



**SGB\_TESTE\_NormasTiposDeTesteEFerramentasDeTeste**

**Versão 1.0**

# Histórico de Revisões

Nome	Alterações	Data	Versão
Felipe Giroto	Criação do Documento	01/11/2012	0.1
Felipe Giroto	Criação dos tópicos 1, 2, 3 e 4	01/11/2012	0.2
Luis Eduardo Rocha	Especificação dos tópicos: 4.1, 4.2, 4.3, 4.4	02/11/2012	0.3
Luis Eduardo Rocha	Especificação dos tópicos: 3.2, 3.4	02/11/2012	0.4
Igor Vinicius dos Santos Silva	Especificação dos tópicos: 1, 2, 2.1, 3.3	02/11/2012	0.5
Igor Vinicius dos Santos Silva	Alterando documento conforme template da Gerencia de Configuração.	02/11/2012	1.0

# Sumário

[1 Introdução](#)

[2 Ferramentas](#)

[2.1 TestLink](#)

[2.1.1 Mantenedores da Ferramenta](#)

[2.1.2 Configurações](#)

[2.2 Selenium Web Test](#)

[2.1.1 Selenium IDE](#)

[2.1.2 Selenium CORE](#)

[2.1.3 Selenium RC](#)

[2.1.4 Selenium Grid](#)

[3 Normas](#)

[3.1 IEEE 829:2008](#)

[3.2 IEEE 1012:2004](#)

[3.3 IEEE 1044:1995](#)

[3.4 IEEE 1465:2004 \(Adaptação da ISO 12119: 1994\)](#)

[4 Tipos e Técnicas de Teste](#)

[4.1 Teste de Requisito](#)

[4.2 Teste Funcional](#)

[4.3 Teste de Regressão](#)

[4.4 Teste de Aceitação](#)

## 1 Introdução

Este documento tem como base definir as diretrizes para a área de teste da fábrica de software para o projeto SGB. Nele contém as ferramentas que serão utilizadas para fazer a cobertura de testes, as normas utilizadas como base para geração de artefatos de teste e também os tipos e técnicas de teste utilizados para os testes ao longo do processo de verificação e validação do produto.

## 2 Ferramentas

Nesta seção estão descritas as ferramentas a serem utilizadas na cobertura dos testes.

### 2.1 TestLink

TestLink é uma ferramenta case de gerenciamento de teste. Essa ferramenta pertence e é mantida pela comunidade de teste de software, em todo o mundo. TestLink é utilizado para a criação e elaboração de Casos de teste e cenários de teste. Nessa ferramenta também é possível vincular os casos de teste de forma hierárquica e a um mesmo grupo de teste. TestLink não é uma ferramenta que executa caso de teste, essa ferramenta é utilizada para documentar os testes que serão executados por outras ferramentas ou manualmente.

#### 2.1.1 Mantenedores da Ferramenta

- Os líderes de equipe
  - Francisco Mancardi - núcleo de desenvolvimento, de designer e revisor
  - Julian Krien - desenvolvedor, coordenação de projetos
- Os membros da equipe
  - Andreas Morsing - desenvolvimento do núcleo, re-factoring

- Martin Havlat - gestão, desenvolvimento, lançamentos, documentação, web-master
- Toshiyuki Kawanishi - Japão coordenador da equipe, localização Japão, desenvolvedor
- Masami Ichikawa - testes, automação de testes
- Amit Khullar - desenvolvedor
- Erik Elof - desenvolvedor
- Andreas Simon - desenvolvedor
- Asiel Brumfield - equipe rastreador administrador, desenvolvedor
- Kevin Levy - desenvolvedor (Reportagem; Requisitos)
- Chad Rosen - Originador do projeto

### 2.1.2 Configurações

Para instalar a ferramenta TestLink necessita-se que já tenha instalado os seguintes softwares:

- Web Service Apache2;
- PHP 5 ou superior;
- MySql 5 ou superior.

## 2.2 Selenium Web Test

O Selenium Web Test é uma ferramenta de automação de testes funcionais para interfaces Web. Desenvolvida e mantida pela OpenQA como software livre utilizando a licença 2.0 da Apache.

Sua principal vantagem é a utilização do próprio navegador web para realização dos testes. É possível realizar o acesso a sistemas através de navegadores como o Firefox, Internet Explorer ou Safari.

O Selenium é uma ferramenta bem completa, pois contempla tanto a parte de geração dos casos de testes quanto a de execução. Atualmente é mais considerado como uma plataforma. Divide-se em Selenium IDE, Selenium RC, Selenium Core e o Selenium Grid.

### 2.1.1 Selenium IDE

O Selenium IDE é um plugin para o Firefox que fornece um ambiente integrado para desenvolvimento de scripts de testes permitindo gravar, editar, executar e depurar os testes. Os scripts podem ser gerados em um padrão próprio do Selenium ou escolher uma das linguagens de programação disponíveis, como Java, C#, PHP, Perl, Ruby, etc. A partir da captura automática das ações do usuário na tela do sistema é possível montar os procedimentos de testes que serão utilizados na execução dos testes .

### 2.1.2 Selenium CORE

O Selenium Core é um módulo que deve ser adicionado ao servidor web para a execução dos scripts na linguagem nativa do Selenium.

### 2.1.3 Selenium RC

O RC (Remote Control) permite a integração com ferramentas de testes unitários (como JUnit) aumentando muito a flexibilidade dos scripts de testes.

#### 2.1.4 Selenium Grid

O Grid é um módulo do Selenium que permite a execução distribuída e em vários ambientes dos scripts de testes.

### 3 Normas

Nesta seção estão descritas as normas a serem utilizadas na cobertura dos testes. Importante ressaltar que as normas serão adaptadas ao contexto da fábrica podendo não ser executadas 100% a risca como estão descritas.

#### 3.1 IEEE 829:2008

O IEEE 829 também conhecido como o Padrão 829 para Documentação de Teste de Software, é um padrão IEEE que especifica a forma de uso de um conjunto de documentos em oito estágios definidos de teste de software, cada estágio potencialmente produzindo seu próprio tipo de documento.

#### 3.2 IEEE 1012:2004

Padrão de Verificação e Validação de software.

Determina se os produtos desenvolvidos dado uma atividade estão conformes dos requisitos daquela atividade e se o software satisfaz o planejamento e as necessidades do sistema. A 1012 é especificado para diferentes níveis de integridade. V&V ajuda na construção da qualidade de um software durante seu ciclo de desenvolvimento, demonstrando se os requisitos do software e do sistema estão corretos, completos, precisos, consistente e testável. Essa norma inclui a análise, avaliação, revisão e inspeção do software e produtos gerados. Essa norma é seguida durante o desenvolvimento do software e não na conclusão.

#### 3.3 IEEE 1044:1995

Padrão de Classificação de Anomalias

A IEEE Std 1044-1993 tem como propósito esquematizar a classificação das anomalias de software. O uso dessa norma no processo de Teste de Software para o projeto SGB, será como guia para elaboração e classificação das ocorrências (erros), defeitos e falhas identificadas na execução das atividades de Teste. Para que possa ser gerenciado o nível da qualidade dos requisitos e dos testes.

#### 3.4 IEEE 1465:2004 (Adaptação da ISO 12119: 1994)

Padrão de Requisitos de Qualidade e de Teste.

Requisitos de qualidade para os pacotes de software e instruções de como testar os pacotes de software em prol dos requisitos estabelecidos. Esse requisitos serão aplicados nos pacotes do software e serão oferecidos e entregados, excluído durante o processo de prorrogação de especificação.

### 4 Tipos e Técnicas de Teste

Nesta seção estão descritas os tipos de teste a serem utilizados na cobertura dos artefatos.

#### **4.1 Teste de Requisito**

Para cada requisito funcional deve ser possível definir um ou mais testes a serem realizados no sistema final para ser possível verificar se o sistema cumpre o requisito na integra. Se este teste não for possível ser definido isso vai significar que o requisito necessita de ser clarificado pois muito provavelmente irá criar problemas no desenvolvimento do produto.

Na definição de cada teste deverá ser tomado em conta os seguintes pontos:

1. identificador do requisito
2. requisitos relacionados
3. descrição do teste
4. problemas na realização do teste
5. comentário e recomendações

#### **4.2 Teste Funcional**

Teste funcional é aquele onde o objetivo é verificar se uma dada implementação está de acordo com a sua respectiva especificação. Também é conhecido como teste “Caixa Preta”. Um tipo de teste funcional é o chamado teste formal. Nele, tanto as especificações como os procedimentos de geração de casos de teste são formalizados.

A identificação dos casos de teste devem se basear única e exclusivamente nas especificações dos componentes de implementação.

Técnicas que serão utilizadas serão:

- Análise de Valores Limites.
- Partição por Equivalência.
- Teste Estado-Transição.
- Testes derivados da Especificação.

#### **4.3 Teste de Regressão**

O teste de regressão é uma técnica do teste de software que consiste na aplicação de testes à versão mais recente do software, para garantir que não surgiram novos defeitos em componentes já testados.

Utilizado durante o desenvolvimento devido a cada iteração, e em uma nova instância de um componente que estará sendo reutilizável.

Se, ao juntar o novo componente ou suas alterações com os componentes restantes do sistema surgirem novos defeitos em componentes inalterados, então considera-se que o sistema regrediu.

#### **4.4 Teste de Aceitação**

Tem por função verificar o sistema em relação aos seus requisitos originais, e às necessidades atuais do usuário.

O teste de aceitação é o teste feito antes da fase de implantação do software alvo. de acordo com a 829 será criado 3 estratégias de teste.

#### Teste de Aceitação formal:

- Executado pelo Usuário final ou por grupo de teste independente
- Funções e recursos devem se conhecidos
- Pode ser automatizado
- Deve ser monitorado e medido

#### Teste Alfa:

No teste Alfa, os procedimentos para executar o teste não são definidos com tanto rigor como no teste de aceitação formal. As funções e as tarefas de negócios a serem exploradas são identificadas e documentadas, mas não há casos de teste específicos para seguir. O testador individual determina o que fazer. Essa abordagem de teste de aceitação não é tão controlada como o teste formal e é mais subjetiva do que o tipo formal.

#### Teste Beta:

O teste beta é o menos controlado das três estratégias de teste de aceitação. Nesse tipo de teste, a quantidade de detalhes, os dados e a abordagem adotada são de inteira responsabilidade do testador individual. Cada testador é responsável por criar o próprio ambiente, selecionar os dados correspondentes e determinar as funções, os recursos ou as tarefas a serem exploradas. Cada um deles é responsável por identificar os próprios critérios que o levarão a aceitar ou rejeitar o sistema no seu estado atual.

O teste beta é implementado por usuários finais, geralmente com pouco ou nenhum gerenciamento por parte da organização de desenvolvimento (ou outra que não seja do usuário final).

## 4.4 Teste de Unidade

Teste de unidade objetiva testar a menor unidade funcional de um código, no caso de projetos orientados a objetos a menor unidade é o método de uma classe. Para esse projeto foram eleitos os dois principais componentes do sistema: Model e o Base. São eles que modelam as regras do negócio. O critério de aceitação é que possua cobertura de testes no mínimo de 90% sobre o Model e o Base.