

# Gerador Burndown

## Sumário de Avaliação de Testes

**Versão <1.0>**

*[Observação: O template a seguir é fornecido para uso com o Rational Unified Process (RUP). O texto entre colchetes e exibido em itálico, em azul (estilo=InfoBlue), é fornecido para orientar o autor e deverá ser excluído antes da publicação do documento. Qualquer parágrafo inserido após esse estilo será definido automaticamente como normal (estilo=BodyText).]*

### Histórico da Revisão

Data	Versão	Descrição	Autor
05/11/2015	1.0	Primeiro documento de testes	Daniel e Geanderson

### Índice Analítico

#### 1. Introdução

##### 1.1 Finalidade

##### 1.2 Escopo

##### 1.3 Definições, Acrônimos e Abreviações

##### 1.4 Referências

##### 1.5 Visão Geral

#### 2. Sumário dos Resultados do Teste

#### 3. Cobertura de Teste Baseada em Requisitos

#### 4. Cobertura Baseada em Códigos

#### 5. Ações Sugeridas

#### 6. Diagramas

# Sumário de Avaliação de Testes

## 1. Introdução

*Este documento visa explicar como a avaliação de testes do sistema conhecido como Gerador Burndown será executado. O Gerador Burndown é um sistema que tem por objetivo gerar gráficos de burndown de horas e pontos para um determinado projeto.*

### 1.1 Finalidade

*A avaliação de testes será utilizada para verificar se o processo de execução do Gerador burndown atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado. Seu principal objetivo é revelar falhas em um produto, de modo que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento de software.*

### 1.2 Escopo

*A intenção é que a avaliação de testes cubra todas as classes implementadas no Gerador Burndown. Apenas os testes necessários para cobrir todas as classes do sistema serão incluídas.*

### 1.3 Definições, Acrônimos e Abreviações

*Gerador BurnDown - Gerador BD*

### 1.4 Referências

*Não consta*

### 1.5 Visão Geral

*O documento de avaliação de testes está dividido da seguinte forma: a Seção 2 descreve um sumário dos resultados do teste, a Seção 3 apresenta a cobertura dos testes baseada nos requisitos, a Seção 4 apresenta a cobertura baseada no código, e por fim, a Seção 5 apresenta as ações sugeridas.*

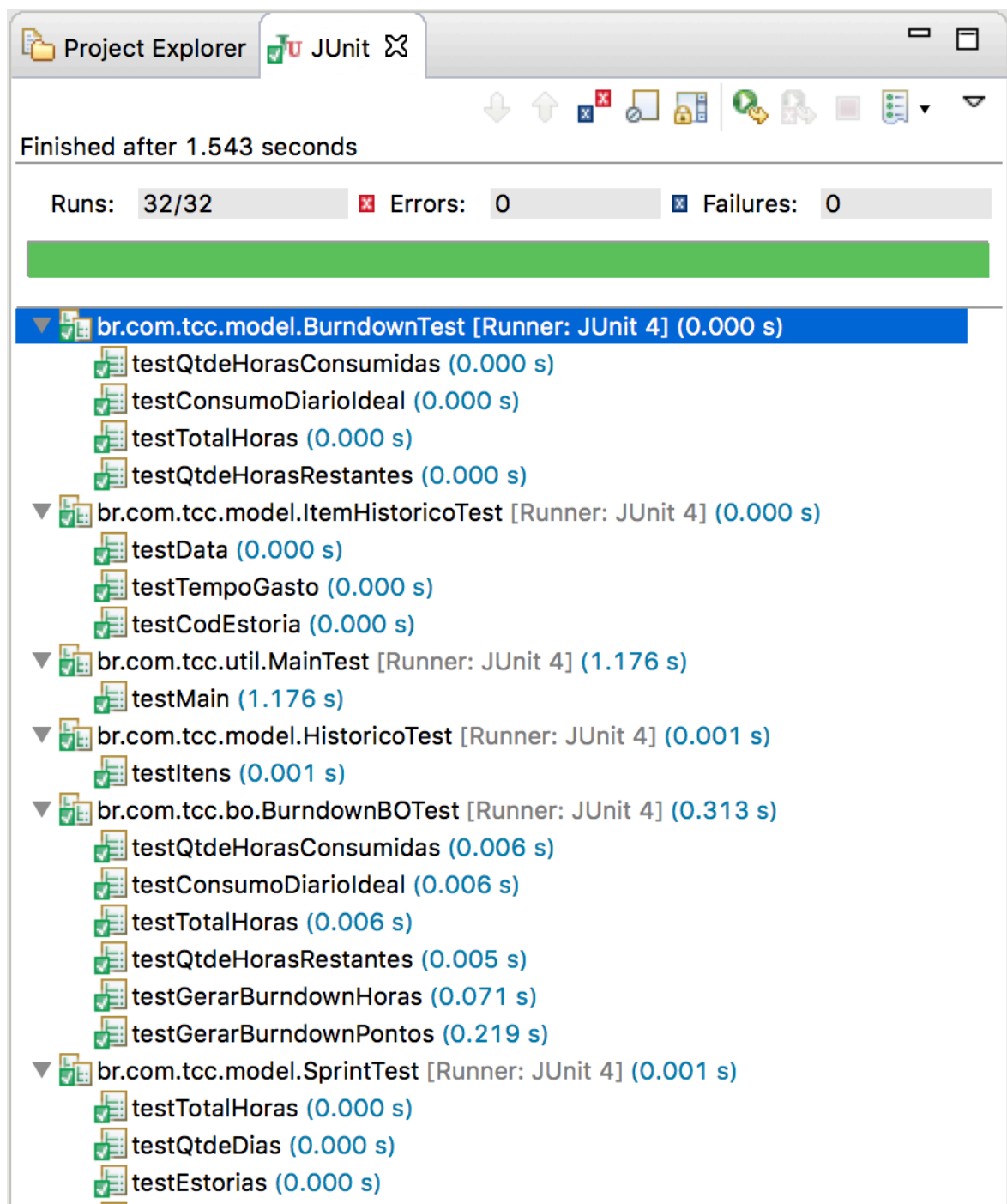
## 2. Sumário dos Resultados do Teste

*As principais classes que compõem o Gerador BD:*

- 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

*A figura abaixo mostra os 32 testes criados para cobrir o Gerador BD:*






### 3. Cobertura de Teste Baseada em Requisitos

*O Gerador Burndown não aceita entradas inseridas por usuários, por essa razão não foi possível criar testes de caixa preta específica para o programa. Entretanto, baseado na especificação de requisitos, foi possível notar que o programa atende completamente a especificação, uma vez que ele realiza exatamente a tarefa que lhe foi atribuída.*

### 4. Cobertura Baseada em Código

*A cobertura por testes baseada no código é mostrada abaixo:*

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

Pode-se observar que 99,8% do código do programa foi coberto por testes.

## 5. Ações Sugeridas

Não há ações relevantes em relação ao Gerador Burndown, apenas um bloco try/catch exemplificado na Seção 6 não foi coberto por nenhum teste.

## 6. Diagramas

A figura abaixo mostra o único trecho de todo o programa que não foi coberto por nenhum teste.

```
@Test
public void testConverteStringParaDate() throws ParseException {
    DateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
    assertEquals((java.util.Date)formatter.parse("20/05/1993"), DataUtil.converteStringPa
    assertEquals(null, DataUtil.converteStringParaDate(null));
    assertEquals(null, DataUtil.converteStringParaDate(""));
    try {
        DataUtil.converteStringParaDate("1121221");
    }
    catch (ParseException e) {
        assertEquals("Unparseable date: \"1121221\"", e.getMessage());
    }
}
```