CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ANA JULIA SOARES DE SOUZA

SISTEMA DE SUGESTÕES PARA CONFIGURAÇÕES DE SERVIDORES WEB APACHE.

MARILIA
2013

ANA JULIA SOARES DE SOUZA

SISTEMA DE SUGESTÕES PARA CONFIGURAÇÕES DE SERVIDORES WEB APACHE.

Trabalho de Curso apresentado ao Curso de Ciência da Computação da Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador:
Prof.Dr.Fábio Dacêncio Pereira

MARILIA
2013

Souza, Ana Julia Soares de Souza.

Sistema de Sugestões para Configurações de ServidoresWeb Apache./ Ana Julia Soares de Souza; orientador: Fabio Dacêncio Pereira. Marilia, SP:[s.n], 2013. 53f

Trabalho de Curso (Graduação em Ciência da Computação) – Curso de Ciência da Computação, Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, Marília, 2013.

1. Servidor Web 2. HTTP Server 3.Vulnerabilidades 4.Sistema de Dicas 5. Httpd.conf 6. Segurança da Informação.

CDD:005.82

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Ana Júlia Soares de Souza

Sistema de Sugestões para Configurações de Servidores Web Apache.

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 9,0 (nove)

Orientador: Fábio D. Pereira

1°. Examinador: Elvis Fusco

2°. Examinador: Rodolfo Barros Chiaramonte

Marília, 02 de dezembro de 2013.

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus.

Aos meus paispelo apoio e conforto nas horas mais difíceis.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela força e inspiração para realização desse trabalho.

Agradeçomeus pais, Jacy e Sylvia por todo apoio, amor e paciência com esta filha que tanto os ama.

Agradeço a Guilhermina (em memoria) pelo amor e todo apoio.

Agradeço a Tia Isolina, Vanderlei e Lourival, por todo apoio a todas as horas que precisei, o meu muito obrigado.

Agradeço à Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília.

Agradeço ao meu orientador, Professor Dr. Fábio Dacêncio Pereira por ter aceitado me orientar e pelo apoio e orientação durante o curso.

Agradeço aos professores que me deram aulas durante o curso.

Agradeço também os amigos e colegas, pois foram ótimas companhias nesses anos de curso.

A todos, muito obrigado!

A persistência é o caminho do êxito.

Charles Chaplin

São as nossas escolhas que revelam o que realmente somos muito mais do que as nossas qualidades.

Alvo Dumbledore

SOUZA, A. J. S. **Sistema de Sugestões para Configurações de Servidores Web Apache**. 2013. 53 f.
Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2013.

RESUMO

É comum que algumas empresas se responsabilizem pela sua própria manutenção e configuração dos servidores, e devido ao despreparo dos administradores, algumas configurações acabam restringindo sua funcionalidade. Com isso, esta pesquisa visa a criação de uma ferramenta que analise toda a parte de configuração de um servidor Web,no caso o Servidor APACHE, auxiliando os administradores a realizar configurações corretas. Neste sentido a ferramenta oferece um sistema de dicas com o qual o administrador pode criar cenários, ou seja, a partir das respostas do administrador a algumas perguntas sobre o arquivo de configuração, são visualizadas dicas sobre o arquivo para criar as configurações apropriadas para cada cenário.

Palavras-chaves-Servidor Web; HTTP Server; Delimitando Informação; Sistema de Dicas; Httpd.conf; Segurança da Informação.

SOUZA, A. J. S. **Sistema de Sugestões para Configurações de Servidores Web Apache**. 2013. 53 f.
Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2013.

ABSTRACT

It is common for some companies take responsibility for their own maintenance and configuration of servers, and due to the unpreparedness of the administrators, some settings are restricting its functionality. With that, this project aims to create a tool to analyze all part of setting up a server, if the server APACHE, assisting administrators to perform correct settings.In this sense the tool offers a hint system with which the administrator can create scenarios, namely from the administrator replies to some questions about the configuration files, are displayed on the file tips to create the appropriate settings for each scenario.

Keywords- Web Server; HTTP Server; Vulnerabilities; Hint System; Httpd.conf; Information security.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Apache:Apache Software Foundation

API: Application Programming Interface

APR: Apache Portable Runtime

CERT.br: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

CIS: Center for Internet Security dos Estados Unidos

CSA: Cloud Security Alliance

CSRF: Cross Site Request Forgery

HTTP: Hypertext Transfer Protocol

IEEE: Institute of Electrical and Electronics Engineers

IDE: Integrated Development Environment

IPA: Information-Technology Promotion Agency

IIS: Internet Information Services

NCSA: National Center for Supercomputing Applications

NIC.br: Núcleo de Informação e Coordenação do Ponto BR.

OWASP: Open Web Application Security Project

TCP: Transmission Control Protocol

UDP : User Datagram Protocol

URL: Uniform Resource Locator

XSS: Cross Site Scripting

WAF: Web Applications Firewall

W3C: World Wide Web Consortium

SUMÁRIO

INTRODUÇÃO

A segurança da informação trata dos processos e controles que preservam os dados que são transmitidos e armazenados em uma organização. Ferreira (2003, p.1) ressalta que toda informação é um ativo que, como qualquer outro ativo importante, tem um valor para a organização e necessita ser protegido. Toda e qualquer informação é considerada importante em uma organização, mas a mesma também está indicada a risco e violação.

Com a evolução da tecnologia, as informações passaram a ser armazenadas nos computadores e muitas organizações deixaram de utilizar os papeis, e com isso a segurança sobre essas informações digitais se tornam maiores Ferreira (2003).

Assim, as aplicações para internet que executam sobre servidores WEB permitem configurações que restringem e determinam sua funcionalidade. Principalmente em servidores dedicados ou semi-dedicados, onde a própria empresa é responsável pela manutenção e configuração dos servidores, pontos vulneráveis são comuns e podem ser explorados por ações maliciosas para acessar ou danificar o sistema.

Este trabalho tem como objetivoa criação de um analisador de configurações em servidores Web,para auxiliar o administrador na configuração de um servidor web, no caso o Servidor *APACHE*.Feito em Java, permite criar configurações para cenários customizados e possibilitaa identificaçãodas falhas mais comuns em configurações do servidor Apache, apresentando sugestões para o administrador.

O Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br), que faz parte do Núcleo de Informação e Coordenação de Ponto BR (NIC.br), publicou informações de administradores e usuários envolvendo redes brasileiras no ano de 2012. Segundo o (Cert.br) as notificações sobre incidentes divulgado no ultimotrimestre de 2012 foi um pouco maior que 155 mil,

As causas são em torno de vulnerabilidades em aplicações *Web*, no qual são hospedados sites e paginas falsas de instituições financeiras, cavalos de Troia e ferramentas utilizadas em ataques a outros servidores.

Em um monitoramento de ataques acumulado durante o fechamento do ano de 2012 apontou um acumulo de 5,48%, a linha amarela é referente a Web, onde essa porcentagem é dada por um caso particular de ataque, visando especificamente o comprometimento de servidores *Web* ou desfigurações de páginas na Internet como ilustrada na figura 1.

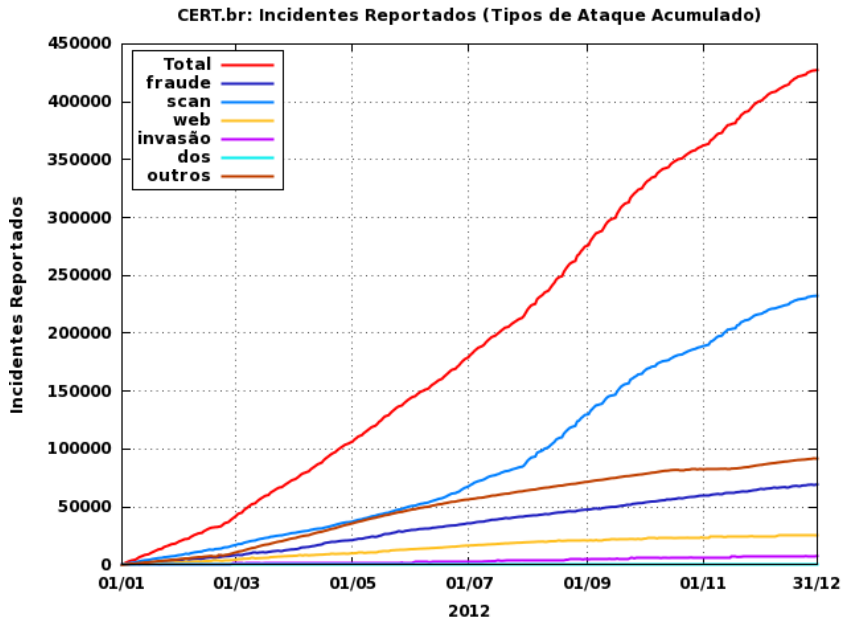


Figura 1- Incidentes Reportados ao CERT.br de Janeiro a Dezembro de 2012

Fonte: CERT.br, 2012.

O Cert.br divulgou, que noprimeiro trimestre de 2013 os ataques aos servidores cresceram 32%, mas que o numero foi menor em comparação ao ultimo trimestre de 2012. Os ataques são dados por tentativa de fraude de senhas dos administradores dos sistemas.

OBJETIVOS

Este trabalho tem como objetivo estudar as principais vulnerabilidadesdo servidor Web Apache e criar uma ferramenta para analisar o arquivo de configuração do Apachevisando a segurança das configurações, criar configurações para cenários customizados e analisar todas as configurações devido aos tipos de erros existentes, fazendo com que possam ser corrigidas as principais fontes de erro de um sistema, dando segurança a empresa e restringindo o acesso de usuários mal intencionados. Entre os objetivos específicos podem-se destacar os seguintes:

- Estudos das configurações doServidor *Web Apache*.
- Criar um *Wizard* (facilitador) para configuração de Servidores WEB de forma segura.
- Explorar recursos de usabilidade.
- Projetar e implementar um sistema de sugestões e dicas de segurança

METODOLOGIA

O projeto foi dividido em três fases principais que contemplam a (i) pesquisa de trabalhos correlatos e tecnologias, (ii) o projeto e desenvolvimento do software e (iii) avaliação dos resultados.

i. Pesquisas de trabalhos correlatos e tecnologias

Nesta etapa foram pesquisadas ferramentas similares que foram utilizadas como base para a arquitetura da solução proposta neste trabalho de conclusão de curso. Além de estudar tecnologias necessárias como a estrutura de configuração de servidores *WEB, APIs*, bibliotecas e IDE de programação para o desenvolvimento da ferramenta.

ii. Desenvolvimentos do Projeto

O desenvolvimento do software está dividido nas seguintes etapas:

- Criar um documento de requisitos do software.
- Definir a especificação do software.
- Definir a arquitetura do software.
- Codificar software em linguagem Java (*IDE Netbeans e APIs* de apoio ao desenvolvimento).

iii. Teste e Análise dos Resultados

Os testes tem como foco principal a análise de usabilidade da ferramenta como instrumento para auxiliar administradores de servidores *WEB* a instalar e implantar serviços *WEB* com segurança.

1. SEGURANÇA DA INFORMAÇÃO NO CONTEXTO DA WEB

Assim como surgem as tecnologias e a evolução dos computadores em armazenar informação, aumentam os riscos de segurança contidos nesses aspectos. Nesta seção são apresentados os aspectos sobre segurança da informação e algumas vulnerabilidades em aplicações web e servidores web.

1.1 Conceitos básicos de segurança de informações.

A segurança da informação é o meio que protege ambientes informatizados. O uso da segurança ajuda a administrar e reduzir ataques, falsificação e invasão ao próprio sistema. A Segurança da Informação busca proteger a informação de diversos tipos de ameaças, minimizar os danos e maximizar o retorno dos investimentos e das oportunidades. (Ferreira, 2008). Suas características são:

- Confidencialidade: garantia de que a informação é acessível somente por pessoas autorizadas;
- Integridade: salvaguarda da exatidão da informação e dos métodos de processamento;
- Disponibilidade: garantia de que os usuários autorizados obtenham acesso à informação e aos ativos correspondentes sempre que necessário.

(Ferreira, 2008).

A segurança da informação se torna importante, pois informações alteradas, não disponíveis, sob o conhecimento de pessoas de má-índole podem comprometer, não apenas a imagem da organização perante terceiros, como também o andamento dos próprios processos organizacionais. (Tribunal De Contas Da União, P.26, 2007).

Para assegurar um bom nível de proteção, é preciso definir o nível de segurança da informação, sendo eles:secreta confidencial, interna e pública(DIAS, 2000, p.53).

- Secreta: Esta informação deve ser preservada e com proteção contra pessoas não autorizadas.
- Confidencial: Esta informação exige restrição, pois pode causar problemas para o negócio, como prejuízos financeiros e vantagem aos concorrentes se for acessada por pessoas não autorizadas.

- Interna: Esta informação deve ser protegida, porém, se for acessada por pessoas não autorizadas, os danos não serão tão grandes.
- Pública: Esta informação não requer restrição, podendo se tornar de conhecimento público, sem que isso traga consequências negativas para a empresa.

1.2 Principais vulnerabilidades em aplicações web.

Segundo a Symantec (Symantec Corporation), houve um aumento de ataques que ocorreram via web no ano de 2012, e os ataques foram direcionados principalmente a pequenas e grandes empresas. O aumento de ataques foi três vezes maior que 2011. De acordo com a Symantec (Figura 2), foram bloqueados cerca de 247 a 350mil ataques baseados em web por dia no ano de 2012,enquanto o mesmo período de 2011 teve de 190 a 300 mil ataques bloqueado por dia.

Figura 2 - Media de Ataques baseados na Web por dia, 2011-2012.
Fonte: Symantec Corporation, 2013.

As vulnerabilidades em aplicações web vêm se tornando comuns, pois nessas aplicações web são encontradas algumas falhas de segurança que facilitam que os códigos sejam alterados, ocasionando roubo de informações.

1.3 OWASP Top Ten.

O grupo *Open Web Application Security Project* é uma organização sem fins lucrativos, que proporciona segurança em aplicações web, para que as pessoas e empresas possam tomar decisões com segurança e combater ataques através da internet.

A fundação realiza eventos em todo o mundo e produz materiais de modo colaborativo, sendo todas as ferramentas, documentos, fóruns e capítulos, livres e abertos a qualquer pessoa empenhada em melhorar a segurança do aplicativo.

O material desenvolvido é utilizado, recomendado e referenciado por diversas organizações,por exemplo:

- *World Wide Web Consortium* (W3C), de âmbito mundial.
- *Center for Internet Security* (CIS) dos Estados Unidos.
- *Cloud Security Alliance* (CSA), de âmbito mundial.
- *Information-Technology* Promotion Agency (IPA), do Japão.
- *Institute of Electrical and Electronics Engineers* (IEEE), de âmbito mundial.
- Banco Central do Brasil.

Sobre a *OWASP* é importante citar um de seus projetos denominado *Top Ten*, que é uma lista, atualizada a cada três anos com as 10 vulnerabilidades mais críticas em aplicações web, escritas por membro de empresas. Os principais tipos de ataques da *Top Ten* contidos na documentação da *OWASP, Top Ten* 2013.

1) *Injection*
2) *Broken Authentication and Session Management*
3) *Cross Site Scripting*(XSS)
4) *Insecure Direct Object References*
5) *Security Misconfiguration*
6) *Sensitive Data Exposure*
7) *Missing FunctionLevel Access Control*
8) *Cross Site Request Forgery*(CSRF)
9) *Using Known Vulnerable Components*
10) *Unvalidated Redirects and Forwards*

De acordo com a lista divulgada pelo Top ten, o item Security Misconfiguration (Configuração Incorreta de Segurança) relata que a segurança exige a definição de uma configuração segura em frameworks, servidor de aplicação, servidor web, banco de dados e

plataforma. As configurações devem ser bem definidas e implementadas, já que geralmente a configuração padrão são inseguras.

1.4 Cert.br

O CERT.br é o Grupo de Resposta a Incidentes de Segurança para a Internet brasileira, faz parte do Núcleo de Informação e Coordenação do Ponto BR, que trata incidentes de segurança em computadores em redes Brasileiras, implantando soluções de segurança.

O Cert.br mantém atualizadas as estatísticas sobre incidentes reportados no Brasil. Suas atividades tendem a dar suporte ao processo de recuperação e análise de ataques, estabelecer trabalhos colaborativos, oferecer treinamentos de segurança, desenvolver documentação e aumentar a capacidade de detecção de incidentes.(Cert.br)

1.5 Ataques a Servidores Web.

Os servidores Web são configurados e atualizados com correções de segurança, mas podem ser comprometidos por conta de vulnerabilidades nas aplicações web.

As aplicações web podem ser mais vulneráveis porque envolvem componentes diferentes do navegador ao servidor web. As vulnerabilidades podem ocorrer em qualquer lugar, mas o servidor está sempre no meio deste ambiente (Wolfgarten, S.; Zaidan,K).

Com foco nos problemas de configuração de um servidor é necessário que se tenha um analisador de configurações permitindo uma configuração customizada.

O Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br), relatou que o primeiro trimestre de 2012 teve um aumento de 8% em relação ao trimestre anterior e uma queda de 4% em relação ao mesmo trimestre de 2011. Os ataques a servidores Web vem crescendo desde 2007, conforme mostra a Figura 3.

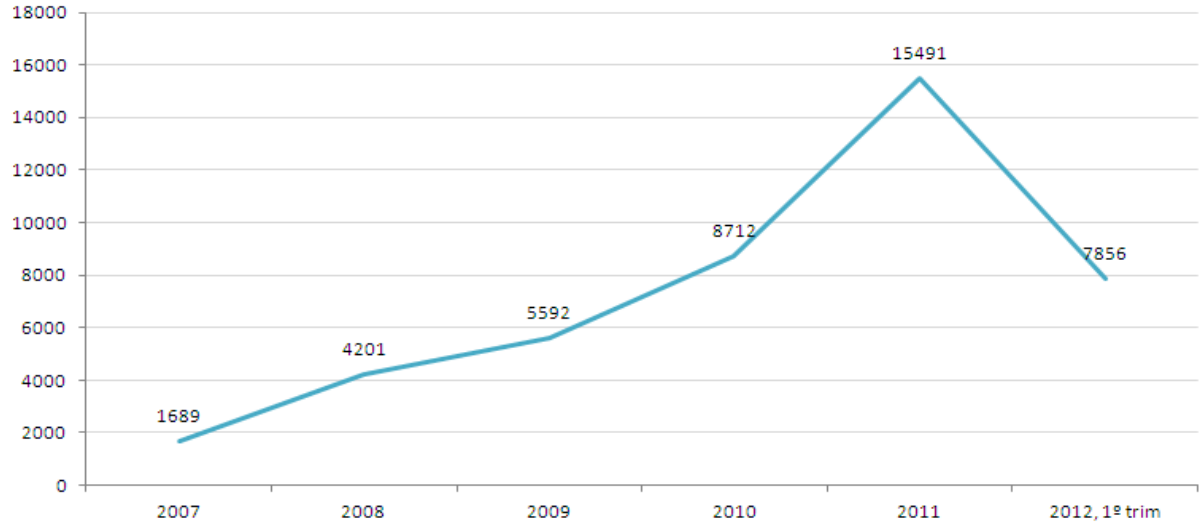


Figura 3- Ataques a Servidores Web.
Fonte: NIC.br - Núcleo de Informação e Coordenação do Ponto BR, 2012.

Hoepers ressalta que este aumento vem crescendo desde 2007 por ataques que exploram falhas de programação em scripts e aplicações adicionadas aos ambientes Web ou desenvolvidas localmente nas organizações.A queda no primeiro trimestre de 2012 é devida algumas tentativas de fraude que foram corrigidas como falhas de programação em scripts .(Cert.br)

1.6 ModSecurity

O *ModSecurity* é um WAF (*Web Applications Firewall)* , que trabalha como um modulo que impede alguns tipos de ataques a servidores, como *Cross-Site Scripting* (XSS) e *SQL Injection.*

O software tem o apoio da empresa *Breach Security*que patrocina a distribuição, foi desenvolvido por Ivan Ristic e possui duas versões de software, sendo uma open-source e uma versão comercial, com suporte profissional que pode ser executado no *Linux, Solaris, FreeBSD, OpenBSD, NetBSD, AIX* e *Windows,* com as próximas versões somente disponíveis para o Apache 2.x. A empresa também fornece regras que garantem a segurança do servidor web.

Isso faz do *ModSecurity* uma boa escolha para proteger servidores web e suas aplicações contra vulnerabilidades.

1.7 Análise de Trabalhos correlatos.

Segundo KLABUNDE (2007),usou de forma prática o uso de tecnologias atuais para uma solução alternativa no monitoramento de servidores web Apache. Onde o usuário é capaz de conectar-se através de qualquer ponto de rede (internet) e fazer o monitoramento de um servidor Apache. Neste projeto o protótipo permite ao usuário fazer consultas de informações de configuração e contabilização do servidor.

A*Red HatEnterprise* possui uma ferramenta de configuração do HTTP da *Red HatEnterprise* que possuicaracterísticas como migrar uma versão anterior httpd.conf para a versão 2.0 do Servidor HTTP Apache, possui uma interface gráfica que possibilita a configuração do arquivo do servidor, como diretivas, host virtuais e autenticações. Desta forma, a ferramenta só permite que a configuração seja feita com os módulos oferecidos pela *Red HatEnterprise*, ou seja, se algum módulo ou parâmetro de configurações for adicionado no arquivo manualmente, não é possível utilizar a ferramenta.

A partir das informações sobre segurança na Web, os servidores Web precisam de segurança devido a sua responsabilidade em armazenamentos de paginas de um site requisitado pelos clientes através de browsers.

2. SERVIDORES WEB

Nesta seção são apresentados os servidores web disponíveis e a funcionalidade do servidor Web Apache.

2.1 Conceitos e tecnologias.

Os servidores web tornaram-se essenciais com a evolução da tecnologia, que é o meio responsável por armazenar e trocar informações com outra máquina. Para realizar este ciclo de troca de mensagem é preciso um cliente que solicite a informação e o servidor que irá receber a solicitação. O cliente utiliza os *browsers* como "*Internet Explorer, Google Chrome*" para realizar a solicitação e o servidor trabalha com softwares para realizar a transferência de informação. Para essa transferência de informação é usado o protocolo *Hypertext Transfer Protocol* (HTTP)de comunicação da Web. Segundo ALECRIM(2006), quando um navegador de internet acessa um site, este faz as solicitações devidas ao servidor Web do mesmo através do HTTP e então recebe o conteúdo correspondente.

Existem alguns servidores disponíveis como: IIS (*Internet Information Services*) da Microsoft, que possui seus servidores baseados em *Windows NT* e aplicação proprietária, o Xitami que é servidor *Open source* desenvolvido como HTTP pessoal e o *Apache HTTP Server*, o servidor mais robusto no mercado.

2.2 Servidor Apache

O *Apache Software Foundation* contem mais de 140 projetos de software de código aberto que contam com suporte organizacional, legal e financeiro.Trata-se de uma organização sem fins lucrativos que proporciona estrutura intelectual e ao mesmo tempo limita a exposição legal potencial para esses projetos.

O projeto Apache HTTP Server nasceu em 1995, obra do então funcionário do NCSA (*National Center for Supercomputing Applications*), Rob McCool, responsável também pela versão 2.2 do Apache.O Apache é um software colaborativo que tem o objetivo de criar um robusto nível comercial, provido de recursos e com livre implementação de código fonte de um servidor HTTP (Web). O projeto é desenvolvido por um grupode voluntários ao redor do mundo, que se comunicam através da Internet e da Web, bem como planejam e desenvolvem tanto o servidor como sua documentação.

Segundo as informações disponibilizadas pela Netcraft, desde 1995 (figura 4), vem crescendo o uso dos domínios da internet utilizando o servidor Web Apache, que tem entre suas principais qualidades a indepedência de plataformas específicas, o que possibilita o suporte para os variados tipos de documentos da Web.

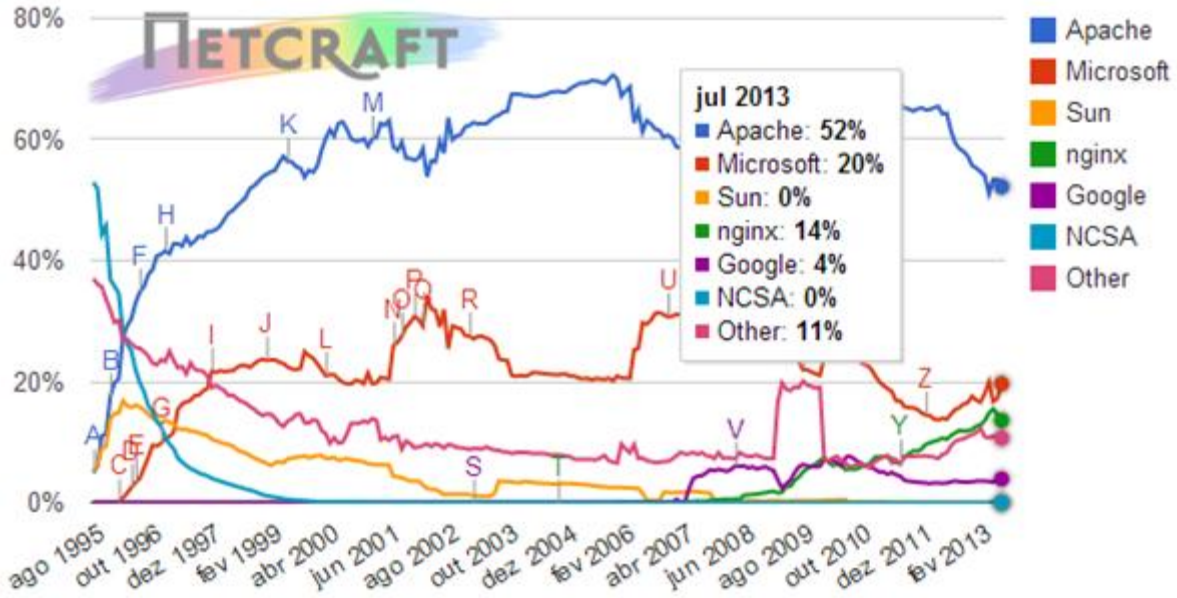


Figura 4 - Market Share for top Servers Across all Domains.
Fonte: News.netcraft, 2013

A independência de plataforma torna o Apache um sistema possível de ser implementado em qualquer sistema operacional. O ambiente de execução *Apache Portable Runtime* (APR) proporciona interface independente de plataforma para o tratamento de arquivos, redes e a portabilidade é garantida no caso de extensão do Apache.Além de sua flexibilidade no detalhamento, a configuração do Apache permite a extensão de sua funcionalidade sem muitas dificuldades.

Considerado um servidor geral que produz respostas a requisições, o Apache não está livre de dependências, situações essas que não o impedem de manipular documentos Web. O núcleo Apache entende que o HTTP, quase sempre implementado em cima de TCP, obedece à requisições baseado nesse último. Já as baseadas em UDP sempre modificam o núcleo.

As requisições devem ser manipuladas de modo que o gancho (reservador de lugar para um grupo específico de funções) represente ações que necessariamente precisam ser processadas como parte do processamento de uma requisição.

Os ganchos são utilizados na tradução de URL para o nome de um arquivo local, para escrever informações de um registro, verificar a identificação de um cliente, direitos de

acesso e o tipo de Mime ao qual a resição se refere. Eles são processados em ordem pré determinada.Os módulos separados fornecem as funções relacionadas aos ganchos. Tanenbaum(2008).

Tanenbaum (2008) ressalta que “todo gancho pode conter um conjunto de funções e cada uma delas deve combinar com o protótipo específico de função (isto é, lista de parâmetros e tipo de retorno)”. O desenvolvedor escreve funções para ganchos específicos, como na (Figura 5).

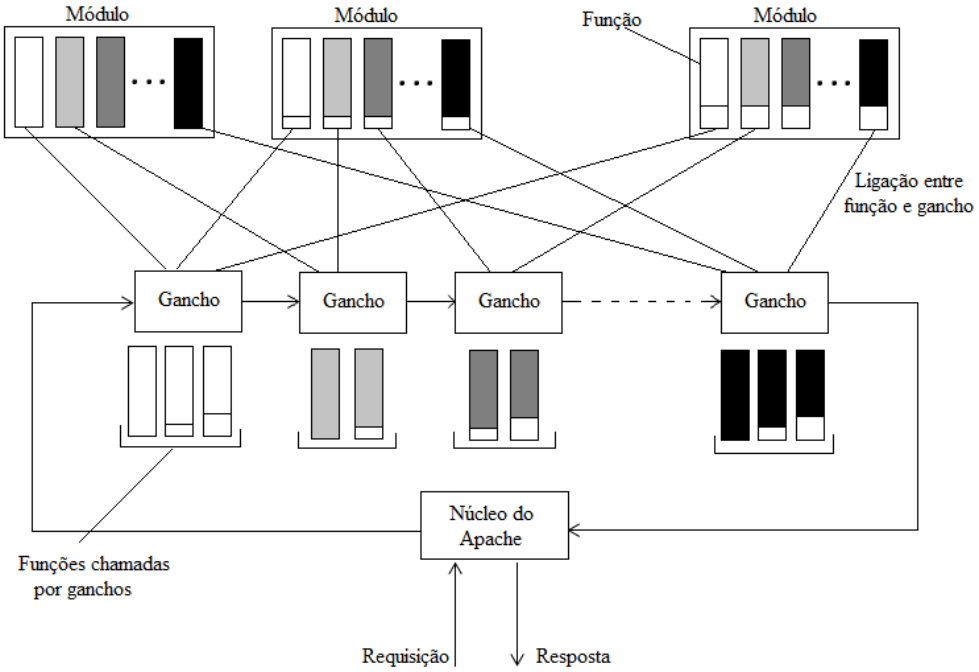


Figura 5 - Organização geral do servidor Web Apache.
Fonte: Sistemas Distribuídos. (Tanenbaum 2008, p.337)

O mesmo especifica qual a função que deve ser anexada ao gancho, mostrado pelas ligações entre funções e ganchos.

2.3 Configuração e arquitetura do servidor Apache.

O Servidor Apache é estruturado em módulos, que podem ser classificados em três categorias, sendo elas: Módulo Base, que contém as funções básicas do Apache, Módulos Multiprocesso,responsáveis pela união com os portas da máquina,e Módulos Adicionais que permitem adicionar qualquer funcionalidade ao servidor. Diaz. C.C.(2007)

Ao instalar o Apache, existem alguns diretórios que são importantes, como:

- **Bin**: São encontrados os arquivos executáveis do Apache.
- **conf:** arquivos de configuração do servidor, como o *httpd.conf.*

- **error:** arquivos com as mensagens de erro do servidor, em várias linguagens.
- **htdocs:** diretório raiz padrão do servidor (Salvam-se as páginas Web).
- **icons:** diretório onde se encontram os ícones que o servidor utiliza.
- **logs:** diretório onde se armazenam os registros de acesso e erros do servidor.
- **manual:** diretório onde se encontra o manual do Apache.
- **proxy:** Diretório com os arquivos do cache do servidor.

2.3.1 Arquivo Httpd.Conf.

O arquivo de configuração do apache se encontra no diretório *conf*, nomeado*httpd.conf*, o mesmo é configurado usando diretivas de configuração onde se encontram todos os parâmetros de funcionamento do servidorde acordo com o site *The Apache Software Foundation.*

No arquivo de configuração, outros arquivos podem ser adicionados usando a diretiva *Include*e para adicionar novas funcionalidades no núcleo do servidor é usado a diretiva *IfDefine* (tabela1).

As alterações no arquivo de configuração só são reconhecidas pelo Apache quando o mesmo for iniciado ou reiniciado.

Tabela 1 – Diretivas adicionais

<IfDefine>	Utilizado para diretivas que são adicionais e só são executadas se o teste for verdadeiro, caso seja falso o bloco é ignorado.
Include	Permite adicionar outros arquivos dentro dos arquivos de configuração do servidor.
TypesConfig	Define o local do arquivo de configuração.

Fonte: Apache.org

O Apache quando instalado possui algumas funcionalidades definidas como padrão no arquivo de configuração. A tabela 2 mostra o conjunto de módulos que está presente no arquivo padrão, os módulos podem ser copilados dinamicamente ou separadamente através da diretiva *LoadModule*. Outras funcionalidades podem ser adicionadas usando o módulo <*IfModule*> que é representado em bloco.

Tabela 2 – Módulos adicionais

<IfModule>	Utilizado para diretivas que são adicionais e só são executadas se o teste for verdadeiro, caso seja falso o bloco é ignorado.
LoadModule	Diretriz que serve para carregar módulos que incluem diferentes funcionalidades.LoadModule nomeModulo localizacaoArquivo

Fonte: Apache.org

As diretivas de funcionamento podem ser aplicadas em todos os servidores ou podem restringir a funcionalidade de alguns diretórios. Para isto,o arquivo é dividido em seções. A tabela 3 mostra as principais diretivas que podem ser aplicadas ao servidor.

Tabela 3 – Diretivas de configuração.

<Directory>	Esta diretriz delimita um conjunto de regras que serão aplicadas somente ao diretório e subdiretórios nomeado.
<DirectoryMatch>	São utilizadas para delimitar um conjunto de diretivas que se aplicam somente ao diretório chamado e subdiretório daquele mesmo diretório.
<Files>	Proporcionam controle de acesso dos arquivos pelo seu nome.
<FilesMatch>	Funciona como o <Files>. No entanto possui uma expressão regular como: `<FilesMatch "\.(gif|jpe?g|png)$">`
<Location>	São utilizadas, pois proporcionam um controle de acesso dos arquivos por meio da URL.
<LocationMatch>	Como o <Location> controla o acesso das URLs, porém permite o uso de expressão regular como: <LocationMatch "/(extra|special)/data">
<VirtualHost>	São aplicadas em host virtual particular. Quando o servidor recebe um pedido de um documento em um host virtual particular, ele usa as diretivas de configuração inclusas na seção <VirtualHost>. Como um endereço IP do host virtual ou um caractere "*".

Fonte: Apache.org

O Apache permite obter um *feedback*da configuração sobre a atividade e o desempenho do servidor (tabela 4), as diretivas para isso acontecer são:

Tabela 4 – Arquivos de Registros.

ErrorLog	Indica em qual arquivo deverão ser gravados os erros que forem encontrados pelo servidor.
LogLevel	Indica o nível das mensagens de erro que serão gravadas no arquivo Log.

Fonte: Apache.org

O arquivo padrão do Apache possui algumas configurações básicas que são geradaspara identificação do servidor e diretivas que controlam as localizações dos arquivos necessários para o funcionamento(tabela 5).

Tabela 5 – Configurações do Servidor.

ServerName	Identifica nome do host e porta que o servidor usa para identificar
ServerAdmim	Identifica qual o e-mail do administrador do servidor.
DocumentRoot	Determina onde são localizadas as páginas que servirão ao servidor
ServerRoot	Determinaem qual arquivo os erros de log serão salvos.

Fonte: Apache.org

A partir dessa etapa tem-se as informações necessárias para uma ferramenta de auxilio para configuração do arquivo httpd.conf do servidor Apache.

3. FERRAMENTA HTTPD EDITOR

Este capítulo apresenta a funcionalidade da ferramenta e os diagramas de comoa ferramenta funciona.

3.1 Descrição Funcional

Este trabalho envolve o desenvolvimento de um software para análise das configurações do servidor *Web Apache*. Neste contexto, foi desenvolvido o software denominado Httpd Editor que faz o seguinte:

- Remove os comentários do arquivo *httpd.conf* identificados por '#'.
- Identifica os atributos e valores dos parâmetros encontrados no arquivo. Posteriormente envia os valores e atributos para o banco de dados, esses parâmetros são mostrados na tela para o administrador poder configurar o arquivo.
- Em uma segunda etapa o administrador responde a perguntas sobre o modo de configurar o arquivo,conforme o preenchimento do arquivo o administrador pode visualizar sugestões de como configurar o arquivo para criar o cenário.
- No final, o administrador pode comparar os arquivos de configuração com o original ou comparar um cenário com a resposta padrão e, a partir do arquivo de erro gerado, configurar um novo arquivo com o cenário.

3.2 Diagrama Caso de Uso.

O caso de uso gerado descreve o papel do administrador com a ferramenta, figura 6.

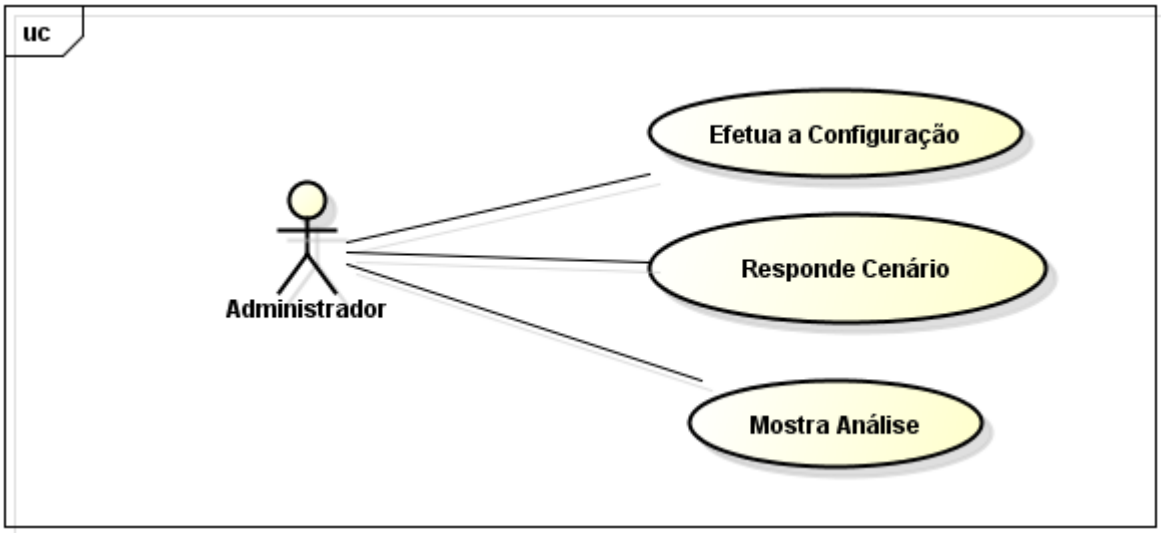


Figura 6 - Diagrama de caso de uso da ferramenta Httpd Editor.
Fonte: Próprio autor, 2013.

3.3 Diagrama de Classe.

Segue o diagrama de classe (figura 7) referente às classes abaixo.

i. **Config**

Identifica o arquivo httpd.conf, além de remover os comentários.

ii. **ConfigCab.**

Janela de consulta dos novos arquivos de configurados.

iii. **Cenário**

Possui as informações para responder as perguntas do cenário, como as dicas de parâmetros e dicas de segurança.

iv. **CenárioRespostaCab**

Janela de repositório dos cenários.

v. **CenarioRespostaItem**

Exibe as informações e resposta dos cenários e dicas.

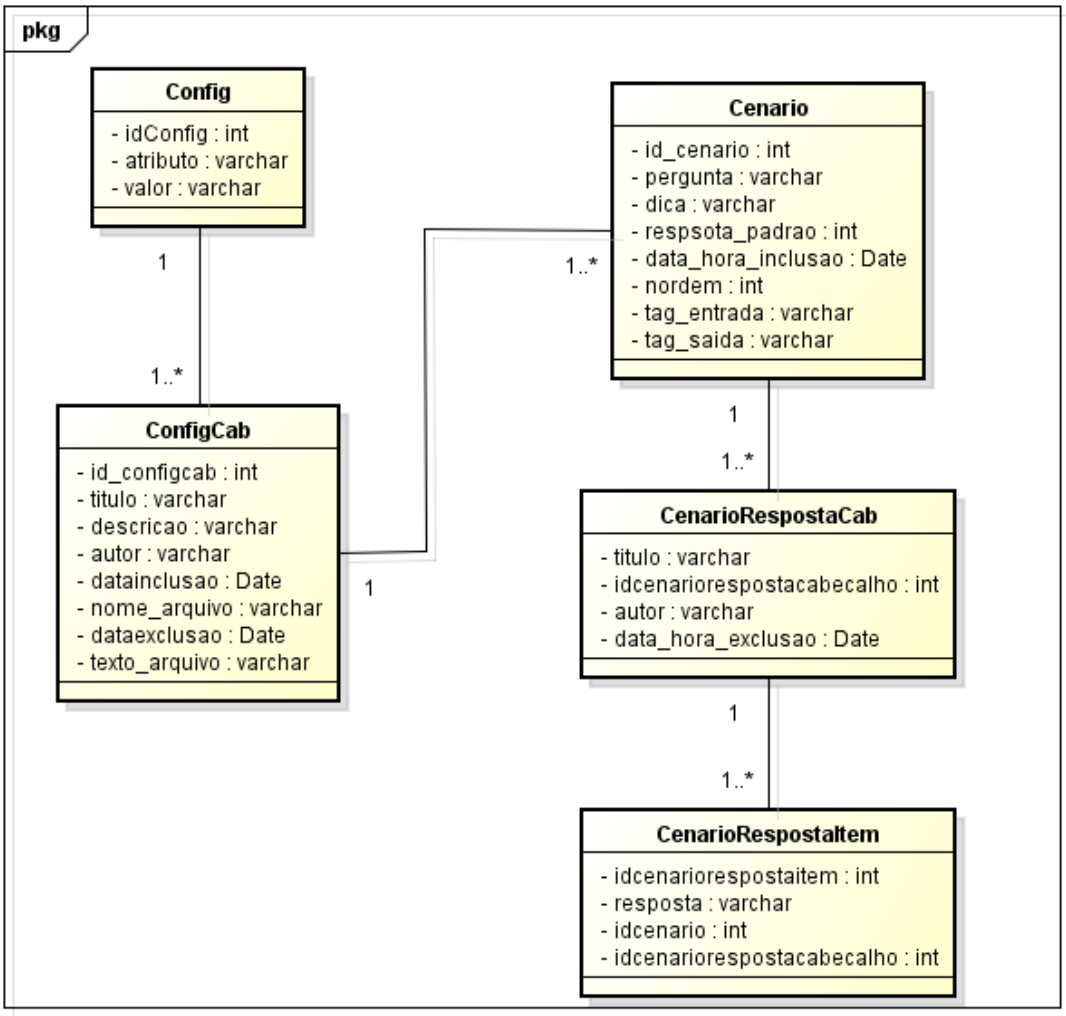


Figura 7 - Diagrama de classe da ferramenta HttpEditor .
Fonte: Próprio autor, 2013

3.4 Diagrama de Sequência: Configuração.

No diagrama de sequência érepresentada a sequência de processos, ou seja, a maneira como foram cadastradas as novas configurações, figura 8.

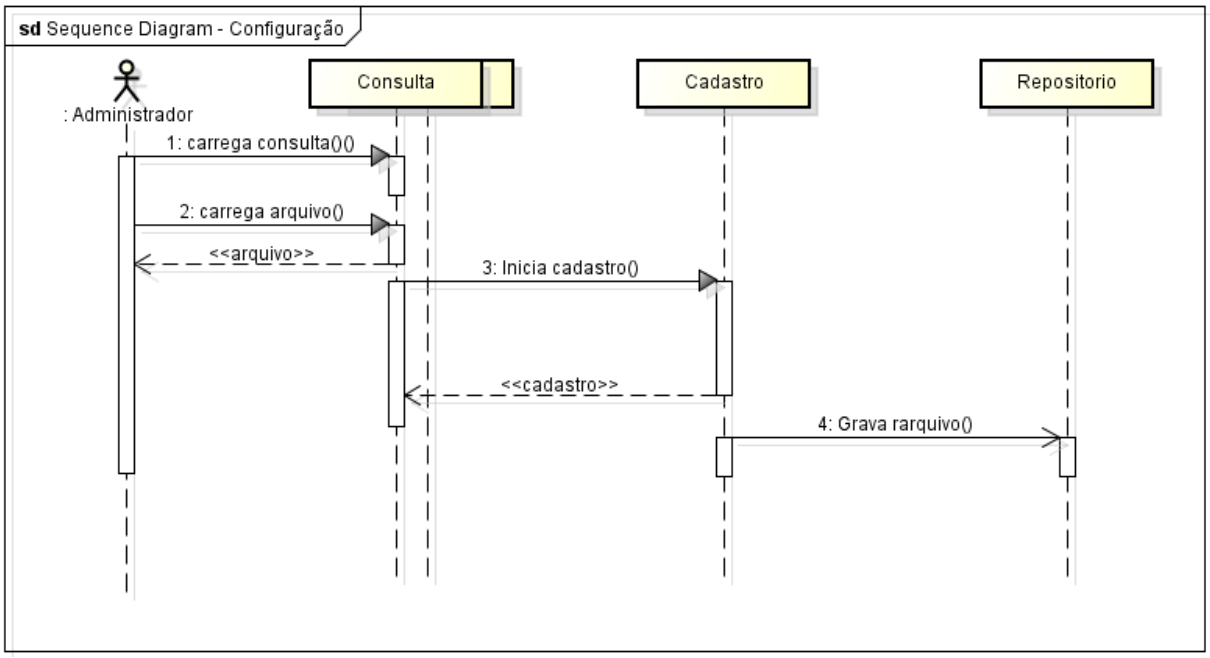


Figura 8 - Diagrama de sequencia: Configuração
Fonte: Próprio autor, 2013.

3.5 Diagrama deSequência: Cenário.

O diagrama de sequência do cenário mostra como os cenários foram realizados, figura 9.

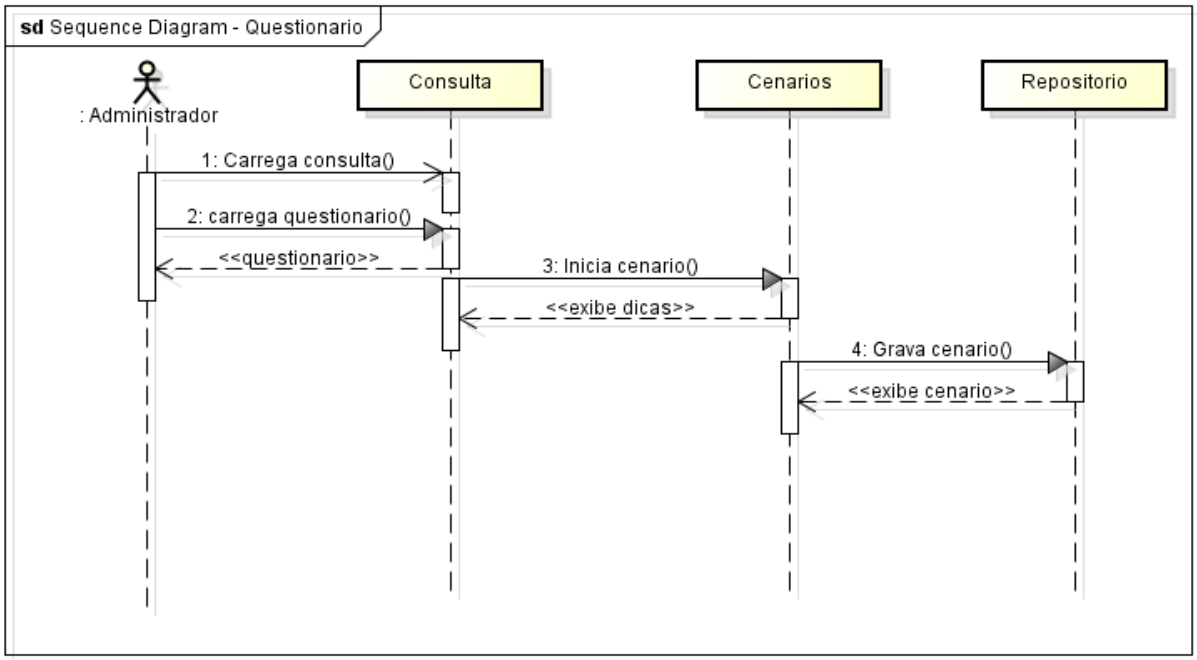


Figura 9 - Diagrama de sequencia: Cenários
Fonte: Próprio autor, 2013.

A figura 10 mostra a modelagem relacional do banco de dados do HttpdEditor.

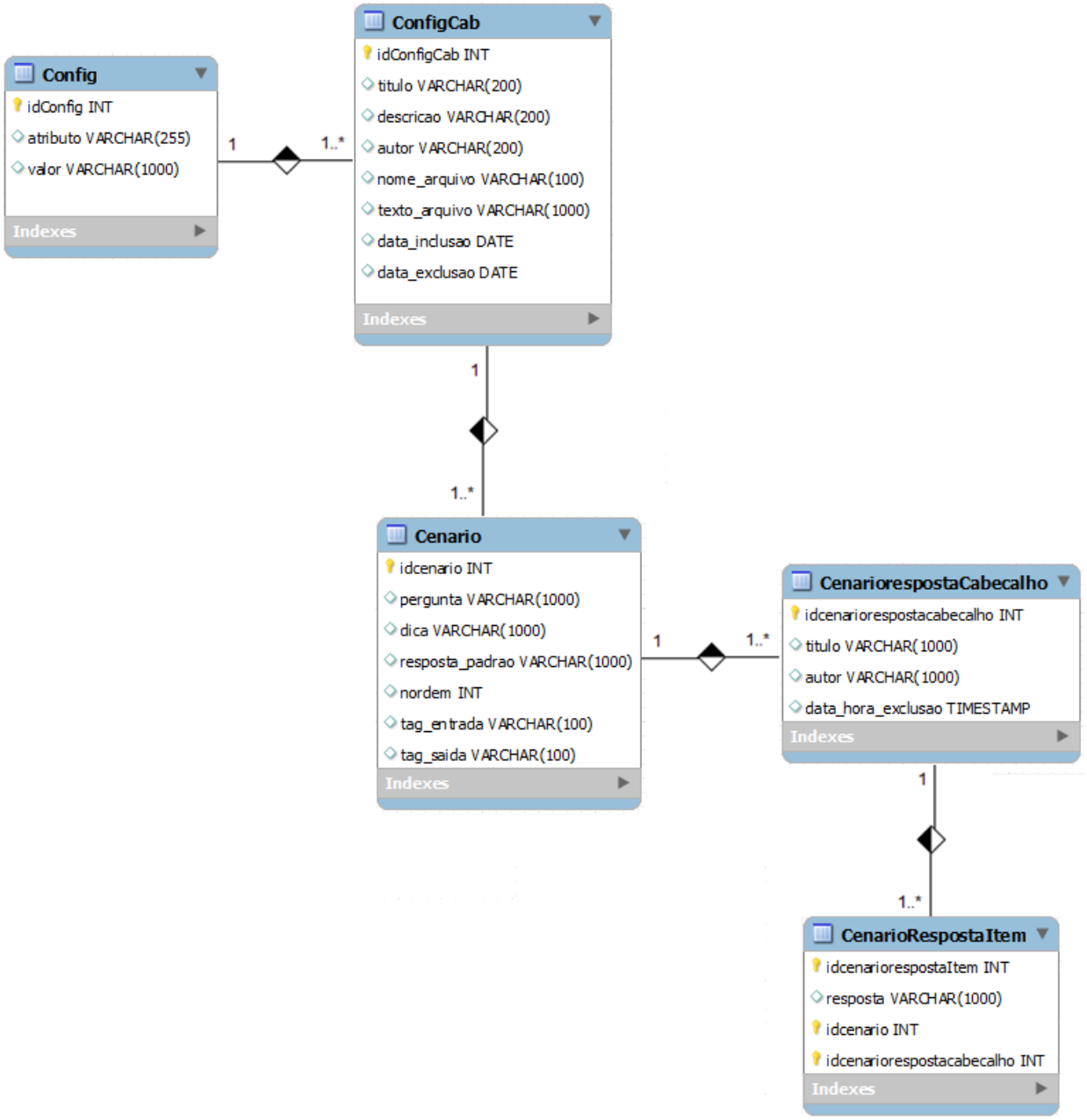


Figura 10 - Modelo Relacional do banco de dados do HttpdEditor.
Fonte: Próprio autor, 2013.

4. MÓDULO DE CONFIGURAÇÃO

Este capítulo apresenta os processos de desenvolvimento da ferramenta HttpdEditor.

4.1 Arquitetura do Protótipo.

Desenvolvido em Java, através da utilização da IDE (*Integrated Development Environment*)*Netbeans* 7.2.

Para desenvolvimento do projeto foi utilizado o Banco de dados Derby (*Apache Derby*). O *Apache Derby*, um subprojeto do Apache, é um banco de dados open source relacional implementado totalmente em Java; executa em qualquer Máquina Virtual Java (JVM) certificada. A figura 11 apresenta a arquitetura do protótipo, facilitando o entendimento de como é o seu funcionamento.

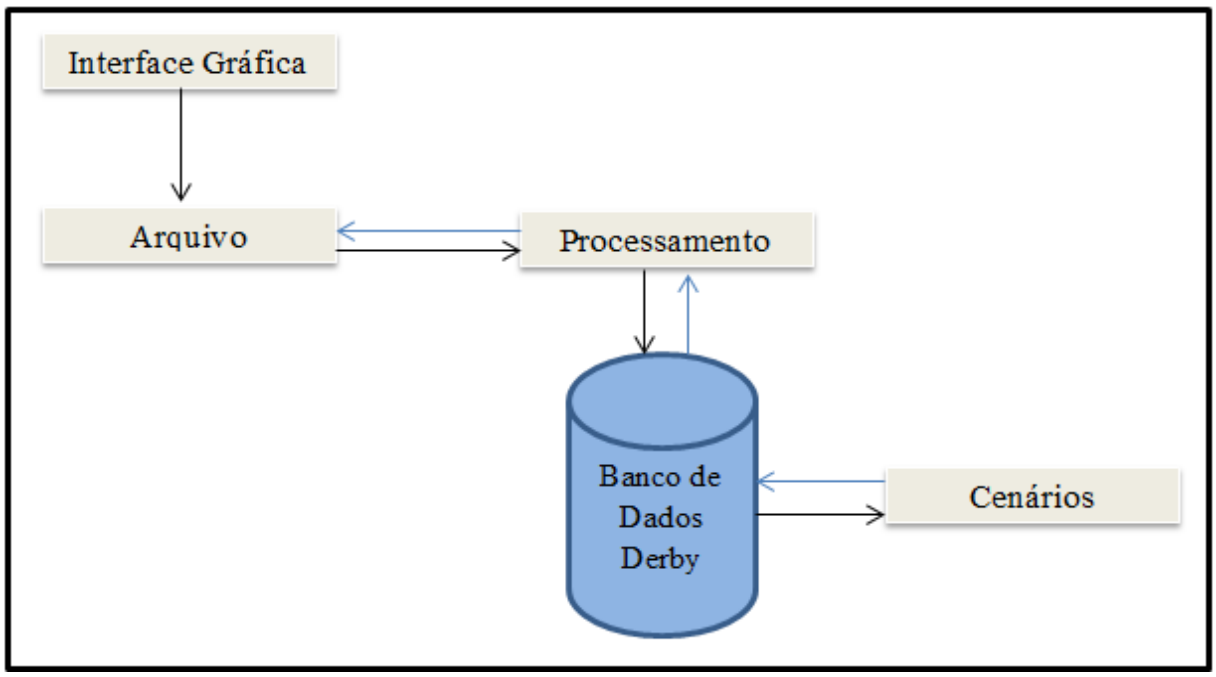


Figura 11 - Modelo conceitual do software proposto.
Fonte: Próprio autor, 2013.

- Arquivo**:** Classe responsável pela leitura do arquivo padrão Httpd.conf
- Processamento: Responsável pela análise no arquivo, identificando valores e atributos dos parâmetros de configuração.
- Banco de Dados: Através do banco de dados Derby todos os parâmetros do arquivo são armazenadas.
- Cenários: Onde o administrador responde as perguntas sobre o arquivo para criar os cenários.

4.1.1 Funcionamento da Ferramenta HttpEditor.

A seguir é mostrado o funcionamento da ferramenta que auxilia o administrador na configuração do Httpd.conf do servidor Web Apache.

4.1.2 Remoção dos Comentários do Arquivo.

A tabela 6 apresenta o arquivo padrão do servidor Apache com os comentários identificados por "#" no arquivo Httpd.conf, em que é realizado a configuração do servidor.

Tabela 6 - Arquivo Httpd.conf.

Entrada
This is the main Apache HTTP server configuration file. It contains the # configuration directives that give the server its instructions. # See <URL:http://httpd.apache.org/docs/2.2> for detailed information. # In particular, see # <URL:http://httpd.apache.org/docs/2.2/mod/directives.html> # for a discussion of each configuration directive. # # Configuration and logfile names: If the filenames you specify for many # of the server's control files begin with "/" (or "drive:/" for Win32), the # server will use that explicit path. If the filenames do *not* begin # with "/", the value of ServerRoot is prepended -- so "logs/foo.log" # with ServerRoot set to "C:/Program Files/Apache Software Foundation/Apache2.2" will be interpreted by the # server as "C:/Program Files/Apache Software Foundation/Apache2.2/logs/foo.log". # # ServerRoot: The top of the directory tree under which the server's # configuration, error, and log files are kept. # # Do not add a slash at the end of the directory path. If you point # ServerRoot at a non-local disk, be sure to point the LockFile directive # at a local disk. If you wish to share the same ServerRoot for multiple # httpd daemons, you will need to change at least LockFile and PidFile. # ServerRoot "C:/Program Files/Apache Software Foundation/Apache2.2" # # Listen: Allows you to bind Apache to specific IP addresses and/or # ports, instead of the default. See also the <VirtualHost> # directive. ## Change this to Listen on specific IP addresses as shown below to # prevent Apache from glomming onto all bound IP addresses. #Listen 12.34.56.78: 80 Listen 80

Fonte: Httpd.conf

Após a análise do arquivo, a tabela 7 apresenta a saída do arquivo a partir da leitura do Httpd.conf, em que é realizada a análise do arquivo, removendo os comentários e salvando o mesmo em um novo arquivo, nomeado Httpd.conf.

Tabela 7 - Saída após remoção de comentários.

Saída
ServerRoot "C:/Program Files/Apache Software Foundation/Apache2.2" Listen 80

Fonte: Próprio autor, 2013.

4.1.3 Identificações dos Parâmetros.

O arquivo Httpd.conf é composto por parâmetros que geram a configuração do servidor *Web Apache*. Para obter a identificação desses parâmetros no arquivo (tabela 8), foi desenvolvido um mecanismo que lê a linha do arquivo e analisaas tag de entrada <> e tag de saída</> identificando os parâmetros e suas funções.

Tabela 8 - Parâmetros e Valores

ServerRoot "C:/Program Files/Apache Software Foundation/Apache2.2" Listen 80	**Atributo** =ServerRoot **Valor** = "C:/Program Files/Apache Software Foundation/Apache2.2" **Atributo** = Listen **Valor** = 80
<IfModule !mpm_netware_module> <IfModule !mpm_winnt_module> User daemon Group daemon </IfModule> </IfModule>	**Atributo** =<IfModule !mpm_netware_module> **Atributo** =<IfModule !mpm_winnt_module> **Valor** =User daemon **Valor** =Group daemon **Atributo** =</IfModule> **Atributo** =</IfModule>
<Directory /> Options FollowSymLinks AllowOverride None Order deny,allow Deny from all </Directory>	**Atributo** =<Directory /> **Valor** =Options FollowSymLinks **Valor** = AllowOverride None **Valor** = Order deny,allow **Valor**= Deny from all **Atributo** =</Directory>

Fonte: Próprio autor, 2013.

4.1.4 Exemplo de Funcionamento da Ferramenta.

Na figura 12 é possível ver a etapa da ferramenta que permite ao administrador criar novas configurações a partir do arquivo padrão. A ferramenta possui uma interface que mostra as configurações criadas pelo administrador.

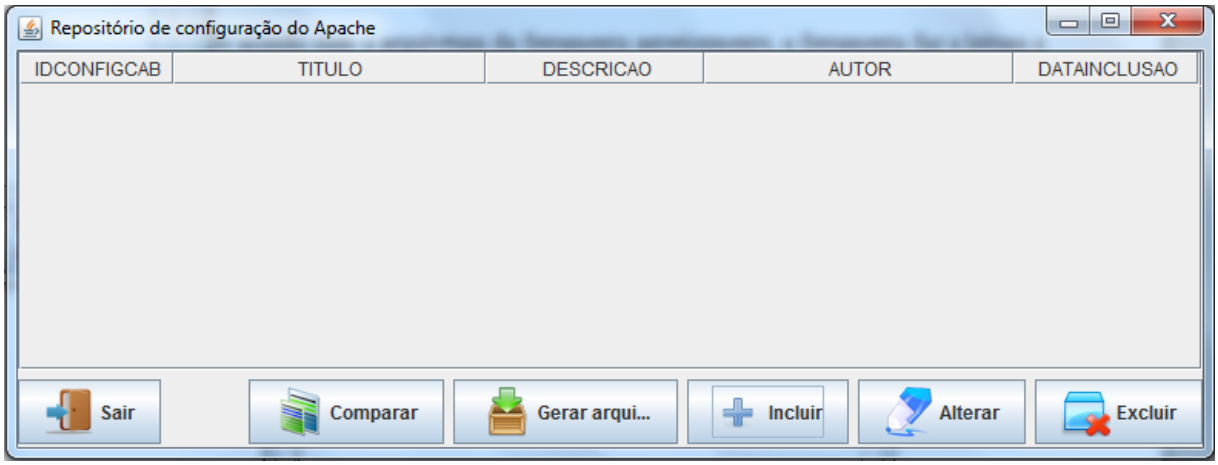


Figura 12 - Repositório de Configuração do Apache
Fonte: Próprio autor, 2013.

De acordo com a arquitetura da ferramenta, é realizada a leitura do arquivo padrão do servidor, onde o arquivo é analisado e inserido no banco de dados. O botão "Incluir" abre a tela de configuração, apresentando ao administrador o arquivo padrão que foi analisado e sem comentários, permitindo que o novo arquivo seja identificado com título, descrição e autor da configuração, onde o mesmo pode acrescentar ou modificar parâmetros dentro do arquivo e salvar as novas informações, conforme pode ser visualizado na figura 13.

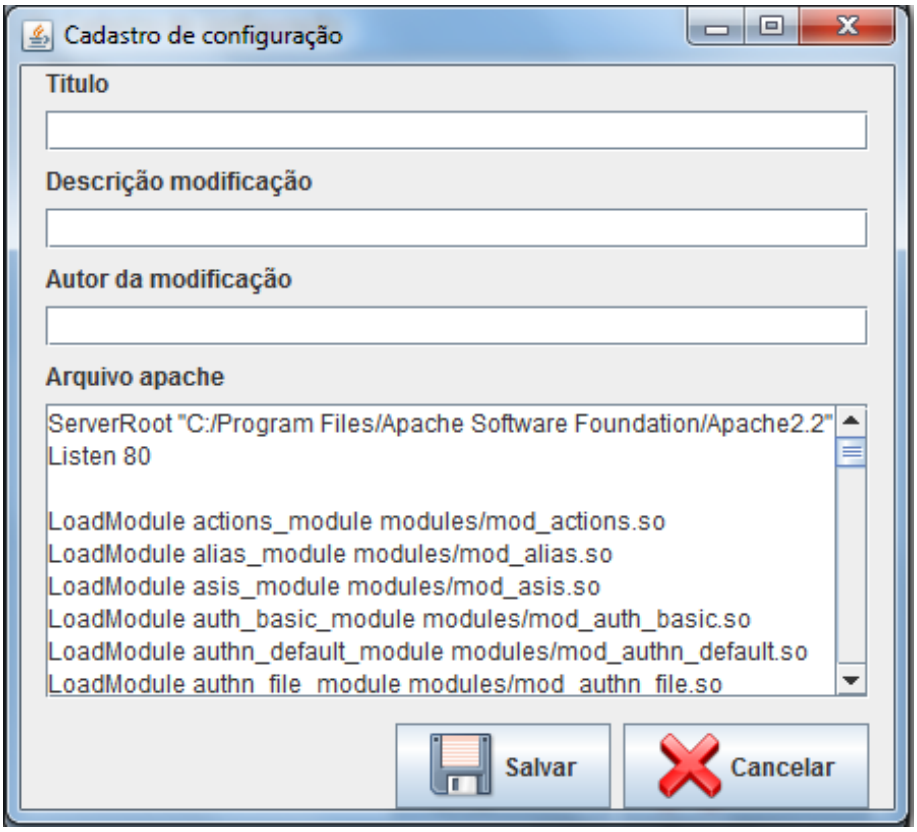


Figura 13 - Cadastro de Configuração
Fonte: Próprio autor, 2013.

Após o administrador salvar os novos arquivos os mesmo são armazenados no repositório, sendo possível visualizar suas informações através das identificações feitas na configuração. O botão "Alterar" permite abrir uma configuração, onde as informações salvas anteriormente são apresentadas ao administrador, possibilitando criar uma nova versão de configuração em cima dos dados armazenados e a nova inserção é identificada pela descrição e data de inclusão, conforme pode ser visualizado na figura 14.

Figura 14 - Repositório com informações alteradas.
Fonte: Próprio autor, 2013.

4.2 Sistema de Sugestão.

A ferramenta HttpEditor contempla com um sistema de sugestões para ajudar e solucionar dúvidas na configuração do arquivo Httpd.conf através de dicas. Ocenário contém 8(oito) perguntaselaboradas através do arquivo httpd.conf e do site do Apache, possibilitando ao administrador realizar uma configuração apropriada.A figura 15mostra a tela de como as informações são apresentadas para o administrador.

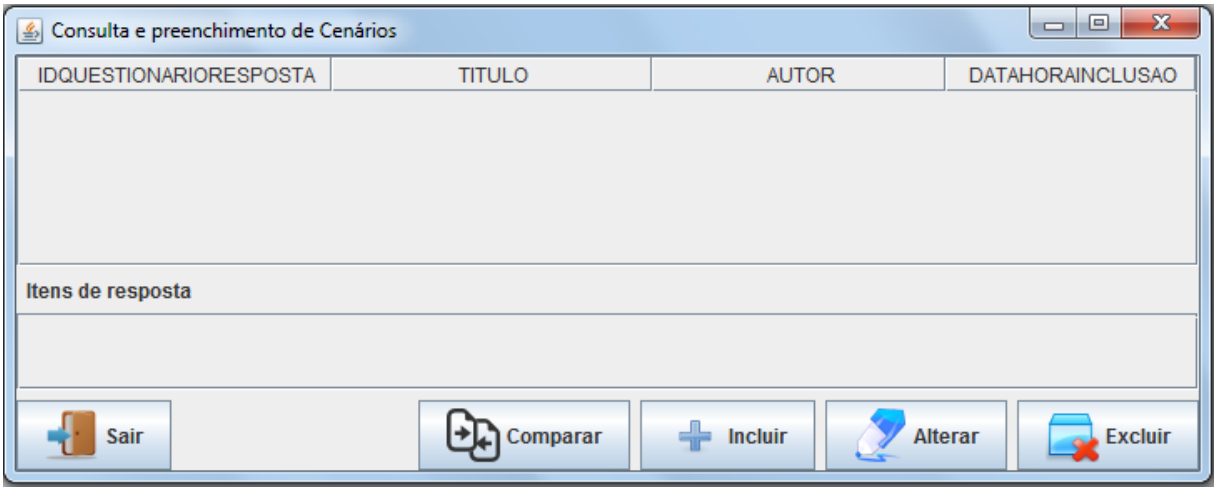


Figura 15 - Repositório de Cenários
Fonte: Próprio autor, 2013.

No sistema de dicas o administrador começa a responder perguntas sobre o arquivo Httpd.conf e com base nas perguntas, o administrador recebe dicas dos parâmetros de configuração para criar o cenário. O botão “Incluir” abre a tela de inclusão de perguntas para o cenário onde o administrador identifica o cenário e preenche as perguntas de acordo com a necessidade da configuração. A imagem 16 representa um exemplo de pergunta com a sugestão de configuração, e logo em seguida o administrador salva e caminha para próxima questão.

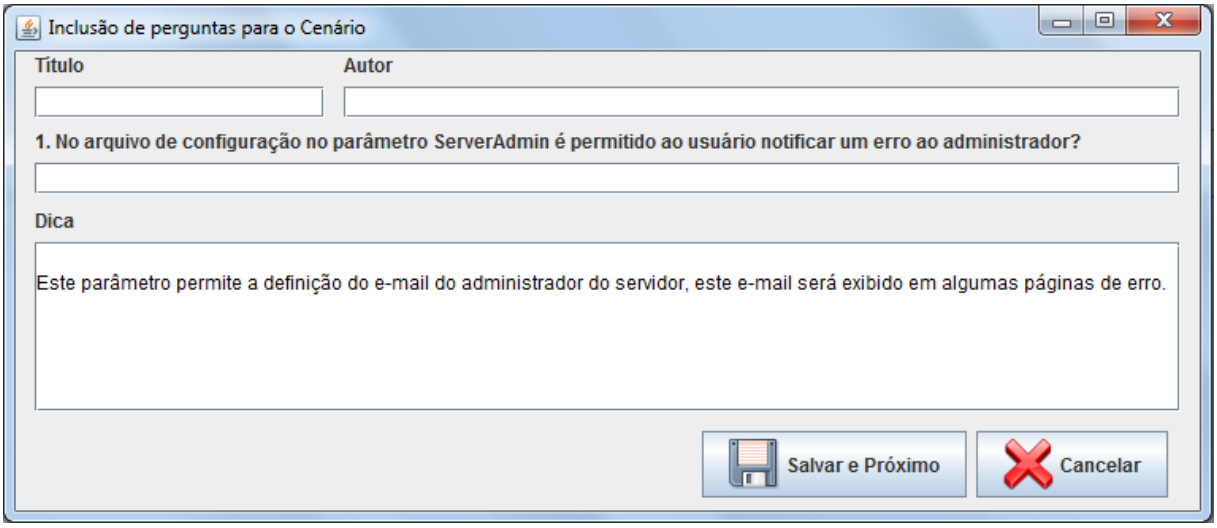


Figura 16 - Inclusão de perguntas para os Cenários.
Fonte: Próprio autor, 2013.

A figura 17 mostra dicas com todas as possibilidades de ativar ou desativar o diretório. O administrador pode escolher as opções, utilizando o modelo do diretório do arquivo padrão ou pode atribuir novas combinações de acordo com a necessidade da configuração.No exemplo o <Directory>...</Directory> possui as seguintes informações geradas pelo arquivo padrão.

<Directory>

Options FollowSymLinks

AllowOverride None

Order deny,allow

Deny from all

</Directory>

Neste caso a diretiva *Options* com a opção *FollowSymLinks* está ativada para receber link simbólico, como a criação de um determinado atalho para um arquivo ou diretório.

Na linha seguinte a diretiva *AllowOverride* com a opção *None* indica se o Apache deve ler os arquivo *.htaccess* neste exemplo está opção está sendo negada com a opção *None.*

Na próxima linha *Order Deny, allow.* O *Order* identifica as regras que o Servidor Apache utiliza para validar um requerimento. Com a opção *deny, allow* a diretiva está sendo negada.E na ultima linha a diretiva *Deny* indica para quais hosts o acesso a qualquer parte do servidor será negado. Neste caso a diretiva é seguida com a opção *from all*, com isso está sendo informado ao servidor para negar o acesso a qualquer parte do servidor a todos os hosts.

Com as dicas, o administrador responde somente o que deseja mudar, no caso não precisa inserir os <Directory>...</Directory>, só informa os valores que pretende alterar ou incluir.

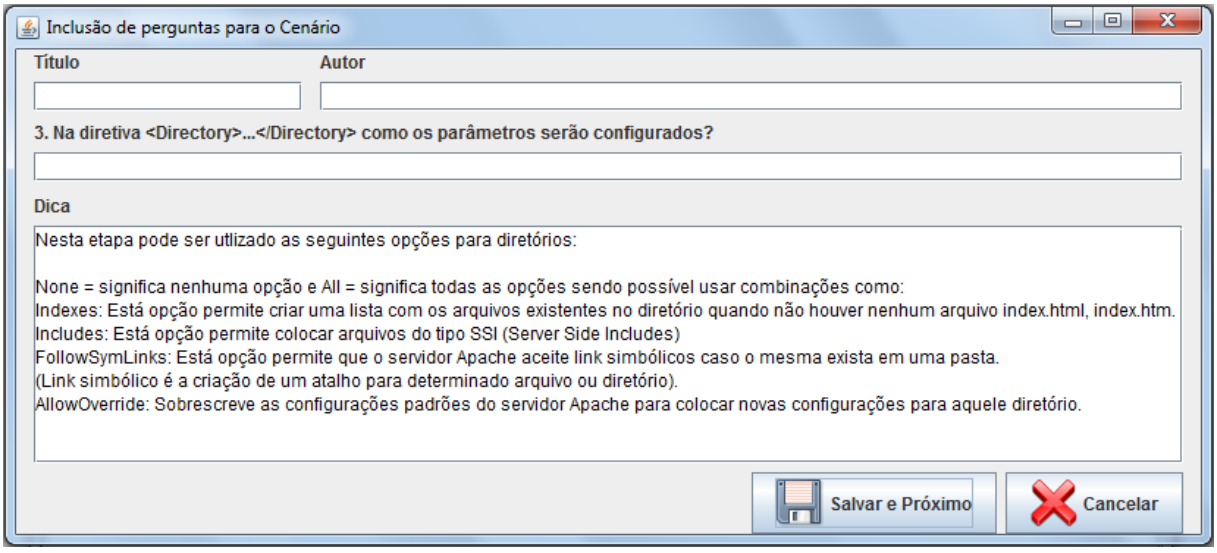


Figura 17 - Inclusão de perguntas para o Cenário.
Fonte: Próprio autor, 2013.

5. RESULTADOS OBTIDOS

Após entender o funcionamento da ferramenta HttpEditor, a partir da leitura do arquivo padrão, processamento dos parâmetros e oscenários respondidos pelo administrador, são abordados nesta seção os resultados obtidos de acordo com os objetivos propostos pelo projeto.

5.1 Configuração.

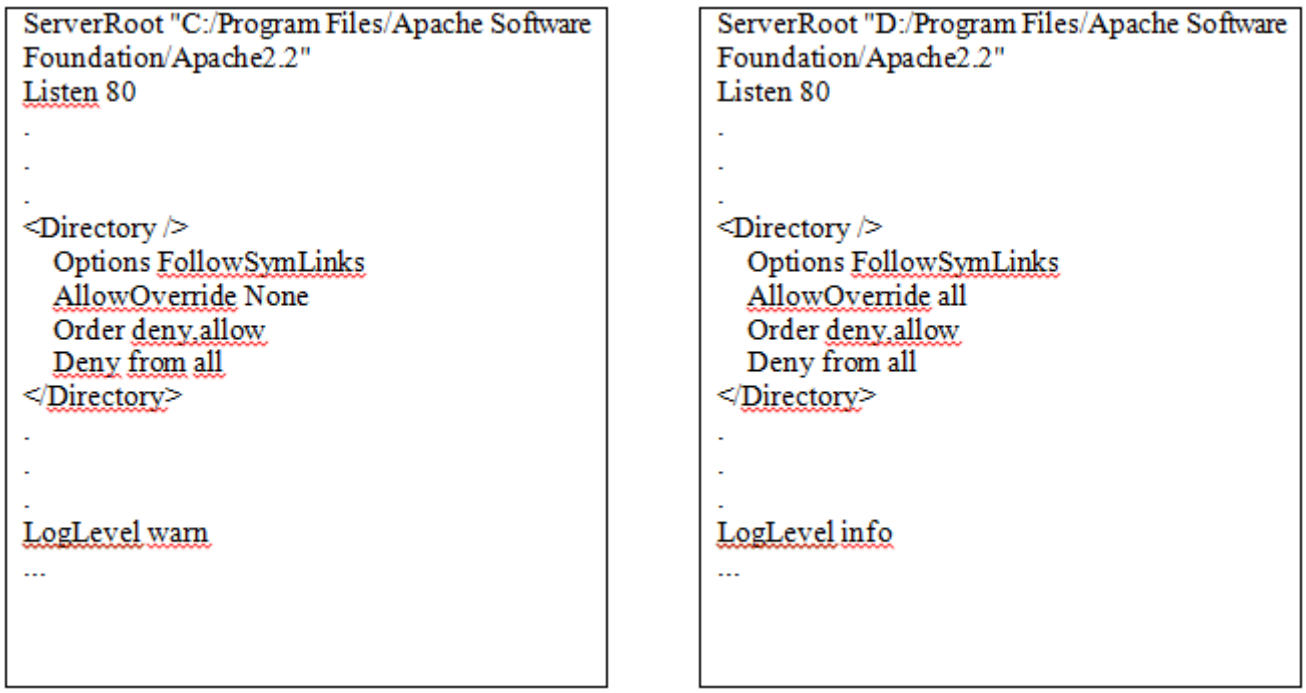
Depois de realizado o cadastro das configurações, o administrador pode realizar a análise dos arquivos e utilizar as configurações adequadas para cada cenário. Na imagem 18, o administrador pode gerar uma comparação entre arquivos dentro do repositório de configuração. A comparação é realizada com os arquivosHttpd.conf e o TesteHome.

Figura 18 - Repositório de Configuração do Apache: Comparar arquivo
Fonte: Próprio autor, 2013.

Para realizar a comparação dos arquivos:

- O administrador acessa a tela de configuração
- Clica no botão comparar
- Seleciona a 1º configuração para realizar a comparação
- Clica no botão comparar
- Seleciona a 2º configuração para realizar a comparação
- Clica no botão comparar
- Se houver diferenças entre os 2(dois) registros, o sistema solicitará a informação do local onde deseja salvar a comparação.

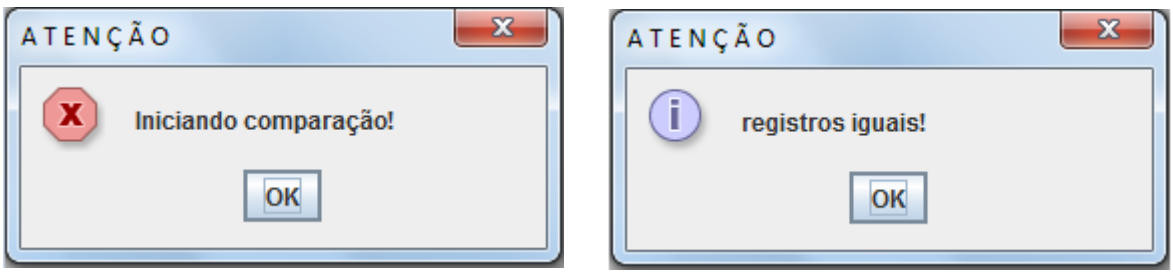
Dessa forma, com as configurações armazenadas, o administrador pode realizar a comparação do arquivo padrão (Figura 19 A)com uma configuração que o administrador tenha cadastrado (Figura 19 B).



A) B)

Figura 19 - A) Http.conf B) TesteHome
Fonte: Próprio autor, 2013.

Neste caso, as informações dos arquivospossuem parâmetros diferentes, ou seja,é informada ao administrador a tela que inicia a comparação (Figura 20 A).Se por acaso o administrador escolher o mesmo arquivo, a ferramenta informa que os arquivos são iguais (Figura 20 B).



A) B)

Figura 20 - A) – Iniciando Comparação B) Registros iguais
Fonte: Próprio autor, 2013.

Em seguida, pedirá ao administrador para informar o local (figura 21) onde os arquivos de erro serão gravados.

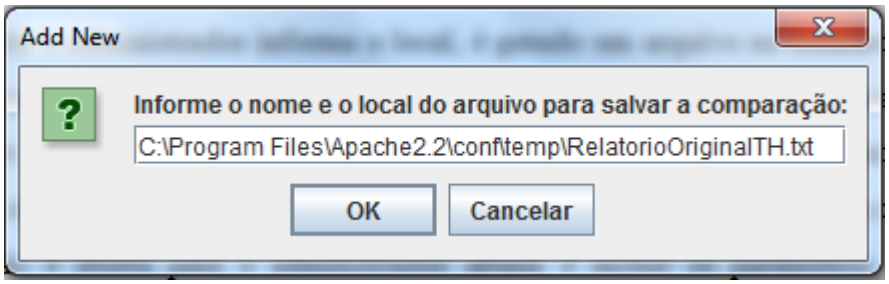


Figura 21 - Salvar a comparação do arquivo
Fonte: Próprio autor, 2013.

Após informar o local, os arquivos são gravados, o arquivo gera os parâmetros que foram modificados na análise feita com arquivo Httpd.conf e o TesteHome, figura 22.

Caso o administrar queira modificar os parâmetros, o mesmo pode alterar a configuração, usando o botão Alterar.

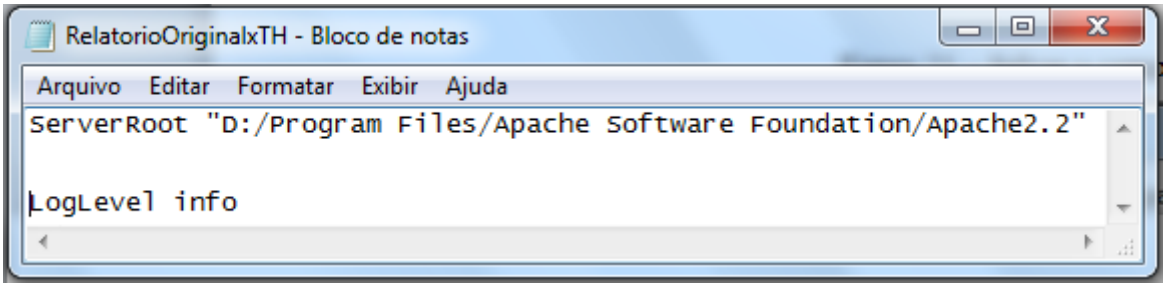


Figura 22 - Relatório OriginalxTH
Fonte: Próprio autor, 2013.

5.2 Cenários.

Para realizar a comparação do cenário com a resposta padrão do arquivo httpd.conf, o administrador visualiza algumas dicas sobre parâmetros correspondentes a cada pergunta e um exemplo de como pode responder a cada pergunta, caso o mesmo queira mudar ou adicionar novos parâmetro.As perguntas e dicaspara responder o cenário são:

1. No arquivo de configuração no parâmetro ServerAdmin é permitido ao usuário notificar um erro ao administrador?

   Sim. Então será exibido o e-mail da pagina de erro.

   Não. Comentar o e-mail para não visualizar a pagina de erro. #desabilitar

- Dica: Este parâmetro permite a definição do e-mail do administrador do servidor, este e-mail será exibido em algumas páginas de erro.

2. No parâmetro ServerName será permitido o uso do domínio DNS ou IP?
   Sim. Terá que inserir o parâmetro ServerName e atribuir DNS ou IP; www.teste.com:80
   Não. Então não será exibido o serverName pois o mesmo já se encontra desabilitado.

- Dica: Este parâmetro é definido automaticamente, se escolher a maneira manual e não obter um domínio DNS utilizar o endereço IP.

3. Na diretiva <Directory>...</Directory> como os parâmetros serão configurados?

```
<Directory />
    Options FollowSymLinks
    AllowOverride all
    Order deny,allow
    Deny from all
</Directory>
```

- Dica: Nesta etapa pode ser utilizado as seguintes opções para diretórios:

  None = significa nenhuma opção e All = significa todas as opções.

  Sendo possível usar combinações como:

  Indexes: Esta opção permite criar uma lista com os arquivos existentes no diretório quando não houver nenhum arquivo index.html, index.htm.

  Includes: Esta opção permite colocar arquivos do tipo SSI (Server Side Includes)

  FollowSymLinks: Esta opção permite que o servidor Apache aceite link simbólico caso o mesmo exista em uma pasta. (Link simbólico é a criação de um atalho para determinado arquivo ou diretório).

  AllowOverride: Sobrescreve as configurações padrões do servidor Apache para colocar novas configurações para aquele diretório.

4. Será permitido o tempo limite da Conexão? Qual será o tempo?
   Sim/Não. Timeout = Tempo escolhido

- Dica: Usar o parâmetro **Timeout** que é definido em segundos, por exemplo Timeout 300 é apropriado para a maioria das situações. Quando habilitado, o mesmo espera por:
  1. A quantidade total de tempo que leva para receber uma requisição GET.
  2. A quantidade de tempo entre o recebimento de pacotes TCP em uma requisição POST ou PUT.
  3. A quantidade de tempo entre ACKs em transmissões de pacotes TCP nas respostas.

5. Serão permitidas conexões persistentes?
   Não/Sim.

- Dica: Usar o parâmetro **KeepAlive***on* permite mais de uma requisição por conexão. **KeepAlive***off*, desativa.

6. Qual o numero máximo de requisições que serão transmitidas durante uma conexão persistente?
   Não.

   MaxKeepAliveRequests: número máximo de conexões persistentes. De 0 a 100.

   Dica: É recomendável que o valor seja alto para um melhor desempenho ou deixar o valor como 0 (zero) para uma quantidade ilimitada.

7. Qual será o número máximo de pedidos simultâneos de clientes que o servidor suportará?
   Não.

- Dica: Usar o parâmetro **MaxClients,** para cada conexão, é criado um processo httpd filho. Após atingir este número máximo de processos, ninguém mais conseguirá se conectar ao servidor web até que um processo filho seja liberado. MaxClients 150

8. Qual o nível das mensagens de erro que serão gravadas no arquivo de log?

- Dica:
  emerg : Grava apenas emergências, geralmente quando o servidor está inutilizável.

  alert : Grava erros que devem ser resolvidos imediatamente.

  crit : Grava erros relacionados a condições críticas, por exemplo, falhas ao tentar abrir um socket.

  warn: Grava erros até o nível de advertência, por exemplo erros recuperáveis pelo servidor.

  notice: Grava mensagens mesmo que não ocorra erro mas que sejam significantes.

  info: Grava até mensagens informativas, indicando soluções para pequenos problemas de performance.

  debug: Grava todas as mensagens, desde erros até a abertura de arquivos de configuração.

Dicas de Segurança:

Modulo ModSecurity

Para ajudar na segurança do servidor é possível adicionar o modulo modsecurity para prevenir alguns tipos de ataques a servidores, como *Cross-Site Scripting* (XSS) e *SQL Injection.*

Ao instalar o pacote do Modsecuriy no Windows requer alguns requisitos.

1. Microsoft Visual Studio C + +
2. Apache 2.2.x
3. Biblioteca PCRE Perl Compatible Regular Expression

O exemplo mostra como é identificado o parâmetro no arquivo http.conf

Ex: <IfModule mod_security.c>
. . .
</IfModule>

Restringir acesso ao Ip.

Para restringir a permissão a Internet para autorizar somente a rede 177.200:

Order Deny, Allow
Deny from all
Allow from 177.200.0.0/26

Por IP, por exemplo:

Order Deny, Allow
Deny from all
Allow from 177.0.0.1

De acordo com dicas sobre os parâmetros do arquivo de configuração e dicas segurança, pode-se destacar como as mais importantes às dicas que negam funcionalidade a diretórios que não são utilizados como visto na questão numero 3(três) e dica do mod_Security que previne alguns tipos de ataques a servidores, visto na dica de segurança.

Com isso, o administrador pode atribuir outros parâmetros e valores para gerar novas configurações de acordo com a necessidade dos cenários. Em seguida, o administrador tem a opção de visualizar todos os cenários e as respostas na tela de consulta e realizar a comparação dos cenários, conforme pode ser visualizado na figura 23.

Neste caso a comparação é realizada com o arquivo TesteEmp, onde o arquivo é comparado com a resposta padrão do arquivo Httpd.conf armazenado no banco de dados.

.

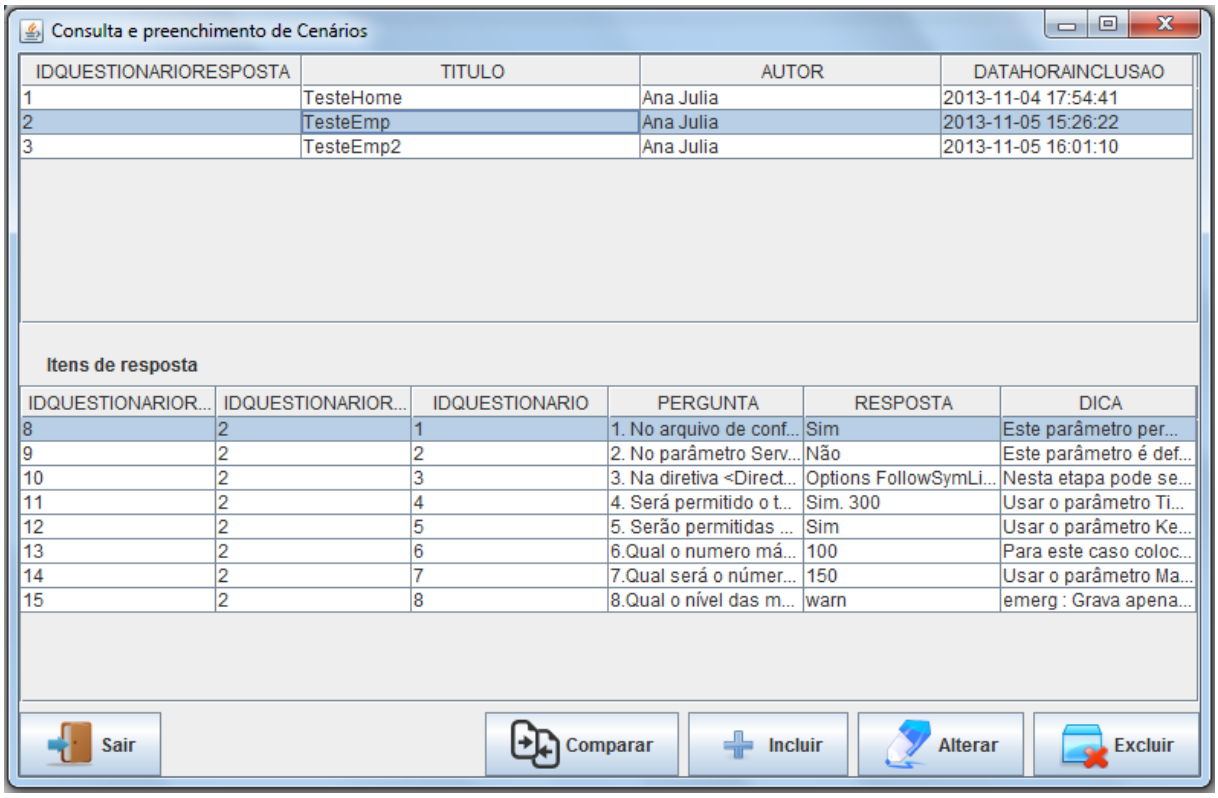


Figura 23- Consulta e preenchimento dos Cenários
Fonte: Próprio autor, 2013.

Assim que escolhido o cenárioque será realizado a comparação, o administrador deve informar o nome e o local onde será salvo o arquivo de erro, conforme pode ser visualizado na figura 24.

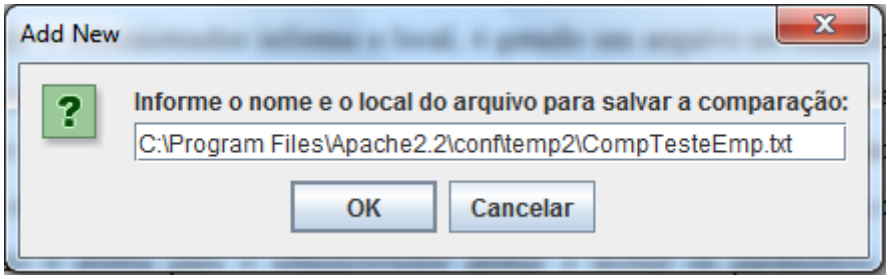


Figura 24 - Salvar a comparação dos Cenários
Fonte: Próprio autor, 2013.

Quando o administrador informa o local, é gerado um arquivo no formato “txt”, que contém as perguntas e respostas dos cenários que são diferentes do arquivo original (figura 25), esse resultado serve para verificar se o administrador realmente deseja alterar e incluir novos parâmetros na configuração. Ao mesmo tempo em que é gerado o arquivo txt,a tela de cadastro de configuração é aberta para o administrador alterar e incluir os parâmetros que foram gerados no arquivo de erro (figura 26) e identificar o título, descrição e autor do arquivo gerado a partir do cenário.

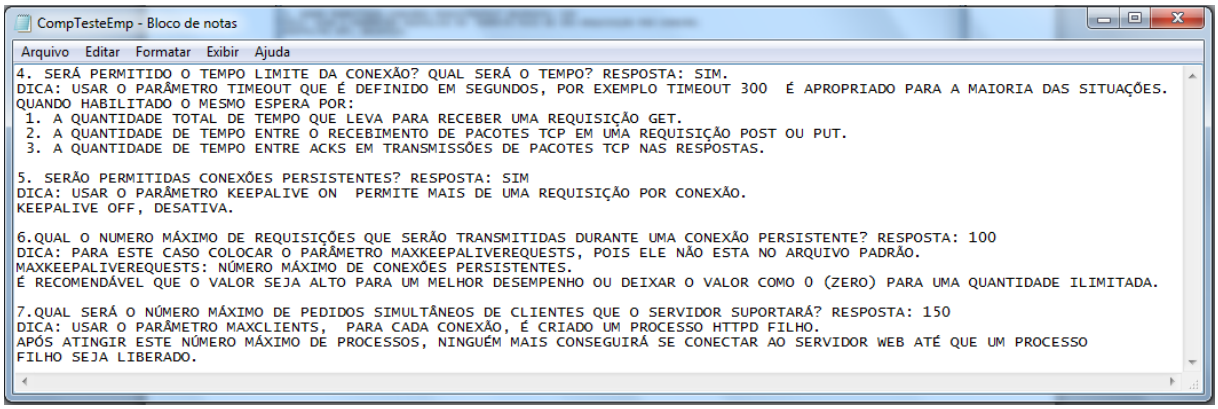


Figura 25 - Relatório CompTesteEmp
Fonte: Próprio autor, 2013.

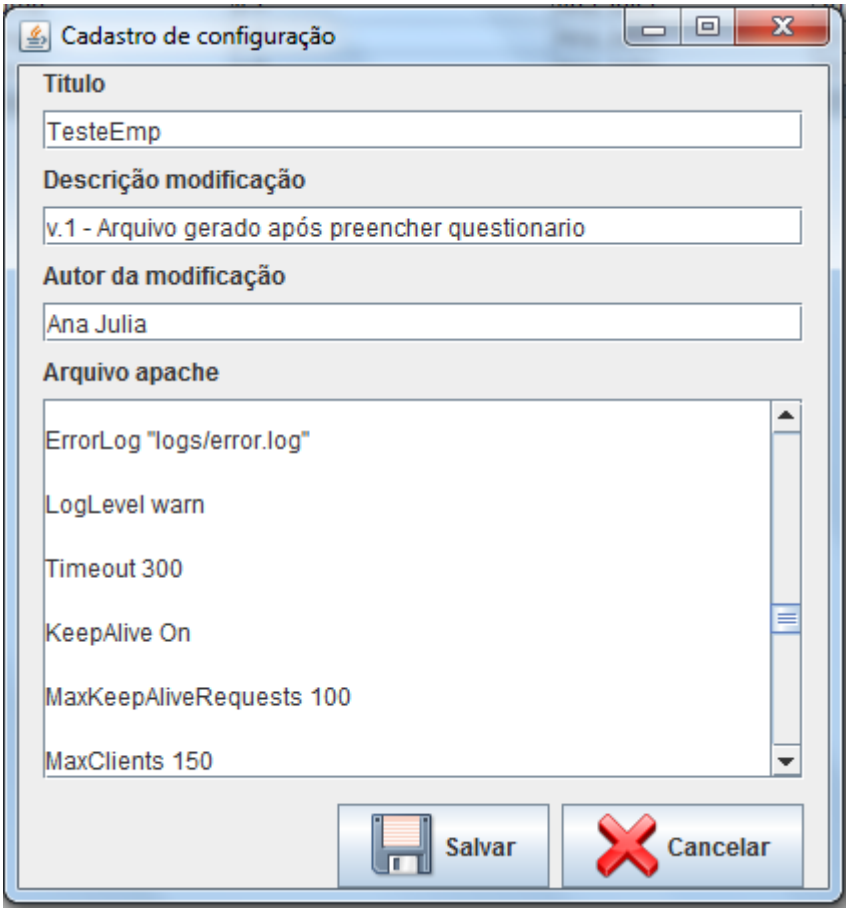


Figura 26 - Cadastro de arquivo gerado pelo Cenário.
Fonte: Próprio autor, 2013.

O arquivo padrão do Httpd.conf é localizado dentro da pasta C:\Users\Apache22\conf, o arquivo quando configurado pelo administrador precisa ser identificado para gerar o arquivo e ser direcionado para executar o servidor, o botão “Gerar arquivo” faz esse processo para gerar o formato que o servidor faz a execução, conforme pode ser visualizado na figura 27.

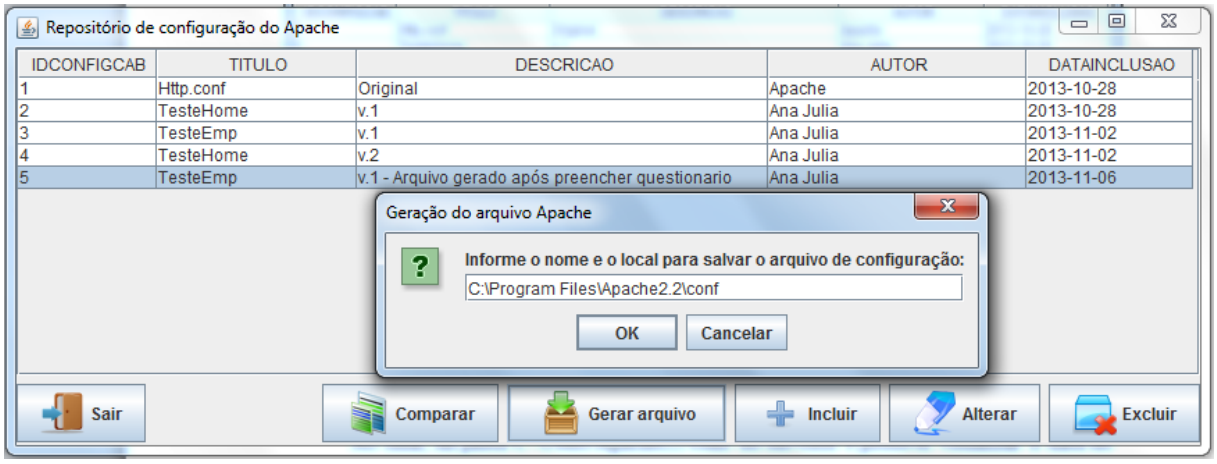


Figura 27 - Gerar Arquivo.
Fonte: Próprio autor, 2013.

Na ferramenta, é possível visualizar outras dicas com o qual o administrador pode optar em usar. A opção de segurança permite que o administrador adicioneo modulo *mod_security* (figura 28) para evitar ataques aos servidores.

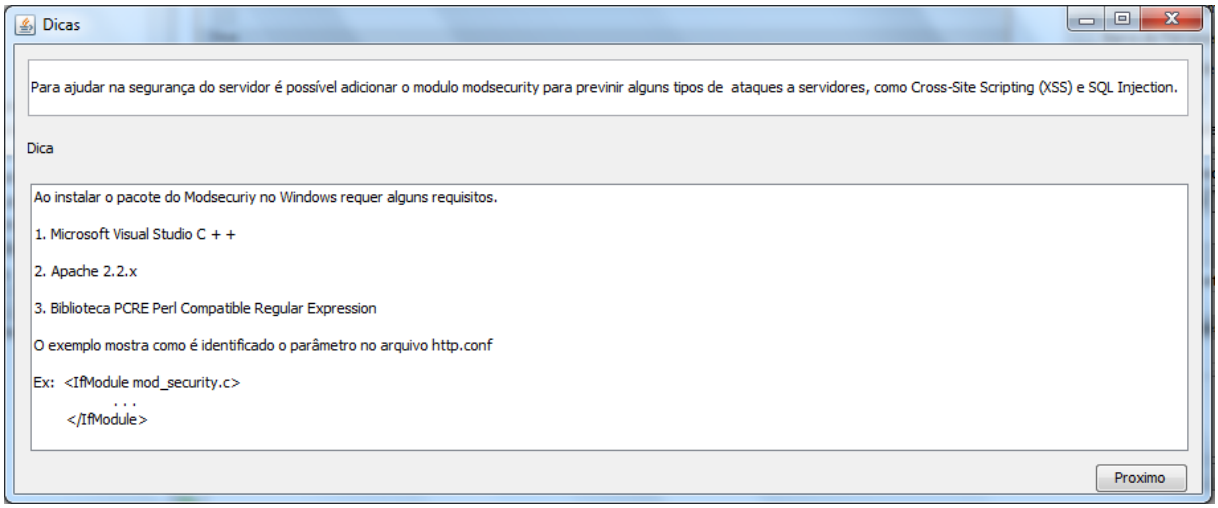


Figura 28 - Dica: Mod_security.
Fonte: Próprio autor, 2013.

Para coleta de resultados foi consultado um adminstrador de servidor para avaliar a ferramenta, onde foi analisado a facilidade em configurar arquivos e análise dos mesmos e as informações do sistema de sugestões. O procedimento utlizado foi a demonstração completa da ferramenta e em seguida foi realizada uma nova configuração feita através das perguntas do cenario onde foram adiconados novos parametros de funcionalidade ao arquivo.

Quanto a usabilidade da ferrametna, a facilidade de cadastrar novos arquivos e criar cenários com base nas dicas, o adminstrador demostrou um alto grau de satisfação. O

adminsitrador também sugeriu para acrescentar mais dicas de seguraça para previnir servidores de atauques maliciosos.

6. CONCLUSÕES

Tendo em vista que algumas empresas são responsáveis pela sua própria manutenção e configuração dos servidores, pode-se notar que alguns casos de falhas acontecem por falta de segurança nos arquivos de configuração, sendo que, em alguns casos o arquivo gerado pelo Apache não satisfaz a necessidade de um determinado ambiente ou às vezes a configuração do arquivo contém informações que não estão sendo utilizadas.

Por isso, foi desenvolvida uma ferramenta que auxilia os administradores a criar configurações a partir do arquivo original do Apache, sendo possível alterar e incluir novos parâmetros e módulos de configuração para cenários customizados, e o administrador pode optar por responder um cenário com perguntas relacionadas ao arquivo, que possibilita ao administrador visualizar algumas dicas de sugestões.

A partir deste trabalho com uma ferramenta desenvolvida para Desktop, espera-se que está ferramenta possa ser desenvolvida para ambiente Web, onde através do meio colaborativo, voluntários possam ajudar a adicionar mais dicas de sugestões no âmbito de aumentar a segurança do servidor de um determinado cenário.

Como trabalho futuro, sugere-se o estudo do servidor GlassFish Server, servidor escrito em Java, o mesmo contém uma ferramenta que gera a configuração padrão ou uma configuração personalizada, com isso, uma análise pode ser gerada sobre a configuração e ajudar administradores a configurar as instruções de instalação para cada método.

Outro trabalho futuro sugerido, é a realização de testes das configurações customizadas para determinado cenário, utilizando o Apache e o GlassFish, validando a melhor configuração e o servidor apropriado para cada aplicação.

REFERENCIAS

ALECRIM, Emerson. Conhecendo o servidor apache. [S.l.], [2006].Disponível em: <http://www.infowester.com/servapach. php>. Acesso em: 26 de Fevereiro de 2013.

ABOUT APACHE. The Apache Software Foundation. Disponível em: < http://httpd.apache.org/ABOUT_APACHE.html> Acesso em 28 de Maio de 2013.

APACHE DERBY.Introdução ao Derby. Disponível em: <http://db.apache.org/derby/docs/dev/pt_BR/getstart/getstartderby.pdf>. Acesso em: 14 de março de 2013.

APACHE HTTP SERVER . The Apache Software Foundation. Disponível em: <http://httpd.apache.org/docs/>Acesso em: 21 de Abril de 2013.

BREACH SECURITY. Disponível em:<http://www.crunchbase.com/company/breach-security>Acesso em 18 de Mio de 2013.

CERT.BR. Estatísticas dos Incidentes Reportados ao CERT.br. Disponível em:< http://www.cert.br/stats/incidentes/2012-jan-dec/tipos-ataque-acumulado.html>. Acesso em: 01 de Março de 2013.

CERT.BR. Estatísticas dos Incidentes Reportados ao CERT.br. Disponível em : <http://www.cert.br/stats/incidentes/2013-apr-jun/analise.html>Acesso em: 20 de Setembro de 2013.

DIAS, Cláudia. Segurança e auditoria da tecnologia da informação. Rio de Janeiro: Axcel Books, 2000.

Diaz. C.C. Instalação e configuração de Apache. Disponivel em:<http://www.criarweb.com/instalacao-configuracao-apache/> Acesso em: 10 de abril de 2013.

ENDRES, R. et al. OWASP Top 10 2013: The Ten Most Critical Web Application Security Risks. 2013.

FERREIRA, Fernando Nicolau Freitas. Segurança da Informação. Rio de Janeiro: Ciência Moderna, 2003.

ICONFIDER, Browse Icons. Disponível em: <www.iconfinder.com> Acesso em: 25 de Outubro de 2013

IIS, Internet Information Services. Disponível em: < http://technet.microsoft.com/pt-br/library/cc753433(v=ws. 10).aspx > Acesso em: 11 de Março de 2013

KLABUNDE, F, Software para Monitoramento de servidores Web Apache. Blumenau, SC, Universidade Regional de Blumenau, 2007.

LUZ, H. J. F. Análise de Vulnerabilidade em Java Web Applications. Marilia, SP: UNIVEM, 2011.

MODSECURITY. Open Source Web Application Firewall. Disponível em: < http://modsecurity.org /> Acesso em: Acesso em 18 de Maio de 2013.

NEWS.NETCRAFT. Internet Security e Data Mining.Disponível em: < http://news.netcraft.com/archives/2013/05/03/may-2013-web-server-survey.html>. Acesso em: 16 de Junho de 2013.

NIC.BR. Ataques a servidores Web. Disponível em < http://nic.br/imprensa/releases/2012/rl-2012--08.htm>Acesso em: 01 de Março de 2013.

OWASP. Open Web Application Security Project. disponível em:< https://www.owasp.org/index.php/Main_Page>

PLACKER. Vulnerabilidade em Aplicações Web. Disponível em: <http://www.securityhacker.org/artigos/item/vulnerabilidades-em-aplicacoes-web>

RED HAT ENTERPRISE LINUX 3: Guia de Administração do Sistema. Disponível em: <http://web.mit.edu/rhel-doc/3/rhel-sag-pt_br-3/ch-httpdconfig.html> Acesso em: 25 de Março de 2013.

SILVA, JOSÉ ALMIR FERREIRA. Servidor Web. Disponível em: <http://www.portaleducacao.com.br/informatica/artigos/17165/como-funciona-um-servidor-web>Acesso em: 28 Abril de 2013.

SYMANTEC CORPORATION. Symantec Internet Security Threat Report: Trends for 2012. Abril 2013. Mountain View, CA.

TANENBAUM, A. S. Sistemas Distribuídos. 2ª edição, paginas 337 à 338 – tópicos 12.2.2

Top Ten. Os dez riszos de segurança mais criticos em aplicações Web. Disponivel em: <http://www.gris.dcc.ufrj.br/OWASP_Top_10_2013_PTBR.pdf> Acesso em: 02de Março de 2013

TRABALHANDO COM O BANCO DE DADOS. Java Derby. Disponível em: <http://netbeans.org/kb/docs/ide/java-db_pt_BR.html>. Acesso em : 14 de Março de 2013.

TRIBUNAL DE CONTAS DA UNIÃO. Boas Praticas em Segurança da Informação, P.26, 2007.Disponível em: < http://portal2.tcu.gov.br/portal/pls/portal/docs/2059162.PDF>Acesso em: 18 de Julho de 2013.

XITAMI. Imatix Corporation. Disponível em: <http://www.xitami.com/ > Acesso em: 11 de Março de 2013.

WOLFGARTEN, S.; ZAIDAN,K; "Proteção de servidores web" Disponível em: <http://lnm.com.br/images/uploads/pdf_aberto/LM_76_70_75_06_capa_apache.pdf > Acesso em: 02 de Maio de 2013