclEmma Java Code Coverage	Eclipse
Gerenciamento de Projeto	Microsoft Project Microsoft Word	Microsoft