



Comunicações por Computador

TP3: Serviço de Resolução de Nomes (DNS)

Guilherme da Silva Amorim Martins A89532
José Pedro Carvalho Costa A89519
Simão Paulo da Gama Castel-Branco e Brito A89482



A89532



A89519



A89482

04 de maio de 2021

1 Parte I: Consultas ao serviço de nomes DNS

1.1 Pergunta a)

Qual o conteúdo do ficheiro `/etc/resolv.conf` e para que serve essa informação?

Este ficheiro contém informação que determina os parâmetros do DNS a resolver.

O DNS resolver permite às aplicações que estão a correr no sistema operativo, traduzir os nomes do domínio para endereços de IP (numéricos) que são necessários para aceder aos recursos da rede local ou à Internet.

```
core@core-VirtualBox:~$ cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search eduroam.uminho.pt
```

Figure 1: Ficheiro "resolv.conf"

1.2 Pergunta b)

Os servidores `www.uminho.pt.` e `www.ubuntu.com.` têm endereços IPv6? Se sim, quais?

Não. O servidor `www.uminho.pt.`, não tem endereços IPv6, no entanto, o servidor `www.ubuntu.com.` tem (`2001:67c:1360:8001::2c`), como se comprova nas figuras seguintes:

```
core@core-VirtualBox:~$ host www.uminho.pt
www.uminho.pt has address 193.137.9.114
```

Figure 2: Endereços do servidor `www.uminho.pt.`

```
core@core-VirtualBox:~$ host www.ubuntu.com
www.ubuntu.com has address 91.189.88.181
www.ubuntu.com has address 91.189.88.180
www.ubuntu.com has IPv6 address 2001:67c:1360:8001::2b
www.ubuntu.com has IPv6 address 2001:67c:1360:8001::2c
```

Figure 3: Endereços do servidor `www.ubuntu.com.`

1.3 Pergunta c)

Quais os servidores de nomes definidos para os domínios: “sapo.pt.”, “pt.” e “.”?

Os servidores de nome para o domínio “sapo.pt.” são “ns2.sapo.pt.”, “dns02.sapo.pt.”, “ns.sapo.pt.” e “dns01.sapo.pt.”, como podemos ver na imagem.

```
core@core-VirtualBox:~$ nslookup
> set type=NS
> sapo.pt.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
sapo.pt nameserver = ns2.sapo.pt.
sapo.pt nameserver = dns02.sapo.pt.
sapo.pt nameserver = ns.sapo.pt.
sapo.pt nameserver = dns01.sapo.pt.

Authoritative answers can be found from:
```

Figure 4: Domínio “sapo.pt.”

Os servidores de nome para o domínio “pt.” são “e.dns.pt.”, “g.dns.pt.”, “b.dns.pt.”, “a.dns.pt.”, “ns2.nic.fr.”, “ns.dns.br.”, “d.dns.pt.”, “h.dns.pt.” e “c.dns.pt.” como se pode verificar na figura em baixo.

```
> pt.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
pt      nameserver = e.dns.pt.
pt      nameserver = g.dns.pt.
pt      nameserver = b.dns.pt.
pt      nameserver = a.dns.pt.
pt      nameserver = ns2.nic.fr.
pt      nameserver = ns.dns.br.
pt      nameserver = d.dns.pt.
pt      nameserver = h.dns.pt.
pt      nameserver = c.dns.pt.
```

Figure 5: Domínio “pt.”

Por último, os servidores de nome para o domínio “.” que são, no total, 13 servidores, encontram-se na figura abaixo.

```
> .
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
.      nameserver = d.root-servers.net.
.      nameserver = g.root-servers.net.
.      nameserver = a.root-servers.net.
.      nameserver = j.root-servers.net.
.      nameserver = m.root-servers.net.
.      nameserver = h.root-servers.net.
.      nameserver = k.root-servers.net.
.      nameserver = c.root-servers.net.
.      nameserver = e.root-servers.net.
.      nameserver = l.root-servers.net.
.      nameserver = f.root-servers.net.
.      nameserver = i.root-servers.net.
.      nameserver = b.root-servers.net.
```

Figure 6: Domínio ”.”

1.4 Pergunta d)

Existe o domínio open.money.? Será que open.money. é um host ou um domínio

Tal como verificamos na imagem a seguir apresentada, o domínio "open.money." existe, uma vez que tem correspondência.

No entanto, não estamos perante um host, mas sim de um domínio que é um servidor email.

```
open.money has address 35.154.208.116
open.money mail is handled by 10 alt3.aspmx.l.google.com.
open.money mail is handled by 5 alt1.aspmx.l.google.com.
open.money mail is handled by 10 mailstore1.secureserver.net.
open.money mail is handled by 10 alt4.aspmx.l.google.com.
open.money mail is handled by 0 smtp.secureserver.net.
open.money mail is handled by 5 alt2.aspmx.l.google.com.
open.money mail is handled by 1 aspmx.l.google.com.
```

Figure 7: Comando host para o domínio "open.money."

1.5 Pergunta e)

Qual é o servidor DNS primário definido para o domínio un.org.? Este servidor primário (master) aceita queries recursivas? Porquê?

Tal como temos feito para resolver outras alíneas, realizamos mais um *nslookup* para descobrir o servidor DNS primário definido para o domínio "un.org.". Achemos então o servidor DNS primário que é o ns1.un.org (origin).

```
core@core-VirtualBox:~$ nslookup
> set type=SOA
> un.org.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
un.org
    origin = ns1.un.org
    mail addr = root.un.org
    serial = 2021041500
    refresh = 1200
    retry = 3600
    expire = 1209600
    minimum = 300
```

Figure 8: Tipo SOA (Start Of Authority) do comando *nslookup*

Para verificarmos se o servidor primário aceita queries recursivas realizamos um dig a este servidor. O resultado foi este:

```
core@core-VirtualBox:~$ dig ns1.un.org

; <>> DiG 9.16.1-Ubuntu <>> ns1.un.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22426
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ns1.un.org.                IN      A
;; ANSWER SECTION:
ns1.un.org.                199     IN      A      157.150.185.28

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: ter abr 20 09:50:41 WEST 2021
;; MSG SIZE rcvd: 55
```

Figure 9: Comando dig ao servidor DNS primário

Concluimos então que este servidor aceita queries recursivas, visto que possui a flag *ra* que se refere a *Recursion Available*, logo este servidor tem recursividade.

1.6 Pergunta f)

Obtenha uma resposta “autoritativa” para a questão anterior

Devido a seguranças na internet da Universidade do Minho, vimo-nos forçados a fazer esta alínea em casa, visto que não conseguíamos ver as *Authorative Answers*. Após realizar o comando *nslookup*:

```
> set type=SOA
> un.org.
Server:      10.255.255.3
Address:     10.255.255.3#53

Non-authoritative answer:
un.org
      origin = ns1.un.org
      mail addr = root.un.org
      serial = 2021042900
      refresh = 1200
      retry = 3600
      expire = 1209600
      minimum = 300

Authoritative answers can be found from:
> set query=ns
> un.org.
Server:      10.255.255.3
Address:     10.255.255.3#53

Non-authoritative answer:
un.org  nameserver = ns2.un.org.
un.org  nameserver = ns1.un.org.
un.org  nameserver = ns3.un.org.

Authoritative answers can be found from:
ns1.un.org  internet address = 157.150.185.28
ns2.un.org  internet address = 157.150.34.57
ns3.un.org  internet address = 157.150.241.25
```

Figure 10: *Authorative answers*

Conseguimos observar os endereços autoritativos *ns1.un.org*, *ns2.un.org* e *ns3.un.org*. Posto isto, tivemos de questionar com o comando *server ns1.un.org* para obtermos a resposta autoritativa.

1.7 Pergunta g)

Onde são entregues as mensagens de correio eletrônico dirigidas a `presidency@eu.eu` ou `presidencia@2021portugal.eu`?

As mensagens dirigidas a esses emails são entregues nos servidores que se obtêm ao inserir os seguintes comandos:

```
core@core-VirtualBox:~$ nslookup
> set type=MX
> eu.eu
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
eu.eu mail exchanger = 20 smtp02.level27.be.
eu.eu mail exchanger = 10 smtp01.level27.be.
```

Figure 11: Servidores de email para "eu.eu"

```
core@core-VirtualBox:~$ nslookup
> set type=MX
> 2021portugal.eu
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
2021portugal.eu mail exchanger = 10 mxg.eu.mpssec.net.
```

Figure 12: Servidores de email para "2021portugal.eu"

Analisando as imagens acima, as mensagens de correio eletrônico dirigidas a `presidency@eu.eu` têm 2 alternativas de entrega. Sendo que o servidor prioritário é o com o valor inferior, então este servidor será o "smtp01.level27.be.". No caso do `presidencia@2021portugal.eu` é mais fácil, já que apenas temos uma opção para o servidor que será o "mxg.eu.mpssec.net.".

1.8 Pergunta h)

Que informação é possível obter, via DNS, acerca de gov.pt?

Para realizar o comando *dig* ao gov.pt, tivemos de descobrir os servidores DNS do domínio gov.pt:

```
core@core-VirtualBox:~$ nslookup
> set type=NS
> gov.pt
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
gov.pt  nameserver = nsp.dnsnode.net.
gov.pt  nameserver = europel.dnsnode.net.
gov.pt  nameserver = dns1.gov.pt.
gov.pt  nameserver = ns02.fccn.pt.
gov.pt  nameserver = a.dns.pt.
```

Figure 13: DNS do domínio gov.pt

Ao analisar a parte da *ANSWER SECTION*, podemos verificar que o servidor tem um IP associado (194.58.198.32), como se verifica na figura abaixo.

```
core@core-VirtualBox:~$ dig nsp.dnsnode.net

; <<>> DiG 9.16.1-Ubuntu <<>> nsp.dnsnode.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56071
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;nsp.dnsnode.net.                IN      A

;; ANSWER SECTION:
nsp.dnsnode.net.                3600    IN      A      194.58.198.32

;; Query time: 1084 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: ter abr 20 10:30:16 WEST 2021
;; MSG SIZE rcvd: 60
```

Figure 14: Comando *dig* ao servidor DNS do gov.pt

1.9 Pergunta i)

1.10 Pergunta j)

Os secundários usam um mecanismo designado por “Transferência de zona” para se atualizarem automaticamente a partir do primário, usando os parâmetros definidos no Record do tipo SOA do domínio. Descreve sucintamente esse mecanismo com base num exemplo concreto (ex: di.uminho.pt ou o domínio cc.pt que vai ser criado na topologia virtual).

A transferência de zona DNS é um tipo de transação de DNS e é um dos muitos mecanismos disponíveis para os administradores replicarem bancos de dados DNS em um conjunto de servidores DNS. Uma transferência de zona usa o TCP para transporte e assume a forma de uma transação cliente-servidor.

O cliente solicitando uma transferência de zona pode ser um servidor slave ou servidor secundário, solicitando dados de um servidor mestre, às vezes chamado de servidor principal. A parte do banco de dados que é replicada é uma zona.

Na 2.^a parte deste trabalho, criamos um domínio cc.pt na topologia virtual e que servirá de exemplo para mostrar parâmetros definidos no Record do tipo SOA (*Start Of Authority*):

```
$TTL      604800
@         IN      SOA      ns.cc.pt. PL05G14.cc.pt. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
                                )
```

Figure 17: Parâmetros do Record do tipo SOA do domínio cc.pt

- **Serial:** É o número de série para esta zona. Se o servidor secundário associado a este número verificar uma diminuição, o servidor irá assumir que a zona foi alterada e iniciará uma transferência de zona.
- **Refresh:** É a quantidade de tempo em segundos que os servidores secundários devem esperar antes de perguntarem aos servidores primários pelo Record do SOA para verem se este foi atualizado.
- **Retry:** É o tempo que o servidor secundário deve esperar para solicitar uma atualização novamente a um servidor de nomes primário que não responde.
- **Expire:** Se um servidor secundário não tiver uma resposta do primário durante este tempo, ele deve parar de responder às queries desta zona.
- **Negative Cache TTL:** É semelhante a um TTL básico, no entanto define especialmetne o valor para respostas TTL negativas. São recomendados periodos de tempo pequenos (15 minutos a 2 horas).

Analisando todas estas informações e consultando o nosso exemplo, podemos verificar que o número de série é 2; sabemos também que o servidor secundário irá entrar em contacto com o primário para ver se este foi atualizado passados 7 dias (604800 segundos); o servidor secundário irá esperar 1 dia (86400 segundos) até solicitar a atualização novamente; este parará de responder após uma inatividade do servidor primário de 28 dias (2419200 segundos); e, por fim, o TTL negativo, neste caso, não será o recomendado, uma vez que é de 7 dias, igual ao *Refresh*.

2 Parte II: Instalação, configuração e teste de um domínio CC.PT

2.1 Configuração do domínio

Na segunda parte do trabalho tivemos como objetivo a criação do domínio *CC.PT*. Para isto, baseá-mo-nos maioritariamente em seguir os passos que nos foram apresentados no enunciado, mas recorrendo também à consulta dos manuais do software *BIND9*, bem como ao conhecimento adquirido nas aulas teóricas.

Primeiramente configurámos o servidor primário. Todos os ficheiros que iremos editar excetuando o primeiro estão localizados na pasta `/home/core/primario`.

Começamos por editar o ficheiro `/etc/hosts` de forma a que este incluisse os registos **10.1.1.1 Server1 ns.cc.pt** e **10.2.2.2 Mercurio ns2.cc.pt** dos servidores primário e secundário, respetivamente.

```
127.0.0.1    localhost
127.0.1.1    core-VirtualBox
10.1.1.1     Server1      ns.cc.pt.
10.2.2.2     Mercurio     ns2.cc.pt

# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Figure 18: Ficheiro `/etc/hosts`

Nos passo 2 editámos o ficheiro `named.conf.options` de modo a termos os servidores do DI (193.136.9.240 e 193.136.19.1) como *forwarders*.

```
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // 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 {
        193.136.9.240;
        193.136.19.1;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; };
};
```

Figure 19: Ficheiro `named.conf.options`

No passo 3 adicionamos ao ficheiro `named.conf` a zona `cc.pt` relativo ao fichero `db.cc.pt`, e as zonas inversas para as redes `10.1.1.*`, `10.2.2.*` e `10.4.4.*`, cada um com o seu ficheiro `bd` relativo. As zonas são do tipo **master**, pois este é o dns primário, e têm a clausula **allow-transfer** com o ip do servidor dns secundario para que este possa transferir os ficheiro `db` e os usar.

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/home/core/primario/named.conf.options";
include "/home/core/primario/named.conf.local";
include "/home/core/primario/named.conf.default-zones";

zone "cc.pt" {
type master;
file "/home/core/primario/db.cc.pt";
allow-transfer { 10.2.2.2; };
};

zone "1.1.10.in-addr.arpa" {
type master;
file "/home/core/primario/db.1-1-10.rev";
allow-transfer { 10.2.2.2; };
};

zone "2.2.10.in-addr.arpa" {
type master;
file "/home/core/primario/db.2-2-10.rev";
allow-transfer { 10.2.2.2; };
};

zone "4.4.10.in-addr.arpa" {
type master;
file "/home/core/primario/db.4-4-10.rev";
allow-transfer { 10.2.2.2; };
};
```

Figure 20: Ficheiro **named.conf**

De seguida criamos o ficheiro **db.cc.pt** à imagem do ficheiro **db.local**, onde adicionamos endereços dos **Name Servers**, dos **Mail Exchange Server** e todos os **A** e **CNAME** records pedidos para registar no enunciado.

```
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns.cc.pt. PL05G14.cc.pt. (
                        3      ; Serial
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        604800  ; Negative Cache TTL
);
; name-servers -> NS records
@         IN      NS       ns.cc.pt.
@         IN      NS       ns2.cc.pt.
; mail servers -> MX
@         IN      MX       10      mail.cc.pt.    ;mail principal
@         IN      MX       20      Server3.cc.pt. ;mail secundario

; name servers - A/CNAME records
;SV1-DNS Primario
Server1 IN      A       10.1.1.1
ns      IN      A       10.1.1.1

;SV2
Server2 IN      A       10.1.1.2
www     IN      A       10.1.1.2
mail    IN      A       10.1.1.2

;SV3
Server3 IN      A       10.1.1.3
pop     IN      CNAME    Server3
imap    IN      CNAME    Server3

;Mercurio-DNS Secundario
ns2     IN      A       10.2.2.2
Mercurio IN      A       10.2.2.2

;Marte e Venus
Marte   IN      A       10.2.2.1
Venus   IN      A       10.2.2.3

;Laptop
Laptop1 IN      A       10.4.4.1
q14     IN      CNAME    Laptop1
```

Figure 21: Ficheiro **db.cc.pt**

No passo 5, baseando-mo-nos no ficheiro **db.127** criamos os ficheiros **db.1-1-10.rev**, **db.2-2-10.rev** e **db.4-4-10.rev** para a pesquisa inversa no domínio.

```

; BIND reverse data file for local loopback interface
;
$TTL 604800
@ IN SOA ns.cc.pt. PL05G14.cc.pt. (
    2 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL

; NS
@ IN NS ns.cc.pt.
@ IN NS ns2.cc.pt.

; PTR Records
;SV1
1 IN PTR ns.cc.pt.
1 IN PTR Server1.cc.pt

;SV2
2 IN PTR Server2.cc.pt.
2 IN PTR www.cc.pt.
2 IN PTR mail.cc.pt.

;SV3
3 IN PTR Server3.cc.pt.
3 IN PTR pop.cc.pt.
3 IN PTR imap.cc.pt.

```

Figure 22: Ficheiro db.1.1.10.rev

```

; BIND reverse data file for local loopback interface
;
$TTL 604800
@ IN SOA ns.cc.pt. PL05G14.cc.pt. (
    2 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL

; NS
@ IN NS ns.cc.pt.
@ IN NS ns2.cc.pt.

; PTR Records
;Mercurio
2 IN PTR ns2.cc.pt.
2 IN PTR Mercurio.cc.pt

;Marte
1 IN PTR Marte.cc.pt.

;SV3
3 IN PTR Venus.cc.pt.

```

Figure 23: Ficheiro db.2.2.10.rev

```

; BIND reverse data file for local loopback interface
;
$TTL 604800
@ IN SOA ns.cc.pt. PL05G14.cc.pt. (
    2 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL

; NS
@ IN NS ns.cc.pt.
@ IN NS ns2.cc.pt.

; PTR Records
;Laptop1
1 IN PTR Laptop1.cc.pt.
1 IN PTR g14.cc.pt

```

Figure 24: Ficheiro db.4.4.10.rev

Para finalizar o servidor primário, verificamos as configurações com os comandos fornecidos. Em seguida configuramos o servidor de dns secundário. Este necessita de menos ficheiros pois irá transferi-los do servidor primário.

No primeiro passo, e tal como fizemos para o servidor primário, alteramos o ficheiro **named.conf.options**, adicionando os mesmos servidores **forwarders** que no ficheiro com o mesmo nome nos ficheiros do servidor primário.

```
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // 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 {
        193.136.9.240;
        193.136.19.1;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; };
};
```

Figure 25: Ficheiro **named.conf.options** do servidor secundario

De seguida alteramos o ficheiro **named.conf** de maneira a incluir as zonas já especificadas na criação dos ficheiros do servidor primário. Desta vez as zonas são do tipo **slave** pois este é o servidor secundario. É especificado também ip do servidor primario na clausula **masters**. Os ficheiros de zona serão guardados em **/var/cache/bind/[nome da zona]**.

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/home/core/secundario/named.conf.options";
include "/home/core/secundario/named.conf.local";
include "/home/core/secundario/named.conf.default-zones";

zone "cc.pt"{
    type slave;
    file "/var/cache/bind/db.cc.pt";
    masters {10.1.1.1; };
};

zone "1.1.10.in-addr.arpa"{
    type slave;
    file "/var/cache/bind/db.1-1-10.rev";
    masters {10.1.1.1; };
};

zone "2.2.10.in-addr.arpa"{
    type slave;
    file "/var/cache/bind/db.2-2-10.rev";
    masters {10.1.1.1; };
};

zone "4.4.10.in-addr.arpa"{
    type slave;
    file "/var/cache/bind/db.4-4-10.rev";
    masters {10.1.1.1; };
};
```

Figure 26: Ficheiro **named.conf** do servidor secundario

Por fim,verificamos as configurações do servidor secundario com os comandos fornecidos.

2.2 Demonstração do funcionamento da topologia

Antes da execução dos testes foi iniciado o servidor primário no nodo **Server1** com o comando **sudo /usr/sbin/named-c/home/core/primario/named.conf -g**, e o servidor secundario no nodo **Mercurio** com o comando **sudo /usr/sbin/named -c /home/core/secundario/named.conf -g**.

Os testes foram todos realizados no Laptop1 por nenhuma razão em específico, podendo-se escolher qualquer nó na topologia. Os mesmos foram realizados com a ferramenta **nslookup** pois foi a nossa ferramenta favorita, achando que nos dava um maior controlo sobre as queries feitas. Estes foram ainda realizados pela ordem de inserção nos ficheiros dos servidores.

Começando pelo servidor primário, perguntamos quais eram os registos do tipo **SOA** (start of authority), sendo a resposta a seguinte:


```

root@Laptop1:/tmp/pycore.32827/Laptop1.conf# nslookup - 10.1.1.1
> set type=SOA
> cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

cc.pt
    origin = ns.cc.pt
    mail addr = PL05G14.cc.pt
    serial = 3
    refresh = 604800
    retry = 86400
    expire = 2419200
    minimum = 604800
>

```

Figure 27: Output das queries com o tipo **SOA**

Em seguida, realizamos queries com os tipos **NS** (DNS name server) e **MX** (mail exchanger server), obtendo as seguintes respostas:

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.1.1.1
> set type=NS
> cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

cc.pt  nameserver = ns.cc.pt.
cc.pt  nameserver = ns2.cc.pt.
>
> set type=MX
> cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

cc.pt  mail exchanger = 10 mail.cc.pt.
cc.pt  mail exchanger = 20 Server3.cc.pt.

```

Figure 28: Output das queries com o tipo **NS** e **MX**

Em terceiro lugar, perguntamos pelos registros do tipo **A** (ipv4) para todos os nomes que tínhamos, recebendo também informação se o nome procurado tem um **canonical name** associado (acontece quando o registro é do tipo **CNAME**).

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.1.1.1
> set type=A
> Server1.cc.pt.
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  Server1.cc.pt
Address: 10.1.1.1
> ns.cc.pt.
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  ns.cc.pt
Address: 10.1.1.1

```

Figure 29: Queries com o tipo **A** para os nomes do Servidor 1

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.1.1.1
> set type=A
> Server2.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  Server2.cc.pt
Address: 10.1.1.2
> www.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  www.cc.pt
Address: 10.1.1.2
> mail.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  mail.cc.pt
Address: 10.1.1.2

```

Figure 30: Queries com o tipo A para os nomes do Servidor 2

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.1.1.1
> set type=A
> Server3.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  Server3.cc.pt
Address: 10.1.1.3
> pop.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

pop.cc.pt    canonical name = Server3.cc.pt.
Name:  Server3.cc.pt
Address: 10.1.1.3
> imap.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

imap.cc.pt   canonical name = Server3.cc.pt.
Name:  Server3.cc.pt
Address: 10.1.1.3

```

Figure 31: Queries com o tipo A para os nomes do Servidor 3

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.1.1.1
> set type=A
> ns2.cc.pt.
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  ns2.cc.pt
Address: 10.2.2.2
> Mercurio.cc.pt.
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  Mercurio.cc.pt
Address: 10.2.2.2
>
> Marte.cc.pt.
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  Marte.cc.pt
Address: 10.2.2.1
>
> Venus.cc.pt.
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  Venus.cc.pt
Address: 10.2.2.3

```

Figure 32: Queries com o tipo A para os nomes da rede 10.2.2.*

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.1.1.1
> set type=A
> Laptop1.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:  Laptop1.cc.pt
Address: 10.4.4.1
> g14.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

g14.cc.pt      canonical name = Laptop1.cc.pt.
Name:  Laptop1.cc.pt
Address: 10.4.4.1

```

Figure 33: Queries com o tipo **A** para o Laptop 1

Para finalizar as queries ao servidor primário, perguntamos pelo inverso dos registros, ou seja, passando o ip de uma maquina descobrir o/os nomes das máquinas.

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.1.1.1
> set type=PTR
> 10.1.1.1
Server:      10.1.1.1
Address:     10.1.1.1#53

1.1.1.10.in-addr.arpa  name = ns.cc.pt.
1.1.1.10.in-addr.arpa  name = Server1.cc.pt,1.1.10.in-addr.arpa.
> 10.1.1.2
Server:      10.1.1.1
Address:     10.1.1.1#53

2.1.1.10.in-addr.arpa  name = www.cc.pt.
2.1.1.10.in-addr.arpa  name = Server2.cc.pt.
2.1.1.10.in-addr.arpa  name = mail.cc.pt.
> 10.1.1.3
Server:      10.1.1.1
Address:     10.1.1.1#53

3.1.1.10.in-addr.arpa  name = imap.cc.pt.
3.1.1.10.in-addr.arpa  name = pop.cc.pt.
3.1.1.10.in-addr.arpa  name = Server3.cc.pt.
>
> 10.2.2.1
Server:      10.1.1.1
Address:     10.1.1.1#53

1.2.2.10.in-addr.arpa  name = Marte.cc.pt.
> 10.2.2.2
Server:      10.1.1.1
Address:     10.1.1.1#53

2.2.2.10.in-addr.arpa  name = Mercurio.cc.pt,2.2.10.in-addr.arpa.
2.2.2.10.in-addr.arpa  name = ns2.cc.pt.
> 10.2.2.3
Server:      10.1.1.1
Address:     10.1.1.1#53

3.2.2.10.in-addr.arpa  name = Venus.cc.pt.
>
> 10.4.4.1
Server:      10.1.1.1
Address:     10.1.1.1#53

1.4.4.10.in-addr.arpa  name = g14.cc.pt,4.4.10.in-addr.arpa.
1.4.4.10.in-addr.arpa  name = Laptop1.cc.pt.

```

Figure 34: Queries com o tipo **PTR**

Realizamos os testes para o servidor secundario pela mesma ordem, sendo mais fácil a comparação. Em primeiro lugar, as queries com o tipo **SOA**:

```

root@Laptop1:/tmp/pycore.32827/Laptop1.conf# nslookup - 10.2.2.2
> set type=SOA
> cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

cc.pt
    origin = ns.cc.pt
    mail addr = PL05G14.cc.pt
    serial = 3
    refresh = 604800
    retry = 86400
    expire = 2419200
    minimum = 604800

```

Figure 35: Output das queries com o tipo **SOA** usando o servidor secundario

Em seguida, as queries com o tipo **NS** e **MX**:

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.2.2.2
> set type=NS
> cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

cc.pt  nameserver = ns.cc.pt.
cc.pt  nameserver = ns2.cc.pt.
>
> set type=MX
> cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

cc.pt  mail exchanger = 10 mail.cc.pt.
cc.pt  mail exchanger = 20 Server3.cc.pt.

```

Figure 36: Queries com o tipo **NS** e **MX** usando o servidor secundario

Em terceiro lugar, as queries com o tipo **A**:

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.2.2.2
> set type=A
> Server1.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:  Server1.cc.pt
Address: 10.1.1.1
> ns.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:  ns.cc.pt
Address: 10.1.1.1

```

Figure 37: Queries com o tipo **A** para os nomes do Servidor 1 usando o servidor de dns secundario

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.2.2.2
> set type=A
> Server2.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:  Server2.cc.pt
Address: 10.1.1.2
> www.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:  www.cc.pt
Address: 10.1.1.2
> mail.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:  mail.cc.pt
Address: 10.1.1.2

```

Figure 38: Queries com o tipo **A** para os nomes do Servidor 2 usando o servidor de dns secundario

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.2.2.2
> set type=A
> Server3.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:   Server3.cc.pt
Address: 10.1.1.3
> pop.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

pop.cc.pt      canonical name = Server3.cc.pt.
Name:   Server3.cc.pt
Address: 10.1.1.3
> imap.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

imap.cc.pt      canonical name = Server3.cc.pt.
Name:   Server3.cc.pt
Address: 10.1.1.3

```

Figure 39: Queries com o tipo **A** para os nomes do Servidor 3 usando o servidor de dns secundario

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.2.2.2
> set type=A
> ns2.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:   ns2.cc.pt
Address: 10.2.2.2
> Mercurio.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:   Mercurio.cc.pt
Address: 10.2.2.2
>
> Marte.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:   Marte.cc.pt
Address: 10.2.2.1
>
> Venus.cc.pt.
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:   Venus.cc.pt
Address: 10.2.2.3

```

Figure 40: Queries com o tipo **A** para os nomes da rede 10.2.2.* usando o servidor de dns secundario

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.2.2.2
> set type=A
> Laptop1.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

Name:   Laptop1.cc.pt
Address: 10.4.4.1
> g14.cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

g14.cc.pt      canonical name = Laptop1.cc.pt.
Name:   Laptop1.cc.pt
Address: 10.4.4.1

```

Figure 41: Queries com o tipo **A** para o Laptop 1 usando o servidor de dns secundario

Por fim, as queries com o tipo **PTR**:

```

root@Laptop1:/tmp/pycore.38053/Laptop1.conf# nslookup - 10.2.2.2
> set type=PTR
> 10.1.1.1
Server:      10.2.2.2
Address:     10.2.2.2#53

1.1.1.10.in-addr.arpa  name = ns.cc.pt.
1.1.1.10.in-addr.arpa  name = Server1.cc.pt.1.1.10.in-addr.arpa.
> 10.1.1.2
Server:      10.2.2.2
Address:     10.2.2.2#53

2.1.1.10.in-addr.arpa  name = mail.cc.pt.
2.1.1.10.in-addr.arpa  name = Server2.cc.pt.
2.1.1.10.in-addr.arpa  name = www.cc.pt.
> 10.1.1.3
Server:      10.2.2.2
Address:     10.2.2.2#53

3.1.1.10.in-addr.arpa  name = imap.cc.pt.
3.1.1.10.in-addr.arpa  name = pop.cc.pt.
3.1.1.10.in-addr.arpa  name = Server3.cc.pt.
>
> 10.2.2.1
Server:      10.2.2.2
Address:     10.2.2.2#53

1.2.2.10.in-addr.arpa  name = Marte.cc.pt.
> 10.2.2.2
Server:      10.2.2.2
Address:     10.2.2.2#53

2.2.2.10.in-addr.arpa  name = Mercurio.cc.pt.2.2.10.in-addr.arpa.
2.2.2.10.in-addr.arpa  name = ns2.cc.pt.
> 10.2.2.3
Server:      10.2.2.2
Address:     10.2.2.2#53

3.2.2.10.in-addr.arpa  name = Venus.cc.pt.
>
> 10.4.4.1
Server:      10.2.2.2
Address:     10.2.2.2#53

1.4.4.10.in-addr.arpa  name = g14.cc.pt.4.4.10.in-addr.arpa.
1.4.4.10.in-addr.arpa  name = Laptop1.cc.pt.

```

Figure 42: Queries com o tipo **PTR** usando o servidor de dns secundario

3 Conclusão

O resultado final deste trabalho é positivo, tendo os objetivos propostos sido cumpridos. Sentimos que conseguimos pôr em prática os conteúdos adquiridos nas aulas teóricas e, ultrapassando as dificuldades que surgiram, adquirir novos conhecimentos.

Para além do que fomos aprendendo nas teóricas, como por exemplo os melhores comandos a usar ao longo do trabalho, sentimos uma necessidade na busca de informação da internet que nos conseguiu guiar em momentos que estivessemos mais perdidos.

Concluindo, o grupo sente que este trabalho foi ótimo para entender e perceber os clientes DNS mais conhecidos incluídos nos sistemas operativos.