# Ferramentas de suporte ao Teste de Software

## Ao longo deste artigo serão apresentadas algumas ferramentas de diferentes características e classificações voltadas para fornecer um amplo suporte ao teste de software.

Ferramentas de suporte ao Teste de Software

Ao longo dos anos, à medida que o teste vem ganhando mais espaço no mercado, diversas ferramentas têm surgido visando à melhoria no gerenciamento e na execução das atividades voltadas para o Teste de Software.

A utilização de ferramentas é um importante aliado para avaliar se o software está exposto a vulnerabilidades. Independente da função, o objetivo gira em torno da simplificação das atividades, controle, e, principalmente, garantia da qualidade em cada uma das etapas de um Processo de Teste. Outro fator relevante é que elas permitem, sobretudo, minimizar os riscos, inconformidades e uma série de falhas que podem ser detectadas durante a execução dos testes.

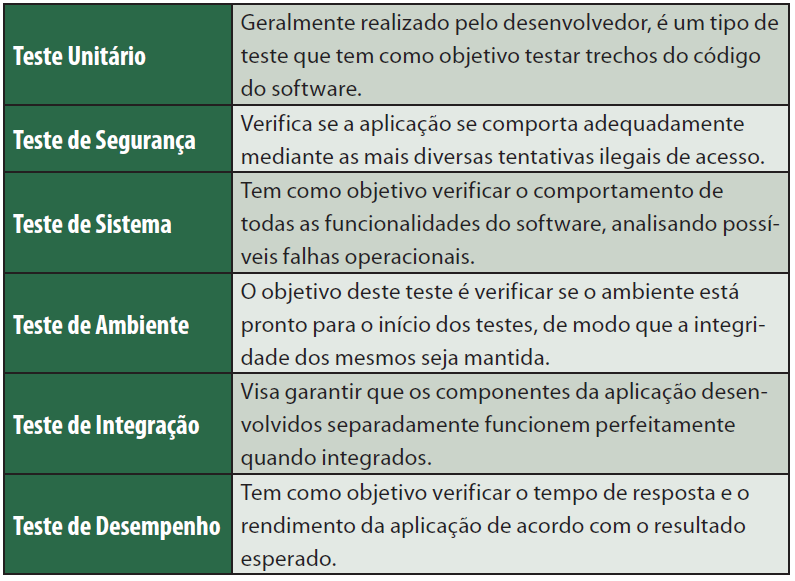
Além de simplificar as atividades, a adoção dessas ferramentas pode aumentar consideravelmente o número de defeitos encontrados, e, consequentemente, agilizar os testes, aprimorando a qualidade das aplicações. Elas também podem levar a melhorias na confiabilidade do software, tornando os testes mais eficazes e produtivos.

Com base nesse contexto, ao longo deste artigo serão apresentadas algumas ferramentas de diferentes características e classificações voltadas para fornecer um amplo suporte ao teste de software.

**Classificação das ferramentas**

As ferramentas são classificadas de acordo com as atividades de teste que elas suportam. Seja para o gerenciamento ou para execução de um teste, elas são categorizadas de acordo com os tipos de teste que executam.

Os tipos de teste, alguns deles demonstrados na **Tabela 1**, possuem diferentes funções, e cada um deles visa avaliar uma determinada característica do software. Um fator interessante é que há ferramentas de suporte que contemplam praticamente todos os tipos de teste existentes.



[**abrir imagem em nova janela**](http://arquivo.devmedia.com.br/REVISTAS/easyjava/imagens/31/2/tab1.png)

**Tabela 1.** Tipos de Teste de Software.

Hoje em dia, algumas ferramentas suportam várias atividades, como gestão dos testes, automação e controle dos defeitos, enquanto outras suportam apenas uma. Mesmo assim, apresentá-las intitulando uma ou outra como “a melhor” não é possível. Esse título vai depender de uma série de fatores como, por exemplo, investimento disponível, aplicabilidade, tipo de software a ser testado, tamanho da equipe e expectativa de retorno. Em função disso, é interessante analisar as opções disponíveis para que seja possível definir qual se enquadra melhor às suas expectativas.

Neste contexto, ao longo desse artigo serão apresentadas diferentes classificações de ferramentas de suporte ao teste, todas com um propósito único: garantir a qualidade do software.

**Ferramentas de gerenciamento de testes**

A gestão de testes possui um papel fundamental dentro de um Processo de Teste, afinal, é responsável pelo planejamento e controle de todas as atividades.

As ferramentas de gerenciamento podem ser utilizadas em todas as etapas do ciclo de vida do teste e têm como grande vantagem a centralização de todas as informações relacionadas à evolução dos testes realizados no software.

Geralmente elas não têm apenas uma função, podendo apresentar distintas aplicabilidades, como priorização dos testes, cronograma de atividades, registro dos resultados, rastreamento do progresso e gerenciamento dos incidentes dos testes.

O objetivo dessas ferramentas está voltado para o controle dos testes, sejam eles manuais ou automáticos. Elas permitem que informações sejam coletadas e acompanhadas, possibilitando uma visão mais ampla para a tomada de decisão, fornecendo artefatos de acordo com os interesses e expectativas.

Essas ferramentas propiciam uma série de vantagens visando à garantia da qualidade do software. Dentre elas, destacamos:

· Permitem um total controle das atividades do Processo de Teste;

· Facilitam a organização, documentação e padronização dos procedimentos executados em cada etapa do ciclo de vida dos testes;

· Auxiliam a tomar decisões mais precisas com base nos resultados obtidos a partir dos testes;

· Possibilitam que os testes sejam rastreados, identificando o status atual de cada uma das atividades de teste.

A seguir serão apresentadas algumas ferramentas voltadas para o gerenciamento dos testes.

**TestLink**

Ferramenta open source que utiliza as plataformas PHP e MYSQL. Permite cadastrar planos de teste e casos de teste, além de controlar a evolução da execução dos testes através da criação de *baselines*, que geram uma base histórica de todos os testes já submetidos no software.

As características da ferramenta incluem:

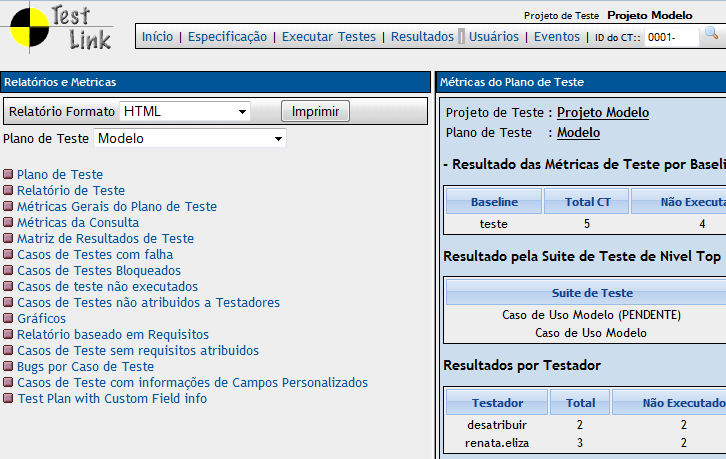
· Gerenciamento de múltiplos projetos;

· Facilidade para criar e gerenciar casos de teste, bem como organizá-los em planos de teste;

· Priorização das tarefas atribuídas;

· Integração com ferramentas de gestão de defeitos, como Bugzilla, Jira e Mantis;

· Fornece diferentes relatórios e gráficos que possibilitam o acompanhamento detalhado dos testes executados, conforme demonstrado na **Figura 1.**



[**abrir imagem em nova janela**](http://arquivo.devmedia.com.br/REVISTAS/easyjava/imagens/31/2/image001.png)

**Figura 1**. Relatórios do Testlink.

**RTH**

Ferramenta web que simplifica o gerenciamento de requisitos, casos de teste, controle de defeitos, planos de teste e análise dos resultados do teste. Possui usabilidade bem intuitiva, permitindo que mesmo usuários inexperientes aprendam rapidamente a utilizá-la.

As características desta ferramenta incluem:

· Gerenciamento de múltiplos projetos;

· Gerenciamento de requisitos;

· Gerenciamento de defeitos;

· Monitoramento do progresso dos testes;

· Integração com o Bugzilla.

**T-Plan Professional**

Ferramenta projetada para facilitar o processo de gerência do teste durante todo o ciclo de vida, do planejamento à execução, desde o teste unitário até o teste de aceitação do usuário.

T-Plan Professional possibilita analisar as métricas coletadas durante a execução dos testes, acompanhar o progresso, definir a prioridade e a profundidade de cobertura dos testes.

As características desta ferramenta incluem:

· Alto nível de comunicação entre os integrantes do projeto;

· Importação e exportação de dados do Word e Excel;

· Fornece e possibilita a customização de inúmeros relatórios;

· Arquitetura aberta, permitindo integrações e extensões;

· Possibilita a reutilização de todos os dados dentro da própria ferramenta.

**QADirector**

É uma ferramenta de plataforma aberta que permite a integração com ferramentas de terceiros. Tais ferramentas viabilizam o rastreamento de defeitos e testes de carga, além da automação dos testes funcionais.

O QADirector viabiliza desde o planejamento dos testes até a análise dos resultados, possibilitando significativos ganhos de produtividade.

As características desta ferramenta incluem:

· Interface simples;

· Capacidade de captura de tela, visando melhor análise dos resultados dos testes;

· Importação de dados do Word e Excel;

· Controle do tempo de execução dos scripts, visando identificar o tempo gasto em cada um dos testes.

**Ferramentas de Gestão de Defeitos**

Garantir qualidade ao software significa também minimizar os riscos e deixar o produto final com o menor número de erros possível. Para isso, a gestão de defeitos torna-se essencial, ao possibilitar tanto o registro quanto o acompanhamento dos *bugs.*

O objetivo de uma ferramenta de gestão de defeitos é registrar os *bugs* encontrados, bem como permitir rastrear as mudanças ocorridas no projeto no decorrer da execução dos testes.

Uma série de vantagens pode ser atribuída ao uso dessas ferramentas. A partir da sua utilização é possível a visualização completa de todo o escopo do projeto, permitindo que os gestores saibam o que está sendo feito e como está o andamento do processo de teste. Além destas, outras vantagens podem ser citadas, tais como:

· Maior visibilidade dos problemas encontrados em cada etapa do processo de desenvolvimento do software;

· Prova formal do problema encontrado através de um relatório detalhado;

· Melhor acompanhamento dos problemas e das implementações desenvolvidas;

· Melhor organização de documentos e arquivos eletrônicos.

A seguir serão apresentadas algumas ferramentas voltadas para a gestão de defeitos.

**Mantis**

Acessado através de um browser, o Mantis é um popular sistema de gerenciamento de defeitos que possui uma baixa curva de aprendizado, é fácil de instalar e também de customizar.

Além disso, é uma ferramenta open source, desenvolvida em PHP e que fornece suporte a bancos de dados como Postgres e MySQL.

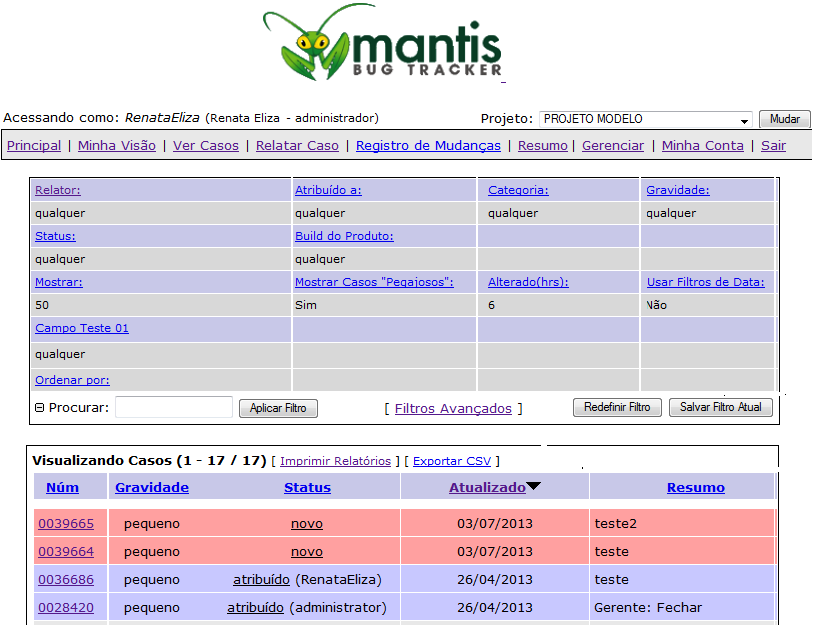
Algumas características relevantes do Mantis são:

· Suporta o gerenciamento de múltiplos projetos;

· Página principal personalizável para cada usuário;

· Permite exportação de dados para planilha.

A **Figura 2** demonstra a tela principal do Mantis, onde todos os bugs registrados em determinado projeto podem ser filtrados e visualizados.



[**abrir imagem em nova janela**](http://arquivo.devmedia.com.br/REVISTAS/easyjava/imagens/31/2/image002.png)

**Figura 2.** Visualização dos casos registrados no Mantis.

**Bugzilla**

O Bugzilla é uma das ferramentas de acompanhamento de bugs mais populares entre as soluções open source disponíveis. Inicialmente, foi criada para apoiar o desenvolvimento do browser Mozilla.

É uma opção que facilita o monitoramento das atividades dos usuários, e hoje em dia tem sido empregada por um grande número de projetos de desenvolvimento de software.

Algumas características relevantes do Bugzilla são:

· Possibilita notificações customizáveis por e-mail;

· Permite adicionar informações nos *bugs* registrados através de e-mail;

· Facilita o monitoramento das atividades dos usuários.

**Scarab**

O Scarab é um sistema de gestão de defeitos multiplataforma de fácil instalação, sendo totalmente personalizável através de um conjunto de páginas administrativas.

Essa ferramenta possibilita um eficiente controle de permissões de acesso aos projetos gerenciados por ela, além de notificar os usuários por e-mail a cada tarefa recebida.

Algumas características relevantes do Scarab são:

· Possibilita a criação de vários projetos;

· Facilidade de manutenção e instalação;

· Possibilita exportar e importar o projeto para outras ferramentas de gerenciamento de defeitos a partir de uma interface XML.

**Trac**

É uma aplicação com estrutura baseada em wiki que possui integração nativa com o SVN (Subversion), oferecendo aos usuários uma excelente interface para o repositório. Além disso, possui capacidade de integração com outros sistemas de controle de versão.

O Trac possibilita a rastreabilidade das alterações realizadas no software através dos links criados por meio da formatação Wiki.

Algumas características relevantes do Trac são:

· Sistema de permissões de acesso simplificado;

· Facilidade de customização;

· Suporta diferentes Bancos de Dados.

**Ferramentas de Teste de Performance, Stress e Carga**

Juntos ou não, os testes de Performance, Carga e Stress visam garantir um desempenho ideal para a aplicação, mantendo a estabilidade, maximizando as transações simultâneas e determinando o comportamento do software sob uma determinada carga de trabalho.

As ferramentas voltadas para esses testes, desde as pagas até as gratuitas, têm como objetivo projetar testes para que seja possível simular diferentes situações no comportamento do software.

Listamos a seguir alguns dos benefícios gerados com a utilização dessas ferramentas:

· Permite determinar quais transações poderão impactar na utilização da aplicação, fornecendo estatísticas tangíveis para análise de toda a equipe.

· Através de métricas precisas, permite uma otimização dos recursos tanto de software quanto de hardware;

· Possibilita a redução dos custos, através da identificação antecipada dos defeitos mais críticos da aplicação;

· Minimiza os riscos relacionados aos requisitos de desempenho, como por exemplo, tempo de inatividade.

A seguir serão apresentadas algumas ferramentas voltadas para os testes de Performance, Carga e Stress.

**WAPT**

É uma ferramenta de testes de Carga e Stress fácil de usar e de baixo custo. Voltada para aplicações web, possibilita a realização de testes em diversos cenários, exibindo o resultado através de diferentes relatórios e gráficos.

Algumas características relevantes são:

· Possibilita a realização de testes remotos, distribuídos e baseados na nuvem;

· Permite o monitoramento do desempenho de bancos de dados;

· Possui suporte ao protocolo HTTPS e outros esquemas de autenticação.

**LoadRunner**

Ferramenta de teste de Carga líder e padrão de mercado. Suporta quase todos os protocolos de rede. Funciona da seguinte maneira: primeiro, captura os dados da aplicação e cria os scripts de teste; em seguida, organiza, executa, gerencia e monitora o teste de carga; por fim, apresenta relatórios que auxiliam na análise e comparação dos resultados.

Algumas características relevantes são:

· Suporta diferentes plataformas, por exemplo, Java e .NET;

· Possui relatórios em formato de tabelas e gráficos que são facilmente compreensíveis;

· Ampla documentação, excelentes tutoriais e suporte ativo da HP.

**JMeter**

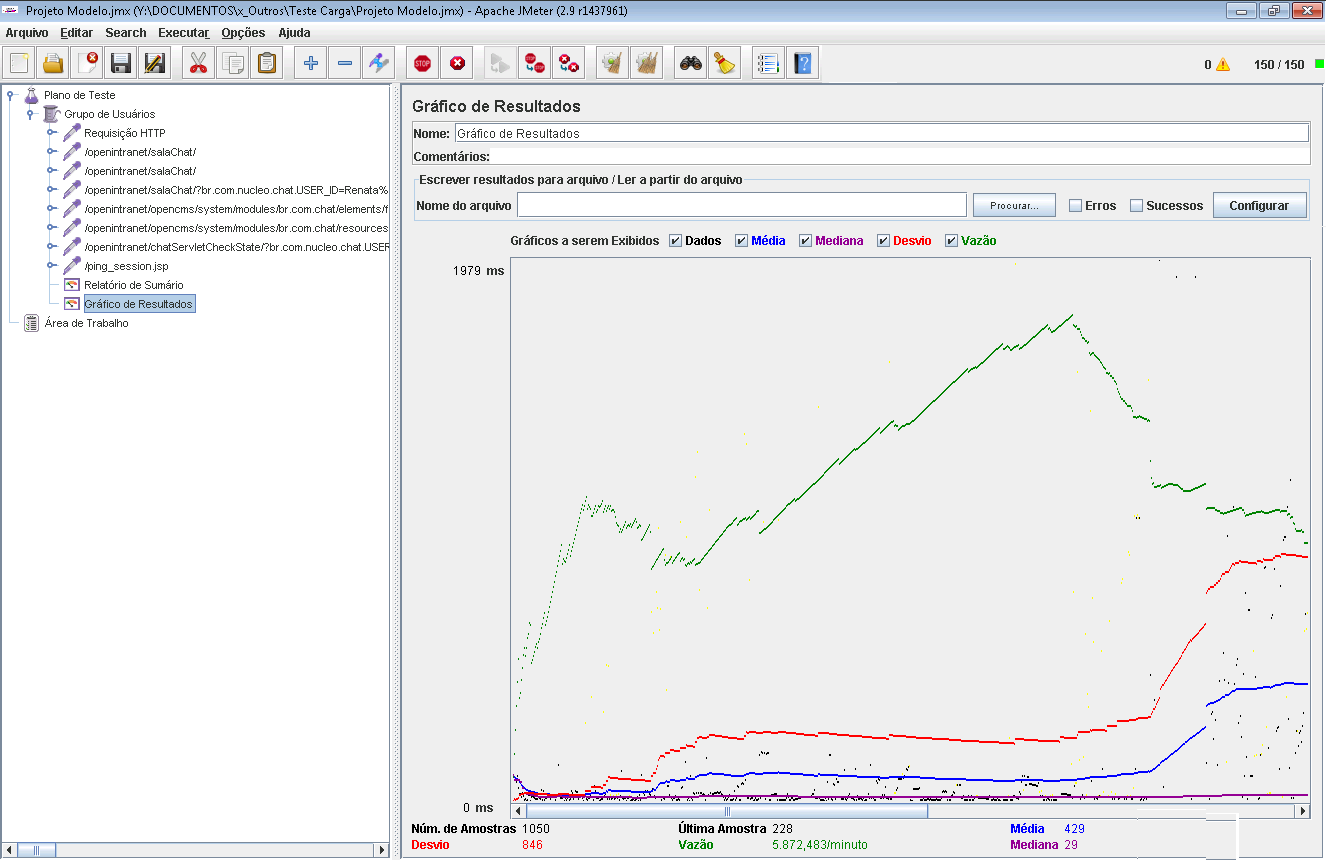
Esta ferramenta é parte do projeto Jakarta da Apache Software Foundation, sendo voltada para a execução de testes de Carga e Stress. Para realização dos testes, o JMeter disponibiliza diversos tipos de requisições, controladores lógicos e listeners. Estes são usados para gerar os resultados dos testes, que são exibidos através de gráficos e tabelas.

Algumas características relevantes são:

· Pode ser usado para testar recursos estáticos e dinâmicos, como servlets, objetos Java e scripts Perl;

· Permite analisar e reexecutar os testes mesmo quando a aplicação estiver off-line;

· Possui uma interface gráfica que propicia uma baixa curva de aprendizado, como pode ser verificado na **Figura 3**.



[**abrir imagem em nova janela**](http://arquivo.devmedia.com.br/REVISTAS/easyjava/imagens/31/2/image003.png)

**Figura 3.** Gráfico de resultados do JMeter.

**LoadUI**

Solução open source voltada para a realização de Teste de Carga. Com um simples arrastar e soltar permite criar, configurar e redistribuir os testes de carga de forma interativa e em tempo real. O LoadUI é composto por vários módulos independentes que podem se integrar para simular diferentes situações, obtendo boletins estatísticos para ajudar a tomar decisões importantes na configuração dos servidores de aplicação e web.

Algumas características relevantes são:

· Fornece análises abrangentes a partir de gráficos, e exibe os dados obtidos em tempo real em relatórios gerados automaticamente;

· Possibilita a exportação de relatórios em diversos formatos, como pdf, doc, xls, HTML e XML;

· Permite um monitoramento interativo do servidor da aplicação em tempo real.

**Ferramentas de Automação de Testes Funcionais**

Automatizar os testes é transformar uma rotina de testes manuais em scripts executados por softwares. A automação pode ser realizada em uma aplicação inteira ou apenas em uma parte dela. Em qualquer das opções, sem vigilância ou com intervenção humana esporádica.

O ideal é que essas ferramentas sejam utilizadas quando a atividade envolver tarefas realizadas com maior frequência. É aconselhável também que a automação seja efetivada quando os testes manuais estabelecidos estiverem maduros, para evitar um possível retrabalho em função de mudanças constantes no software.

As ferramentas de automação mais comuns são as que utilizam a técnica “*Record & Play*”. Essa técnica consiste em gravar as ações executadas pelo usuário enquanto este interage com a interface gráfica do software, convertendo-as em scripts de teste para posterior execução.

As vantagens e desvantagens das ferramentas de automação de testes funcionais são bem contraditórias. Mas no geral, alguns benefícios consideráveis podem ser obtidos a partir da utilização dessas ferramentas, conforme listamos a seguir:

· O retorno rápido dos resultados possibilita que as falhas sejam encontradas mais rapidamente, agilizando o processo de correção dos erros;

· Reduz significativamente o tempo necessário para executar um ciclo de testes, se comparado com a execução humana. A automação pode ser capaz de executar 10, 100, ou até mesmo, 1.000 vezes mais rápido;

· A maioria das ferramentas possibilita a reutilização dos scripts de teste em diferentes versões do software. Em algumas delas, até mesmo quando há mudanças na interface da aplicação;

· Possibilita programar testes sofisticados que trazem informações relevantes que podem ser complicadas de reproduzir manualmente.

A seguir serão apresentadas algumas ferramentas voltadas para a automação dos testes funcionais.

**Selenium**

Ferramenta de automação de testes funcionais para aplicações web, onde os testes são realizados a partir de um *plugin* instalado no Firefox. Este *plugin* permite gravar a navegação do usuário, e depois reproduzi-la de forma automatizada.

É possível utilizar o Selenium por meio de sua interface de gravação de testes, o Selenium IDE, ou através da sua API, que possibilita que os scripts sejam escritos em diferentes linguagens.

Hoje em dia é uma das ferramentas mais conhecidas e utilizadas, sendo considerada uma boa opção por ser gratuita e também por permitir o uso de diferentes linguagens de programação.

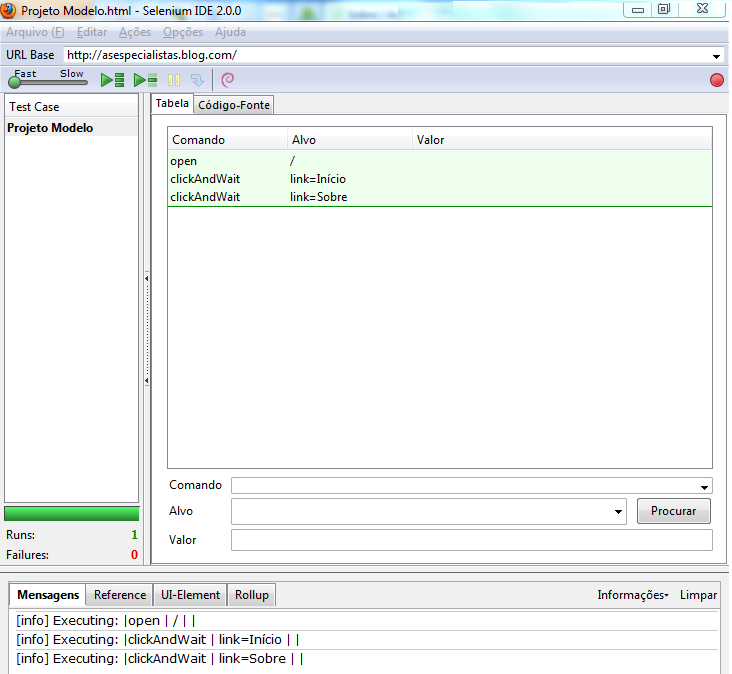
Algumas características relevantes desta ferramenta são:

· Instalação local e simples;

· Os scripts podem ser gerados em um padrão próprio do Selenium ou em outras linguagens de programação, como Java, C#, PHP, Perl e Ruby;

· Não é preciso saber programar.

A **Figura 4** representa um script simples de acesso a uma determinada página.



[**abrir imagem em nova janela**](http://arquivo.devmedia.com.br/REVISTAS/easyjava/imagens/31/2/image004.png)

**Figura 4.** Exemplo de script no Selenium IDE.

**SilkTest**

Essa ferramenta também utiliza a técnica “*Record and Play*” em testes baseados na interface gráfica das aplicações. É uma IDE que viabiliza a gravação, execução, debug e edição dos scripts de teste.

Esses scripts podem ser gerados em diferentes linguagens de programação através de um *plugin* para o Eclipse chamado Silk4j.

Algumas características relevantes desta ferramenta são:

· Os scripts podem ser utilizados para testes em diferentes navegadores;

· As sequências de teste podem ser facilmente modificadas e personalizadas;

· Fornece poderosos recursos de depuração que permitem acompanhar exatamente o que está acontecendo durante a execução do teste;

**Watir**

É uma ferramenta open source, composta por um conjunto de bibliotecas em Ruby, com suporte para aplicações web, independente da linguagem na qual ela foi desenvolvida.

O Watir é leve, simples, flexível e fácil de usar. Seu nome é um acrônimo para Web Application Testing In Ruby.

Algumas características relevantes desta ferramenta são:

· Suporta vários navegadores em diferentes plataformas;

· Utiliza o Ruby, uma linguagem de script moderna e completa que possui uma baixa curva de aprendizado;

· Suporta qualquer aplicação web, não importando a linguagem na qual ela foi desenvolvida.

**Badboy**

É uma ferramenta do tipo “*Record and Play*”, desenvolvida em C++ e voltada para o Windows. Ela possui um *plugin* para o Firefox, mas o navegador padrão para utilização é o Internet Explorer.

O Badboy possibilita o monitoramento das atividades do navegador gravando os eventos mais interessantes. Além disso, é compatível com o JMeter, possui uma ampla documentação e um número cada vez maior de usuários.

Algumas características relevantes desta ferramenta são:

· É uma ferramenta que executa tanto o teste funcional quanto o teste de carga;

· Possui relatórios e gráficos detalhados;

· Baixa curva de aprendizado.

**Ferramentas de Teste de Segurança**

O Teste de Segurança tem como objetivo garantir que o funcionamento da aplicação esteja exatamente como especificado. Verifica também se o software se comporta adequadamente mediante as mais diversas tentativas ilegais de acesso, que visam possíveis vulnerabilidades. Para isso, testa se todos os mecanismos de proteção embutidos na aplicação de fato a protegerão de acessos indevidos.

Hoje em dia, a preocupação com a segurança das aplicações tem crescido, e diversas ferramentas têm sido desenvolvidas na tentativa de evitar invasões e ataques nos softwares. Utilizá-las pode viabilizar a obtenção de uma série de vantagens para a aplicação, por exemplo:

· Propicia a validação dos requisitos do software;

· Permite analisar o risco para determinar todos os fatores que contribuem para sua ocorrência;

· Visa garantir que independente da situação, a informação não será destruída ou corrompida, permitindo que o software se mantenha com um desempenho próximo ao que foi definido pelo cliente;

· Tem como objetivo garantir que o software estará disponível sempre que for necessário, havendo ou não situações anormais de utilização.

Voltadas para aplicações web, muitas são as ferramentas que podem auxiliar na execução dos Testes de Segurança. Neste contexto, a seguir serão apresentadas algumas destas ferramentas.

**Websecurify**

Ferramenta open source que identifica automaticamente as vulnerabilidades das aplicações web. Possibilita a criação de relatórios simples que podem ser exportados em vários formatos.

**Wapiti**

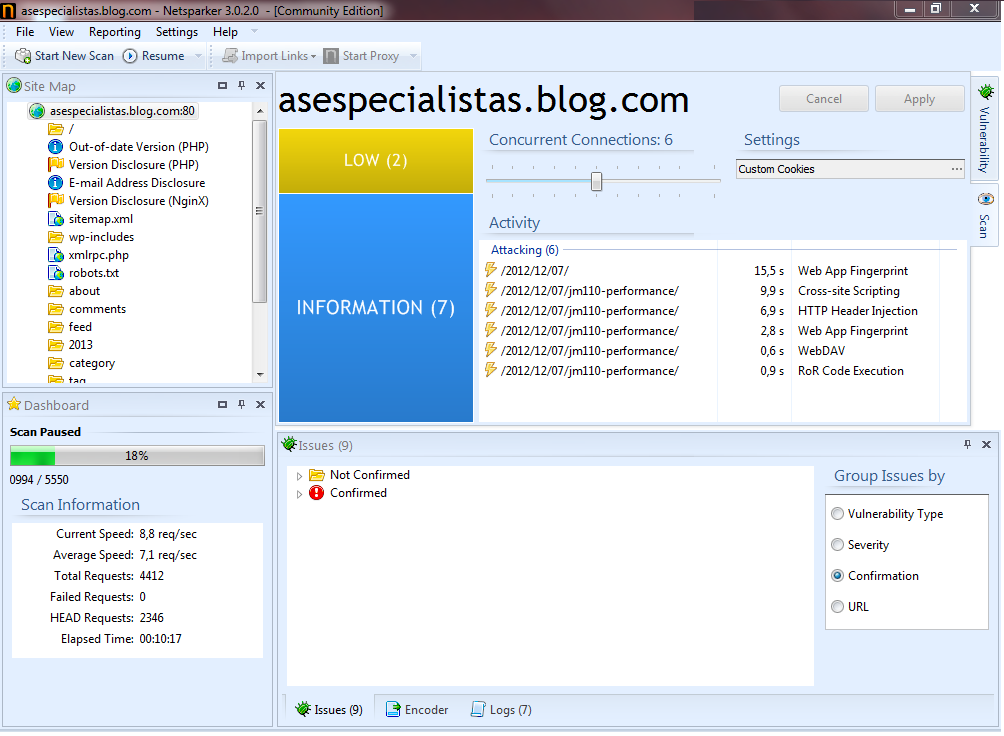
Ferramenta open source que verifica as páginas de sites e aplicativos web procurando por scripts e formulários vulneráveis. É feita em Python e consegue detectar diferentes formas de ataque à segurança do software.

**N-Stalker WAS**

É uma sofisticada ferramenta brasileira para avaliação da segurança do software. Tem a capacidade de verificar até 100 páginas de uma só vez para realização de vários testes de segurança, incluindo XSS.

**Netsparker Community Edition**

É um aplicativo que pode detectar pontos vulneráveis em SQL Injection e XSS. Conforme demonstrado na **Figura 5**, depois de realizar uma varredura em um site, a ferramenta apresenta as soluções para os possíveis problemas encontrados.



[**abrir imagem em nova janela**](http://arquivo.devmedia.com.br/REVISTAS/easyjava/imagens/31/2/image005.png)

**Figura 5.** Exemplo de varredura em um site.

**Conclusão**

As ferramentas de suporte ao teste, através das inúmeras classificações apresentadas nesse artigo, têm como principal objetivo agregar qualidade ao software. Desde que utilizadas de maneira correta, elas permitem que as atividades voltadas para o processo de teste sejam controladas, otimizadas, evidenciadas, documentadas, e, principalmente, organizadas.

Seja para controlar todo o ciclo de vida do processo, gerir os defeitos apresentados ao longo da execução dos testes, avaliar o desempenho e o comportamento da aplicação sob determinadas demandas de carga extrema, agilizar os testes através da automação dos testes funcionais, ou seja para avaliar o quão seguro está o software, as ferramentas de suporte ao teste não vêm para substituir os testadores pelas “máquinas”, elas vêm para auxiliar na melhor execução das atividades de teste.

Essas ferramentas podem ser extremamente úteis para aprimorar a eficiência dos testes e agilizar a repetitiva rotina que envolve as atividades de um Processo de Teste. Mas lembre-se, pagas ou open source, elas devem ser primeiramente avaliadas, de acordo com os respectivos objetivos que se deseja alcançar ao utilizá-las.

**Links**

**Automated testing tool - test across all environments and apps.**  
[**http://www.borland.com/products/silktest/**](http://www.borland.com/products/silktest/)

**Benefits and Types of Software Testing Tools.**  
[**http://www.xing.com/net/smmetrics/testing-beta-trials-tools-**  
**products-a-b-testing-453076/benefits-and-types-of-software-testing-tools-27367116**](http://www.xing.com/net/smmetrics/testing-beta-trials-tools-products-a-b-testing-453076/benefits-and-types-of-software-testing-tools-27367116)

**Conheça o Watir.**  
[**http://qualidadebr.wordpress.com/2011/03/05/conheca-o-watir/**](http://qualidadebr.wordpress.com/2011/03/05/conheca-o-watir/)

**Ferramentas de Suporte a Teste.**  
[**http://www.tiexames.com.br/curso\_Teste\_Software\_CTFL.php**](http://www.tiexames.com.br/curso_Teste_Software_CTFL.php)

**Modelo de Avaliação da Qualidade Funcional de Software.**  
[**http://tconline.feevale.br/tc/files/0002\_1937.pdf**](http://tconline.feevale.br/tc/files/0002_1937.pdf)

**O que é Gerência de Configuração?**  
[**http://pronus.eng.br/artigos\_tutoriais/gerencia\_configuracao/**  
**gerencia\_configuracao.php?pagNum=0**](http://pronus.eng.br/artigos_tutoriais/gerencia_configuracao/gerencia_configuracao.php?pagNum=0)

**Software QA Testing and Test Tool Resources.**  
[**http://www.aptest.com/resources.html**](http://www.aptest.com/resources.html)

**Sistema de Gerenciamento do Teste de Software baseado na Norma ISO/IEC 12207.**  
[**http://www.bc.furb.br/docs/MO/2002/266367\_1\_1.pdf**](http://www.bc.furb.br/docs/MO/2002/266367_1_1.pdf)