# Conteúdo

**1 Anexo A - Testes funcionais e testes de regressão [1] 1**
1.1 Introdução . . . . . 1
1.2 Soluções existentes . . . . . 1
1.2.1 Splinter . . . . . 3
1.2.2 MaxQ . . . . . 4
1.2.3 Sahi . . . . . 5
1.2.4 Badboy . . . . . 6
1.2.5 Watir . . . . . 6
1.2.6 WebTest . . . . . 8
1.2.7 Quick Test Professional . . . . . 9
1.2.8 Selenium . . . . . 10
1.3 Comparação e conclusão . . . . . 11
1.4 Bibliografia . . . . . 14

# Lista de Tabelas

1.1 Comparação das ferramentas para testes funcionais e testes de regressão. . . . . 12

# Lista de Figuras

1.1 Splinter test framework . . . . . 3
1.2 CollabNet MaxQ . . . . . 4
1.3 Sahi Pro . . . . . 5
1.4 Badboy . . . . . 6
1.5 Watir . . . . . 7
1.6 WebTest . . . . . 8
1.7 Quick Test Professional . . . . . 9
1.8 Selenium . . . . . 10

# Capítulo 1

# Anexo A - Testes funcionais e testes de regressão [1]

## 1.1 Introdução

Fornecer software de qualidade não se limita em apenas escolher um bom motor de base de dados, boa estrutura de servidores, arquitetura de sistema bem planejada, linguagens e *frameworks* que a equipe saiba utilizá-los bem. Hoje em dia, com a grande concorrência e a luta para agradar o utilizador, as empresas precisamam ser ágeis e certeiros no investimento do tempo de desenvolvimento em relação a principal parte do software, aquela para a qual vai olhar utilizador e com qual esta vai trabalhar, aquela que trará o sucesso para quem está criando a aplicação é a interface web. Para garantir o funcionamento correto das funcionalidades muitas das vezes as empresas escolham algumas *frameworks* com o objetivo de executar os testes funcionais e testes de regressão. Este documento faz uma análise das ferramentas existentes descrevendo as vantagens e desvantagens de cada uma para este tipo de efeito.

## 1.2 Soluções existentes

As ferramentas analisadas neste documento serão as seguintes:

- Splinter [2]
- MaxQ [3]
- Sahi Pro [4]
- Badboy [5]

- Watir [6]
- WebTest [7]
- Quick Test Professional [8]
- Selenium [9]

Figura 1.1: Splinter test framework

### 1.2.1 Splinter

*Spliter* é uma das várias frameworks de testes para aplicações web que tem o objetivo de automatizar o processo de testes executando as ações que utilizador teria de realizar na página web. Esta ferramenta é escrita em *Python*, é *open source* e seu desenvolvimento é liderado pelo grupo de desenvolvimento *open source Cobrateam*.

As vantagens desta ferramenta são as seguintes:

- *Open source*.
- Suporta differntes *drivers* (*chrome*, *firefox*, *phantomjs*, *zopetest*).
- Permite execução remota.
- Permite simular ações do utilizador.
- Suport para *Javascript*.

As desvantagens são:


Figura 1.2: CollabNet MaxQ

- Não tem suporte nativo para CSS.
- Não permite a execução dos testes em paralelo.
- Não suporta os sistemas de integração contínua.

### 1.2.2 MaxQ

*MaxQ* é uma ferramenta open source desenvolvida por *CollabWeb* com o objetivo de executar testes funcionais páginas web. Esta ferramenta escrita *Java*. Os testes são criados em *scripts Jython*. *Jython* é uma implemetação do *Python*.

As vantagens desta ferramenta são as seguintes:

- É uma ferramenta open source.
- Permite a gravação das ações efetuadas na página web.
- Permite simular ações do utilizador.
- Os scripts podem ser executados usando *Junit*.
- Tem suporte para funcionar através de um proxy server.

As desvantagens são:

- Não suporta ferramentas para gestão e automatizacão de projetos
- Não suporta a execução dos testes em paralelo.
- Não suporta os sistemas de integração contínua.


Figura 1.3: Sahi Pro

### 1.2.3 Sahi

*Sahi* é ferramenta de automação de teste de aplicações Web. Existem dois tipos desta ferramenta: *Sahi* - versão open source com poucas funcionalidades, basicamente server para experementar a ferramenta, e *Sahi Pro* é a versão paga com muitas funcionalidades, de automação de teste de aplicações web.

As vantagens desta ferramenta são as seguintes:

- Permite a gravação das ações efetuadas na página web.
- Permite simular ações do utilizador.
- O formato dos relatórios pode ser igual a *JUnit*.
- Suporta todos os *browsers*.
- Permite guardar relatórios na base de dados.
- Permite gravar as ações efetuadas pelo utilizador.
- Suporta ferramentas para gestão e automatizacão de projetos. (só ant).
- Permite executar comandos *Javascript*.
- Permite executar os testes em paralelo.
- Bom suporte têcnico.

As desvantagens são:

- Versão *open source* tem número limitado das funcinalidades.
- Não fornece suporte para *IFrame*.
- Não suporta a execução dos testes remotamente.
- Não suporta os sistemas de integração contínua.


Figura 1.4: Badboy

### 1.2.4 Badboy

*Badboy* é uma aplicação proprietária, desenhada para automatizar testes de interfaces web utilizando o browser Internet Explorer. Este aspecto é a grande desvantagem apresentada por este produto, pois a maior parte das aplicações são desenhadas para serem utilizadas em múltiplos *browsers*. Por outro lado o Badboy é extensível pois permite que sejam agregadas novas funcionalidades.

As vantagens desta ferramenta são as seguintes:

- Suporta a execução dos testes remotamente.
- Permite simular ações do utilizador.
- Suporta a execução dos testes em paralelo.
- Bom suporte têcnico.

As desvantagens são:

- Ferramenta paga.
- Suporte só para *Windows*.
- Suporte só para um browser, *Microsoft Interner Explorer*.
- Não suporta ferramentas para gestão e automatizacão de projetos.
- Não tem suporta para ser integrada nos sistemas de integração contínua.

### 1.2.5 Watir

*Watir* é uma biblioteca open source utilizada para automatizar testes de aplicações executadas num *browser* através da escrita de *scripts* de teste de fácil interpretação e manutenção. Permite controlar o browser como um utilizador, como por exemplo seleccionar *links*, preencher formulários, pressionar botões, de entre outras operações. Depois de executados os *scripts*, é possível verificar os resultados das acções

Figura 1.5: Watir

executadas tal como o texto ou acção esperada na página web. Trata-se duma biblioteca da família Ruby que suporta aplicações independentemente da tecnologia que foi utilizada para o seu desenvolvimento, também existem bibliotecas para famílias *Java* (*Watij*) e *.NET* (*Watin*).

As vantagens desta ferramenta são as seguintes:

- Suporta vários browsers.
- Suporta vários sistemas operativos.
- Permite executar comandos *Javascript*.
- Suporta vários browsers.
- Bom *API*.
- Bom suporte têcnico.
- Permite simular ações do utilizador.
- Suporta ferramentas para gestão e automatizacão de projetos.

As desvantagens são:

- Não suporta a execução dos testes remotamente.
- Não suporta a execução dos testes em paralelo.
- Não tem suporta para ser integrada nos sistemas de integração contínua.


Figura 1.6: WebTest

### 1.2.6 WebTest

*WebTest* é uma ferramenta *open source* para realização dos testes funcionais das aplicações web. É uma ferramenta simples e rápido a executar os testes, com boa estrutura de relatórios com resultados dos testes.

As vantagens desta ferramenta são as seguintes:

- Pode ser executados em qualquer sistema operativo.
- Permite simular ações do utilizador.
- Permite fazer análise dos erros dos *javascript*.
- Permite executar comandos *Javascript*.
- Suporta *xpath*.
- Extensível.
- Suporta ferramentas para gestão e automatizacão de projetos.
- Bom suporte têcnico.

As desvantagens são:

- Não suporta a execução dos testes remotamente.
- Não suporta a execução dos testes em paralelo.
- Não tem suporta para ser integrada nos sistemas de integração contínua.
- Não suporta páginas web com *AJAX*.


Figura 1.7: Quick Test Professional

### 1.2.7 Quick Test Professional

*Quick Test Professional*(*QTP*) é ferramenta de automação dos testes funcionais e testes de regressão. Esta foi desenvolvida pela *Mercury Interactive* e adquirida pela *HP*. *QTP* permite testar objetos das páginas web e controles ActiveX. Além disso, *QTP* também permite *Java applets* e objetos multimídia, bem como aplicações do *Windows*, *Visual Basic 6* e .NET.

As vantagens desta ferramenta são as seguintes:

- Existe próprio *IDE*.
- Permite simular ações do utilizador.
- Tem suporte para, quase, todos componentes das páginas *web*.
- Suporte para *dialog boxes*.
- Suporte para *upload* dos ficheiros.
- Suporte para todos os *browsers*, incluindo mobile browsers.
- Permite executar comandos *Javascript*.
- Bom suporte têcnico.

As desvantagens são:

- Não suporta a execução dos testes remotamente.
- Não suporta a execução dos testes em paralelo.
- Não suporta ferramentas para gestão e automatizacão de projetos.

Figura 1.8: Selenium

  - Só tem suporte para sistema operativo *Windows*.
  - Não tem suporta para ser integrada nos sistemas de integração contínua.

- Ferramenta paga (4500).

### 1.2.8 Selenium

*Selenium* é um framework para a automatizacão de testes num *browser* em diferentes plataformas. *Selenium* é composto por três ferramentas principais. Cada uma delas desempenha uma função específica no processo de desenvolvimento de teste automatizado para aplicações *web*. Os três componentes são:

- *Selenium-IDE* é uma ambiente integrado de desenvolvimento que permite criar os *scripts* de teste; opera como um *add-on* para o *Firefox* e providencia uma *interface* simples para criar *scripts* de teste e executá-los
- *Selenium-RC* permite aos testers utilizar uma linguagem de programação mais flexível e extensível para desenvolver *scripts* de teste lógicos
- *Selenium-Grid* possibilita que as soluções *Selenium-RC* estendam os seus scripts de testes para múltiplos ambientes remotos.

  As vantagens desta ferramenta são as seguintes:

  - Ferramenta *Open Source*.
  - Existe próprio *IDE*.
  - Permite simular ações do utilizador.
  - Tem suporte para todos componentes das páginas *web*.
  - Suporte para todos os *browsers*, incluindo mobile browsers.
  - Suporta ferramentas para gestão e automatizacão de projetos.

- Suporta todos sistemas operativos.
- Bom suporte têcnico.
- Suporta a integração nos sistemas de integração contínua.
- Permite execução dos testes em paralelo.
- Permite execução dos testes remotamente.
- Permite executar comandos *Javascript*.
- Suporte para *xpath*.
- Suporte nativo para *CSS*.

As desvantagens são:

- Não tem suporte para *upload* dos ficheiros.
- Instável em ambientes complexos.
- Não tem suporte para *dialog boxes*.
- Criação dos *scripts* dos testes exige o conhecimento das linguagens de programação.
- Não existe suporte técnico mas existe uma comunidade muito ativa.

## 1.3 Comparação e conclusão

Todas as ferramentas analisadas tem como objetivo criar e executar testes funcionais e testes de regressão. Para além disso todas as ferramentas permitem simular as ações do utilizador na página web com o objetivo aproximar a execução dos testes com as ações reais do utilizador.

Na tabela a baixo pode ser encontrada comparação das ferramentas analisadas.

| **Ferramentas / Features** | **Splinter** | **MaxQ** | **Sahi** | **Badboy** | **Watir** | **WebTest** | **QTP** | **Selenium** |
|---|---|---|---|---|---|---|---|---|
| Suporte para CSS | N | N | S | N | S | N | S | S |
| Suporte para xpath | N | S | S | N | S | S | S | S |
| Suporte para Javacript | S | S | S | N | S | S | S | S |
| Suporte técnico | N | N | S | S | S | S | S | N |
| Integração com development tools | N | N | S | N | S | N | N | S |
| Integração com ferramentas de integração contínua | N | N | N | N | N | N | N | S |
| Suporte para diferentes browsers | S | S | S | N | S | S | S | S |
| Suporte para diferentes sistemas operativos | S | S | S | N | S | S | N | S |
| Execução testes em paralelo | N | N | S | S | N | N | N | S |
| Execução testes remotamente | S | N | S | S | N | N | N | S |
| Open source? | S | S | N | N | S | S | N | S |
| Gravar ações do utilizador | N | S | S | N | N | N | S | N |

Tabela 1.1: Comparação das ferramentas para testes funcionais e testes de regressão.

Como podemos ver, hoje em dia, muitas das ferramentas open source não sedem, em termos das funcionalidade, as ferramentas pagas. Muitas das ferramentas analisadas (*Splinter*, *MaxQ*, *Sahi*, *Watir*, *WebTest*, *QTP*, *Selenium*) permitam a execução dos comandos *javascript*, que, hoje em dia, estão a ser usados em quase todas as página web.

Todas as ferramentas analisas permitem realizar os testes funcionais, mas testes de regressão só faz sentido realizar só com algumas ferramentas. Faz sentido, executar testes de regressão para as páginas que estão em desenvolvimento, que sempre sofrem atualizações (adição das novas funcionalidades ou correção dos problemas encontrados anteriormente) com o objetivo de garantir que não surgiram novos defeitos em componentes já testados. Neste caso, a ferramenta deve ser capaz de suportar vários sistemas operativos e diferentes *browser*, como por exemplo *Splinter*, *MaxQ*, *Sahi*, *Watir*, *WebTest*, *Selenium*. Com crescimento das funcionalidades na página, o numero dos casos de testes aumenta e portanto o tempo da execução dos testes aumenta. Nesta caso, só algumas ferramentas permitem a execução dos testes em paralelo com o objetivo de diminuir o tempo da execução dos testes, estas ferramentas são *Sahi*, *Selenium*. Para automatizar o processo da execução dos testes, em muitas das vezes, é necessário integrar esta com plataformas de integração continua, como por exemplo *Hudson* ou *Jenkins*. Neste caso só há uma ferramenta que fornece suporte para isso é o *Selenium*.

Todas as ferramentas analisadas servem para executar testes funcionais das aplicações *web*. Cada entidade que desenvolve aplicação *web* deve escolher a ferramenta, conforme o seu processo de desenvolvimento.

## 1.4 Bibliografia

1. S. PRESSMAN, “Software Engineering - A Practitioner’s Approach“, McGrawHill, 2005.

2. Splinter. http://splinter.cobrateam.info. Acesso em Junho 15, 2013.

3. MaxQ. http://maxq.tigris.org. Acesso em Junho 15, 2013.

4. Sahi. http://www.sahi.co.in. Acesso em Junho 15, 2013.

5. Badboy. http://www.badboy.com.au. Acesso em Junho 15, 2013.

6. Watir. http://www,watir.com. Acesso em Junho 15, 2013.

7. WebTest. http://webtest.canoo.com. Acesso em Junho 15, 2013.

8. Quick Test Professional. http://www.hp.com. Acesso em Junho 15, 2013.

9. Selenium. http://www.seleniumhq.org. Acesso em Junho 15, 2013.