Serviço DNS BIND agosto/2015

Instalação e configuração do serviço DNS BIND.

1 - DNS BIND (Berkeley Internet Name Domain)

DNS pode significar Domain Name System ou Domain Name Service. Domain Name System é um sistema que inclui o Domain Name Service.

Domain Name Service é um serviço que mapeia endereço IP e Nome de Domínio Totalmente Qualificado[1] (FQDN) de um para o outro.

Hosts que hospedam DNS são chamados de name servers [servidores de nomes]. Normalmente, os membros da família Unix usam o DNS BIND (Berkeley Internet Name Domain), que é a aplicação mais comum para resolver nomes na Internet.

BIND é uma implementação de código fonte aberto do protocolo DNS, e está em uso em cerca de 75% dos serviços de nomes na Internet.

Este serviço tem a função básica de traduzir em endereços IPs nomes de *hosts* e domínios e vice-versa. Sabemos que nos pacotes TCP, nos seus *headers*, vão o endereço IP de destino e de retorno, porém o usuário não precisa ter a preocupação em saber esses IPs, basta fazer uma consulta [query] ao serviço DNS que ele traduz o IP, dados o nome do *host* e o domínio.

A principal função do serviço de nomes é mapear endereços IPs em nomes lidos pelos

1 - FQDN significa *Fully Qualified Domain Name*. Este é o nome de um determinado host composto por seu hostname seguido do domínio a que pertence. Como exemplo temos um serviço de e-mail de hostname **mail**, pertencente ao domínio **exemplo.com**. Então, o FQDN desse host é **mail.exemplo.com**.

humanos.

Por exemplo, se alguém quiser acessar o site em **www.jairo.pro.br**, pacotes TCP devem ser enviados para a porta 80 de um *host* nesse domínio, porém para qual IP? O serviço DNS responde ao cliente que atualmente **www.jairo.pro.br** encontra-se no IP 187.73.33.34. Isso é chamado de *forward DNS*.

Da mesma forma que obteve o endereço IP do host/domínio, poderia obter o host/domínio dado o IP, que é a parte reversa do serviço DNS. Isso é chamado de *reverse DNS*.

Um outra função do serviço DNS é realizar cache local dos IPs já resolvidos, isso para evitar fazer novas consultas externas a hosts/domínios já traduzidos. Desse modo, atende mais rapidamente aos clientes do serviço e economiza *link* de acesso à internet.

O DNS, Domain Name System, é composto de três partes:

1 – resolvedor [resolver];
2 – serviço de nome [name server];
3 – banco de dados com recursos gravados [database of resource records(RRs)].

Basicamente, o Domain Name System contém um grande database que reside em vários computadores, com o objetivo de identificar os nomes e endereços IPs dos vários hosts e domínios na internet.

O Domain Name System é usado para prover informação ao Domain Name Service, quando houver query [consulta]. O serviço é o ato de fazer query no database, e o sistema é a estrutura e dados que o compõem.

O Domain Name System database é dividido em seções chamadas zones. Os servidores de nomes [name servers] em suas respectivas zones são os responsáveis por responder às queries para as suas zones.

Uma zone é uma subtree [sub árvore] do DNS e é administrado separadamente. Para cada

zone, usualmente existe um name server primário e um ou mais name server secundário. Um name server pode ser autoridade sobre mais de uma zone.

Os nomes [DNS names] são atribuídos através de registros, pela IANA[2] [Internet Assigned Number Authority]. O domain name [nome do domínio] é um nome atribuído para um domínio internet. Por exemplo, uninove.br é o nome do domínio de uma instituição de ensino.

Já a nomeação de hosts dentro do domínio é atribuição dos administradores do domínio.

O acesso ao domain name database ocorre através de um resolver [a partir de uma aplicação cliente]. O resolver envia requisições aos serviços de nomes, que retornam a informação requisitada pelo usuário. A requisição é feita no endereço IP do name server.

O DNS é uma base de dados hierárquica, distribuída **globalmente** e gerenciada **localmente.**

A raiz dessa hierarquia distribuída é constituída pelos servidores-raiz ***root-servers***. Inicialmente os **root-servers** eram apenas sete, todos localizados nos Estados Unidos e geridos pela ***IANA***. Posteriormente, foram ampliados para treze ***root-servers***, dois dos quais ficaram localizados na Europa e um no Japão.

Mas logo depois se viu que treze também era pouco, devido ao rápido crescimento da internet, com consequente aumento substancial na carga de consultas nesses **root-servers**. O problema agora era que, atingido treze **root-servers**, não poderia mais ocorrer ampliação no número de **root-servers** por causa de uma limitação no protocolo DNS.

A solução então foi desenvolver uma técnica chamada *anycast* para clonar os **root-servers** e assim criar servidores espelhos, que operacionalmente não se distinguem dos **root-servers** originais. Esses servidores espelhos foram distribuídos pelo mundo inteiro.

Atualmente já existem cerca de 320 **root-servers** "clonados" distribuídos pelo mundo, e

2 IANA: Internet Assigned Numbers Authority [www.iana.org] é a responsável pela coordenação global do DNS root, endereçamento IP e outros recursos do protocolo IP.

que são atualizados pelos 13 **root-servers** originais.

No Brasil, 14 cidades têm **root-servers**: Belém, Belo Horizonte, Brasília, Campinas, Curitiba, Florianópolis, Fortaleza, Londrina, Natal, Porto Alegre, Rio de Janeiro, Salvador, São José dos Campos e São Paulo.

Essa distribuição regional reduziu a carga nos treze **root-servers** originais e tornou a resolução de nomes mais rápida.

O conteúdo da raiz do DNS está baseado em TLD [Top Level Domain].

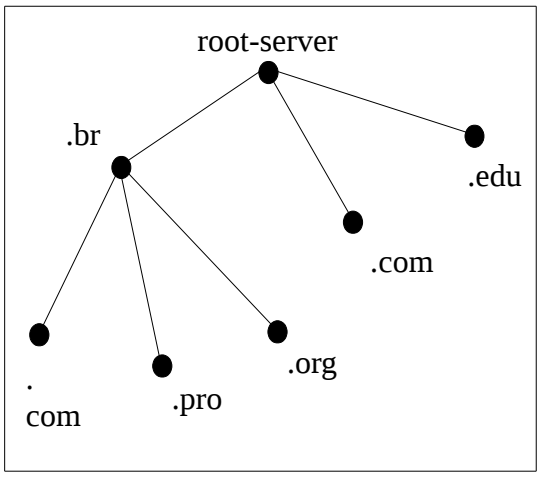
Os TLD de 3 letras foram os primeiros a serem criados e, originalmente, eram domínios Norte-Americanos. Posteriormente, alguns deles foram reclassificados como *gTLD (TLD genéricos*, *mundiais)*. São eles:

.edu: rede acadêmica;
.com: segmento do comércio/indústria;
.gov: governo norte-americano;
.net: atividades de suporte à rede;
.org: organizações não governamentais;
.mil: segmento militar (Arpanet);
.int: organizações internacionais.

Os TLD de 2 letras vieram em 1986. Nesse caso, cada país corresponde a duas letras e são chamados de ccTLD (Country Codes TLD). Alguns exemplos são:

.**ar**: Argentina
.au: Austrália;
.br: Brasil (registrado em 18 de abril de 1989);
.ca: Canadá;
.ch: Suiça;
.cl: Chile;

.de: Alemanha;
.fr: França;
.it: Itália;
.jp: Japão;
.mx: México;
.pt: Portugal;
.ru: Russia;
.tw: República da China;
.us: Estados Unidos da América.



Os ccTLD gozam de autonomia para estabelecer sua árvore hierárquica e para estabelecer sua abrangência e normas de registro. No Brasil, a autoridade sobre o ccTLD está em **registro.br**.

A configuração do serviço DNS envolve diferentes registros. Os principais tipos de registros [os mais comuns] são:

- A: Address, especifica um endereço IP direto;
- AAAA: Address Ipv6, especifica um endereço Ipv6;
- NS: name server, especifica serviços DNS para o domínio ou subdomínio;
- CNAME: Canonical NAME, um apelido para outro hostname;
- MX: Mail eXchanger [ou exchange], o serviço de email;
- PTR: PoinTeR, aponta o hostname/domínio reverso a partir de um endereço IP;
- SOA: Start Of Authority, responsável por respostas autoritativas por um domínio;
- TXT: Registro de texto, com formato arbitrário [serve para diversas funcionalidades];
- LOC: Localização geográfica;
- SRV: Serviços, proporciona a localização de serviços conhecidos;
- DNAME: Domain Alias ou apelido para domínio. É semelhante a CNAME, porém trata de apelido para todo o domínio e não apenas para um hostname.

2 – Funcionamento do DNS

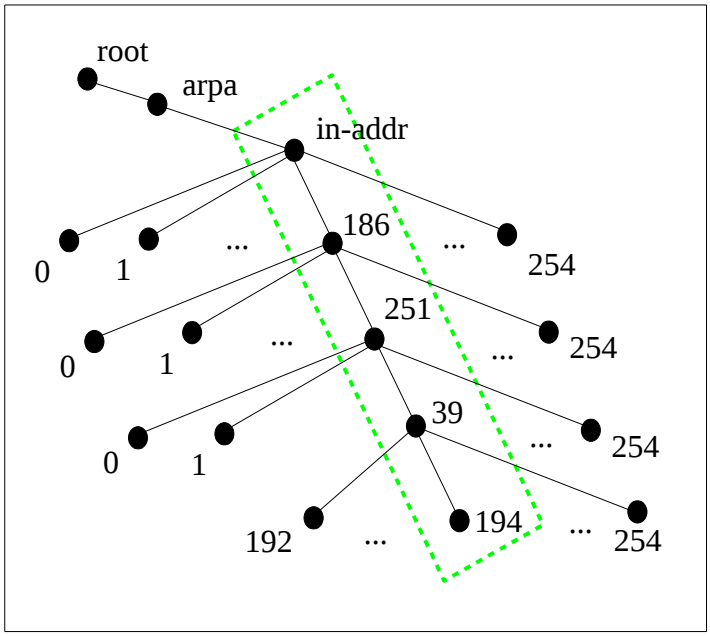
A estrutura hierárquica do DNS funciona como uma árvore invertida, começando pelo root-server [**.**] e seguindo pelos TLD (Top Level Domains), que são divididos em genéricos (gTLD) usados em todo o mundo e os de código de país (ccTLD), que possuem extensões de domínios administrados pelos países.

Para o DNS funcionar corretamente é necessário no mínimo dois servidores: um *master* [primário] e um *slave* [secundário]. O *slave* depende do *master* para as suas atualizações, porém pode responder sem a existência deste. Então, em caso de falha do *master*, o DNS continua funcionando. Operacionalmente, após a instalação e configuração do *master*, configura-se o *slave* especificando quem é o *master* do domínio. Desse modo, qualquer alteração no database do *master* será replicada automaticamente para o *slave*.

Para um determinado domínio na internet podem existir vários *slaves*, porém apenas um *master*.

Quanto ao reverso, ele é controlado pelas entidades que possuem os endereços IP. Nesse caso, o dono pode escolher sub-delegar DNS reverso em uma faixa de IP para alguma outra entidade, que por sua vez também pode sub-delegar. IANA é quem possui todos os endereços IP, então o início dessa sub-delegação ocorre lá.

IANA delega endereçamento de IP aos Registros Internacionais de Internet [RIR: Regional Internet Registry] em 5 regiões: **AfriNIC** [África], **APNIC** [Ásia/Pacífico], **ARIN** [América do Norte], **LACNIC** [América Latina] e **RIPE** [Europa, Oriente Médio e Ásia Central. Por sua vez, esses Registros Internacionais delegam os endereços IP aos provedores de acesso, que por sua vez delegam aos usuários finais.



Para a implementação do reverso, existe um domínio especial reservado chamado **in-addr.arpa**, ao qual todos os endereços IP pertencem. Por exemplo, 192-255.39.251.186.in-addr.arpa para a faixa de IPs 186.251.39.192 a 186.251.39.255. E para chegar aos endereços de host, primeiro terá de perguntar a um **root-server** onde fica 186.0.0.0/8, depois onde fica 186.251.0.0/16 e por último, 186.251.39.0/24. Por exemplo, para obter o reverso do host no IP 186.251.39.194, o caminho seria o seguido na figura ao lado.

3 – Instalação do DNS BIND

Essa instalação será para um serviço standalone, e não inetd.

Para descobrir se o serviço DNS BIND está instalado, uma dica simples é procurar pelo seu script de inicialização em **/etc/init.d**:

```
shell# ls /etc/init.d | grep named
/etc/init.d/named
```

Se não houver saída no comando acima é indicativo de que o serviço **DNS** não está instalado. Nesse caso, instalar com o comando **yum**:

```
shell# yum  install  bind  bind-utils
```

NOTA:

Num Ubuntu, o script de inicialização é "**/etc/init.d/bind9**". E para instalar, o comando é "**apt-get install bind9 dnsutils**".

O pacote bind instala o DNS BIND, e bind-utils instala as aplicações nslookup, dig e nsupdate, entre outras.

Após instalado, verificar se existem os seguintes arquivos:

```
shell# file  /etc/init.d/named
/etc/init.d/named: Bourne shell script text executable
shell# file  /usr/sbin/named
/usr/sbin/named: ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), dynamically linked (uses
shared libs), for GNU/Linux 2.6.18, stripped
shell# file  /var/named/chroot/etc
/var/named/chroot/etc: directory
```

onde:

- **/etc/init.d/named** é o script de inicialização do serviço DNS;
- **/usr/sbin/named** é o executável que ao rodar dá origem ao processo daemon;
- **/var/named/chroot/etc** é o diretório onde ficam as configurações do serviço named. O arquivo de configuração é **named.conf**.

NOTA:

Num Ubuntu, o arquivo de configuração está em "**/etc/bind9/named.conf**".

Para saber se o serviço está rodando, usar o comando **ps** e procurar pelo processo **named**:

```
shell# ps -ef | grep named
```

Se não houver saída no comando acima, indica que o processo não está rodando. Se estivesse rodando, deveria ser parado com o script de inicialização:

```
shell# /etc/init.d/named stop
Parando o named:                           [  OK  ]
```

4 – Configuração da zone aluno.br

Num Linux baseado no Red Hat (CentOS), o diretório de configuração do serviço DNS BIND é **/var/named/chroot/etc**. Vamos entrar nesse diretório como o comando **cd** e depois listar os arquivos que tem lá:

```
shell# cd /var/named/chroot/etc
shell# ls
```

O padrão da instalação do serviço **BIND** na versão 9 no Red Hat (CentOS) é não disponibilizar nenhum arquivo de configuração.

Então, para facilitar e agilizar a configuração da zone **aluno.br**, serão copiados mapas e configurações já preparados e disponíveis em **www.jairo.pro.br/bind93.tar.gz**.

Para baixar esses arquivos será usado o comando **wget**, porém antes precisar acertar a variável **http_proxy**:

```
shell# export  http_proxy=http://RA:SENHA@186.251.39.196:3128
```

Onde:

RA: é o RA do aluno;

SENHA: é a senha de acesso do aluno;

186.251.39.196: é o IP do serviço proxy, que atende na porta **3128** [é um Squid].

Para confirmar se a variável **http_proxy** está correta, usar o comando **echo**:

```
shell# echo  $http_proxy
http://RA:SENHA@186.251.39.196:3128
```

Depois disso, é só baixar o arquivo **bind93.tar.gz** de **www.jairo.pro.br** com o comando **wget**:

```
shell# wget  www.jairo.pro.br/bind93.tar.gz
--2012-10-27 23:27:27--  http://www.jairo.pro.br/bind93.tar.gz
Resolving www.jairo.pro.br... 187.73.33.34
Connecting to www.jairo.pro.br|187.73.33.34|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3862 (3.8K) [application/x-gzip]
Saving to: “bind93.tar.gz”

100%[======================================>] 3,862       --.-K/s   in 0.009s

2012-10-27 23:27:27 (439 KB/s) - “bind93.tar.gz” saved [3862/3862]
```

NOTA:

Se o sistema operacional fosse um Ubuntu [e não CentOS], o arquivo seria **bind9.tar.gz**, e antes de efetuar o download, deveria entrar no diretório **/tmp**.

Por fim, descompactar e extrair o conteúdo do arquivo **bind93.tar.gz** com os comandos **gunzip** e **tar**:

```
shell# gunzip  bind93.tar.gz
shell# tar  -xvf   bind93.tar
aluno.br
localdomain.zone
localhost.zone
localtime
named.broadcast
named.conf
named.conf.local
named.conf.options
named.ip6.local
named.local
named.root
named.root.hints
rev.aluno.br
rndc.conf
```

O arquivo de configuração é o **named.conf**. Para visualizar o conteúdo desse arquivo, usar o comando **more**:

```
shell# more  named.conf
//
// See the BIND Administrator's Reference Manual (ARM) for details, in:
//   file:///usr/share/doc/bind-*/arm/Bv9ARM.html
// Also see the BIND Configuration GUI : /usr/bin/system-config-bind and
// its manual.
//
options
{
        // Those options should be used carefully because they disable port
        // randomization
        // query-source    port 53;
        // query-source-v6 port 53;

        // Put files that named is allowed to write in the data/ directory:
        directory "/var/named"; // the default
        dump-file               "data/cache_dump.db";
        statistics-file         "data/named_stats.txt";
        memstatistics-file      "data/named_mem_stats.txt";

        include "/etc/named.conf.options";
};

include "/etc/named.root.hints";

include "/etc/named.conf.local";

include "/etc/rndc.conf";
```

Repare que as linhas iniciadas por // [duas barras direitas] são linhas comentadas, e que no final desse arquivo tem 4 *includes*, que incluem como configuração também os arquivos **named.conf.options**, **named.root.hints**, **named.conf.local** e **rndc.conf**.

NOTA:

No caso do Ubuntu, o arquivo **bind9.tar.gz** contém **named.conf**, **named.conf.options**, **named.conf.local**, **aluno.br** e **rev.aluno.br**. Basta extrair o conteúdo de **bind9.tar.gz** no diretório **/tmp** e copiar os 5 arquivos acima para **/etc/bind**.

Repare também que, embora o diretório onde estão as configurações seja **/var/named/chroot/etc**, o caminho absoluto para essas *includes*, que aparece em **named.conf**, é **/etc**.

Nessa configuração, o arquivo **named.conf** não precisa ser alterado.

No arquivo **named.conf.options** tem a configuração "forwarders", que deve conter os endereços IP dos servidores de nome da Uninove, atualmente 186.251.39.194 [master] e 186.251.39.195 [slave]. É através dos "forwarders" que o nosso serviço de nomes conseguirá resolver nomes externos, na internet.

Com o comando **more** podemos ver o conteúdo do arquivo **named.conf.options**:

```
shell# more  named.conf.options

        // If there is a firewall between you and nameservers you want
        // to talk to, you might need to uncomment the query-source
        // directive below.  Previous versions of BIND always asked
        // questions using port 53, but BIND 8.1 and later use an unprivileged
        // port by default.

        // query-source address * port 53;

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {
                186.251.39.194;
                186.251.39.195;
        };
```

Então, como poder visto acima, nada precisa ser modificado no arquivo **named.conf.options**.

Já **named.conf.local** inclui a zone **aluno.br**. Isso pode ser visto com o comando **more**:

```
shell# more  named.conf.local
//
// Do any local configuration here
//

// dominio aluno.br
zone "aluno.br" {
        type master;
        file "/etc/aluno.br";
};

// reverso de aluno.br
zone "1.102.10.in-addr.arpa" {
        type master;
        file "/etc/rev.aluno.br";
};
```

Nesse arquivo, talvez seja necessário alterar o endereço da rede no reverso. Na configuração acima está "1.102.10.in-addr.arpa", que se refere à rede 10.102.1.0.

Convém notar também em **named.conf.local** que estamos configurando um DNS master, e que os mapas da zone **aluno.br** estão nos arquivos **aluno.br** e **rev.aluno.br**. No arquivo **rev.aluno.br** está o mapa reverso, para traduzir IPs em nomes.

Desse modo, tudo que resta a fazer agora é [se for o caso] acertar o endereço da rede no arquivo **aluno.br**:

```
shell# more  aluno.br
aluno.br.       IN      SOA             ns1.aluno.br. admin.aluno.br. (
                        2006081401      ; serial
                        28800           ; refresh
                        3600            ; retrai
                        604800          ; expire
                        38400           ; minimum
)

aluno.br.       IN      NS              ns1.aluno.br.
aluno.br.       IN      MX      5       mta.aluno.br.

ns1             IN      A               10.102.1.10
mta             IN      A               10.102.1.250
pop             IN      CNAME           ns1
www             IN      CNAME           mta
```

NOTA:

No final da primeira linha do arquivo **aluno.br**, "admin.aluno.br" refere-se ao e-mail do administrador do DNS, admin@aluno.br.

Convém notar que o registro MX admite prioridades, por exemplo 5, 10, etc. Quanto menor esse número, maior a sua prioridade. Nesse caso, se houvese mais de um serviço de e-mail, o de maior prioridade é que receberia os e-mails.

No arquivo **rev.luno.br** é que está o mapa para resolução reversa:

```
shell# more  rev.aluno.br
@       IN      SOA     ns1.aluno.br. admin.aluno.br. (
        2006081401;
        28800;
        604800;
        604800;
        86400
)
                IN      NS              ns1.aluno.br.
10              IN      PTR             ns1.aluno.br.
250             IN      PTR             mta.aluno.br.
```

Depois de corrigir o arquivo **aluno.br** [se for o caso], é so iniciar o serviço DNS BIND.

5 – Iniciar o serviço DNS BIND

Antes de iniciar o serviço DNS, verificar quais portas TCP estão abertas. Para isso, é necessário a aplicação **nmap** para fazer um scan de portas:

```
shell# nmap  localhost

Starting Nmap 4.76 ( http://nmap.org ) at 2009-09-07 15:31 BRT
Interesting ports on localhost (127.0.0.1):
Not shown: 999 closed ports
PORT    STATE SERVICE
631/tcp open  ipp

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

que mostra apenas a porta 631 [serviço de impressão] aberta.

E também, antes de iniciar o serviço DNS, verificar se existe o processo daemon **named** rodando:

```
shell# ps  -ef  |  grep  named
```

que não deve ter saída, indicando que esse daemon não está rodando.

Agora, então, iniciar o serviço DNS com o script de inicialização:

```
shell# /etc/init.d/named  start
Iniciando o named:                         [  OK  ]
```

Num Ubuntu, o comando seria "**/etc/init.d/bind9 start**".

Para checar se o serviço iniciou corretamente, verificar no arquivo de logs do sistema:

```
shell# tail -30 /var/log/messages
```

NOTA:
Num Ubuntu, o comando seria "**tail -30 /var/log/syslog.log**".

Depois disso, o scan de portas vai mostrar que a porta 53 também está aberta:

```
shell# nmap localhost

Starting Nmap 4.76 ( http://nmap.org ) at 2009-10-12 21:38 BRT
Interesting ports on localhost (127.0.0.1):
Not shown: 998 closed ports
PORT    STATE SERVICE
53/tcp  open  dns
631/tcp open  ipp

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
```

No caso do serviço DNS, ele responde também na porta 53 UDP. Para determinar se essa porta está aberta, usar **nmap -sU** para fazer um scan UDP:

```
shell# nmap -sU localhost

Starting Nmap 4.76 ( http://nmap.org ) at 2009-10-14 22:11 BRT
Interesting ports on localhost (127.0.0.1):
Not shown: 996 closed ports
PORT     STATE         SERVICE
53/udp   open|filtered domain
631/udp  open|filtered ipp

Nmap done: 1 IP address (1 host up) scanned in 1.25 seconds
```

E o comando **ps** vai mostrar que o daemon **named** agora está rodando:

```
shell# ps -ef | grep named
root     2354     1  0 13:41 ?        00:00:00 /usr/sbin/named
```

6 – Testar a resolução de nomes

Para testar, usar as aplicações clientes **nslookup** e **dig** e enviar essa query para localhost, que é onde está o serviço de nomes:

```
shell# nslookup www.aluno.br localhost
Server:         localhost
Address:        127.0.0.1#53

www.aluno.br    canonical name = mta.aluno.br.
Name:   mta.aluno.br
Address: 10.102.1.250
```

```
shell# nslookup mta.aluno.br localhost
Server:         localhost
Address:        127.0.0.1#53

Name:   mta.aluno.br
Address: 10.102.1.250
```

```
shell# nslookup  ns1.aluno.br  localhost
Server:         localhost
Address:        127.0.0.1#53

Name:   ns1.aluno.br
Address: 10.102.1.10
```

```
shell# nslookup  10.102.1.250  localhost
Server:         localhost
Address:        127.0.0.1#53

250.0.102.10.in-addr.arpa       name = mta.aluno.br.
```

```
shell# dig  www.aluno.br  @localhost

; <<>> DiG 9.6.1-P1-RedHat-9.6.1-6.P1.fc11 <<>> www.aluno.br @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45917
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.aluno.br.          IN              A

;; ANSWER SECTION:
www.aluno.br.           38400           IN      CNAME           mta.aluno.br.
mta.aluno.br.           38400           IN      A               10.102.1.250

;; AUTHORITY SECTION:
aluno.br.               38400           IN      NS              ns1.aluno.br.

;; ADDITIONAL SECTION:
ns1.aluno.br.           38400           IN      A               10.102.1.10

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Oct 12 21:09:13 2009
;; MSG SIZE  rcvd: 98
```

Fazer também uma resolução do FQDN www.aluno.br na máquina do colega, onde 10.102.1.XX é o endereço IP onde está o serviço de nomes [substituir XX pelo IP do host]:

```
shell# nslookup  www.aluno.br  10.102.1.XX
Server:         10.102.1.XX
Address:        10.102.1.XX#53

www.aluno.br    canonical name = mta.aluno.br.
Name:   mta.aluno.br
Address: 10.102.1.250
```

Testar também a resolução para um site externo:

```
shell# nslookup  www.jairo.pro.br  localhost

Server:         localhost
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   www.jairo.pro.br
Address: 187.73.33.34
```