UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RDSamurai - Desenvolvimento de um Agente Inteligente para detecção de Gargalos de um Sistema de jogos on-line baseado em servidor web

Trabalho de Graduação

Henrique Vicentini

Santa Maria, RS, Brasil
2005

RDSamurai -
Desenvolvimento de um Agente Inteligente para detecção de Gargalos de um Sistema de jogos on-line baseado em servidor web

por

Henrique Vicentini

Trabalho de Graduação apresentado ao Curso de Ciência da Computação – Bacharelado, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**

Orientadora: Prof. Dra. Roseclea Duarte Medina

Trabalho de Graduação nº 205

Santa Maria, RS, Brasil
2005

Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação

A Comissão Examinadora, abaixo assinada, aprova o Trabalho de Graduação

RDSamurai – Desenvolvimento de um Agente Inteligente para detecção de Gargalos de um Sistema de jogos on-line baseado em servidor web

elaborado por

Henrique Vicentini

como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação

Comissão Examinadora:

_____
Prof. Dra. Roseclea Duarte Medina
(Orientadora)

_____
Prof. João Carlos D. Lima

_____
Prof. Oni Reasilvia de O Sichonany

Santa Maria, 14 de dezembro de 2005.

Agradecimentos

Os agradecimentos são algo de grande importância para um ser humano. Todos sabemos que os nossos esforços seriam em vão caso não o fizéssemos por um propósito, o caminho a ser percorrido em nossa jornada − digo nossa porque a cada passo que dei não estive sozinho − tive a oportunidade de estar ao lado de pessoas grandiosas e amigas as quais tive a honra de conhecer. Cada uma delas será citada nessa pequena demonstração de agradecimento.

O principal agradecimento é ao meu Deus todo poderoso e companheiro nas horas incertas, porque sem ele certamente não haveria tido forças para suportar nos difíceis momentos de problemas na busca pelo conhecimento.

Aos meus amados pais, o meu agradecimento especial, pois eu recebi de vocês, o que eu tenho de mais valor em minha vida que é o meu caráter, obrigado por todas as lutas vencidas juntos, pela força nas horas de dificuldade e pela compreensão nas minhas falhas.

Aos meus avós , Nonno e Nonna, pelo apoio e ajuda nas horas de dificuldades. Declaro-me até hoje uma pessoa de sorte, pois possuo dois pais e duas mães que me ensinaram e me auxiliaram a cada momento em minha vida.

À professora Roseclea, minha orientadora, pelas orientações recebidas e principalmente pelas palavras de incentivo e apoio, pois muitas vezes uma palavra de incentivo vale muito nas horas de dificuldade.

À minha namorada, Liane, pelo apoio e companheirismo nas horas difíceis e pela compreensão pelos momentos em que estive ausente nessa jornada.

Aos meus amigos que fiz no curso, pelas brincadeiras e colaborações para o meu crescimento como pessoa, principalmente aos meus amigos e colegas de trabalho da Decadium Studios, pelo companheirismo, amizade e apoio aos meus projetos.

Aos professores, pelo ensino, orientações e por serem facilitadores na busca do conhecimento.

Enfim, Os meus sinceros agradecimentos a todos que colaboraram para que esse sonho se tornasse realidade.

“Elevo os meus olhos para os montes; de onde me vem o socorro? O meu socorro vem do Senhor, que fez os céus e a terra.”

(Bíblia Sagrada, Salmos 121,1-2)

Resumo

Trabalho de Graduação

Ciência da Computação

Universidade Federal de Santa Maria

RDSamurai –

Desenvolvimento de um Agente Inteligente para detecção de Gargalos de um Sistemade jogos on-line baseado em servidor web

Autor: Henrique Vicentini

Orientadora: Profª Roseclea Duarte Medina

Local e Data da Defesa: Santa Maria, 21 de dezembro de 2005.

O monitoramento de desempenho de servidores web é fundamental para a manutenção da estabilidade do sistema como um todo, porém com o crescimento desse sistema torna-se impraticável para os administradores avaliarem a situação atual do sistema e os gargalos que ele apresenta. Surge assim a necessidade de um software autônomo e de execução contínua para o constante monitoramento do sistema os quais são conhecidos como Agentes Inteligentes. Esse agente tem a atribuição de buscar dados do sistema e entregar para o usuário as informações referentes a possíveis gargalos do sistema. O agente é de grande valia para um processo contínuo de monitoramento, pois devido à mobilidade dos gargalos é muito comum que, uma vez que um seja identificado e reparado, outro apareça. A criação de um agente que seja capaz de detectar problemas de desempenho é de extrema utilidade para futuras aplicações que envolvam sistemas Web. O Agente, nomeado de RDSamurai, atingiu os seus objetivos de constante monitoramento do desempenho na execução de um sistema web, obtendo resultados de grande valia para que a equipe de desenvolvimento do sistema conseguisse acompanhar a evolução do sistema e, quando necessário, aplicar seus esforços na otimização do mesmo. Devido a forma modular que foi implementado o RDSamurai, o mesmo pode ser aplicado a outras atividades de execução contínua, entre elas está previsto o monitoramento dos usuários do sistema Web auxiliando os administradores do sistema a monitorar quem desrespeitar as regras de uso do sistema.

Palavras Chaves: Agentes Inteligentes, Detecção de Gargalos, Sistemas Web.

Sumário

Capítulo 1

1 Introdução

O crescimento da Internet e o uso de computadores pessoais criaram um novo nicho de mercado de entretenimento, os jogos on-line e mais recentemente os jogos on-line baseado em turnos. Esses têm por necessidade um servidor de aplicação para que os usuários possam conectar-se e usufruir do sistema para o seu entretenimento. Essas aplicações do servidor são conhecidas como Sistemas Web, ou Aplicações Web.

Aplicações Web são sistemas acessados por inúmeros usuários e o uso dessas não está restrito a jogos, mas também sistemas de busca, comércio eletrônico etc. O acesso simultâneo a um recurso torna um sistema Web um alvo de gargalos, podendo desestabilizar o sistema de tal forma que o mesmo entra em colapso tornando necessário uma reinicialização para torná-lo operacional novamente. Essa operação limita o crescimento do serviço, atrelando os lucros da empresa ao gargalo do sistema.

Um gargalo é caracterizado como uma parte do sistema que esteja limitando o desempenho do mesmo, segundo Ribas [12]. Muitas vezes os administradores do sistema tomam a decisão precipitada de trocar o servidor por outro com mais recursos, desperdiçando capacidade computacional do servidor e o dinheiro da empresa, onde a eliminação ou atenuação dos gargalos já seria suficiente para tornar o sistema estável novamente. Assim, a eliminação dos gargalos do sistema é de extrema importância antes de tomar qualquer decisão a respeito de substituição de hardware, pois o sistema continuará instável à medida que aumenta o número de usuários do mesmo.

A Decadium Studios desenvolveu um jogo baseado em turnos ambientado no Japão Feudal intitulado Ryudragon[1], o qual utiliza-se de um Servidor Web para disponibilizar o seu serviço. O jogo apresenta atualmente mais de 25.000 (vinte e cinco mil) usuários registrados e pouco mais de 9.000 (nove mil) desses utilizando o sistema (dados do dia primeiro de outubro de 2005). No sistema permanecem, em média, 400 usuários logados, suportados com tranqüilidade no servidor, porém o sistema torna-se instável e lento à medida que o número de usuários ultrapassa 500 jogadores simultâneos. Atualmente esse sistema se encontra sobrecarregado e com impossibilidade de crescimento, o que é insustentável para uma

[1] Acessível em: http://www.ryudragon.com

empresa, visto que limita seus lucros por causa do sistema e não por causa dos usuários do mesmo.

Essa situação tornou-se inadmissível ao iniciar-se um período de *crash test* [15] (realizado no período de 30 de maio de 2005 a 9 de junho de 2005) quando o servidor entrou em colapso, necessitando o pedido de *reboot* físico, o qual não se encontra fisicamente acessível pela equipe de desenvolvimento, pois a mesma terceirizou o serviço de hospedagem. O servidor encontra-se em território internacional (Texas - EUA) assim qualquer medida cabível a ser tomada para ajuste do sistema deve abrir uma chamada para que a mesma seja atendida, muitas vezes retardando as ações da equipe. A expressão *crash test* foi originada na indústria automobilística, a qual buscava maior segurança dos seus veículos e garantias para o consumidor final. Os testes são realizados através de colisões dos veículos portando bonecos em obstáculos fixos a fim de determinar vários fatores que envolvem uma colisão. No caso de jogos a expressão é mais recente e consiste em sobrecarregar um sistema com o intuito de detectar falhas de projeto ou o limite do sistema como um todo.

Assim a necessidade de eliminar os gargalos se torna evidente, eliminando gastos extras com a troca do servidor, melhoria do serviço prestado aos usuários do sistema e viabilizando seu crescimento. Porém surgem as questões: como detectar um gargalo em um sistema Web? Quais os principais subsistemas onde pode surgir um gargalo? A localização é impraticável sem o auxílio de um software para a coleta dessas informações, entrando assim a figura de um agente inteligente para a solução do problema.

Um agente inteligente consiste em um software capaz de interagir com o ambiente em questão, com a finalidade de atingir seus objetivos ou cumprir as tarefas para as quais fora designado, segundo Maes [8].

Esse agente teria a atribuição de buscar dados do sistema e entregar para o usuário as informações referentes a possíveis gargalos do sistema. A partir desses dados a equipe de desenvolvimento tomaria ações referentes à eliminação ou atenuação dos mesmos. Este agente será de grande valia para um processo contínuo de monitoramento de gargalos, pois segundo Ribas [12], devido à mobilidade dos gargalos é muito comum que, uma vez que um gargalo seja identificado e reparado, outro apareça. A criação de um agente que seja capaz de detectar esses gargalos é de extrema utilidade para futuras aplicações que envolvam sistemas Web.

1.1 Objetivos deste trabalho

A proposta desse trabalho é de desvincular o crescimento do jogo Ryudragon dos gargalos que o sistema apresenta, aumentando a rentabilidade do jogo para a empresa Decadium Studios, proporcionando maior diversão aos usuários do jogo e agregando mais clientes para a empresa. Para tal é necessário detectar e remover esses gargalos do sistema.

Será criado um Agente Inteligente direcionado para a detecção de gargalos do sistema como proposta para solucionar o problema de desempenho da aplicação. Essa ferramenta, batizada de RDSamurai[2], servirá para diagnosticar possíveis fragmentos de um sistema que são mais requisitados e que possuem maior tempo de processamento para que uma equipe de desenvolvimento saiba aonde concentrar sua atividade de otimização.

1.2 Organização do texto

A próxima seção é caracterizada pela introdução de conceitos básicos sobre o que o presente documento abordará, além de uma breve introdução do problema a ser solucionado. O assunto de Jogos baseado em turnos inicia a abordagem desse trabalho no capítulo 2, seguido pelo Desempenho de sistemas Web e Gargalos de Sistema que serão apresentados na seção 3. Na seção 4, serão abordados Agentes de Software, suas características, tipos de agentes, utilização de agentes e o emprego de agentes na identificação de gargalos em sistemas web. Na seção 5, serão propostas algumas características do projeto do Agente RDSamurai a ser desenvolvido. No Capítulo 6, será feita uma abordagem do projeto e implementação do Agente RDSamurai. Na seção 7, serão explanados os resultados obtidos pelo sistema. O plano de otimizações será abordado no capítulo 8. No capítulo 9 são apresentadas as conclusões do trabalho e a explanação dos trabalhos futuros. Por último, no capítulo 11 as referências bibliográficas utilizadas no desenvolvimento desse trabalho.

[2] O nome RDSamurai remete à designação RyuDragon Samurai, aonde Ryudragon é o nome do sistema a ser monitorado e Samurai tem o significado de "aquele que serve ao senhor".

Capítulo 2

2 Jogos Web baseado em Turnos (*Turn-Based*)

Um jogo baseado em turnos (*turn-based*), também conhecido como jogo de estratégia baseada em turno, é um jogo dividido em partes visíveis e bem definidas chamadas de turnos. Os jogadores de um *turn-based* permanecem um período analisando a situação atual do seu jogo e da sua estratégia antes de realizar as suas ações no jogo, separando os processos de ação e raciocínio, e escolher a melhor solução que se adéqüe àquele caso. Após todos os jogadores terem feitos suas ações no turno, as mesmas são processadas e tem início um novo turno, como o diagramado na Figura 1 que representa o funcionamento básico de um jogo baseado em turnos:

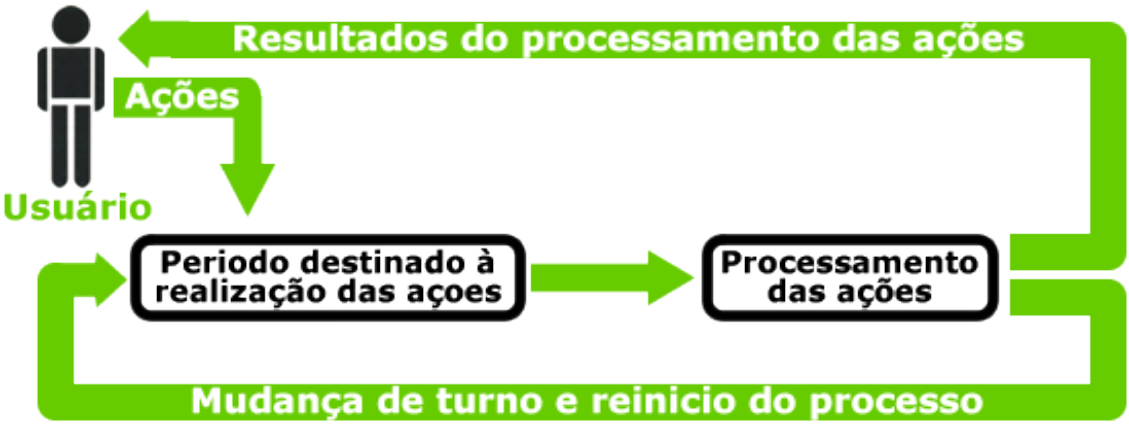

Figura 1: Funcionamento básico de um jogo baseado em turnos.

2.1 O jogo Ryudragon

O jogo on-line **Ryudragon** será apresentado nos próximos dois itens, segundo a descrição formal da Decadium Studios [5]:

O que é o Ryudragon

O Ryudragon é um jogo Web classificado como MMORPG e *turn-based*. Tais jogos apresentam como características principais a grande quantidade de jogadores simultâneos interagindo entre si e o processamento das ações em intervalos fixos de tempo (intervalo de tempo denominado turno).

Por ser um jogo Web, Ryudragon necessita apenas de um navegador e de acesso à Internet para ser jogado. Nenhum *download* é necessário, possibilitando que o usuário o acesse de qualquer local com acesso à rede, como *lan houses, cyber-cafés* e laboratórios de informática, a qualquer hora do dia.

O jogo é ambientado na época do Japão feudal, enfoque diferente de todos os jogos concorrentes, os quais mantêm suas histórias no futuro ou em temas medievais ocidentais, cenários já explorados e bastante desgastados por várias categorias de jogos de computador.

Em Ryudragon, o usuário tem como objetivo levar sua conta (denominada **feudo**) até as posições mais altas no ranking, através de batalhas e alianças com os outros jogadores, até o final da era de dois meses. Após este período, o jogo é reiniciado, e uma nova era se inicia, com as batalhas começando outra vez. Isso possibilita que o interesse do usuário em relação ao jogo não seja perdido, uma vez que ele tem chance de ser o vencedor de quaisquer das eras que participar.

Para incentivar a disputa, os vencedores de cada era ganham prêmios. Atualmente a premiação é voltada, porém não limitada, a produtos de informática como mouses ópticos, teclados sem fio e joysticks. Já utilizamos como prêmios também camisetas, jogos de computador e DVDs de filmes, como Half-Life II, Prince of Pérsia II, Senhor dos Anéis, sendo os dois últimos muito visados pelo público alvo.

Já dentro do jogo, o usuário tem em suas mãos um feudo, que deve ser gerenciado conforme seus critérios, para que se desenvolva e cresça. Ao iniciar o jogo, o usuário é inserido em um clã, que é um grupo de dez jogadores. Os integrantes do clã se tornam aliados e conhecidos, podendo trocar informações importantes sobre combates, conversar sobre táticas e também prestar ajuda um ao outro, como para ataques ou defesas em conjunto. Tal método de jogo tem como objetivo a divulgação: um usuário pode convidar seus amigos para jogarem Ryudragon junto com ele, fazendo com que o número de jogadores cresça. Uma vez em um clã, o jogador pode comprar trabalhadores, para que esses produzam recursos, como arroz, madeira e ouro, e sejam, então, investidos na construção de estruturas, como centros de treinamento de soldados e muralhas. De posse de exércitos, o jogador coordena os ataques feitos a outros jogadores, analisando as melhores táticas junto com seu clã, visando sempre alcançar as primeiras colocações do ranking. Durante a era, o clã do jogador tende a se aliar com outros clãs, aumentando o número de conhecidos e amigos.

O envolvimento do usuário com o jogo é potencializado através da escolha de um dos cinco povos disponíveis na criação do feudo. Estes povos darão um tipo para o feudo, cada um com características diferentes. Tais povos, chamados de origens, tiveram suas informações obtidas por nossa equipe através de pesquisas históricas, todas voltadas à cultura do Japão na época feudal. Isso cria um cenário rico em detalhes, tento em vista que os nomes das unidades, estruturas e origens, bem como a descrição de cada uma, foram colocadas pela Decadium Studios, baseadas e adaptadas nos resultados destas pesquisadas. Cada origem, conforme pesquisado, possuía seus costumes e modos de sobrevivência, o que é retratado em Ryudragon através da diferenciação e especialização de cada origem em uma determinada área, como ataque, defesa, comércio, espionagem e natureza. Cada origem possui vantagens, desvantagens e características únicas, que são analisadas e comentadas pelos jogadores. O usuário possui, então, um leque de possibilidades para escolher o seu estilo de jogo, uma vez que cada origem é diferente. Isto fortifica a idéia de se ter um clã, composto por vários feudos diferentes.

Ao longo da era, o jogador realiza batalhas contra outros jogadores e clãs, para promover o desenvolvimento de seu feudo. Tal desenvolvimento é expresso por dez níveis. Quando inicia o jogo, o feudo se encontra no primeiro nível, com obtenção de recursos limitados e pouco poder de combate. À medida que vai alcançando níveis mais altos, que são proporcionalmente mais difíceis de alcançar, o usuário obtém acesso a novas tecnologias e centros de treinamento, o que incrementa o poder de

combate de seu feudo, bem como lhe dá acesso a novas ferramentas, para que suas táticas de guerra sejam aprimoradas também. A divisão do jogo em níveis tem por objetivo manter o usuário sempre interessado em continuar jogando, mesmo quando o limite dos níveis for alcançado, porque é nesse momento que ele poderá realizar as maiores batalhas contra seus rivais. A curiosidade e a vontade de superar seus concorrentes motiva o usuário a jogar o máximo possível.

O jogador também pode obter dicas e esclarecimento de dúvidas no manual e fórum disponibilizado pelo jogo. O fórum possibilita um entrosamento entre os jogadores, contribuindo para que a comunidade do Ryudragon cresça e se consolide. Integrantes da equipe Ryudragon, bem como jogadores experientes (moderadores), monitoram o fórum, a fim de garantir que as dúvidas sejam sanadas e que o ambiente se mantenha tranqüilo e proveitoso.

Características Gerais do Ryudragon

O Ryudragon está em ascensão no seu segmento de mercado, tendo finalizado sua primeira era com 5800 jogadores cadastrados, a segunda era com 8000 e estando agora em sua terceira era com 12025 (08/08/2005, 08:07) jogadores, tendo sido notado pela mídia regional e estadual como um trabalho sério e de foco definido (vide jornais A Razão, Diário de Santa Maria, Zero Hora, NH e matérias publicadas na rede). Com uma média de 336000 acessos mensais, com cerca de 154000 visitantes (diferentes) por mês, o Ryudragon apresentou um crescimento no número de jogadores de aproximadamente 64%, durante a segunda para a terceira era, o que comprova o âmbito empresarial e o profissionalismo do serviço.

Ryudragon foi desenvolvido, visando tanto o mercado nacional como o internacional, com suporte a várias linguagens, sendo expansível a um número ilimitado de línguas. Atualmente, disponibilizamos o jogo em Português, Inglês e Espanhol. A atualização dos arquivos de linguagem, bem como a inclusão de um novo idioma, é fácil e executado de forma rápida.

Uma característica importante de Ryudragon é sua jogabilidade, que é de fácil entendimento por parte do jogador. A premissa básica, seguida por toda equipe da Decadium Studios durante o desenvolvimento do jogo, foi a facilidade para se jogar. Há manual e tutoriais, bem como textos explicativos, na maioria das áreas do *site*, a fim de integrar e familiarizar o usuário com o jogo de forma rápida e eficiente. Para que tal tarefa fosse realmente cumprida, estudamos a jogabilidade dos jogos existentes no mercado, evitando erros por eles já cometidos, e desenvolvendo métodos próprios e originais. Dentre eles, podemos destacar o tutorial animado, um vídeo que mostra passo a passo como jogar Ryudragon.

A permanência do usuário no jogo foi uma de nossas maiores preocupações. Conforme análises realizadas aos concorrentes nacionais e internacionais, constatamos que os jogadores ficavam frustrados e desistiam de jogar depois de perderem batalhas e terem seus exércitos dizimados por oponentes muito mais fortes que seu poder. Com base nisso, desenvolvemos um sistema de ranking que possibilita apenas ataques entre jogadores que travem, de fato, uma batalha justa. Isso evita abusos e faz com que o jogador se recupere rapidamente depois de vários ataques.

Ryudragon apresenta arte gráfica de qualidade, mesmo sendo um jogo *web*. Tendo em vista que a Decadium Studios desenvolve jogos de computador, possuímos uma equipe exclusiva para o desenvolvimento de material gráfico, composta por modeladores 3D, desenhistas e *designers;* possuímos também contatos com empresas especializadas em desenho. O trabalho de nossa equipe de *design* permitiu que o jogo tivesse quatro estilos (*skins*), sendo possível ao jogador escolher o visual que mais lhe agrada do *site*. A elaboração de novas *skins* segue o mesmo princípio que o sistema de linguagens: a inclusão de uma é feita de forma rápida e eficiente, sendo ilimitado o número de estilos que o *site* pode possuir.

- **Problemas com sobrecarga**

Atualmente o sistema está com mais de 9.000 jogadores (dados de três de outubro de 2005), dos quais, em média, mais de 400 permanecem on-line na aplicação. O sistema encontra-se sobrecarregado, impossibilitando a agregação de novos usuários por não dar suporte para mais jogadores sem desestabilizar o sistema. Assim encontra-se com o crescimento comprometido, inviabilizando campanhas de marketing para o seu crescimento. Limitando o lucro da empresa aos jogadores que estão utilizando o mesmo atualmente.

Como mostrado na Tabela 1, fornecida pela Decadium Studios de um de seus *crash-tests,* podemos verificar que a carga do servidor ao atingir 450 usuários on-line chega ao limite do uso do sistema.

Tabela 1 - Análise dos resultados da carga do processador pelo número de usuários on-line

Usuários on-line	**Carga do Servidor**	
	load	**Uso do processador**
200	0.8	30%
250	1.1	39%
300	1.6	
350	2.4	53%
400	3.5	68%
450	5.6	
		80%
500*	9.5*	99%
		100%*

* Valor atingido em poucas situações e na maioria delas algum dos serviços do servidor caíram.

Capítulo 3

3 Desempenho de Sistemas Web

Segundo Almeida [1], o desempenho é a medida da capacidade de resposta de um sistema.

Segundo Ribas [12], qualquer que seja a causa da queda na capacidade de resposta do sistema, trata-se de um gargalo. É muito comum que, uma vez que um gargalo seja identificado e reparado, outro apareça, o qual não foi percebido por causa da gravidade do gargalo anterior ou porque foi causado pelo reparo do gargalo inicial. Nesta segunda hipótese, o novo gargalo pode ter criado mais demanda a um outro recurso, fazendo com que ele se torne a restrição ao fluxo de trabalho. A detecção de gargalos é o processo de isolar os componentes de hardware ou software que limitam o fluxo de seu trabalho.

O desempenho de um sistema corresponde ao que é requerido do mesmo. No momento em que utilizamos um servidor de arquivos, por exemplo, não necessitamos de um tempo de resposta baixo, mas sim de uma taxa de transferência elevada. Já quando se trata de jogos on-line o tempo de resposta é mais importante do que a taxa de transferência, pois a demora de resposta do sistema afeta a diversão dos jogadores.

3.1 Gargalos de Sistemas

Valendo-se da citação de Punk [6] que constata:

> Por mais que algumas pessoas insistam em aderir à ioga, metralhar os usuários, se separar da esposa e desistir dos projetos, isso não é uma solução. A real solução é tentar localizar o gargalo no desempenho de seu sistema e eliminá-lo.

Um gargalo é a parte do sistema que, no momento, está limitando o fluxo de trabalho. Geralmente, trata-se do consumo excessivo de um recurso específico. A resposta a solicitações de acesso a arquivo, por exemplo, pode ser longa demais para o número de usuários que estejam acessando o servidor. Neste caso, terá sido detectado um sintoma de gargalo, segundo Ribas [12].

Caso um sistema não consiga desempenhar as suas tarefas definidas dentro de um prazo hábil o mesmo entra em um estado de arrasto dos processos, aumentando o número de tarefas na fila de espera e o tempo de resposta da aplicação, caracterizando um aumento do *Lead time*[3].

Os gargalos do sistema aparecem geralmente dentro dos quatro principais recursos de análise e otimização do servidor: memória, processador, subsistema de disco e subsistema de rede, conforme Ribas [12].

3.1.1 Gargalos de Memória

O melhor indicador de um gargalo de memória é uma taxa alta constante de falhas severas de página. Elas ocorrem quando os dados de que um programa precisa não se encontram paginados em memória e devem ser recuperados do disco. Taxas constantes de falha severa de página são um indicador claro de um gargalo de memória.

3.1.2 Gargalos de Processador

Duas das causas mais comuns de gargalos de CPU são os aplicativos e *drivers* com grande uso de CPU e interrupções excessivas que são geradas por componentes inadequados de disco ou do subsistema de rede.

3.1.3 Gargalos do subsistema de disco

Os discos armazenam os programas e os dados que estes processam. Enquanto se espera que um computador responda, freqüentemente é o disco que é o gargalo. Neste caso, o subsistema de disco pode ser o aspecto mais importante do desempenho de E/S (Entrada e Saída), mas os problemas podem ser ocultados por outros fatores, como a falta de memória.

Os contadores de disco estão disponíveis com os objetos Disco lógico e Disco físico. Disco lógico monitora as partições lógicas de unidades físicas. Ele é útil para determinar qual partição está causando a atividade do disco, possivelmente indicando o aplicativo ou serviço

[3] Lead Time corresponde ao tempo computado entre o início da primeira atividade até a conclusão da última, em série de atividades

que está gerando as solicitações. Disco físico monitora unidades individuais de disco rígido e é útil para monitorar unidades de disco como um todo.

3.1.4 Gargalos do subsistema de rede

Os gargalos de rede são uma das mais difíceis áreas a monitorar, por causa da complexidade da maioria das redes atualmente. Vários problemas diferentes podem afetar o desempenho da rede. Ao monitorar a rede, vários contadores e objetos diferentes podem ser monitorados, tais como servidor, redirecionador, segmento de rede e protocolos.

3.1.5 Gargalos de software

Os gargalos de software são caracterizados por regiões de software mal projetadas ou programadas inadequadamente. Os efeitos dos gargalos de software geralmente recaem sobre os sistemas de hardware, gerando gargalos de desempenho em um ou mais subsistemas citados anteriormente.

3.2 Sistemas Web

Segundo Kepler [7], aplicações Web (também conhecidas como Web apps) são programas utilizados por meio de um navegador Web. Toda vez que se faz uma busca no Google, lê um e-mail no Hotmail ou navega pela Amazon, se está utilizando uma aplicação Web.

3.2.1 Características de um Sistema Web

Um Sistema baseado em Web possui características peculiarmente diferentes de outros sistemas baseados na estrutura Cliente-Servidor. Uma aplicação Web pode também ser separada em Cliente e Servidor. Aplicações de sistemas Web possuem características muito importantes para o desenvolvimento de sistemas, que são a portabilidade, acessibilidade, maior controle da segurança da informação e manutenção dos direitos autorais do sistema.

- **Portabilidade**

A portabilidade de uma aplicação Web se dá pela caracterização de sua linguagem de programação. As linguagens de programação voltadas ao desenvolvimento de páginas dinâmicas possuem a característica de serem interpretadas e de serem extremamente portáveis como é o caso do PHP (*PHP Hyper Processor*), JSP (*Java Server Page*), ASP (*Active Server Page*), JavaScript, entre outras menos conhecidas.

As linguagens de programação para o desenvolvimento de aplicações Web são divididas em dois grupos: *Server Side*[16] e *Client Side*[14].

Segundo a Wikipedia[4], Server-side scripting[5] é uma tecnologia de servidores Web na qual uma requisição do usuário é fornecida ao executar diretamente um script no servidor Web para gerar páginas HTML dinâmica. Esta tecnologia é freqüentemente usada para prover páginas Web interativas provenientes de uma interface com bases de dados ou outras formas de armazenagem de informação. Este é diferente do client-side scripting[6], os quais são executados pelo navegador Web, freqüentemente em JavaScript. A principal vantagem dos server-side scripting é a alta capacidade de personalização de resultado para os requerimentos do usuário, assertivas de acesso, ou consultas em bases de dados.

- **Acessibilidade**

A acessibilidade do sistema se dá por um navegador Web comum, pois a apresentação das informações é feita por meio da linguagem de marcação HTML (*hiper Text Markup Language*). A aplicação é carregada do servidor para o cliente, não importando onde o cliente esteja acessando a aplicação, ou qual seja o computador a ser utilizado, podendo ser acessado através de diversos dispositivos diferentes.

A acessibilidade não se restringe em espaço físico. Qualquer dispositivo com conexão a Internet e com um navegador que suporte o padrão HTML é capaz de acessar essa informação de qualquer lugar do mundo, não limitando o acesso a uma aplicação sendo executada no seu dispositivo.

[4] Wikipedia, the free encyclopedia. http://www.wikipedia.org/
[5] Programação em linguagens de script executadas no Servidores
[6] Programação em linguagens de script executadas no Cliente

- **Segurança da Informação**

Segundo o banco Bradesco [2], a segurança da informação é um conjunto de medidas que se constituem basicamente de controles e política de segurança, tendo como objetivo a proteção das informações dos clientes e da empresa, controlando o risco de revelação ou alteração por pessoas não autorizadas.

Em uma aplicação Web, a informação é centralizada no servidor de aplicação. Essa centralização facilita a manutenção da integridade do sistema, focalizando as práticas de segurança no servidor de aplicação, de forma a evitar que outra pessoa, sem as devidas permissões de acesso, venha a acessar as informações contidas no computador em uso pelo usuário do sistema.

- **Direitos Autorais**

As aplicações não são disponibilizadas para os usuários como um software mas como um serviço Web fornecido por um servidor. Assim elas dificilmente são copiadas sem licenças, mantendo a propriedade intelectual do produto para os desenvolvedores do mesmo.

3.2.2 Dimensionamento de um Servidor Web

Uma das preocupações dos projetistas de sistemas Web é a projeção do número de usuários que, possivelmente, farão uso do sistema a ser desenvolvido. Essa informação é um dos principais critérios para a escolha do plano de hospedagem desse sistema.

Um sistema Web bem dimensionado tem que atender determinados parâmetros como: disponibilidade, tempo de resposta, número máximo de usuários on-line, largura da conexão de rede. Esses parâmetros devem ser especificados juntamente com a equipe de desenvolvimento para que exista a conscientização da mesma no dimensionamento do sistema e definições dos critérios de projeto.

Sistemas Baseados em Web são, em geral, sistemas que necessitam de uma boa disponibilidade e confiabilidade. Porém com o aumento de usuários, eles podem tornar-se instáveis e poderão chegar a um determinado ponto que se tornem inseguros, abrindo falhas de segurança.

Capítulo 4

4 Agentes

No meio científico existe uma dificuldade para encontrar um consenso sobre a definição formal do que seja um agente, o qual abranja todos os aspectos possíveis. Assim, encontramos várias definições de agentes, todas válidas e complementares.

Valendo-se da observação de Jennings & Woodridge [17] o qual constatou que:

> "Que a questão o que é um agente? É embaraçosa para a comunidade computacional de agentes da mesma forma como a questão o que é inteligência? É embaraçosa para a comunidade de Inteligência Artificial (IA)."

Segundo o Dicionário Houaiss [6] da língua portuguesa, agente é aquele que ou quem atua, opera, agencia; ou, agente é aquele que ou quem agencia negócios alheios. Baseado nessas definições, agentes possuem duas atribuições distintas: a primeira é algo ou alguém que age independentemente; na segunda atribuição, o agente atua como um coadjuvante ou colaborador para outro personagem do sistema.

Conforme Jennings & Woodridge [17], agentes são entidades de software persistentes, dedicadas a um propósito específico. Persistência distingue agentes de sub-rotinas; agentes têm suas próprias idéias sobre como realizar tarefas e sua própria agenda. Propósito específico os distingue da maioria das demais aplicações; os agentes são tipicamente menores. Porém, pode-se observar outras definições feitas por pesquisadores da área:

De acordo com Cazarini [4], um agente é uma entidade que possui um sistema interno de tomada de decisões. Agindo sobre o mundo e sobre outros agentes que o rodeiam, o qual é capaz de realizar suas tarefas sem a intervenção de outros sistemas ou de alguém, isto é, possui mecanismos próprios de percepção exterior.

Segundo Ketchpel & Genesereth [9], agentes de software são componentes de software capazes de comunicar e cooperar com seus pares por meio de troca de mensagens, usando uma linguagem de comunicação.

De acordo com Maes [8], um agente pode ser definido como um assistente pessoal que colabora com o usuário em uma determinada tarefa.

Segundo Cazarini [16], foi proposta uma hierarquia de agentes e não agentes, conforme mostram a Figura 2 e as definições a seguir:

Figura 2: Classificação hierárquica de agentes e não agentes. Cazarini [4].

- **Entidades:** todos os componentes do mundo são entidades.
- **Objetos:** entidade que possui atributos e métodos.
- **Agentes:** instância de um objeto que possui um propósito ou um conjunto de propósitos
- **Agentes Autônomos:** sistema com capacidade de interação independente e efetiva, com o objetivo de concluir alguma tarefa

4.1 Características dos agentes

Segundo Cazarini [4], equivalente à dificuldade encontrada para definir o que é um Agente, surge a dificuldade de definição de parâmetros, características, funções e restrições totalmente racionais ao conceito de Agente. Assim de um Agente de Software são esperadas três características:

- **Adaptação:** Identifica a capacidade dos agentes se adaptarem aos ambientes e usuários, aprendendo com as experiências;
- **Cooperação:** A utilização de uma linguagem e protocolo padrão entre agentes possibilita a cooperação e a colaboração para alcançar metas comuns;
- **Autonomia:** Agentes realizam suas ações autonomamente a modo de seguir a sua agenda.

Essas características são ilustradas na Figura 3 abaixo:

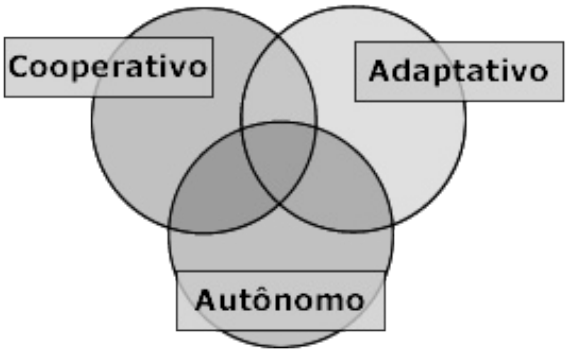

Figura 3: Características de um agente de software, Cazarini [4].

De acordo com Jennings & Woodridge [17], o comportamento dos agentes pode ser definido pelas propriedades que possui. A diferenciação entre agentes é dada pelos comportamentos que apresentam de acordo com suas características. Para que um sistema seja considerado agente, o mesmo não necessita apresentar todas as propriedades, definidas para agentes.

A seguir são apresentadas algumas das propriedades descritas pelos autores:

- **Autonomia:** os Agentes operam sem a intervenção direta de humanos ou outros, e possuem algum tipo de controle sobre suas ações e estado interno;
- **Habilidade social:** agentes interagem com outros agentes (e possivelmente humanos) através de algum tipo de linguagem de comunicação;
- **Reatividade:** agentes entendem seus ambientes e respondem a tempo para se adaptarem as mudanças;
- **Pró-atividade:** agentes não simplesmente respondem ao seu ambiente, eles são capacitados a atuar com iniciativa própria, procurando alcançar uma meta;
- **Continuidade temporal:** agentes permanecem em execução continuamente, ou em funcionamento ativo em primeiro plano, ou passivo em background;
- **Mobilidade:** é a capacidade de um agente de se mover no ambiente através de uma rede de computadores;
- **Cooperação:** é a capacidade que o agente tem de trabalhar conjuntamente com outros agentes, em busca de um objetivo em comum;

- **Benevolência:** é a suposição que o agente não possua metas contraditórias, de modo que todo agente tentará, então, sempre fazer o que é exigido.

4.2 Tipos de agentes

Segundo Bogo [3], os tipos de agentes podem ser classificados de diversas formas. Serão abordadas as classificações quanto à tarefa que executam e quanto às suas características primárias. Todas as classificações podem ser utilizadas em aplicações de agentes, a escolha está relacionada ao objetivo que se procura com a classificação.

4.2.1 Classificação segundo as tarefas que realizam

Segundo Cazarini [4], os agentes podem ser classificados segundo as tarefas realizadas:

- **Agentes de informação:** são agentes dirigidos para obter informação requerida por um agente humano;
- **Agente de entretenimento:** são agentes destinados ao entretenimento, animando e simulando personalidades artificiais;
- **Agentes de aconselhamento:** são agentes que colaboram na execução de tarefas, aconselhando e sugerindo caminhos para a realização da mesma;
- **Agentes assistentes:** são agentes destinados a executar tarefas para o agente humano;
- **Agentes de interface:** são agentes destinados a interagir com os agentes humanos.

4.2.2 Classificação segundo suas características

De acordo com Bogo [3], os agentes podem ser classificados, de acordo com as características primarias, em: agentes inteligentes, agentes de aprendizado, agentes de interface e agentes colaborativos, conforme a Figura 4.

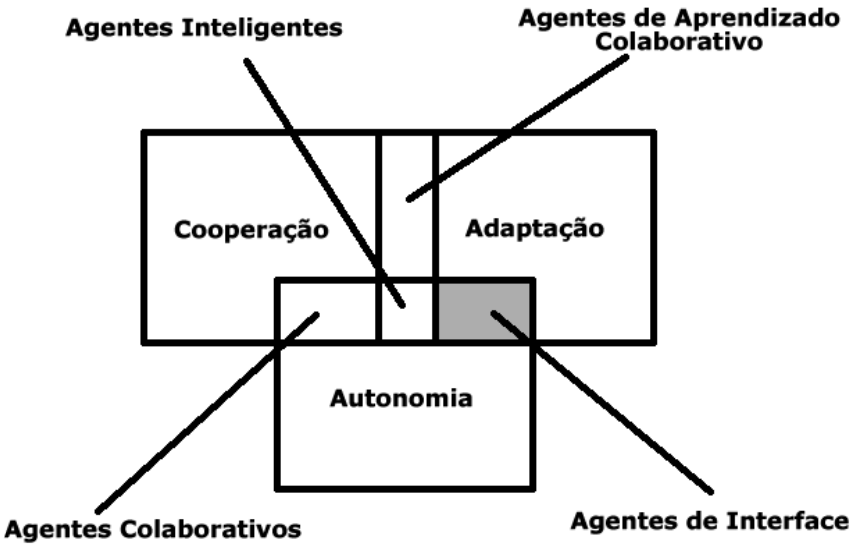

Figura 4: Classificação de agentes segundo suas características, Bogo [3]

4.3 Abordagens na Construção de Agentes

Segundo Maes [8], a idéia de empregar agentes de software na realização de certas tarefas foi introduzida por visionários. Mais recentemente, vários fabricantes de computadores adotaram a utilização de agentes em seus sistemas. Apesar de uma grande quantidade de trabalho ter sido realizado na construção de agentes, atualmente as técnicas existentes não permitem a construção de agentes capazes de gerar interações de alto nível, semelhantes às humanas. Dois principais problemas a serem resolvidos na criação de um agente de software:

- **Competência:** como um agente pode adquirir o conhecimento necessário para decidir quando agir e qual ação tomar;
- **Confiança:** como o usuário pode se sentir confortável ao delegar atividades a um agente.

A primeira abordagem consiste em tornar o programa do usuário um agente de interface. Existem soluções empregando agentes semi-autônomos os quais consistem em uma coleção de regras programadas pelo usuário para o processamento da informação para a realização de uma tarefa em particular. O principal problema desta abordagem é que o critério

de competência não é tratado de um modo adequado. E o usuário deve reconhecer a oportunidade para empregar um agente, tomar a iniciativa de criá-lo, fornecer explicitamente ao agente o conhecimento, e manter atualizadas todas as regras do agente. Nesta abordagem não ocorre o problema de confiança no agente, pois a mesma é provida pelo usuário e é tão confiável quanto sua própria habilidade programacional.

A segunda abordagem, conhecida como "abordagem baseada no conhecimento", consiste em dotar um agente com um extensivo conhecimento específico a um determinado domínio, sobre a aplicação e sobre o usuário. Em tempo de execução, o agente usa seu conhecimento para reconhecer os planos do usuário e encontrar oportunidades para contribuir com ele. Porém, os critérios de competência e confiança constituem um problema nessa abordagem.

O primeiro problema está relacionado com a competência, pois esta abordagem requer uma grande quantidade de trabalho e esforço por parte do engenheiro do conhecimento, pois uma grande quantidade de conhecimento sobre a aplicação e de conhecimento específico tem que ser inserido no conhecimento e base do agente.

O segundo problema é que o conhecimento do agente é fixo, assim, não possibilitando personalizá-lo de acordo com as preferências de cada usuários.

Outro problema relacionado a essa abordagem é a confiança, pois o usuário pode perceber uma perda de controle e de compreensão de sua atividade à medida que é fornecido um agente qualificado, sofisticado e autônomo desde o início do seu uso.

4.4 Propriedades e características do agente de detecção de gargalos RDSamurai

Para a implementação do agente de detecção de gargalo serão adotadas algumas das características dos agentes. Como destacada na Figura 4, o agente terá maior caracterização de seus atributos primários de autonomia e adaptação.

- **Autonomia:** O agente necessita de autonomia para realizar as suas ações de coleta e processamento de informação, desligando um agente humano da tarefa a ser realizada. Os administradores apenas orientarão o agente para que o mesmo atinja seus objetivos;

- **Adaptação:** O agente poderá criar novos parâmetros para seu funcionamento e questionará o administrador sobre a efetivação da mudança. Isso é muito importante no caso dos parâmetros estarem superdimensionados.
- **Cooperação:** Como o agente apresentado é independente de outros sistemas ou agentes, a necessidade de cooperação entre agentes não será necessária, tornando apenas necessária a cooperação com um agente humano, no caso o administrador do sistema.

Classificando o agente conforme a tarefa que o mesmo irá realizar teremos destaque na caracterização do mesmo como:

- **Agente de Informação:** o mesmo irá recolher informação do sistema requerido pelos administradores;
- **Agente de Aconselhamento:** o agente irá filtrar e analisar as informações do sistema e apresentará as seções do sistema que necessitam de melhorias;
- **Agente Assistente:** executará, para os administradores, as ações de coleta e análise das informações;
- **Agente de Interface:** o agente interagirá com agentes humanos (administradores do sistema) para troca de informação.

Capítulo 5

5 Proposta do Agente RDSamurai

A proposta do projeto consiste no uso de Agentes na detecção de gargalos de um Sistema Web.

A seguir será explanado sobre os itens de detecção de gargalos, aplicação de agentes na solução do problema, dados a serem coletados, configurações de parâmetros e análise dos resultados e tomada de decisões.

5.1 Detecção de gargalos

Os gargalos do sistema cobertos pelo agente RDSamurai serão os gargalos de software, os quais afetam o desempenho do sistema como um todo, e sobrecarregam geralmente um dos subsistemas de hardware causando uma avaliação incorreta da capacidade do servidor em que a aplicação está rodando.

Os dados que serão observados serão os de tempo de processamento de cada seção (compreende-se por seção cada página mostrada ao usuário) e o número de acessos àquela seção, a partir dos dados.

- **Tempo de processamento:** O tempo de processamento de um módulo de um sistema implica diretamente no desempenho do sistema, pois quanto maior o tempo de processamento menor será a capacidade de vazão do sistema;
- **Número de acessos:** O número de acesso em um sistema Web a uma determinada seção caracteriza um gargalo, pelo acesso concorrente a um recurso do sistema. Assim essa seção precisa ter o melhor desempenho possível.

5.2 Aplicação do agentes na solução do problema

A aplicabilidade do agente para a detecção de gargalos surge na medida de sua operabilidade. Caracterizando o agente como um software, podemos citar algumas propriedades que ele apresenta, como:

- **Utiliza pouco recurso do sistema:** Pois não adianta um software para detecção de gargalos se aumentar o gargalo do sistema pela sua complexidade;
- **Contínua execução:** O sistema deve ser monitorado constantemente, tornando-se impraticável pela monitoração dos administradores;
- **Autônomos:** Não necessita de intervenção humana, durante sua execução.

5.3 Implementação

5.3.1 Modelagem

O diagrama da Figura 5 demonstra as ações dos atores no cenário de funcionamento do sistema:

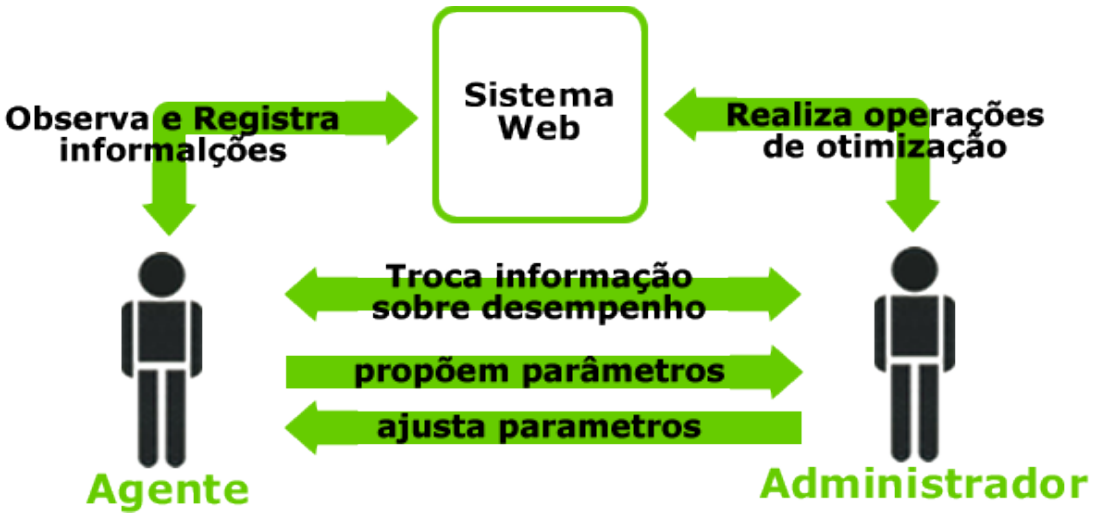

Figura 5: Modelagem do agente proposto

- O RDSamurai observará o sistema Web para obter informações necessárias e as registrará para futuros processamentos.

- O agente processará os dados adquiridos e, caso seja detectada a necessidade de ajustes nos parâmetros de configurações do Agente (os parâmetros de configuração indicam para o agente os seus objetivos e as configurações necessárias para alcançá-los), o mesmo informará o administrador. Caso seja detectado algum possível gargalo, através dos parâmetros definidos, o agente informará o administrador da possível necessidade de otimização do sistema. Caso o agente receba do administrador um novo parâmetro de operação o mesmo ajustará os seus parâmetros para se adequar às novas informações recebidas.
- O Administrador, ao receber uma proposta de parâmetros do agente, analisará a proposta e, caso a mesma esteja coerente, enviará a mensagem de aceitação para o agente; caso contrário, a mesma será rejeitada.
- O Administrador ao receber uma informação sobre possível gargalo do sistema estudará medidas a serem tomadas para melhoria e otimização do mesmo.
- O Administrador, após decidir qual medida de otimização será adotada, atualizará o sistema na tentativa de melhorar o desempenho na seção especificada pelo Agente.

5.3.2 Linguagem

A linguagem PHP [13] será utilizada para a implementação do agente, pelas características a seguir:

- Portável;
- Fácil interação com o sistema do jogo;
- Suporte de acesso a vários bancos de dados;
- Dinâmica;
- Flexível;

5.4 Dados a serem coletados e Configuração de parâmetros

Os Dados a serem coletados são a princípio: tempo de processamento e quantidade de requisições de cada seção.

Os parâmetros do agente serão configurados de acordo com as respostas do sistema, isto é, caso os parâmetros de configuração divirjam dos especificados, o agente tentará propor uma solução. Caso ele consiga uma proposta, essa será encaminhada para a equipe de desenvolvimento para análise e aceitação da mesma. Caso ele não consiga propor, a equipe terá que ajustar os parâmetros de acordo com os dados que o agente obteve.

5.5 Análise dos resultados e tomada de decisões

A partir dos resultados obtidos pelo agente, os mesmos serão analisados pela equipe de desenvolvimento e os administradores tomarão as decisões do que pode ser melhorado e otimizado. As otimizações serão testadas em um servidor local para testar o funcionamento, após um período de testes será feito *upload* para o servidor do sistema.

Capítulo 6

6 Projeto e implementação do RDSamurai

A implementação do RDSamurai foi dada em etapas de acordo com as especificações da modelagem e conforme a necessidade de novos componentes para agregar funcionalidades ao projeto.

6.1 Modelagem da Solução

O sistema do agente foi desenvolvido de maneira a proporcionar uma fácil modificação de configuração e funcionalidades, proporcionando uma ferramenta de fácil modificação e utilização pela equipe de desenvolvimento. Este capítulo tratará a forma pela qual foi implementado o sistema, as principais características, funcionalidade e dificuldades encontradas na implementação.

Os principais componentes do agente são caracterizados pela Figura 6 abaixo:

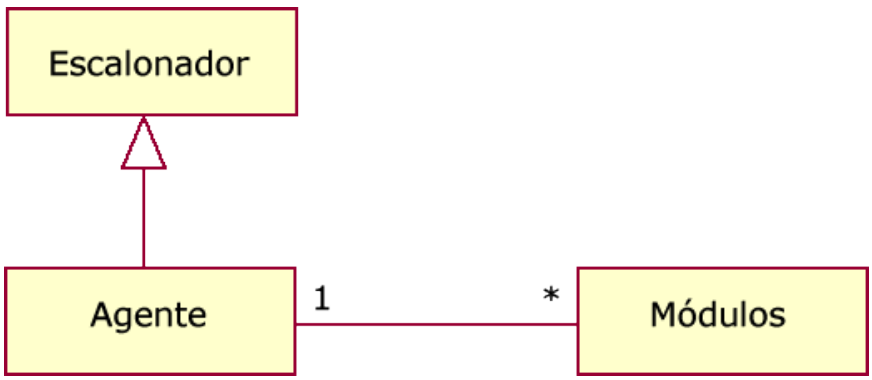

Figura 6: Modelagem básica do RDSamurai

Esses componentes são responsáveis pelo funcionamento básico do agente. Cada um deles será detalhando posteriormente e abaixo é apresentado um breve relato de seu funcionamento:

- **Escalonador:** Responsável pelo escalonamento[7] dos módulos que o RDSamurai está executando, compartilhando o tempo de execução do Agente entre os módulos instalados;
- **Agente:** Responsável pela adição, remoção e execução dos módulos agregados;
- **Módulos:** Responsável pela atribuição das funções ao RDSamurai.

6.2 Diagrama de Classes

6.2.1 Diagrama de Classes do Servidor de Resultados

A Figura 7, abaixo, o diagrama de classes do servidor de resultados. O servidor de resultados utiliza conexões por sockets para realizar as operações relacionadas as operações de gravação e consulta das informações.

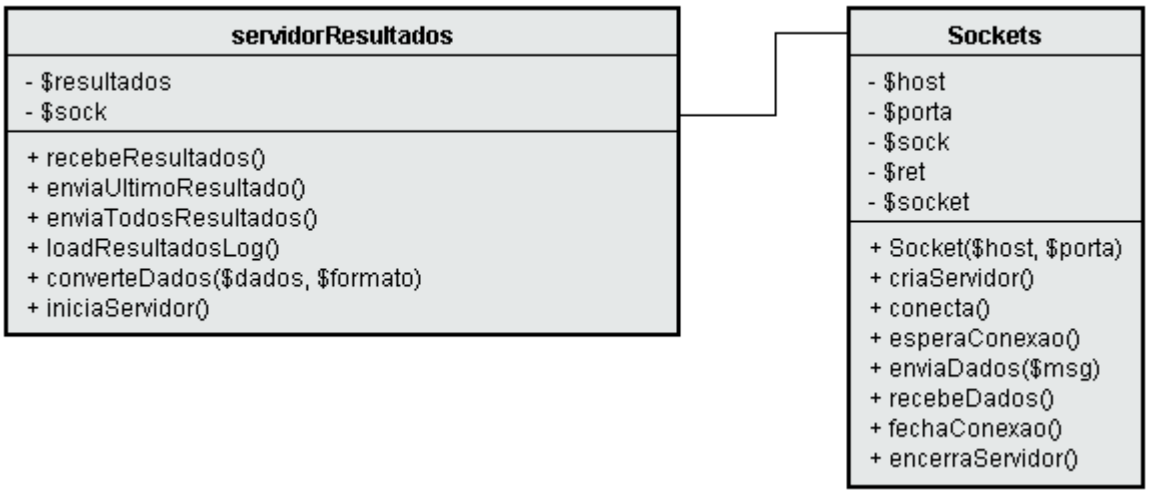

Figura 7: Digrama de classes do Servidor de Resultados

6.2.2 Diagrama de Classes do Agente

A Figura 8, o diagrama de classes do servidor de resultados. O servidor de resultados utiliza conexões por sockets para realizar as operações relacionadas as operações de gravação e consulta das informações.

[7]

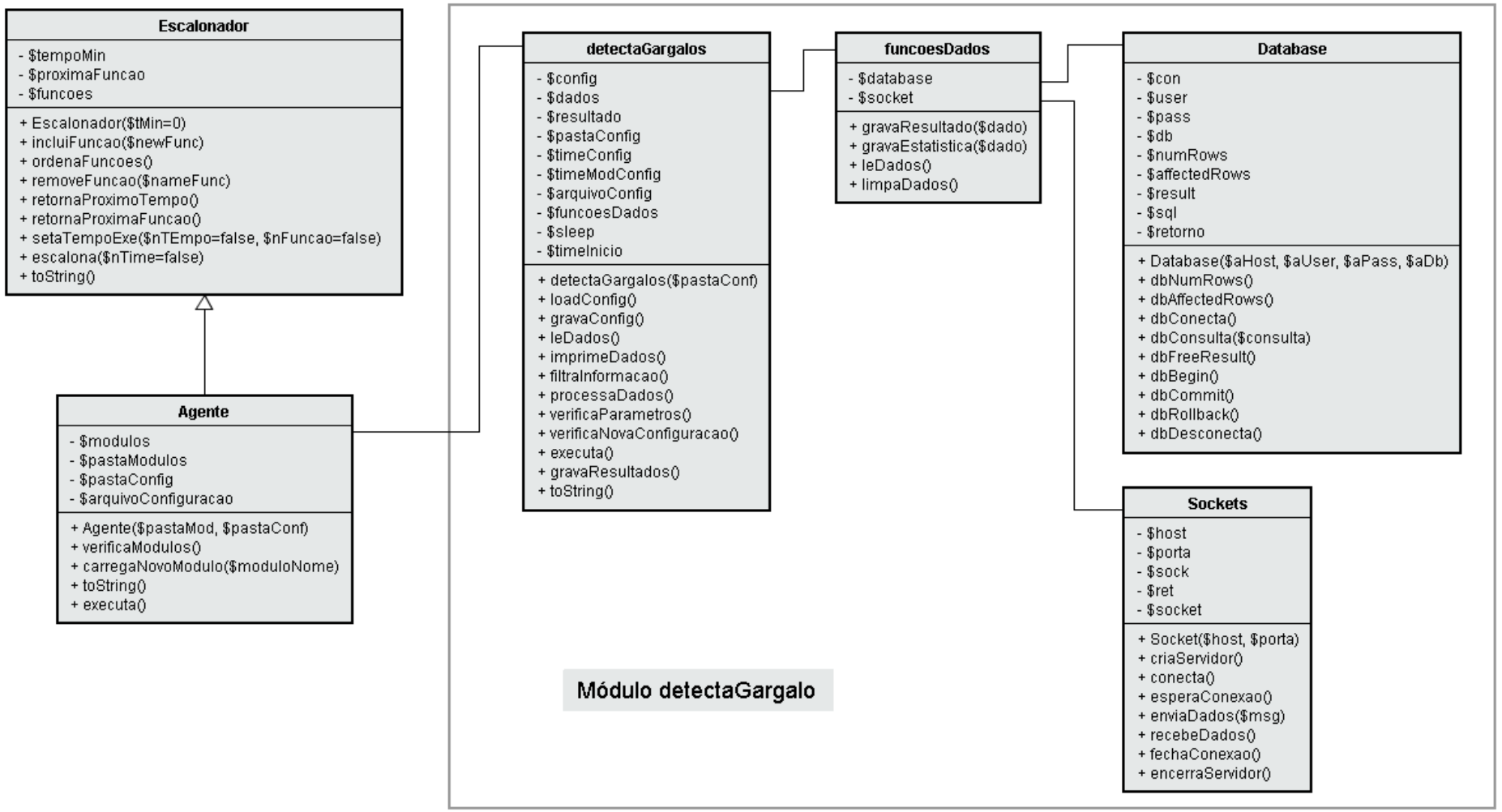

Figura 8: Digrama de classes do RDSamurai

6.3 Classe Agente

O Agente é o módulo central do sistema. O mesmo proporciona as funções de adição, remoção e execução de módulos.

No método `Agente()` (método construtor da classe Agente) são definidos alguns parâmetros para as configurações como o diretório dos módulos (o qual mantém os arquivos dos módulos a serem executados pelo Agente), o diretório de arquivos de configuração (o qual mantém os arquivos de configuração dos módulos e de configuração do Agente) e o arquivo de configuração do Agente.

A adição e remoção de um novo módulo são feitas através da edição do arquivo de configuração do Agente, o qual possui em cada linha o nome do módulo a ser carregado. Assim após a edição das configurações o próprio agente irá verificar a mudança no arquivo e fará as devidas ações de remoção do módulo do Agente ou a adição de um novo módulo.

A execução dos módulos se dá através do retorno do módulo pelo Escalonador a ser executado e chamada do método `executa()` do módulo correspondente.

6.4 Classe Escalonador

A falta de suporte de fluxos de execução (*Threads*) da linguagem PHP acarretou a criação a criação de um escalonador que compartilhe o tempo de execução do Agente entre os módulos instalados. O funcionamento é bem simples, porém supre as necessidades do sistema de "dormir" e "acordar" os módulos de tempos em tempos.

Os métodos ou funções a serem chamadas são cadastradas no escalonador através do método `incluiFuncao($newFunc)` o qual é responsável por colocar a função em uma fila de escalonamento. O Escalonador é responsável por interromper o Agente (*Sleep*) e restaurar sua execução (*Wakeup*). O Escalonador retorna para o Agente qual módulo pediu para ser acordado através do método `escalona()`.

6.5 Módulos do RDSamurai

Os Módulos correspondem às funcionalidades do RDSamurai, cada módulo é responsável por executar a sua tarefa a qual independe de outros módulos. O Agente pode

possuir diversos módulos, porém deve-se tomar cuidado com o acréscimo dos mesmos pois podem sobrecarregar o Agente, fazendo que o mesmo utilize demasiadamente os recursos do sistema.

Para a adição de um novo módulo ao RDSamurai deve-se levar em consideração alguns pré-requisitos, são eles:

- O nome da classe deve ser idêntico ao nome do módulo;
- O nome do arquivo do módulo deve possuir o mesmo nome do módulo, acrescido da terminação "`.php`" e deve ser gravado no diretório de módulos do RDSamurai. O arquivo de módulo dever conter o código programacional da funcionalidade a ser acrescentada ao RDSamurai;
- O módulo deve possuir um arquivo de configuração com o mesmo nome do módulo, acrescido da terminação "`.conf`" e deve ser gravado no diretório correspondente aos arquivos de configuração do Agente;
- A classe deve possuir um método `executar()`, o qual deve retornar o tempo em segundos em que deseja ser executado novamente.

6.5.1 Arquivo de Configuração

Os módulos são subsistemas configuráveis a quente, isso é, não existe a necessidade de parar o RDSamurai para propor novas configurações. Ao realizar a mudança no arquivo de configuração o próprio módulo detecta a mudança e carrega os novos parâmetros do sistema.

6.6 Módulo de detecção de gargalos (`detectaGargalos`)

Para a detecção de gargalos foi adicionado ao RDSamurai um único módulo denominado `detectaGargalos`. O módulo detectaGargalos, assume o papel de recebimento e processamento dos dados provenientes do sistema e gravação dos resultados.

A coleta dos dados se dá através de uma classe `funcoesDados`, a qual tem a responsabilidade de coletar os dados provenientes do sistema e entregá-los para o módulo de detecção de gargalos. Essa classe também é responsável pela gravação das informações no Servidor de Resultados, o qual será tratado posteriormente, trabalhadas pelo módulo para futuro acesso e análise pela equipe de desenvolvimento.

O tratamento dos dados se dá de forma parcial, isso é, de tempos em tempos o módulo é executado e coleta os dados que estão disponíveis para análise. Esses dados são processados e guardados em um buffer interno. O critério de parada para gravação é configurada pelo administrador e se dá de duas formas:

- **Por número de amostras:** Cada acesso a uma seção (no caso de um sistema Web uma seção foi definida como uma página de interface com o usuário) caracteriza uma amostra. Ao atingir um determinado número de amostras pré-definido no arquivo de configuração pela variável `amostras` o módulo gerará os resultados finais e gravará no servidor de dados o qual será tratado posteriormente;
- **Periodicamente:** Ao atingir um determinado período de tempo pré-definido no arquivo de configuração pela variável `amostras_time` o módulo gerará os resultados finais e gravará no servidor de dados.

Análise do tempo de processamento médio de cada seção

Essa análise é feita baseada na média do tempo que cada seção demora a ser processada, o resultado é comparado com um valor pré-definido de (esses valores de configuração foram determinado através de pré-testes do Agente no sistema Web) configuração do módulo de configuração que indica se a média do tempo de processamento ultrapassou o limite pré-especificado pela variável `parametroTPGargalo` do arquivo de configuração do módulo. Caso o mesmo tenha ultrapassado o resultado é destacado pela cor definida em parâmetro `TPMaximoColor` e caso contrário o resultado é marcado com a cor definida pela variável parâmetro `TPNormalColor`. Conforme a Tabela 2 abaixo:

Tabela 2 - Parâmetros de configuração do Tempo de Processamento do RDSamurai

Parâmetro	Valor	Descrição
`parametroTPGargalo`	0.75 s	Parâmetro de configuração do tempo de separação entre o esperado e o gargalo, abaixo ou igual a esse valor se aceita por um tempo de processamento aceitável ou normal, e acima desse tempo configura-se um gargalo ao sistema

Parâmetro	Valor	Descrição
parametroTPNormalColor	white	Parâmetro que indica como será destacado o resultado para um tempo normal de processamento. Todos os parâmetros de cores podem ser escritos no formato de cores do html [10].
parametroTPMaximoColor	red	Parâmetro que indica como será destacado o resultado para um tempo além do tempo de processamento especificado pelo parametroTPGargalo.

Análise do número de acesso médio de cada seção

Essa análise é realizada baseada na percentagem relativa à média esperada[8] do número de acesso a uma determinada seção. Foram definidos cinco níveis de acesso para especificar o grau de acesso a uma determinada seção, esses níveis são definidos pelos parâmetros parametroAcessoX aonde o X varia dos valores zero à quatro. E existem cores correspondentes para os destaques por cores para esses parâmetros as quais são definidas pelos parâmetros parametroA1colorX os quais o X também apresentam valores variando de zero a quatro. Conforme a Tabela 3 abaixo:

Tabela 3 - Parâmetros de configuração da funcionalidade de Acesso do RDSamurai

Parâmetro	Valor	Descrição
parametroAcesso0	0.05 ou 5%	Parâmetro de tempo de processamento normal ou esperado que é definido como 5% acima do tempo médio esperado.
parametroAcesso1	0.1 ou 10%	Parâmetro da primeira faixa de tempo de processamento, que corresponde entre 5% a 10% do tempo médio esperado. Esse valor refere-se a uma faixa de acesso considerada normal.

[8] Definiu-se por média esperada a média do número de acesso às seções do sistema. Essa média é recalculada a cada resultado gerado pelo Agente.

Parâmetro	Valor	Descrição
parametroAcesso2	0.2 ou 20%	Parâmetro da segunda faixa de tempo de processamento, que corresponde entre 10% a 20% do tempo médio esperado. Esse se refere a um valor acima do normal mas ainda não é um valor preocupante.
parametroAcesso3	0.3 ou 30%	Parâmetro da primeira faixa de tempo de processamento, que corresponde entre 20% a 30% do tempo médio esperado. Refere-se a uma faixa acima do normal a qual deve ser mantida sobre monitoramento constante.
parametroAcesso4	0.4 ou 40%	Parâmetro da primeira faixa de tempo de processamento, que corresponde entre 30% a 40% do tempo médio esperado. Essa faixa e acima dela são valores muito preocupantes e representam ótimas regiões para serem otimizadas, pois qualquer ganho em tempo de processamento representa um ganho real total de processamento final.
parametroA1color0	white	Parâmetro referente a cor de destaque correspondente à faixa normal do parâmetro de acesso. Corresponde a cor branca na definição HTML de cores.
parametroA1color1	#90ff90	Parâmetro referente a cor de destaque correspondente a primeira faixa do parâmetro de acesso. Corresponde a cor verde na definição HTML de cores.
parametroA1color2	yellow	Parâmetro referente a cor de destaque correspondente a segunda faixa do parâmetro de acesso. Corresponde a coloração amarelo na definição HTML de cores.

Parâmetro	Valor	Descrição
parametroA1color3	orange	Parâmetro referente a cor de destaque correspondente a terceira faixa do parâmetro de acesso. Corresponde a coloração branca na definição HTML de cores.
parametroA1color4	red	Parâmetro referente a cor de destaque correspondente a quarta faixa do parâmetro de acesso. Corresponde a coloração branca na definição HTML de cores.

A informação proveniente desse processamento dos dados é então gravada em um servidor de dados a partir da escolha do desenvolvedor de qual método ele gostaria de utilizar: por tempo ou por amostras.

6.6.1 Configurações do módulo de detecção de gargalos

A seguir, na Tabela 4, são mostrados alguns parâmetros a serem inicializados pelo arquivo de configuração do módulo de detecção de gargalo, as quais servem para orientar o RDSamurai na realização do seu trabalho.

Tabela 4 - Parâmetros de configuração mais utilizados no RDSamurai

Parâmetro	Descrição
Parâmetros responsáveis pela faixa de atuação do Agente para a análise do Tempo de Processamento das seções	
parametroTPMinimo	tempo de processamento mínimo em segundos
parametroTPMaximo	tempo de processamento máximo em segundos
Parâmetro de gargalo para a análise do Tempo de Processamento	
parametroTPGargalo	gargalo do Tempo de Processamento em segundos
Parâmetros de cores para as faixas do nível de processamento das seções	
parametroTPNormalColor	Cor para faixa normal no formato de cores do HTML [10], abaixo do parametroTPGargalo
parametroTPMaximoColor	cor para faixa crítica no formato de cores do HTML [10], acima do parametroTPGargalo

Parâmetro	Descrição
Parâmetros de controle de execução do módulo	
consultaInicio	Numero de amostras desejada por execução do módulo. É responsável por indicar quantas amostras serão coletadas em média por execução do Agente.
amostras_modo	Modo como serão gravados os resultados: por tempo (amostras_time) ou por amostras (amostras_time).
amostras_time	Define a gravação dos resultados por um período de tempo em segundos
amostras	Define a gravação dos resultados por número de amostras
Parâmetros responsáveis pela faixa de atuação do Agente para a análise do Quantidade de Acessos das seções	
parametroAcessoX	Parâmetro de acesso em índice percentual de acesso além da média desejada de acessos. X como caractere curinga variando de 0 a 4.
parametroA1colorX	Parâmetro de cores das faixas de acesso. X de 0 a 4

6.7 Servidor de Resultados

O servidor de resultados é um sistema independente e separado do Agente criado para o armazenamento das informações provenientes dos módulos do RDSamurai, o seu funcionamento consiste em um simples servidor que atende por uma porta definida pelo administrador (por padrão é utilizada a porta 5050). As operações que o mesmo suporta é de gravação e leitura de resultados.

Ao receber a solicitação de gravação o servidor a efetua de duas maneiras: primeiro a gravação em disco de um log para manter um histórico dos resultados obtidos anteriormente, e segundo a gravação em memória dos resultados obtidos. Para a gravação em memória são mantidos os últimos cinco dados recebidos pelo servidor, esse é utilizado para manter um histórico em memória para a solicitação dos últimos resultados gerados que será uma nova funcionalidade a ser implementada posteriormente para facilitar a análise pelos Administradores.

A solicitação de leitura dos resultados pode ser requerida de duas formas:

- Vetor na linguagem PHP: essa requisição retorna um Array associativo multidimensional o qual é acessível somente para scripts da linguagem php;
- Vetor em XML: esse tipo de requisição visa a utilização dos resultados por sistemas e linguagens diferentes do php. Isso torna os resultados mais acessíveis a outros tipos de softwares que possam vir a ser desenvolvidos com a intenção de visualização dos resultados.

Mais funcionalidades podem ser adicionadas ao Servidor de Dados conforme as necessidades do sistema. Uma das funcionalidades adicionadas foi a de gravação em arquivo dos resultados recebidos pelo RDSamurai e a restauração dos últimos resultados recebidos ao reiniciar o serviço do Servidor de Dados. Essas necessidades foram percebidas no momento em que foi solicitado um resultado e o servidor de dados o havia perdido após uma parada crítica por erro programacional. Uma próxima funcionalidade será a busca por resultados antigos gravados em disco organizados por data de recebimento para solicitação do cliente para a exibição de um resultado específico conforme a necessidade do utilizador.

Capítulo 7

7 Resultados Obtidos

Nesse cápitulo serão apresentados os resultados gerados pelo RDSamurai.

Segue uma descrição do relatório obtido, todos os campos serão explanados para uma melhor compreensão dos resultados. Que serão apresentados na Figura 9.

- **Data:** corresponde a data e hora que foi gerado o relatório. O presente relatório foi gerado no dia 30 de novembro de 2005 às 19h, 04min e 11s. Esse período corresponde a um período destinado a testes entre eras, o sistema estava em média com 400 usuários on-line os quais participaram dos testes junto com a equipe de desenvolvimento;
- **Estatística:** Corresponde aos dados estatísticos das amostras coletadas;
- **tempo Médio de Processamento:** corresponde ao tempo médio de processamento de todas as seções do sistema web. Esse resultado é gerado através da média simples do tempo total de processamento de cada seção e quanto menor o valor encontrado significa que as seções estão sendo processadas mais rapidamente e implica em um aumento na capacidade de requisições para o sistema.
- **mediaAcessoDesejado:** A média de acesso desejado significa a distribuição igualitária dos acessos às seções. Esse é um parâmetro de referencia para verificarmos o desvio do número de acessos que uma determinada seção está recebendo. A média de acesso é o quociente da divisão do número de Amostras (N Amostras) pelo número de seções (n_secoes), como mostrado na equação a seguir:

$$mediaAcessoDesejado = \frac{N\_Amostras}{n\_\sec oes}$$

- **N Amostras:** Corresponde ao número de amostras analisadas, esse dado refere-se a quantia de acessos que foram analisados. O RDSamurai, primeiramente, foi desenvolvido para gerar as análises a partir de um determinado número de amostras, porém constatou-se a necessidade de podermos prever quando seria gerado uma nova análise para que os desenvolvedores pudessem acompanhar o andamento do sistema.

Assim modificou-se o módulo de detecção de gargalos para que o mesmo fosse capaz de gerar análises periodicamente facilitando o trabalho dos desenvolvedores.

- **n_secoes:** Corresponde ao número de seções que foram acessadas pelos jogadores e analisadas pelo RDSamurai, cada seção corresponde a uma página de interface da aplicação com o usuário;
- **secoes do TP(Tempo de Processamento):** Corresponde a uma análise rápida sobre o Tempo de Processamento das seções. O dado que é referido por '**passou'** corresponde a quantas seções passaram do valor pré-estabelecido no parâmetro de configuração do módulo de detecção de gargalos identificado por `parametroTPGargalo` e o resultado referenciado por **'nao_Passou'** indica quantas seções estão abaixo desse valor.
- **Coluna Seções:** Corresponde as seções que foram analisadas. As seções são identificadas pelo nome da página acessada.
- **Coluna nAcessos:** Corresponde aos resultados obtidos referentes a análise do número de acessos em cada seção, segue um exemplo para explicarmos os valores representados: '1581 / 50077 (42.07%)' o primeiro valor (1581) indica quantas vezes a seção foi acessada pelos usuários comparada com o segundo valor (50077) que corresponde ao total de amostras de acessos e por fim o terceiro valor (42.07%) indica qual foi o desvio em relação a média de acessos desejado que para esse resultado é de 1112.82, esse resultado representa o quanto, além do valor esperado de acesso, a seção em análise passou. Os resultados são destacados por um código de cores especificadas no capítulo 6.5.
- **Coluna tempoMProcessamento:** Corresponde ao tempo médio de processamento para aquela seção. Os resultados são destacados por um código de cores especificadas no capítulo 6.5; o valor é resultado da média do tempo de processamento para cada seção e é calculado a partir da divisão do somatório dos tempos de processamento pelo número de seções, como demonstra a equação a seguir:

$$tempoM\,\mathrm{Pr}ocessamento = \frac{\sum_i tempo\_processamento\_\mathrm{sec}ao}{i}$$

7.1 Dados obtidos pelo RDSamurai de detecção de gargalos

Os resultados obtidos pelo agente podem ser analisados a partir da Figura 9 adquirida pela interface do RDSamurai[9]:

30/11/05 às 19:04:11

Estatística

tempo Medio de Processamento	0.52	
mediaAcessoDesejado	1112.82	
N Amostras	50077	
n_secoes	45	
secoes do TP	**passou**	8
	nao_Passou	37

Seções	nAcessos	tempoMProcessamento
altera_dados.php	25 / 50077 (-97.75%)	0.61
altera_email.php	8 / 50077 (-99.28%)	0.25
altera_senha.php	60 / 50077 (-94.61%)	0.25
denuncias.php	5 / 50077 (-99.55%)	0.15
em_beta.php	46 / 50077 (-95.87%)	0.21
forum_interno.php	283 / 50077 (-74.57%)	0.39
forum_interno_envia_msg_grupo.php	54 / 50077 (-95.15%)	0.64
forum_interno_envia_msg_individual.php	475 / 50077 (-57.32%)	0.62
forum_interno_msg_grupos.php	1581 / 50077 (42.07%)	1.02
forum_interno_msg_individuais.php	290 / 50077 (-73.94%)	0.90
index.php	5024 / 50077 (351.46%)	0.75
login.php	942 / 50077 (-15.35%)	0.21
noticias.php	4 / 50077 (-99.64%)	0.13
origem_zero.php	17 / 50077 (-98.47%)	0.85
premiacao.php	63 / 50077 (-94.34%)	0.19

[9] Pode ser acessada em: http://www.ryudragon.com/agente/

problemas_senha.php	22 / 50077 (-98.02%)	0.16
publicidade.php	25 / 50077 (-97.75%)	0.34
ranking.php	4803 / 50077 (331.61%)	1.27
ranking_clas.php	563 / 50077 (-49.41%)	0.62
reenvio.php	93 / 50077 (-91.64%)	0.21
registro.php	645 / 50077 (-42.04%)	0.95
regras.php	7 / 50077 (-99.37%)	0.56
reportar_bugs.php	26 / 50077 (-97.66%)	0.25
sentinela.php	73 / 50077 (-93.44%)	0.35
suporte.php	14 / 50077 (-98.74%)	0.34
tatsujin.php	63 / 50077 (-94.34%)	0.27
teste_load.php	2 / 50077 (-99.82%)	0.09
usercerco.php	149 / 50077 (-86.61%)	0.60
usercla.php	501 / 50077 (-54.98%)	0.68
usercla_publico.php	209 / 50077 (-81.22%)	0.59
usercomercio.php	1183 / 50077 (6.31%)	0.57
userdragao.php	32 / 50077 (-97.12%)	0.48
userespionagem.php	1681 / 50077 (51.06%)	0.71
userespionagem_ocorridas.php	342 / 50077 (-69.27%)	0.61
userextras.php	61 / 50077 (-94.52%)	0.57
userfeudo.php	6451 / 50077 (479.70%)	0.60
userheroi.php	267 / 50077 (-76.01%)	0.39
userhistoricofeudo.php	1234 / 50077 (10.89%)	1.09
usermanual.php	121 / 50077 (-89.13%)	0.81
userprofile.php	1053 / 50077 (-5.38%)	0.46
userrelatoriotropas.php	4129 / 50077 (271.04%)	0.46
userresumo.php	7503 / 50077 (574.23%)	0.83
usertropas.php	766 / 50077 (-31.17%)	0.56
userunidades.php	9174 / 50077 (724.39%)	0.69
versao.php	8 / 50077 (-99.28%)	0.14

Figura 9: Resultados obtidos pelo agente RDSamurai.

7.2 Interpretando as Informações Obtidas Pelo RDSamurai

Nessa seção será abordada a interpretação das informações obtidas no resultado previamente citado. Analisaremos as informações retornadas pelo agente e a importância dessas para encontrar-se as possíveis soluções para os problemas de gargalo do sistema.

A metodologia a ser utilizada obedecerá a seguinte ordem: será feita uma análise das amostras coletadas, dos resultados gerados pela quantidade de acessos às seções do sistema e, por fim, uma análise dos resultados dos tempos de processamento das seções do sistema.

7.2.1 Análise das Amostras e Tempo de Coleta:

O conjunto de resultados foi obtido a partir da análise de 50.077 amostras. As amostras foram geradas em um período aproximado de três horas isso corresponde a uma média de 278 acessos por minuto.

Uma análise simples pode ser feita para verificar-se que o servidor web mantém mais de um processo na fila de execução. Como o servidor processa 4,63 páginas por segundo (análise feita pelo número de amostras pelo tempo de coleta), e cada página leva em média 0,52 segundos, isso é, 1,92 páginas por segundo, isso representa uma fila de 2,71 páginas a cada segundo em espera para ser processada. Considerando o paralelismo de execução o servidor consegue servir as necessidades, porém aumentando o tempo de resposta do sistema para os usuários.

Para diminuirmos essa sobrecarga é de grande importância diminuirmos o tempo de processamento das páginas que se encontram sobrecarregadas, ou pelo número de acesso ou pelo tempo de processamento.

7.2.2 Análise dos Resultados do Número de Acesso das Seções

A análise do número de acessos a uma determinada seção é importante para analisar o comprometimento com o desempenho do sistema pelo número de acessos à mesma. Uma seção que seja muito acessada, mesmo possuindo um tempo de processamento aceitável ou normal, pode representar um gargalo, pois a mesma está com um nível de acesso elevado e qualquer tempo de processamento economizado para essa seção é de grande importância no desempenho geral do sistema. Como demonstrado no exemplo a seguir:

userunidades.php

tempo de processamento: 0,69 s;

número de acessos: 9174

tempo de uso do servidor: 6330,06 s

Caso consigamos reduzir em 0,05s o tempo de processamento da seção userunidades.php para o mesmo número de acessos teríamos o tempo de uso do servidor de:

tempo de uso do servidor: 5871,36

Representando uma economia de 458,7 s, no qual o servidor poderá estar alocando os seus recursos para outras requisições dos usuários.

Análise de algumas seções através dos resultados obtidos:

- `forum_interno_msg_grupos.php (42.07%)`: Seção responsável pelo sistema de mensagens internas do grupo no jogo. A quantia de acessos é considerada alta, ultrapassou a ultima faixa definida pelo gargalo de acessos e chegou ao valor de 42,07% além do valor esperado de 1113 acessos por seção. O número de acessos pode ser diminuído criando mecanismo de marcação de tópicos lido;
- `index.php (351.46%)`: Página inicial do sistema, todos os usuários passam por ela para acessar o jogo, e portanto implica em um grande número de acesso da mesma. Pode ser otimizada diminuindo o tempo de processamento sendo gerada uma página estática e atualizada periodicamente. Mantendo o seu caráter informativo, através da publicação de notícias referentes ao jogo, e de interface inicial com o usuário;
- `ranking.php (331.61%)`: Corresponde a página de colocação dos jogadores. É uma das seções mais importantes do sistema pois é a partir dela que são realizadas muitas das ações de jogo como: espionar, atacar, mandar mensagens, ver informações e convidar o jogador para fazer parte do clã. Essa página representa a parte central de todo o sistema e é fundamental para a realizações de ações importantes no jogo. A mesma passou por várias otimizações: como reduções de consultas ao banco de dados e simplificação dos resultados obtidos, porém ainda é uma parte do sistema que deve estar em constante monitoramento pois além de conter muitos acessos, a mesma possui um tempo de processamento elevado, como veremos na análise do tempo de processamento das seções;
- `userresumo.php (574.23%)`: Página principal do jogo, aonde contém um resumo da situação atual de todo o feudo do jogador. Essa página é uma das mais

acessada pelo sistema e corresponde a 15% dos acessos, é portanto um ponto fundamental para análise e cuidados com desempenho do sistema.

- `userunidades.php (724.39%):` Essa seção foi uma surpresa para a equipe de desenvolvimento pois essa página é responsável pelo treinamento de unidades e não era esperado tantos acessos à mesma, cerca de 18% dos acessos totais, isso demonstra o valor do uso do RDSamurai para monitoramento de acesso do sistema, pois essa seção não teria prioridades para otimizações do sistema.

7.2.3 Análise dos Resultados do Tempo de Processamento das Seções

A análise do tempo de processamento das seções é muito importante para visualizar em que está sendo despendido o uso do servidor. Uma página que leve muito tempo para ser processada pode acarretar no aumento na fila de espera de processamento do servidor o que implica na sobrecarga do mesmo. Essa sobrecarga pode ser agravada caso a quantidade de acessos àquela determinada seção seja alto, pois a página, além de exigir em tempo de processamento, for muito acessada, pode afetar a estabilidade do servidor em função da sobrecarga do sistema.

Análise de algumas seções através dos resultados obtidos:

- `forum_interno_msg_grupos.php (1.02):` A página do sistema de mensagens de grupo não era esperada como sendo uma página problemática, mas se demonstrou responsável também pelos acessos ao sistema, pois a mesma apresenta um tempo de processamento de pouco mais de um segundo;
- `ranking.php (1.27 s):` A página de *ranking* é novamente apontada como uma responsável direta no desempenho do sistema, a análise implica em um tempo de processamento médio de 1,27 segundos para cada acesso à seção. A página de ranking também não era esperada como gargalo na análise pelo tempo de processamento pela equipe de desenvolvimento, pois já havia passado por várias otimizações de código e reduções de consultas;
- `userhistoricofeudo.php (1.09):` página responsável por mostrar o histórico das ações do jogador no feudo, por esse motivo possui grande número de acessos ao banco de dados. Essa seção apresente grande potencial para ser otimizada com a redução de consultas ao banco de dados e otimização do código.

7.2.4 Análise Global dos Resultados

Uma análise global dos resultados obtidos foi feita para possibilitar o desenvolvimento de um plano de otimizações. Será destacada a página pela combinação dos resultados da análise do número de acessos e do tempo médio de processamento de cada seção.

Essa análise é importantíssima para a geração do plano de otimização. Através do produto do tempo médio de processamento e a quantia de acesso a cada seção encontra-se o tempo total de uso do servidor designado ao processamento da mesma.

Conhecendo todos esses valores pode ser calculada a percentagem do uso do tempo de processador por seção, como pode-se ver os resultados mais importantes gerados na Tabela 5 a seguir:

Tabela 5 - Análise Geral dos resultados obtidos pelo RDSamurai.

Seções	**nAcessos**	**tempoMProcessamento**	**uso do servidor pela seção**
userunidades.php	9174	0,69	17,12%
userresumo.php	7503	0,83	16,85%
ranking.php	4803	1,27	16,50%
userfeudo.php	6451	0,6	10,47%
index.php	5024	0,75	10,19%
userrelatoriotropas.php	4129	0,46	5,14%

As seções a serem otimizadas por ordem de prioridade são as seguintes, segundo a análise dos resultados da Tabela 5:

1. **userunidades.php:** como citado anteriormente, essa seção foi uma das surpresas para os desenvolvedores. Pois é a página mais acessada de todo o sistema e responsável por mais de 17% do uso do tempo de processamento do servidor,. Não é um gargalo pelo tempo de processamento de cada seção, mas pelo conjunto da grande quantia de acessos e o tempo de processamento, assim o mínimo tempo melhorado com uma otimização implica em uma grande redução no uso do servidor pela seção;
2. **userresumo.php:** uma das principais páginas do jogo e responsável por trazer várias informações para o jogador através do resumo de acontecimentos e do estado atual do feudo;
3. **ranking.php:** verificando a quantidade de acesso e o tempo de processamento consumido em média por acesso a página, o *ranking* está entre as páginas candidatas a ser escolhida para melhorar o desempenho do sistema, ainda mais por apresentar um tempo de processamento elevado a sua otimização pode ajudar muito no tempo de resposta das páginas acessadas, pois diminuiria o tempo na fila de execução;

Capítulo 8

8 Plano de Otimizações

Em uma reunião entre os desenvolvedores, para uma análise dos resultados obtidos pelo RDSamurai, definiu-se um plano de otimização do sistema. E Constatou-se que os resultados obtidos através do RDSamurai foram muito bons e de grande valia para orientar a equipe no processo de otimização do sistema. As seções críticas foram bem definidas pelo RDSamurai e os esforços a serem empregados nas otimizações serão nas seções que:

- Apresentarem gargalos pelo número excessivo de acesso e de processamento;
- Posteriormente às Seções em que apresentam o tempo de processamento elevado e quantia de acesso normal
- Finalmente serão tratadas as seções que apresentam um tempo de processamento satisfatório porém são muito acessadas.

As otimizações da seção passarão por uma criteriosa análise para que sejam encontradas possíveis melhorias no código do sistema e ao acesso ao banco de dados.

8.1 Otimização do Banco de Dados

A otimização de consultas ao banco de dados constituirá de melhorias nas consultas e na geração de tabelas temporárias de *cache* as quais melhorarão o desempenho das partes do sistema em que façam acesso a esses dados. Entre as melhorias de consultas ao banco de dados podemos citar:

- **Redução de consultas:** remover consultas redundantes ao banco de dados para uma mesma seção do sistema, evitando assim acesso ao sistema de arquivos e o excesso de tempo para retornar os valores já acessados previamente. Esse processo pode trazer bons resultados pois no período de projeto do sistema a equipe não tomou os devidos cuidados de analisar quais consultas poderiam ser suprimidas, o que provavelmente trará ao sistema bons resultados de otimização;
- **Substituição de consultas:** essa melhoria consiste em substituir consultas complexas por consultas mais simples e que demorem menos tempo para serem processadas, foi

analisado pelos desenvolvedores o uso de mais de uma consulta simples para realizar o mesmo trabalho que uma complexa, a qual trouxe bons resultados nos testes realizados. As consultas complexas exigem a garantia de acesso de leitura a várias tabelas simultaneamente no servidor de banco de dados, o que pode ser um processo demorado pois implica na espera pela liberação de todas as tabelas para a realização da consulta. Enquanto que nas consultas mais simplificadas não necessitam de *lock* de leitura em várias tabelas, mas sim somente na que esta sendo utilizada.

- **Criação de tabelas temporárias:** para dados que são constantemente acessados em grupos (consultas com a junção de várias tabelas), a criação de tabelas temporárias que mantenham esses dados agrupados é de extrema importância para a melhoria do desempenho nas otimizações do sistema pois evita o uso de consultas complexas com junções de várias tabelas do banco de dados através da criação de uma tabela única com todos os dados necessários, o que traz a redução de complexidade na consulta a essa tabela.

8.2 Otimização do Código da Aplicação

A aplicação possui várias bibliotecas, e para uma seção não é necessário o carregamento de todas elas, pois as partes do sistema não necessitam conhecer todas as funções ou bibliotecas do sistema. Faremos uso dessa dinamicidade da linguagem php para carregarmos em cada seção somente as bibliotecas necessárias para a realização da tarefa, evitando assim a sobrecarga de inclusão de bibliotecas pelo sistema.

Outra principal otimização é uma análise criteriosa das bibliotecas utilizadas em cada seção e a sua responsabilidade pelo desempenho geral do sistema. Serão buscados setores de código mal estruturado e que possam ser reescritos de forma a otimizar a função desempenhada pelo mesmo.

Capítulo 9

9 Conclusão

De acordo com os resultados obtidos, o RDSamurai atingiu plenamente seus objetivos de auxiliar na identificação de gargalos em um servidor Web.

Os resultados gerados pelo mesmo auxiliaram a equipe de desenvolvimento na criação de um plano de otimização do sistema. Esses resultados foram muito importantes pois foram constatadas seções do sistema nas quais a equipe de desenvolvimento não esperava encontrar um gargalo e o mesmo se mostrou eficaz na detecção desses gargalos, aonde uma equipe humana teria dificuldades de detectá-los ou levaria muito tempo para chegar aos resultados.

Ao observarem-se as características do RDSamurai, pode-se remeter o seu uso em outros segmentos de monitoramento, promovendo a continuidade do sistema e será de grande valia a ajuda prestada para os desenvolvedores, pois caso não elimine o monitoramento manual o mesmo diminuirá muito, aumentando a produtividade da equipe detendo-a em atividades de maior prioridade, além de que terá um integrante a mais nas atividades corriqueiras.

Além do valor da informação retornada pelo RDSamurai a equipe ganhou um grande conhecimento humano, pois foi necessário o trabalho em equipe na designação das tarefas de otimização para que o sistema retornasse a um estado estável. E esse trabalho em equipe é um conhecimento muito importante e valorizado no trabalho em grandes organizações pois indica o grau de maturidade profissional da equipe e principalmente dos profissionais envolvidos.

9.1 Trabalhos futuros

Pela característica de expansibilidade do RDSamurai o mesmo provê a reutilização de seu sistema para outros usos como forma independente e autônoma de execução. A equipe de desenvolvimento optou por adotá-lo em seus futuros projetos de monitoramento de usuários e a adição da funcionalidade de análise citada no capítulo 7.2.4.

9.1.1 Monitoramento de Usuários

A constante utilização de ações proibidas no sistema do jogo, de forma prejudicial pelos usuários do Sistema web, traz grandes problemas de confiabilidade do jogo pelos demais jogadores. Uma das formas que o jogador possui de trapacear no jogo é criando múltiplas contas e as atacando, criando assim um ambiente facilitado de melhorar o seu jogo sem riscos, nem perdas. O principal objetivo, discutido entre os desenvolvedores, é a criação de um módulo de detecção dos usuários que desrespeitam as regras de conduta do jogo. Facilitando a análise dos jogadores e mantendo uma maior confiabilidade ética ao jogo pelos jogadores sérios que estão dispostos a se divertir ao utilizar o sistema.

Esse módulo será implementado a partir de janeiro de 2006 e o seu funcionamento se dará pela busca por jogadores. Caso o sistema seja capaz de analisar um jogador por minuto isso implica em um rastreamento de todos os usuários a cada oito dias[10]. Isso significa agregarmos valor ao produto através da confiabilidade dos usuários no sistema e pela sua satisfação ao serviço prestado.

9.1.2 Análise da percentagem de uso do servidor

A adição da funcionalidade de análise de resultado descrita no capítulo 7.2.ao próprio RDSamurai se mostrou muito importante para o diagnostico dos gargalos do sistema. A adição de tal ferramenta ao sistema garante maior praticidade para a análise das seções a serem otimizadas e mais uma característica que pode ser analisada pela equipe de desenvolvimento para o auxilio na escolha de qual seção prender os esforços de otimização.

[10] Oito dias levando em consideração a análise de um jogador por minuto e com o número de 12.000 jogadores que estavam ativos no período de três de novembro de 2005.

10 Referências Bibliográficas

[1] ALMEIDA, Vergílio A. F. Análise e Modelagem de Desempenho de Sistemas de Computação. Disponível em: http://www2.dcc.ufmg.br/~virgilio/download/perf_aula1.pdf. Acesso em: 01/10/2005.

[2] BANCO BRADESCO S/A; O que é Segurança da Informação. Em: http://www.bradesco.com.br/br/seguranca_informacao/oquee.html. Acesso em: 28/09/2005.

[3] BOGO. Luis Henrique. CRIAÇÃO DE COMUNIDADES VIRTUAIS A PARTIR DE AGENTES INTELIGENTES: UMA APLICAÇÃO EM E-LEARNING. Mestrado em engenharia de produção, Universidade Federal de Santa Catarina, Santa Catarina, 2003.

[4] CAZARINI, E. W. Agentes Inteligentes. Artigo apresentado em português no livro Interacting with Virtual Environments, Ed. John Wiley & Sons, capítulo 15: Virtual Actors and Virtual Environments, 1994.

[5] DECADIUM; Ryudragon Descrição Formal, descrição enviada à equipe de relações publicas do Terra (www.terra.com.br), Santa Maria Agosto de 2005.

[6] HOUAISS, A.; Dicionário Houaiss da língua portuguesa. São Paulo: Objetiva, 2001.

[7] KEPLER PROJECT. Kepler: uma plataforma de desenvolvimento Web para lua. Em: http://www.keplerproject.org/wiki/BR/KeplerFAQ. Acesso em 02/10/2005

[8] Maes, Pattie; Agents that Reduce Work and Information Overload; Communication of the ACM, Julho de 1994.

[9] MOREIRA, D. A. e WALCZOWSKI, L. T. Using Software Agents to Generate VLSI Layouts; IEEE Expert Intelligent Systems, Vol. 12, No. 6, November/December 1997, Pág: 26-32

[10] Nunes, Chimène Dias; Tabela de Cores para Linguagem HTML; Disponível em: http://www.geocities.com/ensinandohtml/cores_html.htm; Acesso em: 07/12/2005;

[11] PUNK, Piter; Linux Magazine. São Paulo, Outubro 2004. Pág 68-70

[12] RIBAS, Júlio César da Costa. Acordo de nível de serviço service level agreement - sla, 2000. Disponível em: "http://www3.cefetsc.edu.br/julio/paginas/pesquisa/Adm_SLA.PDF". Acesso em 23/08/2005.

[13] SCHWENDIMAN, Blake. PHP 4: Guia do Programador. São Paulo: Ciência Moderna, 2001.

[14] WIKIPEDIA, THE FREE ENCYCLOPEDIA; Client-side scripting - Wikipedia, the free encyclopedia em: http://en.wikipedia.org/wiki/Client-side_scripting. Acesso em: 01/10/2005.

[15] WIKIPEDIA, THE FREE ENCYCLOPEDIA; Crash test - Wikipedia, the free encyclopedia. Disponível em: http://en.wikipedia.org/wiki/Crash_test. Acesso em: 05/09/2005.

[16] WIKIPEDIA, THE FREE ENCYCLOPEDIA; Server-side scripting - Wikipedia, the free encyclopedia em: http://en.wikipedia.org/wiki/Server-side_scripting. Acesso em: 01/10/2005.

[17] WOOLDRIDGE, Michael; JENNINGS, Nicholas R. Itelligent Agentes: Theory and Practice. Knowledge Engineering Review, Outubro de 1994.