

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

**GERENCIADOR DE DNS VIA WEB**

**TRABALHO DE GRADUAÇÃO**

**Leandro Isaias Lucca**

**Santa Maria, RS, Brasil  
2007**

# **GERENCIADOR DE DNS VIA WEB**

**por**

**LEANDRO ISAIAS LUCCA**

[restinga@inf.ufsm.br](mailto:restinga@inf.ufsm.br)

Trabalho de Graduação  
Curso de Ciência da Computação  
Universidade Federal de Santa Maria (UFSM)

Requisito parcial para obtenção do grau de  
**Bacharel em Ciência da Computação**

**Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Roseclea Duarte Medina**

**Trabalho de Graduação N° 228  
Santa Maria, RS, Brasil  
2007**

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

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

**GERENCIADOR DE DNS VIA WEB**

**Elaborado por**

**LEANDRO ISAIAS LUCCA**  
[restinga@inf.ufsm.br](mailto:restinga@inf.ufsm.br)

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

**Comissão Examinadora**

---

**Prof<sup>a</sup>. Dr<sup>a</sup>. Roseclea Duarte Medina**  
**(Orientadora)**

---

**Prof<sup>a</sup>. Dr<sup>a</sup>. Andrea Schwertner Charão**

---

**Prof<sup>a</sup>. Mestre Oni Reasilvia de O. Sichonany**

**Santa Maria, RS, Brasil**  
**2007**

## AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me dado força em todas as etapas da minha vida.

Agradeço a meus pais Luiz e Catarina pelo apoio, amor e carinho que sempre tiveram comigo. Eles sempre estiveram do meu lado em inúmeros momentos de minha vida, me deram força para que eu pudesse lutar, então dedico a eles este trabalho.

Ao meu irmão Daniel, embora estivesse sempre querendo utilizar o computador quando eu precisava fazer trabalhos, agradeço pelo carinho e compreensão.

A minha namorada Carina, que sempre me deu forças, me entendendo nas horas difíceis dessa caminhada, sempre tentando me ajudar com suas palavras de estímulo. Nininha, agradeço por ter sido tão prestativa comigo, agradeço pelo amor e carinho que sempre dedicastes a mim. Este TG não é só meu, também é um pouco teu. Muito obrigado!

Agradeço também a meus familiares e amigos que sempre estiveram por perto para me dar apoio nessa caminhada. Em especial gostaria de agradecer ao Diego e ao Cechin pela amizade que sempre tivemos durante a faculdade. Mas, não poderia deixar de agradecer ao seu Rogério e a dona Salete que apostaram e confiaram em nossos potenciais sempre nos dando o maior apoio e compreendendo os momentos que não podíamos nos dedicar totalmente ao trabalho.

Aos professores, agradeço por darem um rumo as nossas vidas, nos auxiliando sempre que necesserário. Em especial devo agradecer as professoras Rose, Andréa e Oni, por estarem sempre me ajudando nessa caminhada.

*“Meu muito obrigado a todos que de uma maneira ou de outra me deram força para que eu estivesse aqui neste momento.”*

## RESUMO

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO  
TRABALHO DE GRADUAÇÃO

### GERENCIADOR DE DNS VIA WEB

Autor: Leandro Isaias Lucca  
Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Roseclea Duarte Medina  
Local e Data de Defesa: Santa Maria, 27 de Fevereiro de 2007

Com a grande evolução da Internet, fez-se necessária a implementação de um método que pudesse encontrar qualquer componente de rede. Essa idéia foi expandida com a criação de um sistema de resolução de nomes de domínio: o protocolo DNS (*Domain Name System*).

O DNS é um dos serviços mais críticos em redes de computadores. Qualquer falha no mesmo pode vir acarretar a perda parcial ou total na funcionalidade de outros serviços.

O desenvolvimento da atual ferramenta, foi implementado para a *web* através de uma interface claramente interativa e amigável de modo a tornar-se prática e eficiente para o usuário final do sistema, visando o gerenciamento do servidor de DNS intitulado BIND (*Berkeley Internet Name Domain*) que é o mais utilizado atualmente.

A implementação do sistema foi efetivada em linguagem PHP, utilizando-se HTML, XML e Javascript de modo a usufruir de suas funcionalidades segundo as reais necessidades encontradas no decorrer do desenvolvimento.

Após a implementação foi efetuada uma seqüência de testes até a conclusão e a implantação do mesmo em versão beta (versão de teste).

Concluindo, o presente projeto pretende se adequar as reais necessidades dos usuários do BIND, através de um gerenciamento simples e dinâmico, de modo a facilitar suas vidas, abstraindo dos mesmos uma imensa gama de conceitos necessários.

**Palavras-chave:** protocolo; DNS; BIND; gerenciamento; WEB; IP.

# **ABSTRACT**

FEDERAL UNIVERSITY OF SANTA MARIA  
COURSE OF COMPUTER SCIENCE  
GRADUATION WORK

## **WEB DNS MANAGER**

Author: Leandro Isaias Lucca  
Advisor: Prof<sup>a</sup>. Dr<sup>a</sup>. Roseclea Duarte Medina  
Local and Defense Date: Santa Maria, February 27, 2007

With the great evolution of the Internet, the implementation of a method became necessary that could find any network component. This idea was expanded with the creation of a system of domain names resolution: DNS (Domain Name System) protocol.

The DNS is one of the most critical services in computer networks. Any fail in the same can come to cause the average loss or total the functionality of other services.

The development of the current tool, was implemented for web through an interactive and clearly friendly interface in order to become practical and efficient for the final user of the system, being aimed at the management of the DNS server intitled BIND (Berkeley Internet Name Domain) that it is used currently.

The implementation of the system was accomplished in language PHP, using itself HTML, XML and Javascript in order to usufruct of its functionalities according real necessities found in elapsing of the development.

After the implementation was effected a sequence of tests until the conclusion and the implantation of the same in beta version (test version).

Concluding, the present project intends to adjust the real necessities of the users of the BIND, through a simple and dynamic management, in order to facilitate its lives, abstracting from same an immense gamma of necessary concepts.

**Keywords:** protocol; DNS; BIND; management; WEB; IP.

# ÍNDICE

<b>LISTA DE FIGURAS .....</b>	<b>9</b>
<b>LISTA DE TABELAS .....</b>	<b>11</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>12</b>
<b>1 INTRODUÇÃO .....</b>	<b>13</b>
<b>2 REFERENCIAL TEÓRICO.....</b>	<b>15</b>
2.1 DNS .....	15
2.1.1 Padrões Determinantes .....	16
2.1.2 Zona.....	16
2.1.3 Domínio.....	17
2.1.4 Subdomínio .....	17
2.1.5 Nome de Domínio/Subdomínio .....	17
2.1.6 Comparativo: Zona X Domínio X Subdomínio .....	18
2.1.7 Tipos .....	19
2.1.8 Servidor de <i>Cache</i> .....	19
2.1.9 Servidor Primário ou <i>Master</i> (Mestre) .....	21
2.1.10 Servidor Secundário ou <i>Slave</i> (Escravo).....	21
2.1.11 Estrutura Organizacional e Hierarquia do Banco de Dados .....	22
2.1.12 Registro de Recurso .....	23
2.1.13 DNS Direto .....	24
2.1.14 DNS Reverso .....	24
2.1.15 DNS Dinâmico.....	25
2.1.16 Funcionamento Prático.....	26
2.2 BIND .....	27
2.2.1 Encaminhamentos ( <i>forwarders</i> ) .....	27
2.2.2 Update Dinâmico e DNS Seguro.....	28
2.2.3 Transferência Incremental de Zona ou Incremental Zone Transfers (IXFR).....	28
2.2.4 Arquivo de inicialização ( <i>named.conf</i> ).....	28
2.2.5 Arquivo de <i>cache</i> ( <i>raiz</i> ).....	30
2.2.6 Formato do arquivo de zona .....	31
2.2.7 Arquivo de configuração de zona direta .....	33
2.2.8 Arquivo de configuração de zona reversa .....	33
2.2.9 Arquivos de interface local.....	34
2.2.10 RNDG.....	34
<b>3 TRABALHOS RELACIONADOS .....</b>	<b>36</b>
3.1 FreeDNS.....	36

3.2	Zoneedit.....	36
3.3	XName.....	36
3.4	Webmin.....	37
<b>4</b>	<b>DESENVOLVIMENTO .....</b>	<b>38</b>
4.1	Estruturação do Código-Fonte.....	38
4.2	Autenticação e Segurança.....	40
4.3	Layout.....	41
4.4	Usuários e Permissões.....	42
4.5	Definição da Interface de Gerenciamento.....	43
4.5.1	Interface de Gerenciamento no Modo Administrador.....	44
4.5.2	Interface de Gerenciamento no Modo Usuário.....	47
4.6	Topologia.....	51
4.7	Comunicação entre os Servidores de Nome .....	52
4.8	Atualizações Dinâmicas.....	52
4.9	Acesso Diretamente ao BIND.....	52
4.10	Documentação.....	53
4.11	Implantação e Considerações Finais.....	54
<b>5</b>	<b>RESULTADOS.....</b>	<b>55</b>
5.1	Testes de Software.....	55
5.2	Comparativo entre Ferramentas de Gerenciamento de DNS.....	57
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>59</b>
<b>7</b>	<b>TRABALHOS FUTUROS.....</b>	<b>61</b>
<b>8</b>	<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>62</b>
	<b>GLOSSÁRIO.....</b>	<b>66</b>
	<b>ANEXOS .....</b>	<b>71</b>
	ANEXO I - Exemplo de Arquivo de Inicialização “named.conf” incluindo Arquivo de Encaminhamentos.....	72
	ANEXO II - Exemplo de Arquivo de Configuração de Zona Reversa.....	75
	ANEXO III - Exemplo de Arquivo Cache (Raiz).....	76
	ANEXO IV - Exemplo de Arquivos de Interface Local.....	78

## LISTA DE FIGURAS

<b>Figura 1</b>	DNS básico.....	15
<b>Figura 2</b>	Comparativo entre os tipos de servidores de DNS (a) Servidor de <i>Cache</i> (b) Servidor Primário (c) Servidor Secundário.....	19
<b>Figura 3</b>	Sistema de consulta a servidores de <i>cache</i> de DNS.....	20
<b>Figura 4</b>	Servidor secundário interagindo corretamente com o primário.....	22
<b>Figura 5</b>	Demonstração de um exemplo simples de hierarquia em árvore de domínios do banco de dados do DNS.....	23
<b>Figura 6</b>	Demonstração de um exemplo completo de hierarquia de árvore de domínios do banco de dados do DNS.....	23
<b>Figura 7</b>	DNS direto .....	24
<b>Figura 8</b>	Apresentação fictícia passo a passo da “conversa” cliente-servidor para a resolução de nomes .....	25
<b>Figura 9</b>	DNS reverso .....	25
<b>Figura 10</b>	Processo de interrogação ao DNS em seqüência da resolução do nome <code>www.clubedasredes.eti.br</code> .....	26
<b>Figura 11</b>	Exemplo do arquivo de configuração “ <code>named.conf</code> ” em um servidor de nomes somente autoritativo .....	30
<b>Figura 12</b>	Exemplo do arquivo de configuração “ <code>named.conf</code> ” em um servidor de nomes somente <i>cache</i> .....	30
<b>Figura 13</b>	Formato de linhas de registro de recurso .....	31
<b>Figura 14</b>	Arquivo de configuração da zona direta <code>teste.com.br</code> .....	34
<b>Figura 15</b>	Arquivo de configuração da zona reversa <code>168.194.1.in-addr.arpa</code> .....	34
<b>Figura 16</b>	Estruturações do código-fonte do GDNS.....	39
<b>Figura 17</b>	Modelo estrutural do parser.....	40
<b>Figura 18</b>	Tela de autenticação em área restrita .....	41
<b>Figura 19</b>	Tela inicial do GDNS .....	42
<b>Figura 20</b>	Visão geral da interface de gerenciamento do GDNS .....	43
<b>Figura 21</b>	Privilégios/funções no modo administrador .....	44
<b>Figura 22</b>	Tela de configuração estrutural do GDNS no modo administrador.....	45
<b>Figura 23</b>	Tela de criação e gerenciamento de usuários do GDNS no modo administrador.....	45

<b>Figura 24</b> Tela de criação e gerenciamento de zonas do GDNS no modo administrador.....	46
<b>Figura 25</b> Tela de configuração inicial do GDNS no modo administrador .....	47
<b>Figura 26</b> Tela de encaminhamentos do GDNS no modo administrador.....	47
<b>Figura 27</b> Privilégios no modo usuário.....	48
<b>Figura 28</b> Tela de informações pessoais do GDNS no modo usuário .....	49
<b>Figura 29</b> Tela de configurações iniciais do GDNS no modo usuário.....	49
<b>Figura 30</b> Tela de criação e gerenciamento de zonas do GDNS no modo usuário .	49
<b>Figura 31</b> Tela de criação e gerenciamento de recursos de rede.....	50
<b>Figura 32</b> Diagrama de seqüência da interface de criação e gerenciamento de recursos de rede. ....	50
<b>Figura 33</b> Topologia de organização, em forma de anel, dos servidores de nome..	51
<b>Figura 34</b> Tela de ajuda sobre configurações estruturais do GDNS.....	53

## LISTA DE TABELAS

<b>Tabela 1</b>	Comparativo entre as nomenclaturas Zona, Domínio e Subdomínio.....	18
<b>Tabela 2</b>	Regras básicas de configuração do arquivo de zonas .....	31
<b>Tabela 3</b>	Comparativo entre as ferramentas de gerenciamento existentes no mercado e a ferramenta desenvolvida neste projeto.....	58

## LISTA DE ABREVIATURAS E SIGLAS

<b>A</b>	Address
<b>ACL</b>	Access Control List
<b>CNAME</b>	Canonical Name
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>DNS</b>	Domain Name System
<b>GDNS</b>	Gerenciador de DNS
<b>HINFO</b>	Hardware Information
<b>HTML</b>	HyperText Markup Language
<b>ICANN</b>	Internet Corporation for Assigned Names and Numbers
<b>IP</b>	Internet Protocol
<b>MX</b>	Mail Exchanger
<b>NS</b>	Name Server
<b>OSI</b>	Open Systems Interconnection
<b>PHP</b>	Hypertext Preprocessor
<b>PTR</b>	Pointer
<b>RFC</b>	Request for Comments
<b>RNDC</b>	Remote Name Daemon Control
<b>RP</b>	Responsible People
<b>SOA</b>	Start of Authority
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol
<b>TTL</b>	Time To Live
<b>TXT</b>	Text
<b>URL</b>	Universal Resource Locator
<b>WEB</b>	World Wide Web (WWW)
<b>XML</b>	EXtensible Markup Language

## 1 INTRODUÇÃO

Devido à grande evolução da Internet nos últimos anos e o considerável aumento do uso de *intranets*, fez-se necessária uma organização mais estruturada dos componentes de ambas as redes, para que os mesmos pudessem efetuar suas comunicações de forma mais rápida e precisa.

Um problema que se encadeou juntamente com o aumento e conseqüentemente com a evolução das redes de computadores foi: o que e como fazer para identificar um computador numa rede imensa como a Internet? A solução para este problema foi obtida utilizando-se a arquitetura TCP/IP (TORRES [29]) e aplicações desenvolvidas com o propósito de identificar componentes na rede através do protocolo IP.

O protocolo IP é utilizado em redes de computadores para fazer a identificação de cada computador em uma determinada rede. A esta identificação dá-se o nome de IP (TANENBAUM [26]). Então para que um computador possa ser localizado por outro em uma rede é necessário que o outro computador tenha conhecimento sobre o número IP do computador que quer acessar.

Uma tarefa difícil para um ser humano é a possibilidade de memorizar números. A partir disso, surgiu a idéia de atribuir nomes para cada componente de rede, sugerindo assim uma forma mais eficiente de se manter a identificação com uma grande quantidade de IPs.

O surgimento desta idéia gerou um grande problema: como gerenciar uma grande quantidade de nomes em uma rede como a Internet? A proposta que melhor se adaptou a essa idéia e que é utilizada atualmente foi a criação de um sistema de resolução de nomes de domínio ou DNS, que tem a finalidade de resolver nomes de computadores ligados a uma rede.

O DNS é estruturado em um modelo cliente-servidor (TANENBAUM & STEEN [28]), onde é possível a qualquer componente da rede, encontrar qualquer outro que esteja disponível e ainda de maneira rápida, eficiente e descentralizada. O cliente faz uma requisição a um servidor global, depois a um sub-global e assim por diante, até que o referido componente seja localizado.

O foco principal deste projeto está centrado no servidor de DNS. Será utilizado o BIND (*Berkeley Internet Name Domain*), desenvolvido pela Internet

Software Consortium [14] por ser o mais popularmente conhecido.

Enfim, tendo como base a importância do gerenciamento e do bom funcionamento das redes de computadores, o presente projeto visa integrar o servidor BIND com a praticidade e comodidade que, de acordo com TANENBAUM [23], a *WEB* proporciona, criando assim uma interface mais amigável para o gerenciamento do DNS.

Diante da crescente demanda do mercado de *webhosting*, torna-se evidente a necessidade de um gerenciamento conforme propõe este projeto e para que isto ocorra é imprescindível que seja feita uma correta configuração do servidor de DNS, evitando assim problemas graves pela falha na resolução de nomes.

Essa necessidade despertou uma grande motivação para o desenvolvimento de uma ferramenta para auxiliar no gerenciamento do DNS, eliminando o custo com licença da maioria dos gerenciadores proprietários existentes no mercado em uma interface mais simples, prática e eficiente.

Outro estímulo que também pode ser citado é a necessidade da empresa Conplan Sistemas, a qual não utiliza nenhum gerenciador prático em seus servidores de hospedagem.

De maneira geral, foi visado o desenvolvimento *WEB* da ferramenta denominada GDNS, para o gerenciamento do servidor de DNS intitulado BIND, com o propósito de facilitar e simplificar a utilização do mesmo, diminuindo assim o custo e aumentando a flexibilidade para adequação em servidores de hospedagem.

O desenvolvimento baseado na *WEB* permitirá o gerenciamento a partir de qualquer parte do mundo e a qualquer hora, tornando-o mais prático e eficiente com o uso de uma interface amigável.

Enfim, visando as reais necessidades de uma ferramenta de gerenciamento nestes moldes, a mesma foi desenvolvida no decorrer da organização deste documento. Na seção 2 será tratado o referencial teórico que é a base para o desenvolvimento a ser tratado na seção 3. Nessa seção será apresentado o desenvolvimento seguido das etapas necessárias e do modelo de implementação e organização do sistema. Já a seção 4 mostrará os resultados obtidos após o desenvolvimento da ferramenta em questão. Nas seções seguintes, será efetuada a conclusão do trabalho, as tendências de desenvolvimento de trabalhos futuros e os anexos.

## 2 REFERENCIAL TEÓRICO

Nesta seção será analisada a parte conceitual e técnica dos processos que estarão envolvidos com a metodologia de implementação do gerenciador de DNS.

### 2.1 DNS

Basicamente um servidor de nomes converte nomes de *hosts* em endereços IP, ou vice-versa, como pode ser visto na Figura 1.

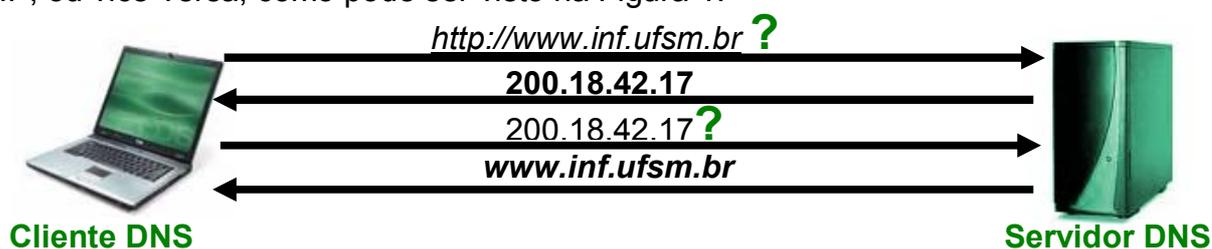


Figura 1 DNS básico

Segundo TANENBAUM [24], a essência do DNS é a criação de um esquema hierárquico de nomes baseado no domínio de um sistema de bancos de dados distribuídos para implementar esse esquema de nomenclatura. Ele é usado principalmente para mapear nomes de *hosts* em endereços IP, porém também pode ser usado para outros objetivos como mapear endereços IP para nome de *hosts*, identificar os serviços disponíveis na rede, identificar o responsável por tal componente ou até mesmo descrever informações sobre o hardware e software utilizados pelo *host* em questão.

Ainda, segundo TANENBAUM [24], o DNS é constituído por servidores de nomes (*nameservers*), e resolvedores (*resolvers*). O resolvedor é um procedimento de biblioteca que quando invocado por um programa aplicativo, entra em contato com o servidor de nomes passando a ele como parâmetro o seu nome. O servidor de nomes retorna o endereço IP após efetuar a requisição do resolvedor, que o retorna para a aplicação que fez a chamada.

O DNS é o responsável pela maioria das confusões e problemas básicos da Internet, sendo provavelmente o mais importante serviço que define a qualidade de um provedor.

Conforme TANENBAUM [24], o protocolo DNS pertence a camada de

aplicação, portanto é de interação com o usuário, ou seja, é o usuário que determina como o DNS funciona. Assim existe a necessidade de configuração por parte do usuário, o que não é uma tarefa fácil devido a grande gama de opções implementadas pelos softwares de gerenciamento.

Devido a grande aceitação do BIND pela maioria dos servidores de DNS da Internet, suas facilidades e flexibilidades, foi implementado um gerenciador do servidor de DNS de forma simples, prática e ainda, via WEB, utilizando todas as vantagens que o sistema, como um todo, pudesse nos oferecer.

Antes de configurar um servidor DNS, é necessário entender a gama de conceitos relacionados com o mesmo, evitando assim problemas com o funcionamento dos serviços da rede. Pensando nisso descreveremos nas subseções a seguir as informações necessárias para a correta compreensão do DNS.

### **2.1.1 Padrões Determinantes**

Os padrões atuais que ditam as regras para o DNS são baseados principalmente nas seguintes RFCs:

- RFC1034 (MOCKAPETRIS [20]) – Nomes de Domínio – Conceitos, facilidades e instalação;
- RFC1035 (MOCKAPETRIS [19]) – Nomes de Domínio – Implementação e especificação.

A descrição das RFCs acima citadas, não será apresentadas no decorrer deste projeto, visto que a implementação será estruturada em dependência com o BIND e de acordo com a Internet Systems Consortium, Inc [13], o mesmo já possui suas funcionalidades e sua especificação baseado nas RFCs acima citadas.

### **2.1.2 Zona**

Zona pode ser entendida como a região sob o domínio de certa entidade ou área com nomes e endereços IP correlacionados.

Conforme a Internet Systems Consortium, Inc [13], uma zona é definida como uma região da árvore de domínios que abrange informação necessárias ao servidor de nomes para a delegação de nomes para um certo domínio.

### 2.1.3 Domínio

De acordo com a Wikipédia, a enciclopédia livre [33], domínio é um nome que serve para localizar e identificar conjuntos de computadores na Internet. O que não é muito preciso em relação ao uso para localização e identificação de computadores ou até mesmo dispositivos móveis em rede local.

Com isso, domínio, genericamente pode ser entendido como o espaço computacional que abrange de forma hierárquica nomes de *hosts* em suas respectivas zonas.

### 2.1.4 Subdomínio

Conforme afirma LANDEMAINE [16], a parte do endereço que segue o domínio na hierarquia é o subdomínio. O mesmo é o responsável pela divisão heterogênea do domínio em partes correlacionadas.

### 2.1.5 Nome de Domínio/Subdomínio

O nome do subdomínio segue as mesmas regras de nome que o do domínio, levando em consideração que o subdomínio está abaixo na hierarquia do domínio.

De acordo com TANENBAUM [25], o nome de um componente de domínio (*label*) deve conter, no máximo, 63 caracteres, não podendo exceder a um total de 255 caracteres para o nome completo de domínio ou subdomínio.

Tomando como base a realidade de nomes na Internet, nomes não devem conter acentos ou caracteres especiais, bem como não importa a caixa da letra, ou seja, são *case insensitive*. Um *label* pode ser um conjunto de caracteres que inicia com uma letra, terminando em letra ou número, podendo conter letras, números ou hífens no meio. Um nome é composto de *labels* separados por pontos.

Em um servidor de nomes, a distribuição e a hierarquia do banco de dados é estruturada no modelo de árvore.

Segundo a distribuidora de software Conectiva Linux [8] e de acordo com a hierarquia de árvore na qual a distribuição do banco de dados do DNS é estruturada, quando o domínio raiz aparece no nome de domínio de um ramo, o nome parece terminar com um ponto, quando, na verdade, ele termina com um ponto e o domínio raiz, que é representado por uma seqüência nula de caracteres. Assim, alguns

programas interpretam um ponto no final de um nome de domínio como significando que o nome de domínio é absoluto. Um nome de domínio absoluto identifica a localização do ramo na hierarquia sem ambigüidade. Nomes não seguidos por pontos são chamados de nomes relativos, já que eles não especificam sua localização exata na árvore, podendo, assim, ser relativos a diversos pontos dela.

Nomes de domínio podem ser comparados com sistemas de arquivos de um sistema operacional. Estes nomes seguem a uma hierarquia interna e por isso são facilmente localizados em toda estrutura, o que facilita a organização dos nomes de domínio.

### 2.1.6 Comparativo: Zona X Domínio X Subdomínio

Estas nomenclaturas estão intimamente relacionadas e exigem uma correta distinção. Para isso recorreremos à Tabela 1, a qual faz o comparativo entre as três nomenclaturas e suas peculiaridades.

**Tabela 1** Comparativo entre as nomenclaturas Zona, Domínio e Subdomínio

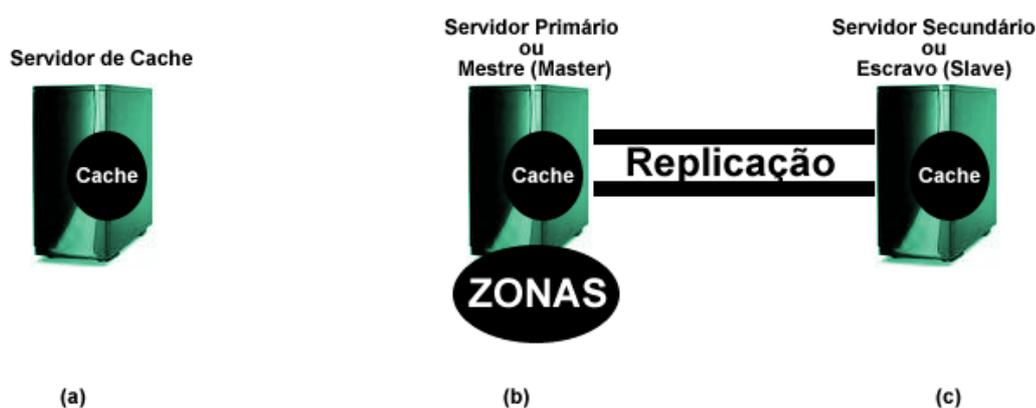
<b>Zona</b>	<b>Domínio</b>	<b>Subdomínio</b>
Área delegada para um domínio, sendo ela parte do domínio ou o domínio como um todo. Por outro lado, pode ser considerada a região sobre o domínio.	Espaço de nomes onde podem ser encontradas várias zonas. Nome utilizado para identificar e organizar zonas com o mesmo propósito.	Parte do domínio que o restringe a um determinado propósito. Subdivisão do domínio com outro nome como um novo ramo na árvore de nomes de domínio.
Ex.: Zona .BR Zona .UFISM.BR	Ex.: ufsm.br	Ex.: inf.ufsm.br
Hierarquia interna ao domínio e ao subdomínio.	Hierarquia externa.	Hierarquia interna ao domínio.
Mantém domínios, subdomínios e aliases.	Mantém subdomínios e aliases.	Mantém aliases.
Nomenclatura variável.	Nomenclatura restrita.	Nomenclatura restrita.
Utilizada pelo DNS.	Utilizada no cotidiano.	Utilizada no cotidiano.
Faz analogia com arquivo em um sistema de arquivos.	Faz analogia com diretório em um sistema de arquivos.	Faz analogia com sub-diretório em um sistema de arquivos.

Coordenada pelo administrador do sistema do servidor DNS.	Coordenado pelo ICANN [12] , o qual, segundo o guia de registro de domínio e hospedagem de sites: 1º Site [1] é responsável por coordenar o funcionamento de domínios na Internet.	Coordenada pelo administrador do domínio pelo qual o subdomínio deve ser mantido.
---	--	---

### 2.1.7 Tipos

Segundo FreeBSD Handbook [9], servidores de nome normalmente aparecem em duas formas: como servidor de nomes autoritativo e, como um servidor de nomes somente *cache*. Um servidor de nome autoritativo é necessário quando alguém quer servir informação DNS para o mundo, respondendo de forma autoritativa a pesquisas. Já um servidor de nomes somente *cache* é necessário quando se deseja uma resposta mais rápida de um servidor DNS para nomes que já foram recentemente resolvidos, reduzindo assim o tráfego da rede de uma forma geral.

Os servidores de nome autoritativos são subdivididos em servidor mestre ou primário (Figura 2 (b)) e servidor escravo ou secundário (Figura 2 (c)). O servidor secundário é uma réplica do servidor primário. Já o não autoritativo é o servidor de *cache* (Figura 2 (a)), pois ele não resolve definitivamente os nomes.

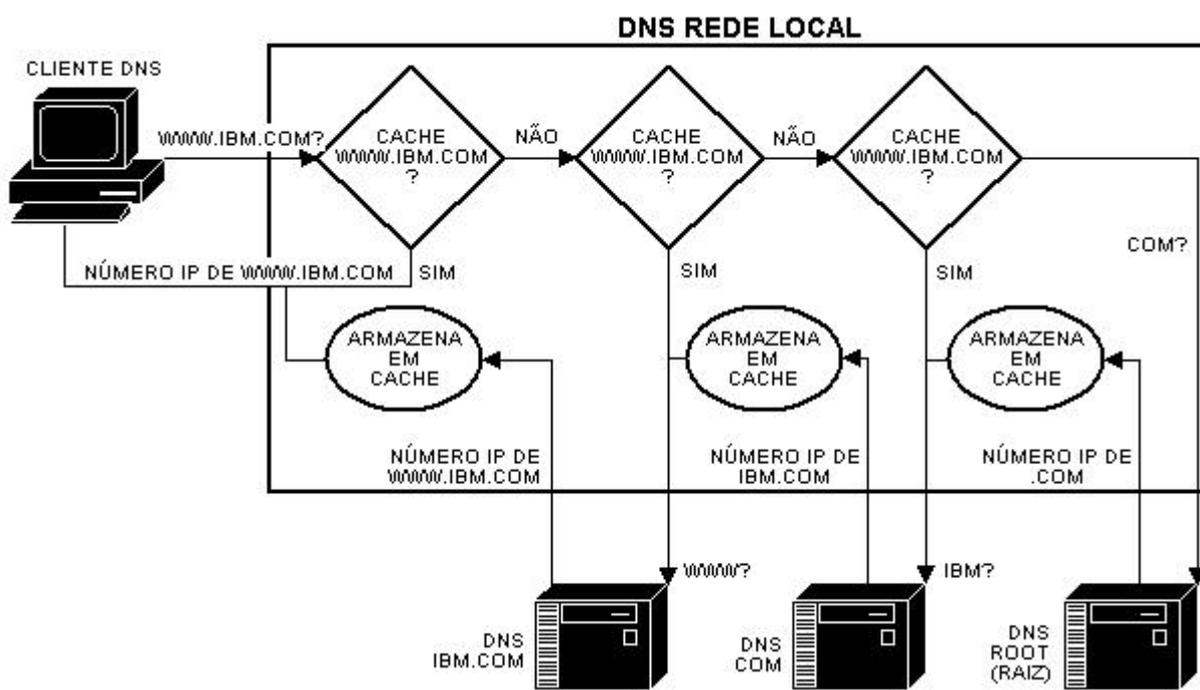


**Figura 2** Comparativo entre os tipos de servidores de DNS (a) Servidor de *Cache* (b) Servidor Primário (c) Servidor Secundário

### 2.1.8 Servidor de *Cache*

De acordo com a Internet System Consortium [13], um servidor de *cache* é

responsável por armazenar todas as consultas realizadas em memória *cache* por um certo período denominado TTL, de forma que quando este tempo expira, a informação da consulta é retirada do próprio *cache*. Assim, quando o cliente necessita de uma informação, se o servidor de nomes já possui a consulta em *cache*, este não irá consultar seus superiores na hierarquia, evitando consultar um servidor raiz ou outro servidor acima na hierarquia, em toda a consulta, como mostrado na Figura 3.



Fonte: FUGIOKA [10]

**Figura 3** Sistema de consulta a servidores de *cache* de DNS

Conforme afirma a distribuidora de software Conectiva Linux [7], um servidor de nomes procurando uma máquina ou componente da rede pode ter de enviar um número considerável de pesquisas até encontrá-la. Enquanto faz isso, ele recebe uma grande quantidade de informações sobre os outros servidores, as quais são guardadas (colocadas em *cache*) para acelerar futuras pesquisas. Na próxima vez em que o servidor de nomes tiver de pesquisar algo sobre isso, o processo será acelerado. Ainda, mesmo que o servidor de *cache* não tenha a resposta armazenada, ele pode ter informações sobre o servidor de nomes responsável por sua zona.

De acordo com o que foi detalhado anteriormente, o servidor de *cache* é muito útil quando se requer praticidade, rapidez e redução do consumo de tráfego na

rede.

### **2.1.9 Servidor Primário ou *Master* (Mestre)**

Segundo a Internet Systems Consortium, Inc [13], o servidor primário ou mestre pode ser definido como um servidor autoritativo (servidor autorizado para a manutenção de zonas de domínio), ou seja, que mantém a base de dados de determinadas zonas. Este servidor além de fazer consultas a servidores raízes também é consultado por outros servidores DNS como responsável pelas zonas em que foi delegado.

Sendo, o servidor primário responsável (autorizado) pelo conjunto de zonas, ele passa a ser a peça mais importante quando se trata de DNS, mas sozinho não é tolerante a falhas, sejam elas de software ou de hardware. A solução para isso foi criar uma réplica do servidor mestre chamada de servidor secundário ou escravo (*slave*) que veremos na subsecção 2.1.10 a seguir.

### **2.1.10 Servidor Secundário ou *Slave* (Escravo)**

Segundo a Internet Systems Consortium, Inc [13], o servidor secundário é uma réplica do servidor primário que carrega as informações do conteúdo das zonas de um servidor primário através da transferência de zonas. Em caso de falha no servidor primário, o servidor secundário entrará em execução tornando-se autoridade sobre a zona, assumindo as funções de resolução de nomes do servidor primário.

Então se pode afirmar que o servidor secundário entrará em atividade a qualquer momento dependendo de possíveis falhas no sistema. Basta para isso garantir que o mesmo esteja corretamente configurado.

Com os servidores DNS primário e secundário, corretamente configurados e em perfeito estado de funcionamento, como mostra a Figura 4, pode-se notar um maior desempenho na resolução de nomes ao passo que serão evitadas possíveis falhas de qualquer natureza através de técnicas de tolerância e replicação.

Normalmente, fica sob a responsabilidade do administrador do sistema a correta configuração e o correto estado de funcionamento do servidor de nomes de forma a obter o melhor resultado possível. Para isso é necessário que o mesmo tenha consciência e conhecimento sobre os procedimentos envolvidos.



**Figura 4** Servidor secundário interagindo corretamente com o primário

### 2.1.11 Estrutura Organizacional e Hierarquia do Banco de Dados

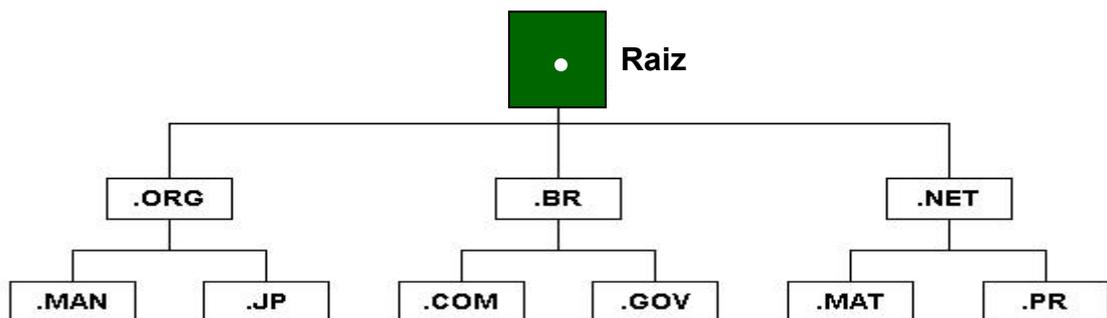
A estrutura organizacional do banco de dados dos servidores DNS é uma hierarquia distribuída em todo o mundo, onde cada nível mantém informações sobre outros principais dos quais precisa para resolver todos os nomes.

De acordo com TANENBAUM [25], o banco de dados do DNS é distribuído através de seus domínios distribuídos em uma estrutura de árvore onde a informações sobre cada um são descritas. As folhas da árvore representam domínios que não possuem subdomínios. Existem dois tipos de domínios de nível superior: genéricos e de países. Os domínios genéricos determinam o tipo de organização, como por exemplo “.org”, “.com”, etc. Os domínios de países determinam o país como por exemplo “.br”, “.us”, etc.

Para que o DNS possa resolver nomes há um ponto de partida (raiz) de pesquisa. A partir deste ponto, a consulta é repassada para um dos servidores delegados por este servidor (servidor raiz ou servidor “.”), os quais delegam sucessivamente outros servidores, até que seja delegado definitivamente o servidor autorizado.

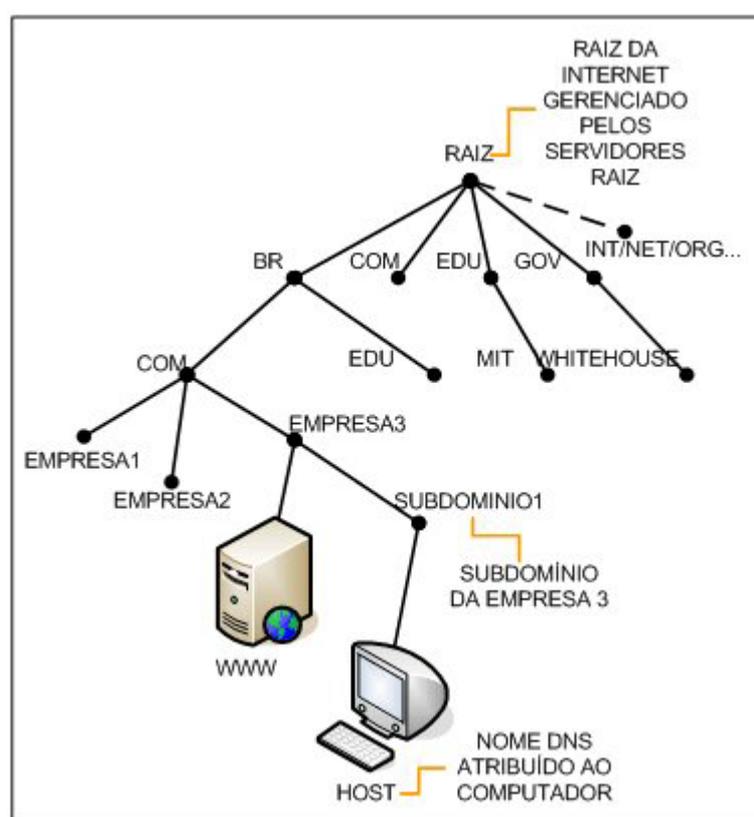
A hierarquia DNS inicia com o domínio “.” (ponto) que é o servidor raiz como está ilustrado na Figura 5, se proliferando de acordo com uma árvore, como pode ser visto na Figura 6.

A hierarquia da Figura 5 é continuada até se chegar ao domínio ou subdomínio desejado conforme demonstrado na Figura 6, onde, partindo-se do domínio raiz é encontrado um nome de *host* atribuído a certo computador de um certo subdomínio do domínio “Empresa3” que está sob o domínio genérico “com” e sob o domínio “br” do país Brasil.



Fonte: FUGIOKA [10] (adaptado)

**Figura 5** Demonstração de um exemplo simples de hierarquia em árvore de domínios do banco de dados do DNS



Fonte: SANTOS [21]

**Figura 6** Demonstração de um exemplo completo de hierarquia de árvore de domínios do banco de dados do DNS

### 2.1.12 Registro de Recurso

Segundo TANENBAUM [27], todo domínio pode ter um conjunto de registros de recursos, sendo o mais comum para um único *host* o registro que indica seu endereço IP, mas também existem outros tipos, sendo que todos eles compõem a base de dados do DNS, mantendo todas as informações referentes a uma certa zona.

### 2.1.13 DNS Direto

O DNS direto é um modelo que como o próprio nome sugere, efetua o mapeamento de nomes para endereços IP. Basicamente é o modelo utilizado para a resolução e identificação de nomes que não exijam confirmação de endereço. Na Figura 7 é ilustrado um exemplo, atualmente real de DNS direto.

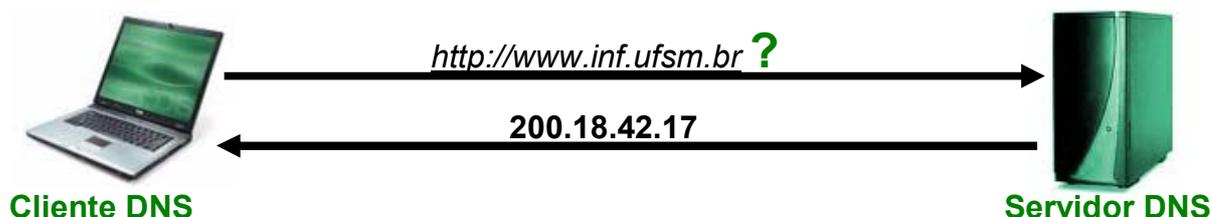


Figura 7 DNS direto

### 2.1.14 DNS Reverso

O DNS reverso é um modelo oposto ao DNS direto, o qual efetua o mapeamento de endereços IP para nomes.

Para apresentar a necessidade do DNS reverso são utilizadas as citações de GODOY [11]:

Imagine-se saindo com alguém (ou seus filhos saindo com alguém) que diz chamar-se Maria. Todas as vezes que você chama "Maria!", esta pessoa olha para você com um sorriso.

Agora, imagine que todas as vezes que chama "Joana!", Maria também olha para eles e responde com um sorriso. O mesmo sorriso para Gabriela, Carla e outros nomes. Estranho, não? Você teria segurança?

Segundo GODOY [11], toda vez que é solicitada a visualização de um site ou é efetuado o envio de um e-mail, o DNS cliente começa a conversar com um ou vários servidores de DNS. Para ilustrar esta "conversa", o autor descreve a apresentação figurada da mesma, passo a passo conforme a Figura 8.

```

Seu computador: Olá www.g2ctech.com! Eu sou a máquina
200.200.200.200. Muito prazer!

Servidor: Olá, 200.200.200.200. Um instante que vou perguntar ao DNS
qual o seu nome, mas enquanto isto, diga-me você qual é.

Seu computador: Eu sou a máquina abobrinha.chuchu.com.br

DNS: 200.200.200.200 é a máquina abobrinha.chuchu.com.br

```

**Servidor:** OK. A sua informação bate com a do DNS. Você é realmente quem diz ser neste caso específico. Vou anotar teu nome nos registros enquanto você navega por aqui.

Fonte: GODOY [11] (adaptado)

**Figura 8** Apresentação fictícia passo a passo da “conversa” cliente-servidor para a resolução de nomes

Embora a apresentação acima não seja formal, ela ilustra claramente os passos seguidos para a interrogação de DNS reverso. É esta interrogação que determina e ilustra todo o processo garantindo maior segurança de confirmação na resolução de nomes.

Na Figura 9, temos um exemplo, atualmente real de DNS reverso.



**Figura 9** DNS reverso

### 2.1.15 DNS Dinâmico

Segundo SuperDNS [22], DNS dinâmico é simplesmente um modo de prender um *host* estático em um endereço de IP dinâmico. Na Internet, existe um número limitado de endereços de IP. Quando você conectar ao seu provedor de acesso, a você é atribuído um endereço de IP temporário para você utilizar enquanto estiver conectado. Uma vez que você desconecta, este endereço IP será utilizado por outra pessoa, e assim por diante. Se você quiser que seu computador seja acessado externamente, será necessário divulgar o seu endereço IP atual.

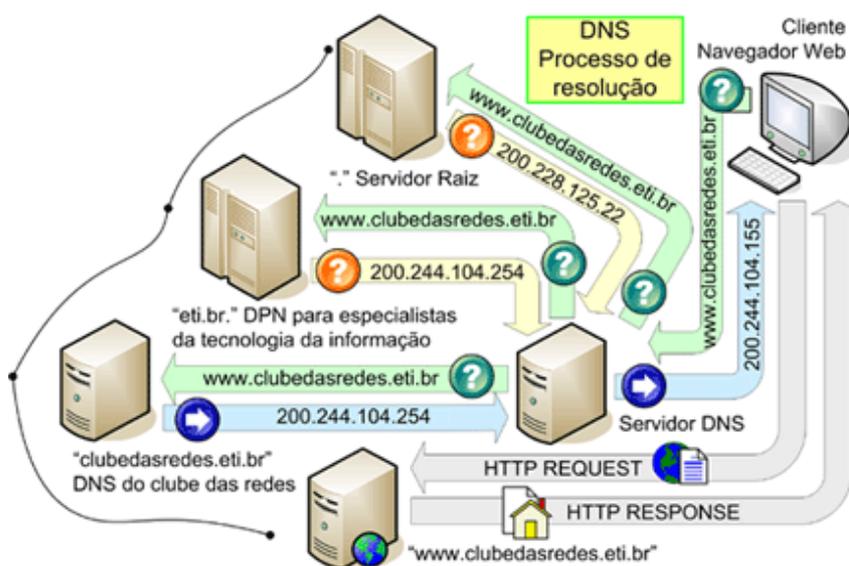
Tendo em mente a idéia básica do DNS dinâmico, podemos afirmar que ele é necessário em *websites* pessoais ou em pequenos serviços de teste, bem como assistência remota ou outras aplicações. Todavia, o DNS dinâmico vai além, podendo ser utilizado por administradores de rede para acompanhar os processos de suporte remoto, além de inúmeros outros.

O DNS dinâmico é implementado tomando-se como base a RFC2136 (VIXIE [30]) que determina todos os processos de atualização dinâmica.

### 2.1.16 Funcionamento Prático

De acordo com MClint - NetBSD [18], para efetuar uma interrogação do DNS, um aplicativo contata um servidor DNS que provê, por sua vez, a contatação dos servidores responsáveis pelo domínio ao qual se refere o nome a ser resolvido. Na realidade, a interrogação segue um percurso hierárquico, partindo dos *root servers*, até chegar ao nível de detalhamento desejado.

Para o melhor entendimento do funcionamento de uma interrogação DNS, exemplificaremos conforme SANTOS [21], uma forma de encontrar o nome "clubedasredes.eti.br" na Internet, podendo ser necessário fazer pesquisa em até quatro servidores de nomes. Inicialmente, é consultado o servidor raiz, que então, indica o servidor de nomes responsável pelo ".br" que já possui informações sobre o domínio ".eti.br", que indica em seguida o servidor responsável pelo domínio "clubedasredes.eti.br". O mesmo, resolve o nome "www.clubedasredes.eti.br" respondendo o endereço IP para o cliente conforme a Figura 10.



Fonte: SANTOS [21]

**Figura 10** Processo de interrogação ao DNS em seqüência da resolução do nome www.clubedasredes.eti.br

Ainda tomando como base SANTOS [21], não há necessidade de se fazer pesquisas a outros servidores, pois o servidor local que atende o *host* cliente que requisitou a pesquisa, guarda a informação em seu *cache*, tão logo ele faça a primeira pesquisa em um nome que não esteja no seu domínio, sendo assim, as futuras pesquisas serão fornecidas de maneira instantânea.

## 2.2 BIND

Segundo a Internet Systems Consortium, Inc [13], o BIND é uma implementação do protocolo DNS que provê a implementação livre e aberta de praticamente todos os componentes do DNS, incluindo:

- Uma biblioteca *resolver* DNS, responsável pela resolução de nomes;
- Um servidor DNS (*named*);
- Ferramentas para verificação da configuração e do funcionamento do DNS.

O BIND possui alguns arquivos de configuração que por padrão vêm pré-configurados para o funcionamento de um servidor de *cache*. Por outro lado para configurarmos um servidor autoritativo, precisamos necessariamente, modificar tais arquivos, os quais segundo a Internet Systems Consortium, Inc [13], são:

- Arquivo de inicialização “*named.conf*” incluindo arquivo de encaminhamentos (ANEXO I);
- Arquivo de configuração de zona direta (ANEXO II);
- Arquivo de configuração de zona reversa (ANEXO III);
- Arquivo *cache* ou raiz (ANEXO IV);
- Arquivos de interface local (ANEXO V).

Não serão apresentados os passos de instalação do BIND, partindo do pressuposto de que o administrador da rede ou a distribuição do sistema operacional já os tenha efetuado corretamente. Para o perfeito entendimento do BIND e seus arquivos de configuração, incluindo ferramentas de automação, as subseções a seguir descrevem detalhadamente a documentação necessária para a correta manutenção do servidor de DNS.

### 2.2.1 Encaminhamentos (*forwarders*)

Encaminhamentos ou *forwarders*, conforme MClint – NetBSD [18], são endereços IP de servidores de *cache* ou autoritativos que são responsáveis pela efetuação das interrogações de acordo com uma lista chamada de lista de *forwarders* ou lista de encaminhamentos. Desse modo serão eles a se encarregar do *cache* das informações, ao invés do servidor local. No caso em que os *forwarders* não consigam responder a uma interrogação, então o servidor procurará encontrar a

resposta por conta própria. Esta articulação reduz a carga de trabalho do servidor local.

### 2.2.2 Update Dinâmico e DNS Seguro

De acordo com a Internet Systems Consortium, Inc [13], o update dinâmico é um método para adição, alteração e exclusão de registros em um servidor de nomes mestre através do envio de mensagens especiais, sendo habilitado zona por zona, conforme será mostrado na Figura 11 abaixo. A segurança do DNS dinâmico é efetivada através da transação de chaves criptografadas por um canal seguro chamado de DNS seguro. Todas as alterações efetivadas através do DNS seguro são mantidas em um arquivo chamado jornal de zonas, com extensão “.jnl”, até que a alteração seja concluída e transferida para o arquivo de zona e o arquivo .jnl seja removido quando o novo endereço IP já estará acessível para o domínio requerente.

### 2.2.3 Transferência Incremental de Zona ou Incremental Zone Transfers (IXFR)

Segundo a Internet Systems Consortium, Inc [13], a transferência incremental de zona é uma alternativa para que não seja preciso efetuar a transferência de uma zona completa (AXFR ou full zone transfer) e sim, somente as alterações feitas nos registros de zona. Para usar transferência incremental de zona deve-se setar a opção `ixfr-from-differences` conforme mostrado na Figura 11.

### 2.2.4 Arquivo de inicialização (named.conf)

O arquivo de inicialização é responsável pela definição dos arquivos a serem lidos pelo servidor de nomes e pode ser visto detalhadamente em suas configurações principais. Conforme descreve a Internet Systems Consortium, Inc [13], essas configurações podem ser subdivididas em dois tipos de servidores: autoritativo de acordo com a Figura 11 e *cache* de acordo com a Figura 12.

```
options {
    directory "/etc/namedb"; // Diretório de trabalho do servidor de nomes
    allow-query { any; }; // Quem tem permissão para efetuar uma consulta ao servidor de nomes
    // any: qualquer endereço
    recursion no; // Determina se pode prover ou não serviços recursivos
    notify no; // Determina o padrão global de notificação
    forward only; //Determina que o servidor de nomes somente percorre os forwarders e nunca
    inicia nenhuma consulta por conta própria
    ixfr-from-differences yes; // Determina o uso de transferência incremental de zona
```

```

    dnssec-enable yes; // Determina o uso de DNS seguro
    include "/etc/named.d/forwarders.conf"; // Diretório de localização dos encaminhamentos de
zona. Também pode ser utilizado encaminhamento interno para a zona atual, basta usar no lugar do
include o forwards {[endereços IP]};

};

zone "." in { // Descrição da zona raiz
    type hint; // Tipo da zona no servidor de nomes. É aceitável hint, forward, master (mestre) ou
slave (escravo)
    file "root.hint"; // Localização dentro do diretório "directory" acima especificado do arquivo de
zona para a zona atual

};

zone "localhost" in { // Descrição da zona direta para o host local
    type master;
    file "localhost.zone";
    notify no; // Notificação que serve para indicar se o mestre avisa ou não o escravo quando
houver alguma modificação no arquivo de zona dado
};

zone "0.0.127.in-addr.arpa" in { // Descrição da zona reversa para o host local
    type master;
    file "localhost.rev";
    notify no;
};

zone "example.com" in { // Descrição de uma zona direta de exemplo chamada "example.com"
    type master;
    file "example.com.db";
    allow-transfer { // endereços IP de servidores escravos com permissão de transferência de
zona para a zona atual, ou seja, esta zona somente poderá ser transferida para os slaves abaixo:
        192.168.4.14;
        192.168.5.53;
    };
    allow-query { // endereços IP ou ACLs de servidores permitidos para fazer consultas na zona
atual
        192.168.4.19;
    };
};

zone "example2.com" in { // Descrição de uma zona direta de exemplo chamada "example2.com"
utilizando DNS dinâmico
    type master;
    file "example2.com.db";
    allow-update { //permite update dinâmico através de uma chave de criptografia usando DNS
seguro. Neste caso estamos usando a mesma chave do rndc, mas poderia ser criada qualquer outra
chave caso fosse necessário.
        key "rndc-key";
    };
};

zone "encaminhamento.com" in { // Descrição de uma zona de encaminhamento chamada
"encaminhamento.com"
    type forward;
    forwarders { // Encaminhamentos internos, somente será utilizado encaminhamento para esta
zona utilizando o endereço IP abaixo
        192.168.4.1;

```

```

    }
}

zone "eng.example.com" in { // Descrição de uma zona para um subdomínio eng da zona anterior,
porém como uma zona slave
    type slave;
    file "eng.example.com.bk";
    masters { 192.168.4.12; }; // Uma zona do tipo slave deve apontar para o endereço IP do
servidor master
};

key rndc-key { // chave de controle chama de rndc-key utilizada pelo DNS seguro
    algorithm hmac-md5;
    secret "0ygfdr2i9B8bslztQ2i0OQ==";
};

controls { // Descrição do componente que pode manter o controle por uma certa porta sobre uma
certa chave
    inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { rndc-key; };
};

```

*Fonte: Internet Systems Consortium, Inc [13] (adaptado)*

**Figura 11** Exemplo do arquivo de configuração “named.conf” em um servidor de nomes somente autoritativo

```

acl corpnets { 192.168.4.0/24; 192.168.7.0/24; }; // acls são listas de endereços IP que controlam os
acessos
options {
    directory "/etc/namedb"; // Diretório de trabalho do servidor de nomes

    allow-query { corpnets; }; // Quem tem permissão para efetuar uma consulta ao servidor de
nomes
// corpnets: acl que controla acesso a um determinado grupo de
endereços IP
};

zone "0.0.127.in-addr.arpa" in { // Descrição da zona reversa para o host local
    type master;
    file "localhost.rev";
    notify no;
};

```

*Fonte: Internet Systems Consortium, Inc [13] (adaptado)*

**Figura 12** Exemplo do arquivo de configuração “named.conf” em um servidor de nomes somente cache

Não serão demonstradas, nesta subseção, as demais configurações do arquivo “named.conf”, visto que as mesmas serão utilizadas de acordo com o padrão preestabelecido pelo BIND.

### 2.2.5 Arquivo de cache (raiz)

Este arquivo contém informações sobre os servidores de nome de domínio “raiz” espalhados pelo mundo. O mesmo é um arquivo padrão e faz parte do BIND

como um arquivo de redirecionamentos ou encaminhamentos. Semelhante a um arquivo de zona, o arquivo de *cache* informa o nome e o endereço IP de cada um dos 13 servidores raiz espalhados pelo mundo, fazendo com os mesmos um balanceamento de carga.

### 2.2.6 Formato do arquivo de zona

O formato de um arquivo de zona do BIND é iniciado com \$TTL ou tempo de vida do arquivo de zona, ou seja, o tempo que o arquivo de zona deve ser mantido em *cache* pelos servidores DNS depois da efetivação da consulta, seguido de linhas com os registros de recursos de acordo com a Figura 13.

<b>Nome</b>	TTL(opcional)	<b>IN tipo_registro</b>	prioridade(opcional)	<b>dados_registro</b>
-------------	---------------	-------------------------	----------------------	-----------------------

*Fonte: Internet Systems Consortium [13] (adaptado)*

**Figura 13** Formato de linhas de registro de recurso

Segundo TANENBAUM [27] e tendo como base a Figura 13, o nome informa o domínio ao qual este registro se aplica, sendo a chave de pesquisa primária utilizada para atender às consultas. Já o TTL que é um parâmetro opcional que define a duração que este registro de recurso deve ser mantido em *cache* por outros servidores DNS externos que fizerem consulta a este servidor DNS, fornecendo assim, uma indicação de estabilidade do registro. O IN sempre deve ser usado em registros de recursos, pois indica que o nome do registro **está** relacionado com a Internet. In é o tipo de classe característica de redes TCP/IP, utilizadas atualmente. O tipo\_registro define o registro fonte. E prioridade que é um campo opcional define o valor de cada dado\_registro, enquanto dados\_registro definem os dados de cada tipo de registro.

Existem algumas regras básicas que servem para qualquer registro de recurso e são explicadas de acordo com a Tabela 2.

**Tabela 2** Regras básicas de configuração do arquivo de zonas

<b>Regra</b>	<b>Descrição</b>
@ (arroba)	Substitui o identificador da zona, ou seja, ao invés de colocar o domínio identificador da zonas pode-se usar o @.
“.” (ponto)	Todo o nome deve ser sucedido de um ponto ao final para indicar o nome de domínio completo. Caso o ponto não suceda o nome, o mesmo

	será sucedido pelo nome identificador da zona.
Tabulação ou espaço em branco	Repete o valor do campo anteriormente acima.
Endereço de e-mail	Como @ é um caractere especial que substitui o identificador de zona, substitui-se o @ por ponto “.”.
Tempos	Todos os tempos, no arquivo de zonas é descrito por padrão em segundos, mas pode descrito em outras unidades com o uso de curingas.
; (ponto e vírgula)	Considera até o final da linha como comentários.

Fonte: MClint - NetBSD [18] e Internet Systems Consortium, Inc [13] (adaptado)

TANENBAUM [27] descreve os registros de recursos mais frequentemente utilizados, os quais serão abordados no decorrer deste projeto. Segundo ele e de acordo com o formato da Figura 13, cada registro de recurso será detalhado com o seu respectivo formato, como segue:

- SOA – início da zona autoritativa sendo um registro único por zona, iniciando com um nome do registro que deve ser o nome do identificador da zona ou @. Os dados deste registro são o servidor de DNS mestre seguido do e-mail do administrador do mesmo, possuindo ainda os seguintes parâmetros:
  - *Serial* ou número de série que serve para identificar uma mudança no arquivo de zona atual. Segundo MClint - NetBSD [18], uma alternativa é escolher uma codificação baseada na data da última modificação seguida de um contador de modificações, por exemplo: 2007011002 significando a segunda alteração no dia 10 de 01 de 2007;
  - *Refresh* ou atualização que é o intervalo de tempo (por padrão em segundos) que o servidor de nomes secundário deve verificar a validade de seus dados;
  - *Retry* ou re-tentar que é o tempo (por padrão em segundos) que o servidor de nomes secundário deve esperar para consultar novamente um servidor primário quando este falhar;
  - *Expires* ou tempo de expiração que é o tempo (por padrão em segundos) que o servidor de nomes secundário pode ficar sem contato com o primário até os dados não serem mais considerados válidos;

- *TTL* mínimo ou tempo de vida mínimo é o tempo (por padrão em segundos) mínimo que os outros servidores que efetuarem consultas neste podem considerar os dados fornecidos como válidos.
- NS (*Name Server*) – utilizado para cada servidor de nomes do domínio, ou seja, servidor de nomes autoritativo;
- A (*Address*) – utilizado para fazer o mapeamento entre nome e endereço IP;
- CNAME (*Canonical Name*) – utilizado para a definição de aliases, ou seja, nomes alternativos para um determinado *host*;
- MX (*Mail Exchanger*) – utilizado para a definição de repassadores de e-mail, ou seja, servidores de e-mail.
- PTR (*Pointer*) – utilizado para definir a tradução de um endereço IP para um nome de *host*;
- HINFO (*Host Information*) – utilizado para armazenar informações sobre o *host* dado (hardware sem espaços, sistema operacional sem espaços);
- RP (*Responsible People*) – utilizado para armazenar informações sobre a pessoa responsável do *host* dado (e-mail sem espaços, nome ou apelido sem espaços);
- TXT (*text*) – utilizado para armazenar informações na forma de texto sobre o *host* dado.

### 2.2.7 Arquivo de configuração de zona direta

O arquivo de configuração de zona direta é o responsável pelo mapeamento direto de zona e é mostrado com dados, apenas ilustrativos, de exemplo conforme a Figura 14.

### 2.2.8 Arquivo de configuração de zona reversa

O arquivo de configuração de zona reversa é o responsável pelo mapeamento reverso é mostrado com dados, apenas ilustrativos, de exemplo conforme a Figura 15.

## 2.2.9 Arquivos de interface local

Segundo a Internet Systems Consortium, Inc [13], os arquivos de interface local descrevem a configuração da zona direta e também da zona reversa do *host localhost* ou do endereço IP 127.0.0.1.

```

$TTL 2d
@      IN      SOA     ns1.teste.com.br. root.teste.com.br. (
                          2006122802
                          3h
                          1h
                          1w
                          1d )

@      IN      NS     ns1
@      IN      NS     ns2

localhost  IN  A      127.0.0.1
ns2        IN  A      168.194.1.11
mail       IN  A      168.194.1.12
mail1     IN  A      168.194.1.13
@         IN  A      168.194.1.10

@         IN  MX     10   mail
@         IN  MX     1   mail1

webmail    IN  CNAME   mail
www        IN  CNAME   @
ns1        IN  CNAME   @

;subdominio escolha
escolha    IN  NS     ns1

hwinfo     IN  HINFO   hardware so
@          IN  TXT     "txt teste"
leandro    IN  RP     restinga.inf.ufsm.br. restinga

```

Fonte: Internet Systems Consortium, Inc [13] (adaptado)

**Figura 14** Arquivo de configuração da zona direta teste.com.br

```

$TTL 1W
@      IN      SOA     ns1.teste.com.br. root.teste.com.br. (
                          2006122801
                          2D
                          4H
                          6W
                          1W )

@      IN      NS     ns1.teste.com.br.
@      IN      NS     ns2.teste.com.br.

10     IN      PTR     teste.com.br.
11     IN      PTR     ns2.teste.com.br.
12     IN      PTR     mail.teste.com.br.
13     IN      PTR     mail1.teste.com.br.

```

Fonte: Internet Systems Consortium, Inc [13] (adaptado)

**Figura 15** Arquivo de configuração da zona reversa 168.194.1.in-addr.arpa

## 2.2.10 RNDC

Segundo CAMPOS [6], RNDC é um utilitário que controla as operações do

servidor BIND. O RNDC comunica-se com o servidor BIND através de uma conexão TCP/IP, opcionalmente enviando os comandos autenticados com assinatura digital.

### 3 TRABALHOS RELACIONADOS

Esta seção apresentará as ferramentas de gerenciamento relacionadas, dentre as principais existentes no mercado, seguidas de suas respectivas análises.

#### 3.1 FreeDNS

O FreeDNS [5] é um robusto gerenciador de DNS que suporta DNS dinâmico e Estático, gerenciamento de subdomínios, redirecionamento eficiente de URL, DNS reverso, além de inúmeras outras funcionalidades.

As principais desvantagens do FreeDNS são as restrições da versão livre. Com ela somente é possível o gerenciamento de 5 *hostnames* compartilhados e 20 subdomínios por domínio. Além disso, outra desvantagem é a inexistência de alguma opção de tradução de linguagem do sistema de inglês para português do Brasil, o que pode dificultar o aproveitamento do mesmo.

#### 3.2 Zoneedit

O Zoneedit [38] suporta gerenciamento de DNS tanto estático quanto dinâmico, gerenciando zonas em uma utilização das opções básicas de administração na sua versão livre, permitindo somente administração de no máximo 5 zonas.

O principal problema do Zoneedit é a diversidade de serviços fracamente relacionados, como por exemplo, encaminhamento de e-mail. Outro fator importante é a escassez do gerenciamento de inúmeros registros de recursos suportados pela versão mais recente do BIND, assim como o gerenciamento de zonas apenas no servidor mestre em uma interface de uso crítica para o usuário.

#### 3.3 XName

O XName [37] é um robusto gerenciador de DNS que suporta DNS dinâmico e Estático, gerenciamento de subdomínios, múltiplos usuários e grupos, DNS reverso, notificações automáticas, gerenciamento completo de zonas além de inúmeras outras funcionalidades e ferramentas para automatização de tarefas.

A principal desvantagem é a inexistência de alguma opção de tradução de

linguagem do sistema de inglês/francês para português do Brasil, o que pode dificultar o aproveitamento do mesmo. Outra desvantagem é a interface de difícil utilização.

### **3.4 Webmin**

O Webmin [31] é uma ferramenta de administração gráfica, escrita por Jamie Cameron, utilizando a linguagem Perl. Ela foi projetada para ser uma ferramenta de administração leve, funcional, e que possa ser facilmente estendida. A ferramenta está disponível hoje em vários idiomas, e está sendo considerada a ferramenta oficial de administração em vários sistemas operacionais e distribuições.

O Webmin é um painel de administração de servidores e, além de possuir inúmeros recursos, apresenta um gerenciador do servidor de DNS BIND. Este gerenciador é bem robusto, possui tradução para português do Brasil, gerencia domínios e subdomínio de maneira completa e ilimitada, além de possuir um vasta documentação.

A desvantagem do Webmin é que a sua versão atual apenas gerencia a versão 8 do BIND e não a versão 9 que é a versão mais atualizada. Outra desvantagem é a de a interface exigir um conhecimento mais avançado do administrador de rede, tendo ele que conhecer profundamente a parte conceitual dos registros de recursos para a criação de uma zona, o que foge ao seu hábito comum.

## 4 DESENVOLVIMENTO

Nesta seção, apresenta-se a implementação e o desenvolvimento da ferramenta intitulada GDNS, seguidos dos detalhes dos processos agregados.

Inicialmente foi desenvolvido um estudo geral sobre o protocolo DNS e a arquitetura TCP/IP incluindo as RFCs que os regulamentam. Posteriormente, foi feito um estudo geral sobre o BIND, seus recursos, sua melhor utilização, configurações principais e a determinação do melhor modelo possível para o caso. Estes estudos se desenvolveram de acordo com a seção anterior no decorrer do referencial teórico e serviram como base para as decisões de desenvolvimento aqui tomadas.

Após os estudos já referenciados, foi feita a análise de layout e estruturação da interface a ser utilizada, bem como a determinação e modularização da aplicação na tentativa de sempre deixá-la o mais simples possível.

Como já mencionado, a interface será considerada um passo de extrema importância, já que o sistema tem o intuito de ser acessado de maneira simples e objetiva.

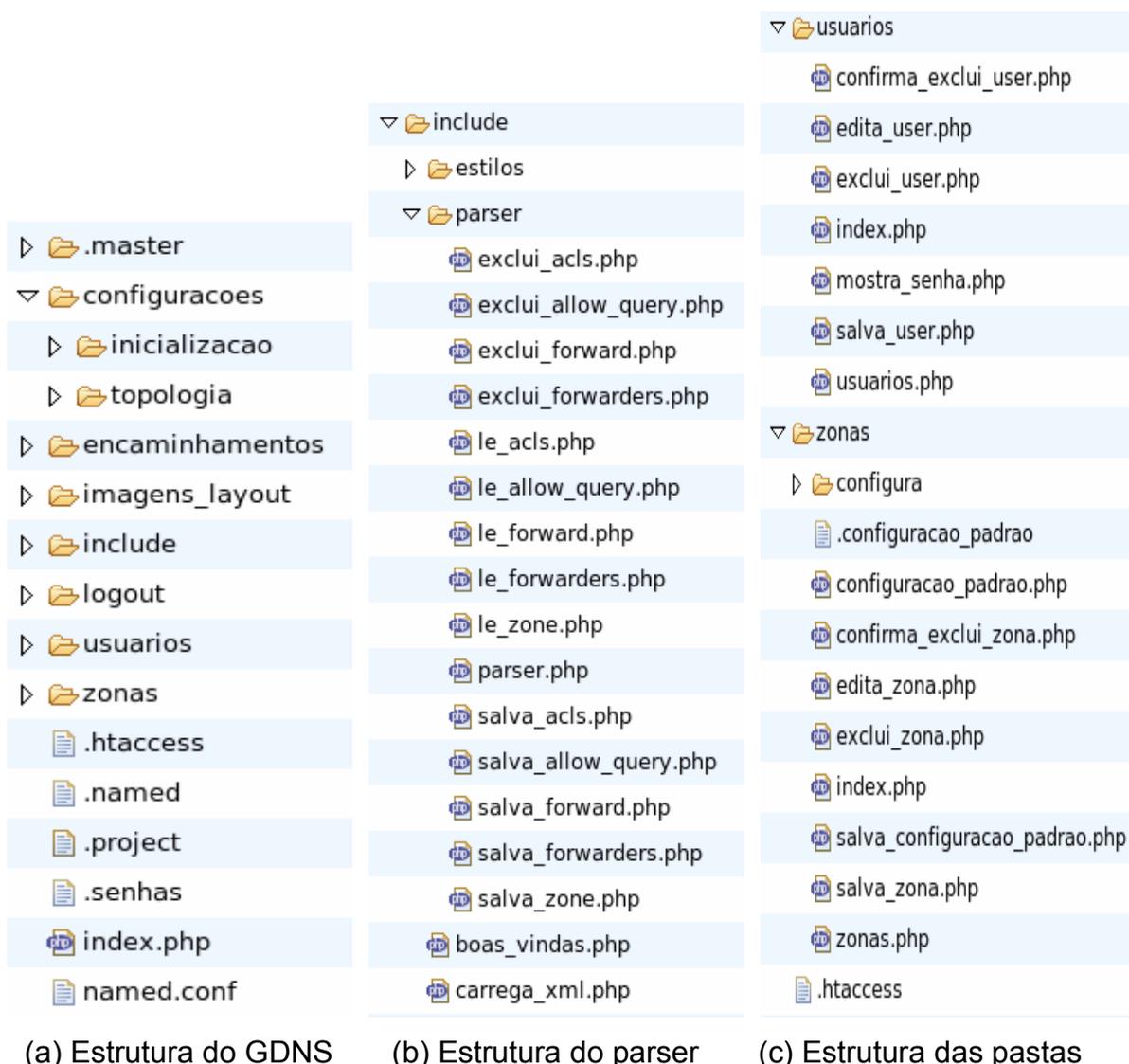
Em seqüência da interface, foi desenvolvida a implementação do sistema principalmente em linguagem PHP, utilizando-se HTML, XML e JavaScript de maneira a usufruir o que existe de mais prático em cada uma, usando inclusive as experiências do desenvolvedor.

Por fim, foram executados testes relativos ao software com a finalidade de reduzir erros de implementação, melhorando assim a qualidade do sistema. Após foi elaborada a documentação, a qual ficou disponível internamente na aplicação. Finalmente, o sistema foi disponibilizado para a utilização em ambiente real em fase de beta-teste, sendo que posteriormente será amplamente difundido.

A metodologia de implementação e desenvolvimento será determinada nas subseções seguintes, de modo a demonstrar os passos desenvolvidos na efetivação deste projeto.

### 4.1 Estruturação do Código-Fonte

O código está estruturado de acordo com a Figura 16 que mostra partes do código-fonte do sistema e suas principais funcionalidades, bem como a hierarquia dos arquivos envolvidos.



(a) Estrutura do GDNS

(b) Estrutura do parser

(c) Estrutura das pastas

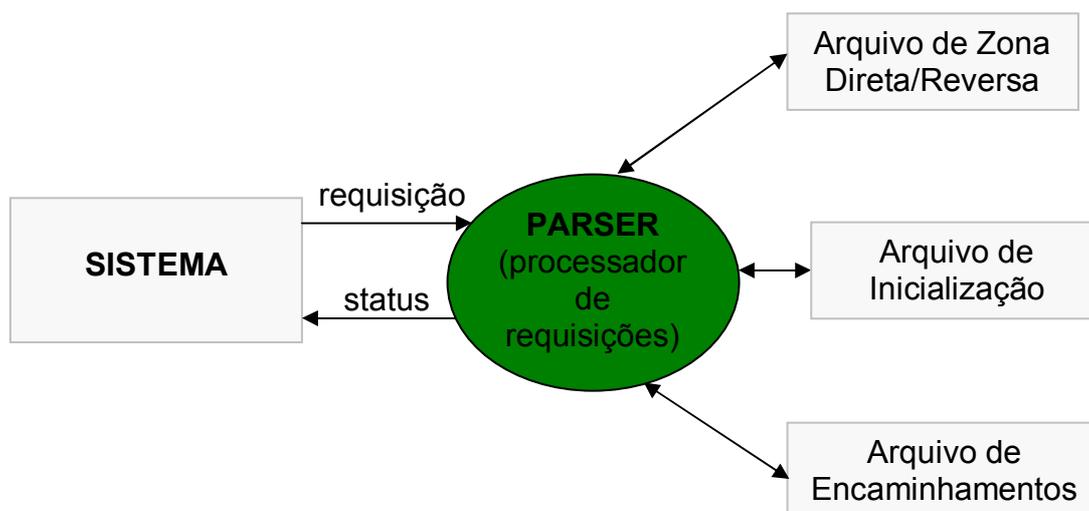
**Figura 16** Estruturações do código-fonte do GDNS

Na Figura 16 (a), que apresenta a estrutura do GDNS, pode ser vista a estrutura de organização do código-fonte, onde as pastas remetem a mesma estrutura do menu.

Na Figura 16 (b) é apresentado o “parser”. Ele é o responsável por manipular os arquivos de configuração do BIND, lendo, interpretando e efetuando todas as operações necessárias para a manutenção dos arquivos:

- “Named.conf” ou arquivo de inicialização;
- De zona direta e reversa;
- De encaminhamentos.

O modelo estrutural do parser é ilustrado na Figura 17 e tem por finalidade principal demonstrar a estrutura e sua funcionalidade.



**Figura 17** Modelo estrutural do parser

De acordo com a Figura 16 (c), que apresenta a estrutura de pastas do usuário e das zonas, sempre existe um arquivo de código chamado de `index.php` que é o responsável pela manutenção da sessão e por acessar o gerenciador. Assim todas as requisições, independente do gênero (edição, exclusão, adição, etc.) são tratadas diretamente pelos arquivos “`index.php`” das subpastas de acordo com a estrutura da Figura 16 (a), os quais distribuem as tarefas para os outros módulos do código e se encarregam da verificação de permissões.

## 4.2 Autenticação e Segurança

A autenticação será efetuada no GDNS. O sistema irá autenticar diretamente na máquina servidora, utilizando os usuários cadastrados no arquivo de senhas gerenciado pelo `htaccess`. Essa autenticação é totalmente efetivada pelo apache juntamente com o `htaccess`, os quais isolam a área do GDNS de acesso não autorizado. Uma ilustração do processo de autenticação é apresentada na Figura 18.

Após a autenticação, cada usuário receberá uma sessão que o identificará durante todo o processo de atuação no sistema. Essa sessão permitirá ao usuário, através de um processo de migração entre os servidores de nomes da rede, uma manutenção independente de localização, porém dependente de autorização do usuário sobre o servidor.



**Figura 18** Tela de autenticação em área restrita

O processo de migração envolve várias questões de segurança que serão tratadas utilizando-se DNS seguro, com o auxílio da ferramenta RNDG.

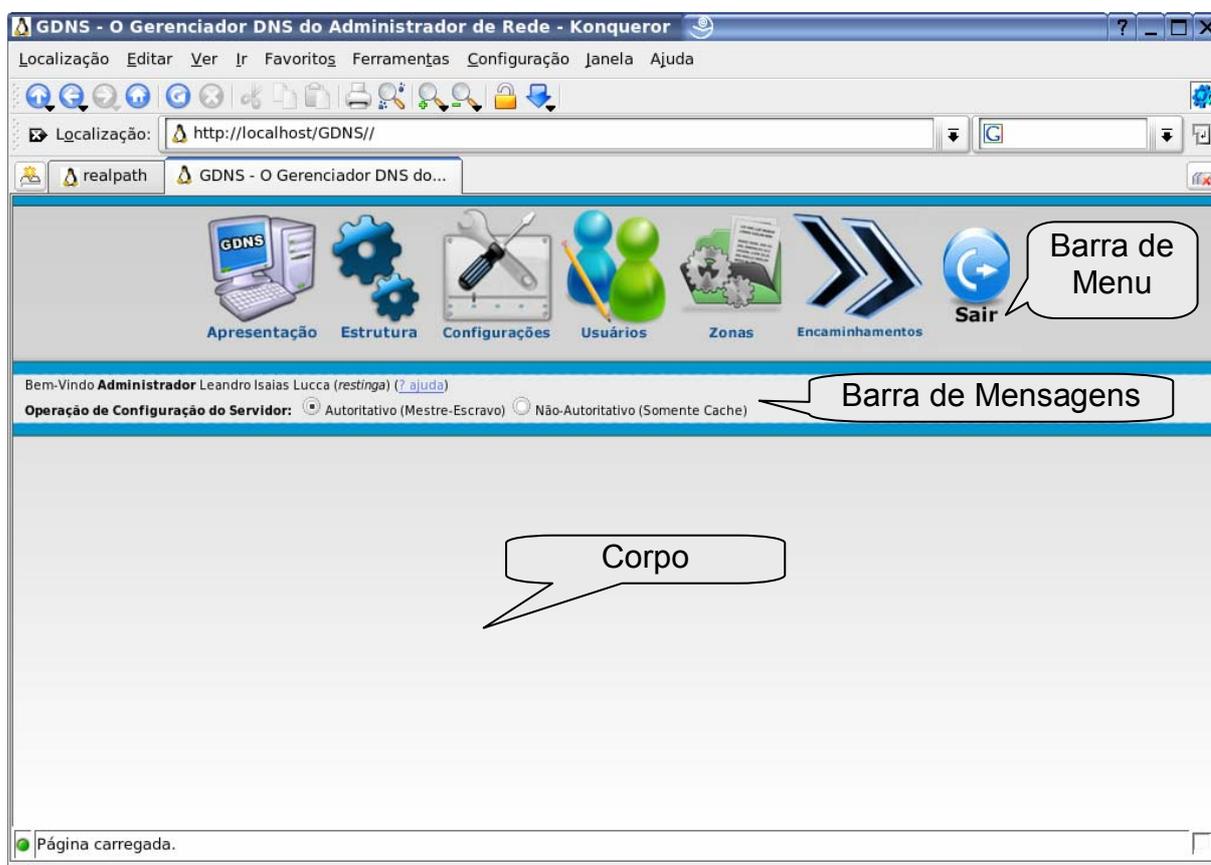
### 4.3 Layout

O layout do GDNS utiliza a técnica cross-browser. A apresentação do mesmo pode ser vista de acordo com a tela inicial do GDNS ilustrada na Figura 19.

Considerando a Figura 19, na barra ao cabeçalho do sistema temos o menu de operações, onde é possível acessar todas as telas de gerenciamento do sistema. O menu apresentação, que é a própria tela inicial, apresenta a barra de boas vindas ao usuário. O menu estrutura, apresenta a estrutura organizacional do sistema, onde é configurada a topologia dos servidores de nome. O menu configurações, apresenta configurações iniciais do sistema, tais como, ACLs e controle de permissões de consulta. O menu zonas apresenta a lista de zonas e suas opções padrão de gerenciamento, bem como opções de configuração geral, incluindo a manutenção interna da própria zona. Já o menu encaminhamentos é responsável pelos encaminhamentos do servidor de nomes e o menu sair, permite ao usuário eliminar as sessões que lhe dão permissão de acesso ao sistema.

A barra de mensagens é destinada a mensagens de boas-vindas, mensagens de sucesso, falha ou confirmação de operação, bem como mensagens com

documentação de ajuda on-line.



**Figura 19** Tela inicial do GDNS

Por fim, o corpo do layout, apresenta a área de trabalho do sistema, onde todas as operações e dados são mostrados a partir da sessão indicada no menu.

#### 4.4 Usuários e Permissões

Os usuários pertencem a duas classes distintas:

- Usuários com privilégio de administração que controlam todas as configurações globais, as zonas e as ações dos usuários comuns;
- Usuários comuns que têm privilégio básico de controle sobre a zona de seu domínio.

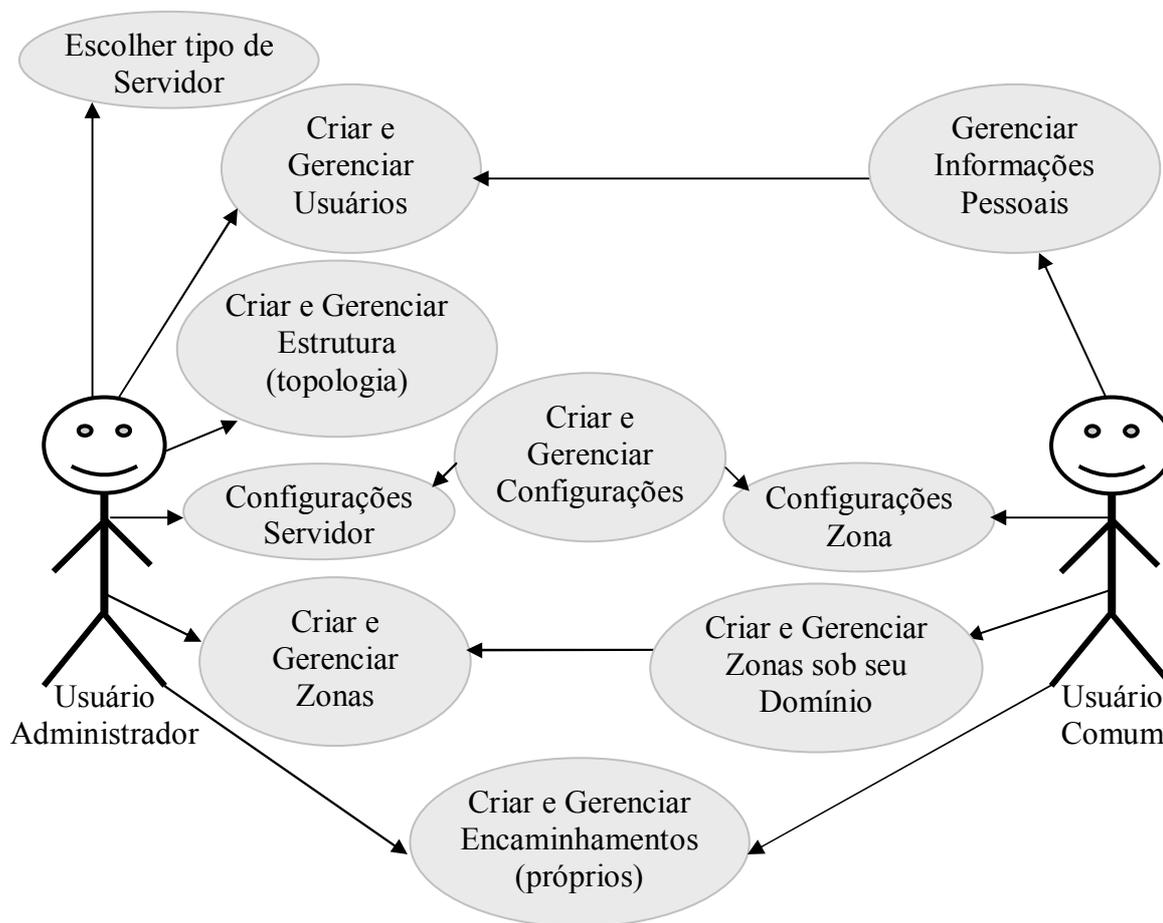
O usuário com privilégio de administrador, além de efetuar a manutenção dos usuários comuns, também é responsável por controlar as permissões de cada um, determinando e verificando o correto gerenciamento do servidor de nomes.

#### 4.5 Definição da Interface de Gerenciamento

A interface de gerenciamento foi considerada o ponto principal no desenvolvimento do presente projeto. Esta questão foi analisada, devido à dificuldade dos usuários perante a compreensão da interface dos diversos gerenciadores de DNS como o Zoneedit, o XName, o FreeDNS e o Webmin.

A ferramenta de gerenciamento foi implementada com o intuito de prover uma interface que abstraia do usuário a parte conceitual determinada pelo DNS, utilizando termos comuns a administradores de redes em geral. Isso vai muito além do que é apresentado pelas ferramentas avaliadas.

Uma visão geral da interface de gerenciamento pode ser vista de acordo com o diagrama de casos de uso, descrito na Figura 20.



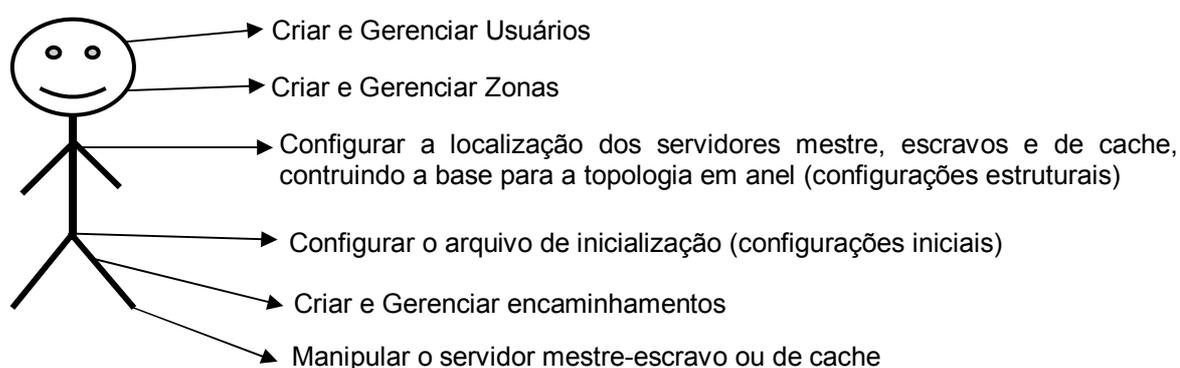
**Figura 20** Visão geral da interface de gerenciamento do GDNS

As subseções seguintes demonstram o modelo de interface para cada usuário do sistema em questão.

#### 4.5.1 Interface de Gerenciamento no Modo Administrador

O modo administrador é destinado ao usuário que mantém privilégios avançados.

Diferentemente do usuário comum, o administrador tem privilégios e funções de acordo com a Figura 21.



**Figura 21** Privilégios/funções no modo administrador

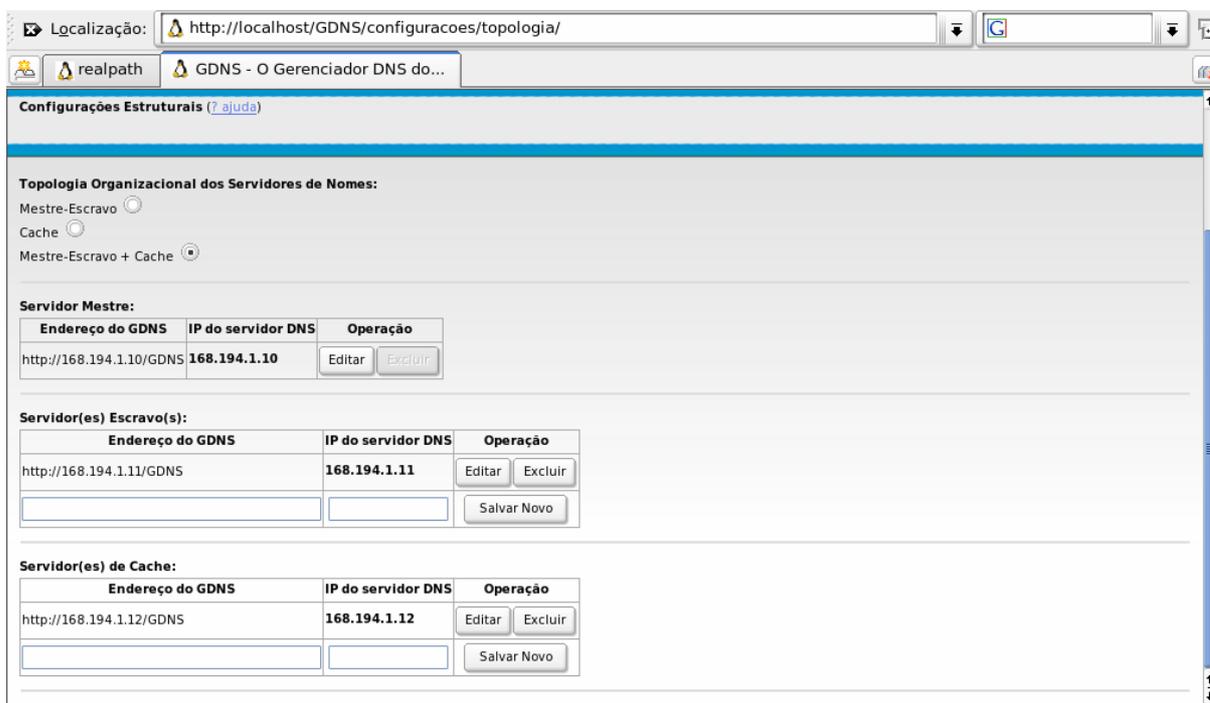
Ao acessar o sistema, o administrador deve escolher a opção de configuração do servidor, desde que a configuração estrutural esteja habilitada para o modo de manipulação mestre-escravo + *cache*. A opção de configuração permite ao administrador optar pelo servidor que desejar configurar. A configuração estrutural deve estar configurada e é expressamente necessário que seja feita no momento da instalação do sistema. A tela de configuração estrutural pode ser vista de acordo com a Figura 22.

O privilégio de criar e gerenciar usuários pode ser visto como um cadastro de usuários do sistema, onde o administrador cadastra o nome do usuário, tipo, login e senha para efeitos de autenticação, uma breve descrição e o respectivo e-mail para o posterior cadastramento automático do responsável da zona. Esse privilégio é ilustrado pela tela de criação e gerenciamento de usuário do GDNS na Figura 23.

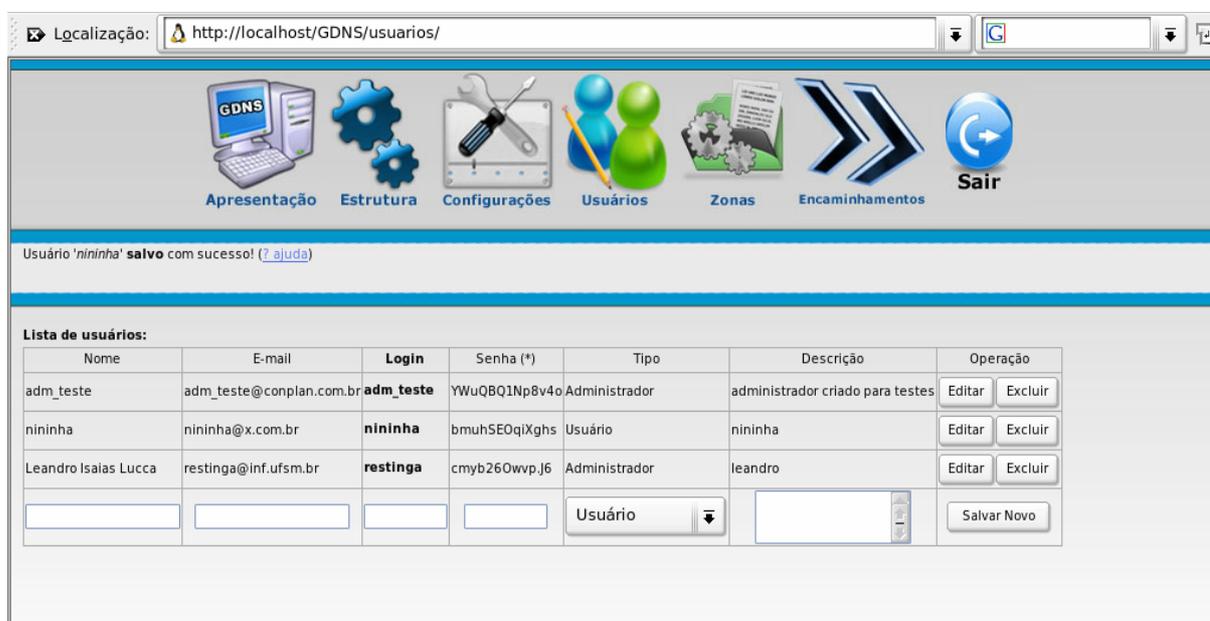
Já o privilégio de criar e gerenciar zonas vai um pouco além. Neste caso o administrador cadastra a zona com o nome de domínio pertencente ao usuário, criando assim os arquivos de configuração de zona direta e reversa padrão de acordo com os dados do usuário e da respectiva zona. Esses arquivos de zona deverão, obrigatoriamente, sofrer alterações por parte do usuário comum, o qual passará a ser o responsável pelo seu gerenciamento.

O administrador poderá criar zonas do tipo mestre-escravo ou de

encaminhamento. O parâmetro “*allow-transfer*” será alterado automaticamente de acordo com o endereço IP cadastrado nas configurações estruturais.



**Figura 22** Tela de configuração estrutural do GDNS no modo administrador



**Figura 23** Tela de criação e gerenciamento de usuários do GDNS no modo administrador

A criação e gerenciamento de zonas pode ser visualizada para maiores detalhes de acordo com a tela da Figura 24.

Localização: <http://localhost/GDNS/zonas/>

Gerenciamento de Zonas ([? ajuda](#))

**Configuração Padrão para a criação de novas zonas:**

Tempo de Vida (TTL)	Tempo de Atualização (Refresh)	Tempo de Retentativa (Retry)	Tempo de Expiração (Expires)	Operação
38400 seg.	10800 seg.	3600 seg.	604800 seg.	Gravar Alterações

**Zonas de Domínio / Subdomínio:**

Domínio/Subdomínio	Usuário Responsável	Tipo de Zona	Faixa de IPs	Operação
teste.com.br	nininha	Mestre-Escravo	168.194.1	Editar Excluir Configurar
xls.teste.com.br	nininha	Encaminhamento	168.194.4	Editar Excluir Configurar
teste1.teste.com.br	nininha	Mestre-Escravo	168.194.2	Salvar Novo

**Figura 24** Tela de criação e gerenciamento de zonas do GDNS no modo administrador

Embora o BIND suporte alternância de servidores mestre e escravo por zona, ou seja, para uma certa zona o servidor mestre de outra pode ser escravo e vice-versa, essa flexibilidade não será utilizada por afetar o caráter organizacional do sistema.

Outro aspecto importante a ser relatado na criação de zona é que todas as zonas do tipo mestre-escravo serão criadas com o parâmetro “*notify*” setado como “*yes*”, assim todas as zonas notificarão o servidor escravo quando forem propagadas suas alterações. O parâmetro “*recursion*” será configurado para “*yes*” no caso de ser um servidor autoritativo e “*no*” no caso de ser um servidor de *cache*. Por isso, se torna imprescindível configurar a localização dos servidores mestre, escravos e de *cache*, na construção da base estrutural da topologia de servidores.

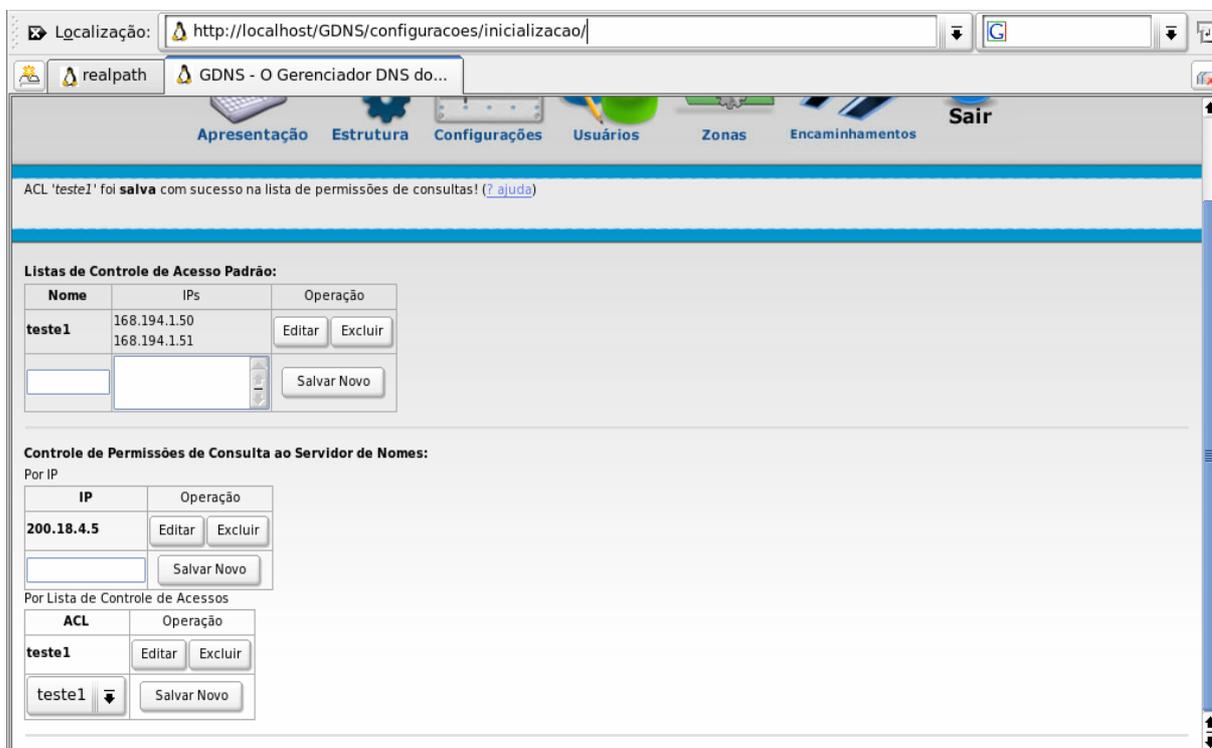
Tomando como base os privilégios de alteração do arquivo de inicialização, o administrador pode:

- Criar e gerenciar ACLs globais;
- Alterar o parâmetro “*allow-query*” que controla as permissões de consulta ao servidor de nomes, para os valores “*any*” que permitem consultas de qualquer *host*, ou uma lista de IPs, ou ainda uma ACL;

Os privilégios de gerenciamento do arquivo de inicialização são configurados no GDNS de acordo com a tela de configuração inicial apresentada na Figura 25.

Por fim, o administrador também tem privilégios para a configuração global de

encaminhamentos do servidor de nomes, bem como a alteração do parâmetro “forward” que controle os encaminhamentos, para os valor “only” permitindo assim, somente encaminhamentos. Essa tela de configuração pode ser vista pela Figura 25.



**Figura 25** Tela de configuração inicial do GDNS no modo administrador

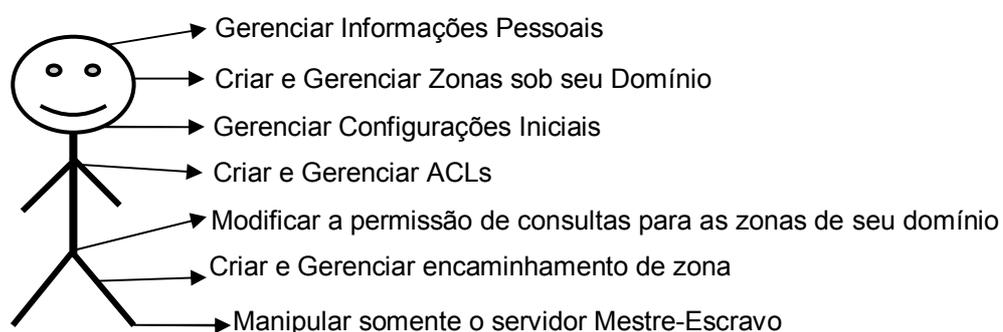


**Figura 26** Tela de encaminhamentos do GDNS no modo administrador

#### 4.5.2 Interface de Gerenciamento no Modo Usuário

O modo usuário é destinado ao usuário comum mantendo os privilégios

mínimos necessários. Tais privilégios são ilustrados abaixo de acordo com a Figura 27.



**Figura 27** Privilégios no modo usuário

O privilégio de gerenciar informações pessoais pode ser visto com um simples arquivamento de dados pessoais, onde o usuário tem apenas permissão para a edição dos dados. Já o privilégio de criar e gerenciar zonas pode ser visto como um cadastro de zonas, bem como sua configuração completa. O cadastramento de novas zonas ocorre de acordo com a necessidade de um subdomínio para o domínio sob o qual o usuário mantém o controle. O gerenciamento de zonas pode ser efetuado, pelo usuário, sob as zonas mantidas pelo mesmo, podendo efetuar alterações em seus recursos de acordo com os serviços necessários. O usuário somente tem permissão para criação de zona do tipo mestre-escravo. Assim como no modo administrador, no modo usuário abstrai-se a criação da zona reversa, ou seja, ambas, a zona direta e a reversa, são criadas juntamente de forma automática. Os registros de recursos também são abstraídos dos usuários.

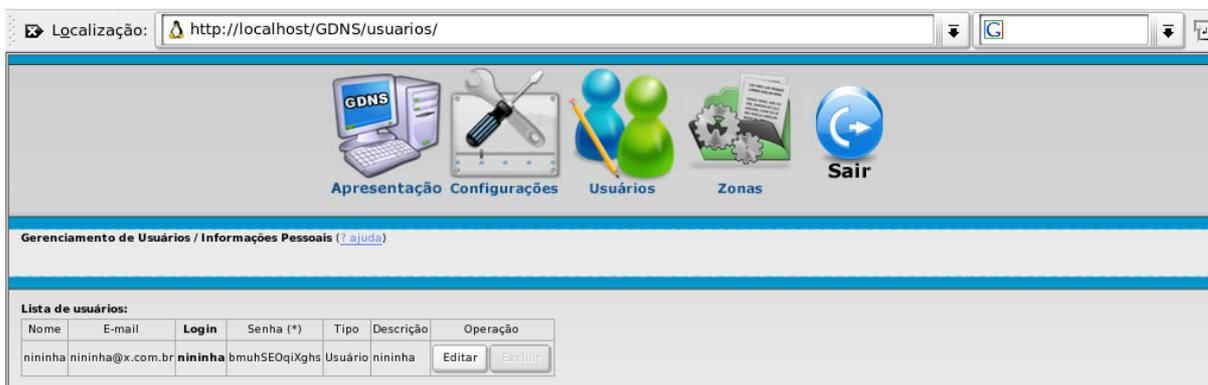
O modelo de gerenciamento utilizado para a criação e alteração de zonas é de suma importância para o correto gerenciamento do servidor DNS, por isso a interface com o usuário comum será detalhada passo-a-passo, bem como todos os dados requisitados por ela de acordo com:

- Informações pessoais do usuário;
- Configurações iniciais;
- Criação e gerenciamento de zonas;
- Criação e gerenciamento de recursos de rede.

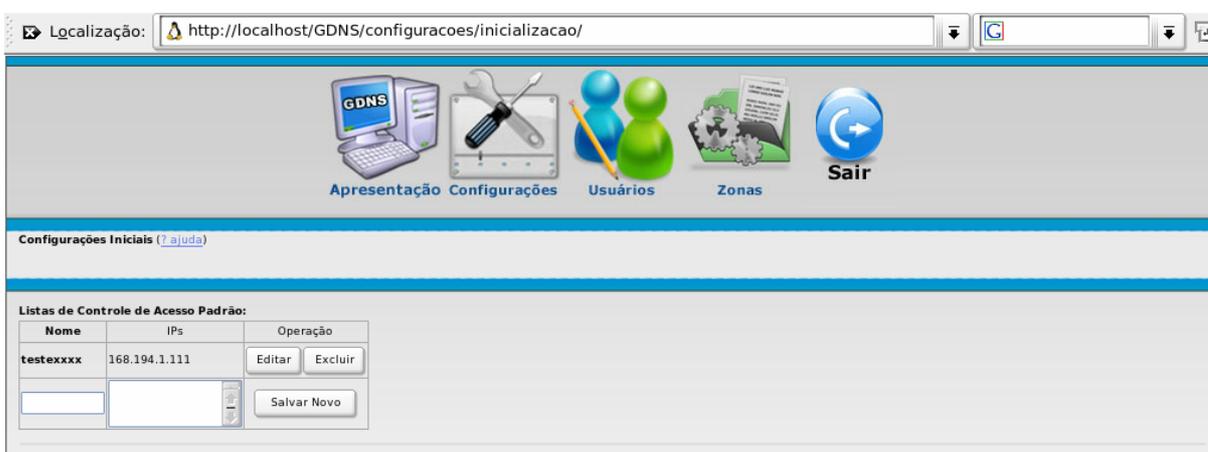
A interface de informações pessoais pode ser visualizada de acordo com a tela demonstrada na Figura 28.

A interface de configurações iniciais dos serviços de nome pode ser

visualizada de acordo com a tela demonstrada na Figura 29.



**Figura 28** Tela de informações pessoais do GDNS no modo usuário



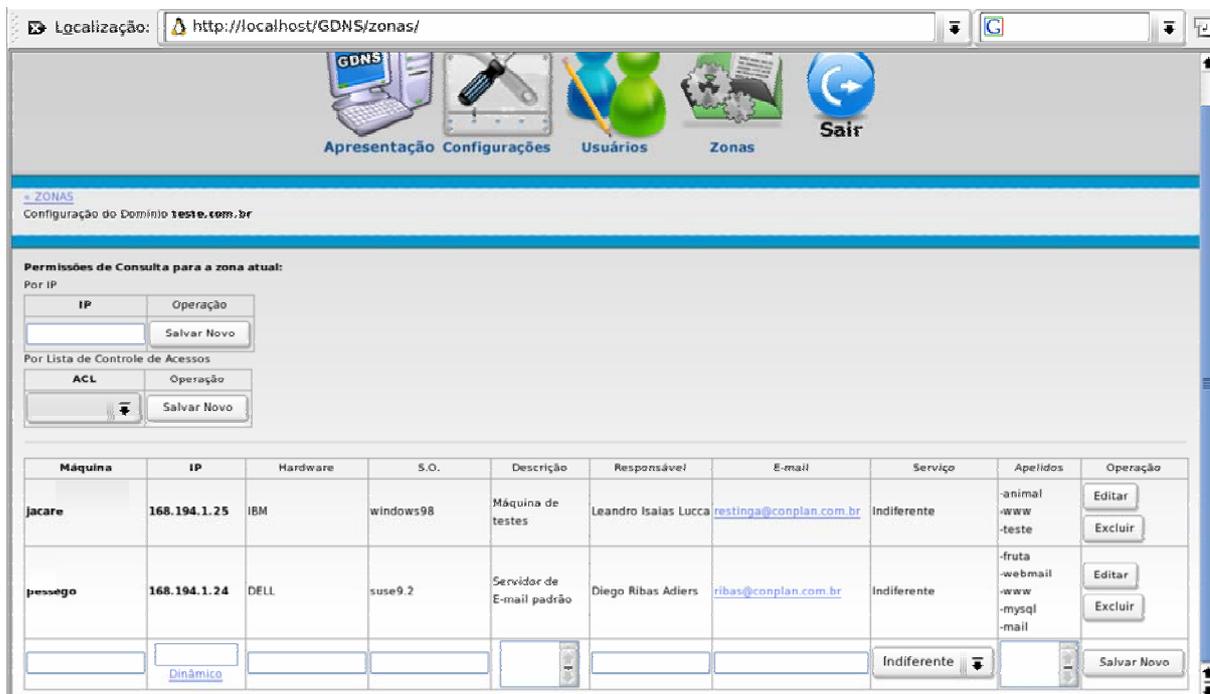
**Figura 29** Tela de configurações iniciais do GDNS no modo usuário

A interface de criação e gerenciamento de zonas pode ser visualizada de acordo com a tela demonstrada na Figura 30.



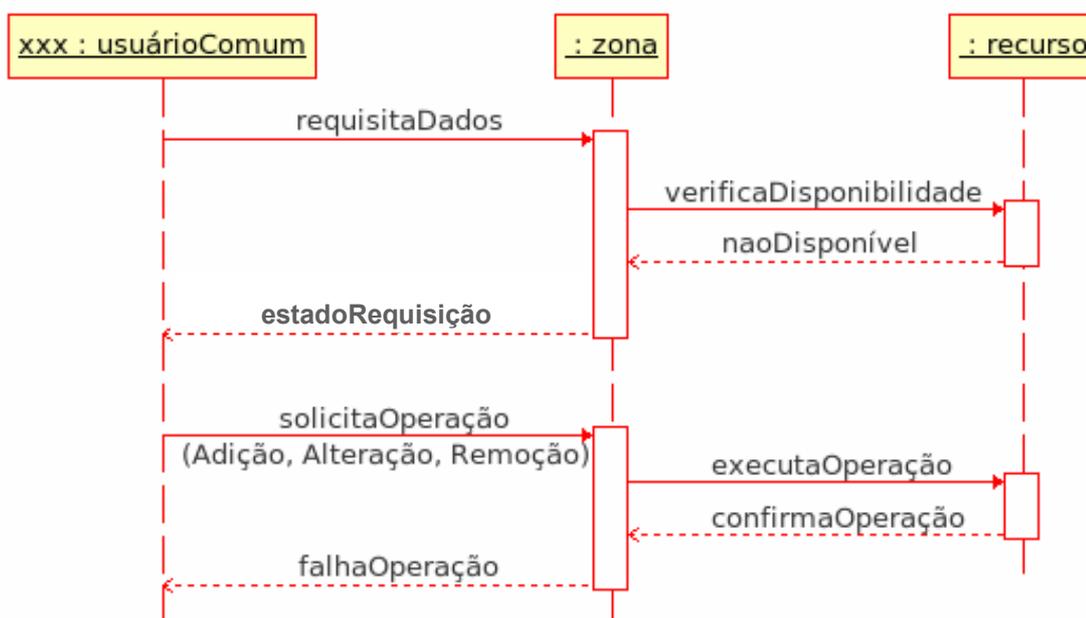
**Figura 30** Tela de criação e gerenciamento de zonas do GDNS no modo usuário

A interface de criação e gerenciamento de recursos de rede, que é considerada a interface de maior importância, pois ela será a mais utilizada pelos usuários comuns, pode ser visualizada de acordo com a Figura 31.



**Figura 31** Tela de criação e gerenciamento de recursos de rede

Para ilustrar melhor o funcionamento da interface de criação e gerenciamento de recursos de rede, segue o diagrama de seqüência de acordo com a Figura 32.



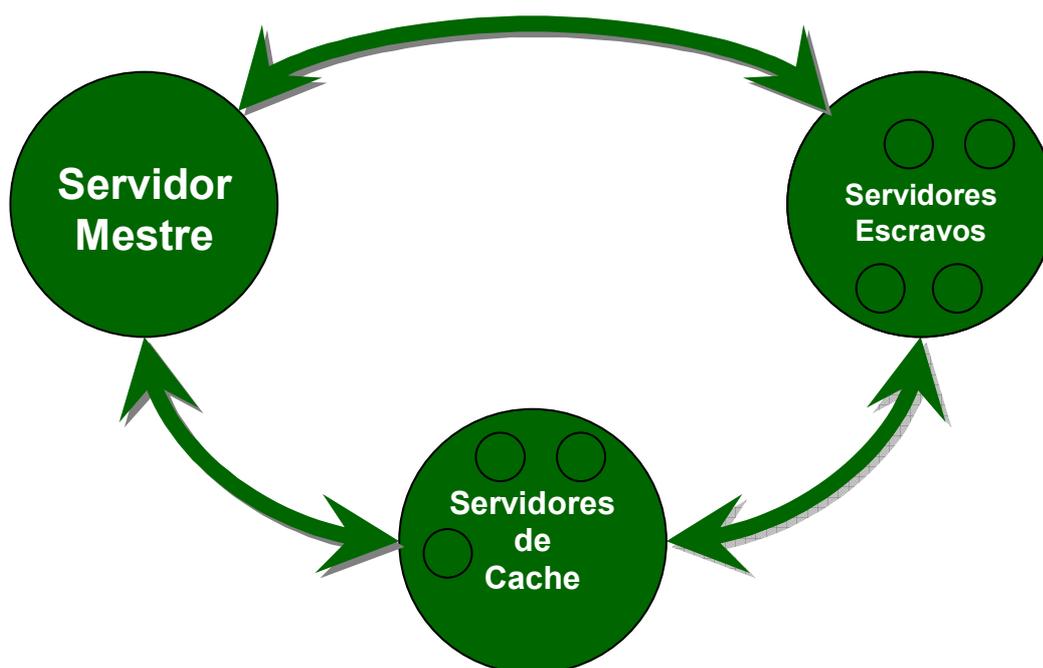
**Figura 32** Diagrama de seqüência da interface de criação e gerenciamento de recursos de rede.

Outro ponto importante é a criação e gerenciamento de ACLs conforme pode ser visto na Figura 31. Esse ponto está intimamente relacionado com a modificação na permissão de consultas das zonas. Uma consulta pode ser feita somente por um grupo de *hosts* (ACLs ou endereços IP) ou por apenas um, sendo que por padrão uma consulta será feita por qualquer *host*.

Por fim, o gerenciamento de encaminhamento de zona também é um recurso disponível ao usuário comum e efetuado zona a zona, ou seja, cada zona possui seus encaminhamentos próprios em caso de ser uma zona do tipo encaminhamento.

#### 4.6 Topologia

Os servidores de DNS devem estar organizados para o bom funcionamento do GDNS como um todo. Essa organização deve seguir uma topologia de acordo com a Figura 33 abaixo:



**Figura 33** Topologia de organização, em forma de anel, dos servidores de nome

A topologia na forma de anel indica como deve ser feita a organização dos servidores de nomes. O GDNS irá efetuar a manutenção do BIND no local onde o usuário estiver autenticado, partindo do pressuposto de que sejam obedecidas as rotas descritas na topologia.

O usuário poderá autenticar em qualquer servidor de nomes, seja ele, mestre, escravo ou somente *cache*. Essa autenticação somente será efetivada caso o

usuário tenha permissão para utilizar o gerenciador no servidor onde fez a tentativa de autenticação. Após a autenticação, o usuário será direcionado para o servidor que necessitar configurar, porém o responsável por este direcionamento é o próprio sistema.

De acordo com a flexibilidade de autenticação e a praticidade que a topologia em forma de anel proporciona, torna-se possível abstrair a localização dos serviços do usuário, esteja ele no modo administrador ou usuário comum, apenas movendo sessões dentro do anel de servidores de nome.

#### **4.7 Comunicação entre os Servidores de Nome**

A comunicação ocorre através do uso da ferramenta RNDC, da troca de chaves entre os servidores em questão e da migração de sessões de autenticação.

Para a efetivação das configurações, os servidores efetuam migração através de troca de sessões, obtendo assim acesso ao RNDC local para fazer o recarregamento de zonas e de configurações globais, através do comando "*rndc reload*". Feito isso as alterações entram em funcionamento e o servidor de nomes fica atualizado, garantindo assim consistência dos dados dos arquivos em questão.

#### **4.8 Atualizações Dinâmicas**

As atualizações dinâmicas envolvidas com o DNS dinâmico deverão ser feitas através do carregamento de scripts externos solicitados ao usuário, os quais, de acordo com o que regulamenta o DNS seguro, usam chaves criptografadas para fazer trafegar as informações. Tais scripts são varridos na procura de falhas ou até mesmo tentativas de inserção de códigos maliciosos, de modo que a submissão do arquivo é efetuada somente em caso de sucesso na varredura.

#### **4.9 Acesso Diretamente ao BIND**

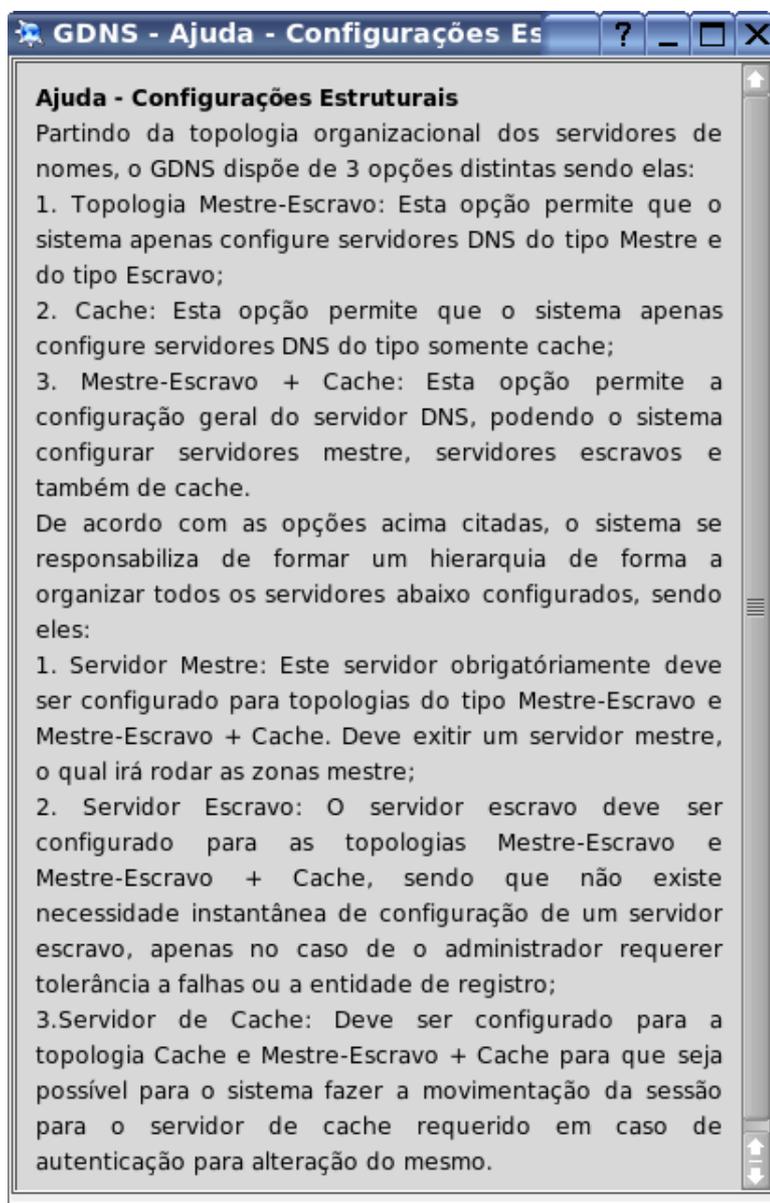
O acesso ao BIND será controlado através do RNDC. Com essa ferramenta o sistema poderá permitir aos usuários (com o devido privilégio interno), a reconfiguração, o recarregamento das zonas, os testes e estatísticas do serviço, bem como reinicialização do BIND.

Servidores DHCP, embora não estejam em questão neste projeto, poderão

utilizar as chaves de segurança para controlar remotamente o BIND, alterando IPs de *hosts* que utilizem DNS dinâmico. Essa alteração poderá ser feita através do contato com scripts que serão submetidos ao sistema de gerenciamento.

#### 4.10 Documentação

A documentação foi integrada ao próprio sistema através da utilização de ajuda on-line (*help on-line*), ou seja, toda a informação necessária foi disponibilizada de modo que o usuário pode obtê-la de acordo com a sua real necessidade. Uma tela de ajuda do sistema pode ser vista como exemplo de documentação na ilustração apresentada pela Figura 34.



**Figura 34** Tela de ajuda sobre configurações estruturais do GDNS

#### **4.11 Implantação e Considerações Finais**

O sistema, inicialmente, foi implantado nos servidores de hospedagem da empresa Conplan Sistemas, podendo ser difundido e implantado por outras empresas ou instituições, de acordo com a aceitação do mesmo.

## 5 RESULTADOS

Os resultados foram obtidos através dos testes efetuados no GDNS de acordo com a subseção 5.1. Tais testes estão descritos juntamente com os resultados obtidos dos mesmos. Após os testes foi efetuado um comparativo para medir as funcionalidades do GDNS ao lado dos principais gerenciadores de DNS existentes.

### 5.1 Testes de Software

Os testes de software têm uma grande importância. Foram feitos testes de sistema, onde o sistema foi varrido em busca de falhas através da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes foram executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia.

De acordo com os testes de sistema, primeiramente foi testado o sistema de autenticação, tendo como dados de teste, nomes de usuário e senha em branco, inexistentes no arquivo de senhas de usuários (arquivo que mantém as informações do usuário) e aleatórios digitados manualmente. Já era esperado que o sistema de autenticação não falhasse partindo do pressuposto que o Apache o teria implementado e testado de forma correta, já que o mesmo mantém o sistema de autenticação a partir da configuração dada.

Foi efetivado o teste de segurança no sistema em busca de falhas na passagem de parâmetros no navegador, falhas de postagem de variáveis externas e falhas de recuperação de arquivos-fonte para *download* diretamente pelo navegador e por “*wget*”. Essas falhas não foram encontradas devido às configurações da diretiva de segurança do servidor PHP estar corretamente configurada e o sistema ser implementado nos moldes propostos seguindo as descrições do PHP.

Outro teste, de extrema importância foi o teste de interface. O mesmo foi realizado efetuando o teste da interface nos navegadores Mozilla, Firefox, Netscape, Internet Explorer e Konqueror nas suas versões mais atuais. Os testes foram efetuados com o intuito de encontrar incompatibilidades entre os navegadores. Neste caso, alguns erros foram encontrados e corrigidos, incluindo em sua maioria erros de Javascript, devido cada navegador ter uma forma específica de

tratar a linguagem. O Javascript foi um dos problemas já esperado e mais comuns, o qual gerou a necessidade de testar qual navegador implementava qual método para que assim o sistema ficasse totalmente *cross-browser*.

Também foi efetuado o teste de consistência dos dados em relação aos arquivos de configuração do BIND. Esse teste foi efetuado adicionando, alterando e removendo dados fictícios e atentando para os arquivos de configuração do BIND de forma que todas as operações fossem efetuadas de acordo com a proposta contida no manual do BIND. Para isso os arquivos foram checados através da ferramenta *named-checkconf* e *named-checkzone* de acordo com a Internet Systems Consortium [13]. Nessa fase de checagem, os arquivos de configuração sofreram inúmeros problemas de escrita em locais incorretos, principalmente no final do arquivo. Esses problemas foram corrigidos e ocorreram devido ao ponteiro do arquivo estar em locais incorretos e ao alto uso de expressões regulares nos testes (expressões regulares causam muita ilegibilidade no código-fonte), mas isso já era esperado.

O teste do funcionamento integrado dos servidores de nome foi efetuado em uma rede com três computadores interligados, de modo que cada um fosse respectivamente servidor mestre, escravo e de *cache* e que as sessões migrassem entre os dois últimos, os três ou os dois primeiros. Um problema inesperado foi encontrado na transmissão da sessão devido a configuração do firewall de um dos computadores que mantinha o servidor mestre. O problema estava no *socket* de conexão com o servidor que não conseguia se comunicar com a porta 8080 (porta atual do servidor web em questão). O problema foi resolvido transmitindo-se através do navegador o arquivo de sessão de forma criptografada para posterior verificação de privilégios e autorização ou não pelo sistema.

Por último, foi efetuado o teste dos arquivos de encaminhamento e de informações sobre usuários, topologia e ACLs, os quais não apresentaram nenhum problema de acordo com os dados cadastrados, alterados e excluídos. Isso já era esperado devido a pequena complexidade destes arquivos e o pequeno número de informações distintas que os mesmos mantém.

Todos os testes de sistema foram efetuados tanto no modo administrador quanto no usuário e foram devidamente corrigidos e retestados de acordo com a necessidade.

Além dos testes de sistema e tomando como base a importância da interface,

foram feitos testes de aceitação. Nestes testes deve-se escolher um grupo de usuários que já tenham algum conhecimento sobre o assunto, de modo que estes sejam responsáveis pela efetivação de testes de sistema e apontem falhas. Foi escolhido então um grupo restrito de dois usuários, mas especificamente administradores de rede que se propuseram. Estes administradores de rede já foram responsáveis pela administração de servidores do Núcleo de Ciência da Computação da UFSM e atualmente trabalham na área, por isso têm certo domínio sobre redes de computadores.

Um dos componentes responsáveis pelos testes de aceitação levantou a hipótese de a migração de sessões entre os servidores de nomes da topologia estar falha devido a sessão estar migrando sem segurança pela rede, então o problema foi resolvido utilizando-se criptografia do tipo MD5 para fazer trafegar a sessão de um servidor para outro.

Outro responsável pelos testes de aceitação apontou a necessidade de uma interface em telas distintas para o gerenciamento dos recursos de cada zona, o que foi modificado no sistema através da colocação de um botão de postagem intitulado “configurar” que remete diretamente para as configurações específicas da zona selecionada.

O auxílio por parte dos responsáveis pelos testes de aceitação reforçou a conferência do sistema, o que permitirá a elevação do nível de satisfação dos clientes ou usuários finais.

Por fim, o sistema foi implantado em versão beta-teste, para que os usuários encontrem possíveis defeitos ou falhas do software e comuniquem ao desenvolvedor.

## **5.2 Comparativo entre Ferramentas de Gerenciamento de DNS**

As ferramentas descritas na seção 3 foram avaliadas com o intuito de examinar suas peculiaridades em comparação com a ferramenta desenvolvida neste projeto, no caso, o GDNS. A Tabela 3 ilustra com clareza as principais características dessas ferramentas em um comparativo com o GDNS, ressaltando os pontos característicos necessários para o gerenciamento de um servidor de DNS em seus recursos mínimos.

Todas as ferramentas de gerenciamento podem ser vistas como excelentes

ferramentas, porém algumas peculiaridades as distinguem da ferramenta desenvolvida, sendo essa última, mais abrangente e menos restrita a questões de flexibilidade, compatibilidade e segurança.

**Tabela 3** Comparativo entre as ferramentas de gerenciamento existentes no mercado e a ferramenta desenvolvida neste projeto

<b>Característica</b>	<b>FreeDNS</b>	<b>Zoneedit</b>	<b>XName</b>	<b>Webmin</b>	<b><u>GDNS</u></b>
Suporte a DNS estático	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
Suporte a DNS dinâmico	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
Gerenciamento de zonas de domínio e subdomínio	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
Controle de encaminhamentos	<b>X</b>		<b>X</b>	<b>X</b>	<b>X</b>
Gerenciamento de usuários	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
Controle de um número ilimitado de zonas			<b>X</b>	<b>X</b>	<b>X</b>
Tradução para português do Brasil				<b>X</b>	<b>X</b>
Vasta gama de registros de recursos são suportados para configuração	<b>X</b>		<b>X</b>	<b>X</b>	<b>X</b>
Gerenciamento de múltiplos servidores DNS em uma só interface de interação com o usuário					<b>X</b>
Controle distinto de permissões de usuários em grupos de usuários			<b>X</b>		<b>X</b>
Versão ilimitada de recursos				<b>X</b>	<b>X</b>
Interface compatível com o linguajar comum de administradores de rede e de fácil utilização					<b>X</b>
Controle de configurações iniciais do arquivo de configuração "named.conf"				<b>X</b>	<b>X</b>

## 6 Conclusão

O DNS é um serviço crítico do qual praticamente todos os outros serviços são dependentes. Pode-se dizer que o DNS é a base da Internet assim como, analogamente, nossos documentos dão a base teórica e conceitual para nossa existência.

Por ser muito requisitado o servidor DNS pode causar inúmeros problemas em uma rede de computadores desde que não esteja devidamente implantado e configurado.

Um servidor de DNS robusto como o BIND exige uma grande gama de conhecimentos sobre suas funcionalidades, sejam elas gerais ou específicas. Gerenciar um servidor de forma correta e eficiente vai muito além de simples configurações de nomes em zonas de DNS direto, porém não é uma tarefa fácil de ser cumprida por administradores de rede com pouca experiência ou com pouco tempo para se informar sobre a melhor forma de manutenção.

Em sua implantação, o GDNS agilizou o processo de configuração de um servidor DNS, reduzindo consideravelmente o tempo de configuração, de forma simples, objetiva, flexível, com eliminação de custos com licença e ainda em linguagem português do Brasil.

Os resultados atingidos seguiram satisfatoriamente o que se esperava. Dentre eles podemos citar o resultado dos testes de aceitação, bem como o sucesso na correção das falhas iniciais do sistema, os quais foram descritos na seção 5.1, que trata os testes de software. Estes últimos tornaram o sistema mais estável e menos sujeito a falhas.

A documentação do GDNS também está disponibilizada de fácil acesso e com uma linguagem de grande aceitação para os administradores de rede.

O método de autenticação com o usuário, usando htaccess do Apache, facilitou bastante a implementação de questões relacionadas à segurança do sistema, deixando-o confiável e eliminando do mesmo a responsabilidade de garantir a segurança, visto que com esse método de autenticação, o apache é o principal responsável.

O GDNS, em sua interface simples e objetiva, bem como seu desenvolvimento via *web*, permitem ao administrador de rede, uma interação mais

habitual com o sistema, podendo assim desfrutar de todas as funcionalidades que um servidor de DNS possa oferecer, de modo a torná-lo útil em um curto espaço de tempo e com uma enorme facilidade. Ainda podendo promover ao usuário uma manutenção mais efetiva do servidor de nomes.

Partindo da necessidade de adaptação e adição de serviços, o GDNS disponibiliza através da sua estrutura de desenvolvimento, um código-fonte totalmente customizável, com a possibilidade de uma simples adição de módulos ao sistema. Essa flexibilidade de gerenciamento de módulos ao sistema se deve intensamente a estrutura de arquivos conforme mostra a estruturação do código-fonte de acordo com a seção 4.1.

Embora nem todos os recursos do BIND tivessem sido utilizados, esta versão do sistema ficou bastante ampla e capaz de sanar qualquer dificuldade comum que uma rede venha a ter em relação à resolução de nomes de domínio.

O desenvolvimento e implantação do GDNS permitiram a ampliação de conhecimentos em relação a redes de computadores em geral, bem como promoveram uma outra visão a respeito de modelagem, organização e estruturação de um sistema em geral, admitindo outra visão sobre as reais necessidades de engenharia e qualidade de software, bem como, sobre automação de tarefas relacionadas a redes de computadores.

O GDNS então, passa a ser uma ferramenta que abrange todas as funcionalidades citadas no decorrer deste projeto, sendo assim um gerenciador de DNS com grandes chances de destaque no mercado e com grande tendência de ampliações e adaptações para administração de inúmeros outros serviços correlacionados.

Este projeto deverá ter uma contribuição significativa para o gerenciamento de redes no âmbito de DNS usando BIND ou até mesmo outras aplicações do mesmo propósito, servindo inclusive como alternativas para painéis de gerenciamento de servidores e serviços.

Enfim, as reais potencialidades deste projeto são de extrema importância para o gerenciamento do serviço mais conflitante e extenso no qual se baseia a Internet: O DNS.

## 7 TRABALHOS FUTUROS

Como trabalhos futuros pretende-se seguir a mesma linha de desenvolvimento, ampliando primeiramente, o número de registros de recursos suportados para configuração pelo GDNS até alcançar a abrangência de todos os suportados pela versão mais atual do BIND.

Em seqüência, pretende-se automatizar as máquinas da rede para que, juntamente com o servidor DHCP, o DNS possa interagir e ser automaticamente configurado.

Por fim, outra linha de desenvolvimento que se pretende seguir é a abrangência do GDNS nos moldes tendentes do mercado de softwares de rede, para que o sistema possa ser implantado em painéis comerciais de gerenciamento de redes de computadores.

## 8 REFERENCIAS BIBLIOGRÁFICAS

[1] 1º Site. **O que é Domínio.** Disponível em: <[http://www.1site.com.br/dominio/oquee\\_dominio.php](http://www.1site.com.br/dominio/oquee_dominio.php)>. Acesso em: 03 de janeiro de 2007.

[2] Apache Software Foundation. **Tutorial do Apache: arquivos .htaccess.** Disponível em: <<http://httpd.apache.org/docs/trunk/howto/htaccess.html>>. Acesso em: 14 de fevereiro de 2007.

[3] BOS, Bert. **XML in 10 Points.** Acessado em <<http://www.w3.org/XML/1999/XML-in-10-points.html>>. W3C, 13 de Novembro de 2001. Acesso em: 14 de fevereiro de 2007.

[4] ALVAREZ, Miguel A. **O que é PHP.** Disponível em <<http://www.criarweb.com/artigos/202.php>>. <criarweb>. Acesso em: 11 de outubro de 2006

[5] ANDERSON, Joshua. **FreeDNS - Free DNS - Dynamic DNS - Static DNS subdomain and domain hosting.** Disponível em: <<http://freedns.afraid.org>>. Acesso em: 10 de outubro de 2006.

[6] CAMPOS, Leonardo S. M. **RNDC utilizando assinatura digital para acessar o servidor BIND.** Disponível em: <<http://www.agatetepe.com.br/tutorial1543.html>>. Acesso em: 19 de dezembro de 2006.

[7] Conectiva Linux. **Configuração de Cache.** Disponível em: <[http://www.conectiva.com/doc/livros/online/10.0/servidor/pt\\_BR/ch09s05.html](http://www.conectiva.com/doc/livros/online/10.0/servidor/pt_BR/ch09s05.html)>. Acesso em: 19 de dezembro de 2006.

[8] Conectiva Linux. **Funcionamento do DNS.** Disponível em: <[http://www.conectiva.com/doc/livros/online/10.0/servidor/pt\\_BR/ch09s02.html](http://www.conectiva.com/doc/livros/online/10.0/servidor/pt_BR/ch09s02.html)>. Acesso em: 19 de dezembro de 2006.

[9] FreeBSD Handbook. **DNS.** Disponível em: <<http://www.openit.com.br/freebsd->

[hb/network-dns.html](#)>. Acesso em: 27 de novembro de 2006.

[10] FUGIOKA, Sérgio. **Administração de Redes Linux**. Disponível em: <<http://wiki.sintectus.com/bin/view/GrupoLinux/AdministacaoDeRedesLinux>>. Acesso em: 29 de dezembro 2006.

[11] GODOY, Jorge. **A Importância do DNS e DNS Reverso**. Disponível em: <<http://www.g2ctech.com/artigos/dns-reverso.html>>. G2C Tech Consultoria Ltda, 15 de janeiro de 2003. Acesso em: 19 de dezembro de 2006.

[12] ICANN - The Internet Corporation for Assigned Names and Numbers. **O ICANN**. Disponível em: <<http://www.icann.org/>>. Acesso em: 08 de janeiro de 2007.

[13] Internet Systems Consortium, Inc. **BIND 9 Administrator Reference Manual**. Disponível em: <<http://www.isc.org/index.pl?/sw/bind/>>. 04 de novembro de 2005. Acesso em: 18 de dezembro de 2006.

[14] Internet Systems Consortium, Inc. **ISC BIND**. Disponível em: <<http://www.isc.org>>. Acesso em: 09 de outubro de 2006.

[15] INTERney. **O que é HTML? Para que serve?** Disponível em: <<http://www.interney.net/blogfaq.php?p=6541494>>. Acesso em: 11 de outubro de 2006.

[16] LANDEMAINE, Charles André. **O que é um subdomínio**. Disponível em: <<http://www.auriance.net/faq/index.php?sid=43001&lang=pt&action=artikel&cat=1&id=1&artlang=pt>>. Auriance, 18 de outubro de 2004. Acesso em: 02 de janeiro de 2007.

[17] LISBOA, Izaias. **O que é JavaScript**. Disponível em: <<http://www.codefactory.com.br/tutoJS/CodeFactory-Tutorial-JavaScript.htm>>. Acesso em: 13 de fevereiro de 2007

[18] MCLink - NetBSD. **O Domain Name System**. Disponível em: <<http://www.mclink.it/personal/MG2508/nbsdbra/chap-dns.html>>. Acesso em: 26 de

dezembro de 2006.

[19] MOCKAPETRIS, P. **Domain Names - Implementations Specification, RFC-1035**. USC/Information Sciences Institute, novembro de 1987. Disponível em: <<http://tools.ietf.org/html/rfc1035>>. Acesso em: 03 de janeiro de 2007.

[20] MOCKAPETRIS, P. **Domains Names - Concepts and Facilities, RFC-1034**. USC/Information Sciences Institute, novembro de 1987. Disponível em: <<http://tools.ietf.org/html/rfc1034>>. Acesso em: 03 de janeiro de 2007.

[21] SANTOS, Luiz C. **DNS**. Disponível em: <<http://www.abusar.org/dns.html>>. ABUSAR – Associação Brasileira dos Usuários de Acesso Rápido, maio de 2004. Acesso em: 19 de dezembro de 2006.

[22] SuperDNS :: Sua solução Brasileira para DNS dinâmico! **FAQ – Perguntas Frequentes: O que é DNS dinâmico**. Disponível em: <<http://www.superdns.com.br/faq.shtml>>. Acesso em: 09 de janeiro de 2007.

[23] TANENBAUM, Andrew S. **A WORLD WIDE WEB**. Redes de Computadores. 4ª ed. Rio de Janeiro: Campus, 2003. p. 651-717.

[24] TANENBAUM, Andrew S. **DNS – DOMAIN NAME SYSTEM**. Redes de Computadores. 4ª ed. Rio de Janeiro: Campus, 2003. p. 616-617.

[25] TANENBAUM, Andrew S. **O espaço de nomes do DNS**. Redes de Computadores. 4ª ed. Rio de Janeiro: Campus, 2003. p. 617-619.

[26] TANENBAUM, Andrew S. **O protocolo IP**. Redes de Computadores. 4ª ed. Rio de Janeiro: Campus, 2003. p. 461-464.

[27] TANENBAUM, Andrew S. **Registros de Recursos**. Redes de Computadores. 4ª ed. Rio de Janeiro: Campus, 2003. p. 620-623.

[28] TANENBAUM, Andrew S.; STEEN, Maarten V. **THE CLIENT-SERVER MODEL**. Distributed Systems – Principles and Paradigms. 1ª ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. p. 42-53.

[29] TORRES, Gabriel. **TCP/IP**. Redes de Computadores – Curso Completo. 1ª ed. Rio de Janeiro: Axcel Books do Brasil, 2001. p. 63-68.

[30] VIXIE, P. **Dynamic Updates in the Domain Name System, RFC 2136**. USC/Information Sciences Institute, abril de 1997. Disponível em: <<http://www.ietf.org/rfc/rfc2136>>. Acesso em: 09 de janeiro de 2007.

[31] Webmin. **Webmin**. Disponível em: <<http://www.webmin.com>>. Acesso em: 18 de fevereiro de 2007.

[32] Wikipédia, a enciclopédia livre. **DHCP**. Disponível em: <<http://pt.wikipedia.org/wiki/DHCP>>. Acesso em: 18 de fevereiro de 2007.

[33] Wikipédia, a enciclopédia livre. **Domínio**. Disponível em: <<http://pt.wikipedia.org/wiki/Domínio>>. Acesso em: 02 de janeiro de 2007.

[34] Wikipédia, a enciclopédia livre. **Modelo OSI**. Disponível em: <[http://pt.wikipedia.org/wiki/Modelo\\_osi](http://pt.wikipedia.org/wiki/Modelo_osi)>. Acesso em: 14 de fevereiro de 2007.

[35] Wikipédia, a enciclopédia livre. **PHP**. Disponível em: <<http://pt.wikipedia.org/wiki/PHP>>. Acesso em: 13 de fevereiro de 2007.

[36] Wikipédia, a enciclopédia livre. **RFC**. Disponível em: <<http://pt.wikipedia.org/wiki/RFC>>. Acesso em: 02 de janeiro de 2007.

[37] XName site. **Free DNS Hosting Service**. Disponível em: <<http://www.xname.org>>. Acesso em: 10 de outubro de 2006.

[38] ZoneEdit. **Free DNS service – Easy, web-based domain manager**. Disponível em: <<http://www.zoneedit.com>>. Acesso em: 10 de outubro de 2006.

## GLOSSÁRIO

**ACL** - Lista de Controle de Acessos.

**Aliases** - Apelidos para serviços ou componentes de uma rede, com uma determinada finalidade.

**Apache** - Servidor de páginas *web* comumente utilizado em servidores linux, porém disponível para outros sistemas operacionais.

**Arquivos-Fonte** - Arquivos cujo conteúdo contém código-fonte de certo programa.

**Árvore** - Estrutura de dados muito comum em programação estruturada, a qual descreve uma hierarquia de memória onde dados podem ser temporariamente armazenados.

**Balanceamento de Carga** - Imposição de que todos os servidores estejam sob as mesmas condições de utilização (carga).

**Beta-teste** - Tempo de teste do sistema, onde os usuários irão avisar sobre possíveis erros e o sistema como um todo será amplamente monitorado.

**Biblioteca** - Junção de código com uma finalidade comum.

**Cache** - Área responsável pelo armazenamento de informações em memória para uso posterior.

**Camada de Aplicação** - É a camada de interação direta com o usuário e a de nível mais alto do modelo OSI.

**Chaves Criptografadas** - Sequência mascarada de caracteres com o intuito de apenas torna-la legível a que tiver os devidos privilégios para lê-la.

**Código-Fonte** – Código em linguagem de programação, legível à programadores que dominam a linguagem em questão.

**Componentes da Rede** - Equipamentos de informática como microcomputadores, pda's, laptops, aparelhos celulares, entre outros.

**Conplan Sistemas** - Empresa que presta consultoria e planejamento no desenvolvimento de sistemas de TI. Acessível através do endereço eletrônico: <http://www.conplan.com.br>.

**Cross-browser** - É uma referência à habilidade de uma aplicação *web* de suportar múltiplos navegadores.

**DHCP** - Protocolo de configuração de host dinâmico. Segundo Wikipédia, a enciclopédia livre [32], o DHCP é um protocolo de serviço TCP/IP que oferece configuração dinâmica de terminais, com concessão de endereços IP de *host* e outros parâmetros de configuração para clientes de rede.

**Download** - Processo de recepção de dados em um transferência de dados de um computador para outro.

**Expressões Regulares** – São expressões utilizadas em várias linguagens de programação com o intuito de, com o simples uso de caracteres coringas, expressarem testes ou imposições.

**Firewall** - Sistema de segurança que protege um sistema operacional contra danos causados por pessoas mal-intencionadas.

**Host** - Identificação de um computador ou da interface de rede de um componente da mesma.

**Hostname** - Nome de *host*.

**Htaccess** - Segundo o Apache Software Foundation [2], os arquivos *.htaccess* permitem invocar um sistema de autenticação através de arquivos de senha diretamente do servidor. Esta facilidade é executada sobre o apache e permite uma autenticação rápida e segura sobre um certo grupo de arquivos.

**Insensitive** - Sem diferenciamento da caixa das letras, ou seja, não importa se a letra é maiúscula ou minúscula.

**Interface** - Seqüência de janelas/telas que determinam os passos a seguir pelo usuário de modo que o mesmo consiga interagir e obter o resultado efetivamente solicitado.

**Interface Amigável** - Interface simples, de fácil localização dos recursos suportados, totalmente prática e intuitiva, permitindo que o usuário possa fazer tudo o que necessita de forma claramente interativa.

**Internet** - Rede mundial de computadores interconectados com inúmeros propósitos.

**Interrogação** - Solicitação de endereço IP ou nome ao servidor DNS.

**Intranets** - Redes internas de computadores, utilizadas por empresas e/ou centros de atividades afins.

**IP** - Protocolo pertencente à camada de rede de acordo com o modelo OSI.

**JavaScript** - Segundo Lisboa [17], O JavaScript é uma linguagem de programação baseada na linguagem JAVA e destinada ao uso em páginas *web* ou em servidores *web*. Seus códigos são inseridos nas páginas HTML sendo executadas pelo navegador do cliente.

**Localhost** - *Host* de acesso local.

**Mapeamento Direto** - Indicação de correspondência com o DNS direto.

**Mapeamento Reverso** - Indicação de correspondência com o DNS reverso.

**Modelo Cliente-Servidor** - Modelo organizacional utilizado em redes de computadores. O cliente é responsável por fazer requisições ao servidor, o qual aceita ou rejeita as requisições e executa-as de acordo com a solicitação do cliente.

**Modelo OSI** - De acordo com a Wikipédia, a enciclopédia livre [34], camadas OSI ou interconexão de sistemas abertos, é um conjunto de padrões ISO relativo à comunicação de dados descrito hierarquicamente do nível mais baixo (próximo ao hardware) para o nível mais alto (próximo ao software/usuário). O modelo OSI é subdividido em 7 camadas, sendo elas: física, de enlace de dados, de rede, de transporte, de sessão, de apresentação e de aplicação.

**Nó** - Parte integrante da árvore responsável pelo armazenamento de um certo dado.

**PHP** - Segundo a Wikipédia, a enciclopédia livre [35], PHP é uma linguagem de programação de computadores interpretada, *opensource* e muito utilizada para gerar conteúdo dinâmico na *Web*.

**Provedor** - Órgão ou entidade responsável por disponibilizar aos usuários serviços de rede e/ou Internet

**Raiz** - Topo da estrutura em árvore de onde partirão as ramificações.

**Ramo** - Nó da estrutura de árvore que está abaixo de um nó raiz ou de outro ramo.

**Réplica** - Cópia fiel de hardware e/ou de software, a qual tem as mesmas funcionalidades, porém não fica em atividade até ser solicitada.

**Resolução de Nomes** - Processo de tradução de um nome de *host* para o seu endereço IP correspondente.

**RFC** - Segundo Wikipédia, a enciclopédia livre [36], o acrônimo também pode significar *Request for Change* ou traduzindo para o português, requisição de mudança. RFCs são responsáveis pela determinação de normas que ditam a organização ou estruturação para certas informações.

**Root Servers** - Servidores de nível mais alto na hierarquia de domínios.

**Script** - Pequeno trecho de código com a função de efetuar alguma tarefa em algum sistema.

**Setar** - Atribuir um valor.

**Socket** - Responsável por criar um canal de transmissão de dados entre duas máquinas.

**Softwares de Gerenciamento** - Programas de computador ou aplicações que se destinam a solução computacional de um problema executando suas tarefas em outro programa ou em arquivos que necessitem manutenção.

**TCP/IP** - Arquitetura descrita no modelo OSI, como um grupo de camadas utilizadas em redes de computadores com o propósito de facilitar a organização e o controle na transmissão de dados a partir do protocolo IP da camada de rede.

**Tolerância à Falhas** - Um sistema tolerante à falhas é um sistema capaz de funcionar mesmo na presença de falhas, abstraindo a ocorrência das mesmas do usuário.

**Topologia** - Forma de organização.

**TTL** - Corresponde ao tempo de vida.

**WEB** - Redução de nome ou apelido para *World Wide Web* (WWW), porém com o mesmo significado.

**Webhosting** - Servidor com o intuito de gerenciar e hospedar sites web.

**Websites** - Sites ou páginas que divulgam informações pela Internet.

**Websites Pessoais** - Websites sob domínio de uma pessoa física.

**Wget** - Comando utilizado em sistemas operacionais linux e unix com o intuito de fazer download completo de serviços disponíveis em conexões TCP.

**XML** - De acordo com BOS [3], XML é uma família de tecnologias com um crescente número de módulos que oferecem serviços úteis para a manipulação, organização e hierarquização de dados.

**Zona Direta** - É a zona responsável pelo mapeamento direto.

**Zona Reversa** - É a zona responsável pelo mapeamento reverso.

## **ANEXOS**

## ANEXO I - Exemplo de Arquivo de Inicialização "named.conf" incluindo Arquivo de Encaminhamentos

```
acl testel { #restinga
    168.194.1.50;
    168.194.1.51;
};
acl testexxxx { #nininha
    168.194.1.111;
};
options {

    # The directory statement defines the name server's working directory
    directory "/var/lib/named";

    # Write dump and statistics file to the log subdirectory. The
    # pathnames are relative to the chroot jail.

    dump-file "/var/log/named_dump.db";
    statistics-file "/var/log/named.stats";

    # The forwarders record contains a list of servers to which queries
    # should be forwarded. Enable this line and modify the IP address to
    # your provider's name server. Up to three servers may be listed.

    # The listen-on record contains a list of local network interfaces to
    # listen on. Optionally the port can be specified. Default is to
    # listen on all interfaces found on your system. The default port is
    # 53.

    #listen-on port 53 { 127.0.0.1; };

    # The listen-on-v6 record enables or disables listening on IPv6
    # interfaces. Allowed values are 'any' and 'none' or a list of
    # addresses.

    listen-on-v6 { any; };

    # The allow-query record contains a list of networks or IP addresses
    # to accept and deny queries from. The default is to allow queries
    # from all hosts.

    allow-query {
        200.18.4.5;
        testel;
    };

    #forward only;

    recursion yes;
    ixfr-from-differences yes;
    dnssec-enable yes;
    notify yes;
    include "/etc/named.d/forwarders.conf";
};
```

```

zone "." in {
    type hint;
    file "root.hint";
};

zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

# Include the meta include file generated by createNamedConfInclude. This
# includes all files as configured in NAMED_CONF_INCLUDE_FILES from
# /etc/sysconfig/named

include "/etc/named.conf.include";
logging {
    category default { default_syslog; };
};

key rndc-key {
    algorithm hmac-md5;
    secret "0ygfdr2i9B8bsIztQ2i00Q==";
};
controls {
    inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { rndc-key; };
};

zone "teste.com.br" in { #nininha
    type master;
    file "master/teste.com.br";
};
zone "1.194.168.in-addr.arpa" in { #nininha
    type master;
    file "master/168.194.1.zone";
};

```

Arquivo de Inicialização "named.conf" com a zona teste.com.br do servidor mestre

```

forwarders {
    168.194.1.60;
};

```

Arquivo de Encaminhamentos

## Exemplo de Arquivo de Configuração de Zona Direta

```

$TTL 38400
@      IN      SOA      ns1.teste.com.br. nininha.x.com.br. (
                          2007021801
                          10800
                          3600
                          604800
                          1w )

@      IN      NS       ns1
@      IN      NS       ns2

localhost  IN  A       127.0.0.1
ns2        IN  A       168.194.1.11
jacare     IN  A       168.194.1.25
pessego    IN  A       168.194.1.24
@          IN  A       168.194.1.10

animal     IN  CNAME    jacare
www        IN  CNAME    jacare
teste      IN  CNAME    jacare
fruta      IN  CNAME    pessego
webmail    IN  CNAME    pessego
www        IN  CNAME    pessego
mysql      IN  CNAME    pessego
mail       IN  CNAME    pessego

jacare     IN  HINFO    IBM windows98
pessego    IN  HINFO    DELL suse9_2

jacare     IN  TXT      "Máquina de testes"
pessego    IN  TXT      "Servidor de E-mail padrão"

jacare     IN  RP       restinga.conplan.com.br. Leandro_Isaias_Lucca
pessego    IN  RP       ribas.conplan.com.br.   Diego_Ribas_Adiers

```

Arquivo de Configuração de Zona Direta do Domínio "teste.com.br"

**ANEXO II - Exemplo de Arquivo de Configuração de Zona Reversa**

```
$TTL 38400
@      IN      SOA      ns1.teste.com.br. nininha.x.com.br. (
                                2007021801
                                10800
                                3600
                                604800
                                1w )

@      IN NS     ns1.teste.com.br.
@      IN NS     ns2.teste.com.br.

10     IN PTR    teste.com.br.
11     IN PTR    ns2.teste.com.br.
25     IN PTR    jacare.teste.com.br.
24     IN PTR    pessego.teste.com.br.
```

Arquivo de Configuração de Zona Reversa da Faixa de IPs 168.194.1

### ANEXO III - Exemplo de Arquivo Cache (Raiz)

```

; This file holds the information on root name servers needed to
; initialize cache of Internet domain name servers
; (e.g. reference this file in the "cache . <file>"
; configuration file of BIND domain name servers).
;
; This file is made available by InterNIC
; under anonymous FTP as
; file /domain/named.root
; on server FTP.INTERNIC.NET
; -OR- RS.INTERNIC.NET
;
; last update: Jan 29, 2004
; related version of root zone: 2004012900
;
;
; formerly NS.INTERNIC.NET
;
. 3600000 IN NS A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4
;
; formerly NS1.ISI.EDU
;
. 3600000 NS B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000 A 192.228.79.201
;
; formerly C.PSI.NET
;
. 3600000 NS C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000 A 192.33.4.12
;
; formerly TERP.UMD.EDU
;
. 3600000 NS D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000 A 128.8.10.90
;
; formerly NS.NASA.GOV
;
. 3600000 NS E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET. 3600000 A 192.203.230.10
;
; formerly NS.ISC.ORG
;
. 3600000 NS F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET. 3600000 A 192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
. 3600000 NS G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET. 3600000 A 192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
. 3600000 NS H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET. 3600000 A 128.63.2.53
;
; formerly NIC.NORDU.NET
;

```

```
.                3600000    NS      I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET. 3600000    A       192.36.148.17
;
; operated by VeriSign, Inc.
;
.                3600000    NS      J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET. 3600000    A       192.58.128.30
;
; operated by RIPE NCC
;
.                3600000    NS      K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET. 3600000    A       193.0.14.129
;
; operated by ICANN
;
.                3600000    NS      L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET. 3600000    A       198.32.64.12
;
; operated by WIDE
;
.                3600000    NS      M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET. 3600000    A       202.12.27.33
; End of File
```

Arquivo Cache com os 13 Servidores de Nomes RAIZ

## ANEXO IV - Exemplo de Arquivos de Interface Local

```

$TTL 1W
@           IN SOA      @      root (
                42          ; serial (d. adams)
                2D          ; refresh
                4H          ; retry
                6W          ; expiry
                1W )        ; minimum

                IN NS     @
                IN A      127.0.0.1

```

Arquivo de Interface Local de Zona Direta

```

$TTL 1W
@           IN SOA      localhost.  root.localhost. (
                42          ; serial (d. adams)
                2D          ; refresh
                4H          ; retry
                6W          ; expiry
                1W )        ; minimum

                IN NS     localhost.
1           IN PTR      localhost.

```

Arquivo de Interface Local de Zona Reversa