

## Comunicações por computador Grupo 5 - PL1

7 de maio de 2021

### TP3: Serviço de Resolução de Nomes (DNS)



Luís Martins (A89600)



Renata Teixeira (A89611)



Tiago Alves (A89554)

# 1 Questões e respostas

## 1.1 Parte I: Consultas ao serviço de nomes DNS

Com base no manual das aplicações (ex: *man nslookup* ou *man dig*) e no material de suporte procure responder às seguintes questões:

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

Uma vez que utilizamos uma máquina virtual, o ficheiro */etc/resolv.conf*, inicialmente, continha apenas um nameserver (nameserver 127.0.0.53), sendo que, para a realização deste TP, de acordo com o professor, adicionamos o nameserver 193.137.16.65.

Os nameservers são endereços de internet que o *resolver* deve questionar.

O *options* permite que certas variáveis internas do *resolver* possam ser modificadas, sendo que a opção *edns0* permite suporte para extensões DNS.

O *search*, é uma lista de pesquisa para consulta de um host-name, cuja entrada é o nome local do domínio ("eduroam.uminho.pt").

```
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 193.137.16.65
nameserver 127.0.0.53
options edns0 trust-ad
search eduroam.uminho.pt
core@core-VirtualBox:~$
```

Figura 1: Conteúdo do ficheiro */etc/resolv.conf*

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

Executando o comando *host* concluímos que não, apenas o servidor *www.ubuntu.com* tem endereços IPV6. Como demonstrado na seguinte figura:

```
core@core-VirtualBox:~$ host www.uminho.pt.  
www.uminho.pt has address 193.137.9.114  
core@core-VirtualBox:~$ host www.ubuntu.com  
www.ubuntu.com has address 91.189.88.180  
www.ubuntu.com has address 91.189.88.181  
www.ubuntu.com has IPv6 address 2001:67c:1360:8001::2c  
www.ubuntu.com has IPv6 address 2001:67c:1360:8001::2b  
core@core-VirtualBox:~$
```

Figura 2: Endereços de ambos os servidores

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

Para o domínio “*sapo.pt.*” estão definidos os servidores:

- ns.sapo.pt, cujo endereço IP é 212.55.154.202.
- ns2.sapo.pt, cujo endereço IP é 212.55.154.194.
- dns01.sapo.pt, cujo endereço IP é 213.13.28.116.
- dns02.sapo.pt, cujo endereço IP é 213.13.30.116.

```
core@core-VirtualBox:~$ nslookup  
> set type=NS  
> sapo.pt.  
Server:      193.137.16.65  
Address:     193.137.16.65#53  
  
Non-authoritative answer:  
sapo.pt nameserver = ns.sapo.pt.  
sapo.pt nameserver = dns02.sapo.pt.  
sapo.pt nameserver = dns01.sapo.pt.  
sapo.pt nameserver = ns2.sapo.pt.  
  
Authoritative answers can be found from:  
ns.sapo.pt      internet address = 212.55.154.202  
ns2.sapo.pt     internet address = 212.55.154.194  
dns01.sapo.pt   internet address = 213.13.28.116  
dns02.sapo.pt   internet address = 213.13.30.116  
dns01.sapo.pt   has AAAA address 2001:8a0:2106:4:213:13:28:116  
dns02.sapo.pt   has AAAA address 2001:8a0:2206:4:213:13:30:116
```

Figura 3: Servidores do domínio “*sapo.pt.*”

Por outro lado, para o domínio "pt.", estão definidos 9 servidores, como podemos observar na secção "Non-authoritative answer", da seguinte imagem.

```
> pt.  
Server:      193.137.16.65  
Address:     193.137.16.65#53  
  
Non-authoritative answer:  
pt      nameserver = g.dns.pt.  
pt      nameserver = a.dns.pt.  
pt      nameserver = h.dns.pt.  
pt      nameserver = d.dns.pt.  
pt      nameserver = ns2.nic.fr.  
pt      nameserver = c.dns.pt.  
pt      nameserver = b.dns.pt.  
pt      nameserver = ns.dns.br.  
pt      nameserver = e.dns.pt.
```

Figura 4: Servidores do domínio "pt."

Por último, para o domínio ".", estão definidos 13 servidores, seguindo o mesmo raciocínio do domínio anterior, os servidores deste estão representados na figura seguinte.

```
> .  
Server:      193.137.16.65  
Address:     193.137.16.65#53  
  
Non-authoritative answer:  
.      nameserver = c.root-servers.net.  
.      nameserver = d.root-servers.net.  
.      nameserver = f.root-servers.net.  
.      nameserver = g.root-servers.net.  
.      nameserver = k.root-servers.net.  
.      nameserver = a.root-servers.net.  
.      nameserver = l.root-servers.net.  
.      nameserver = b.root-servers.net.  
.      nameserver = m.root-servers.net.  
.      nameserver = e.root-servers.net.  
.      nameserver = h.root-servers.net.  
.      nameserver = i.root-servers.net.  
.      nameserver = j.root-servers.net.
```

Figura 5: Servidores do domínio "."

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

Podemos confirmar que o domínio "open.money" existe, visto que tem correspondência.

Para além disso, podemos também afirmar que este domínio não é um *host*, todavia é um servidor de e-mail, podendo esta afirmação ser comprovada pela execução do comando "host open.money".

```
core@core-VirtualBox:~$ host open.money.  
open.money has address 35.154.208.116  
open.money mail is handled by 10 alt4.aspmx.l.google.com.  
open.money mail is handled by 1 aspmx.l.google.com.  
open.money mail is handled by 10 mailstore1.secureserver.net.  
open.money mail is handled by 10 alt3.aspmx.l.google.com.  
open.money mail is handled by 5 alt2.aspmx.l.google.com.  
open.money mail is handled by 5 alt1.aspmx.l.google.com.  
open.money mail is handled by 0 smtp.secureserver.net.  
core@core-VirtualBox:~$
```

Figura 6: Execução do comando "host open.money."

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

O servidor "ns1.un.org" é o servidor DNS primário definido para o domínio un.org. Uma vez mais, observando a secção "Non-authoritative answer" e a sua sub-secção "origin", podemos confirmar a resposta dada.

```
core@core-VirtualBox:~$ nslookup  
> set type=SOA  
> un.org.  
Server:          193.137.16.65  
Address:         193.137.16.65#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
```

Figura 7: Execução do comando "nslookup" do tipo SOA para o domínio un.org.

Uma vez que ao executarmos o comando "dig ns1.un.org." obtemos o resultado da seguinte imagem, podemos confirmar que este servidor primário aceita queries recursivas.

```
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: 5822
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2e90e007b4ff7e0a31bfcd6d609154e6b4845d43fe549391 (good)
;; QUESTION SECTION:
;ns1.un.org.                IN      A

;; ANSWER SECTION:
ns1.un.org.                300     IN      A      157.150.185.28
```

F Obtenha uma resposta “*autoritativa*” para a questão anterior.

De forma a obter o pedido, isto é, uma resposta “autoritativa” para a questão anterior, executamos o comando *nslookup* do tipo SOA, obtendo o seguinte:

```
core@core-VirtualBox:~$ nslookup
> set type=SOA
> un.org.
Server:          193.137.16.65
Address:         193.137.16.65#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:
un.org  nameserver = ns1.un.org.
un.org  nameserver = ns2.un.org.
un.org  nameserver = ns3.un.org.
ns3.un.org    internet address = 157.150.241.25
ns1.un.org    internet address = 157.150.185.28
ns2.un.org    internet address = 157.150.34.57
>
```

Figura 8: Execução do comando nslookup do tipo SOA para o domínio un.org.

Com esta execução obtivemos endereços autoritativos: *ns1.un.org.*, *ns2.un.org.* e *ns3.un.org.*. Após isso, questionamos estes endereços executando o comando "server ns1.un.org", de forma a selecionar esse servidor, e escrevendo novamente "un.org.", obtendo a seguinte resposta:

```
> server ns1.un.org
Default server: ns1.un.org
Address: 157.150.185.28#53
> un.org.
Server:      ns1.un.org
Address:     157.150.185.28#53

un.org
  origin = ns1.un.org
  mail addr = root.un.org
  serial = 2021042900
  refresh = 1200
  retry = 3600
  expire = 1209600
  minimum = 300
>
```

Figura 9: Questionar o endereço autoritário ns1.un.org

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

Para obter os servidores que gerem os e-mails enviados executamos os seguintes comandos:

```
> set type=MX
> 2021portugal.eu
Server:      193.137.16.65
Address:     193.137.16.65#53

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

Figura 10: Servidores de mail para a presidência.

```

> eu.eu
Server:      193.137.16.65
Address:     193.137.16.65#53

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

```

Figura 11: Servidores de mail para presidency.

A partir desta informação, concluímos que as mensagens para `presidencia@2021portugal.eu` serão entregues em `mxg.eu.mpssec.net`.

No caso de `presidency@eu.eu`, esta tem duas opções. Como `smtp01.level27.be` tem o valor 10, significa que este servidor é o prioritário, recebendo a mensagem. Caso este não esteja disponível, a mensagem será enviada para `smtp02.level27.be`.

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

Executando o comando "dig gov.pt ANY" obtivemos o seguinte resultado:

```

core@core-VirtualBox:~$ dig gov.pt ANY

;<<>> DiG 9.16.1-Ubuntu <<>> gov.pt ANY
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 32786
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: udp: 65494
;; QUESTION SECTION:
;gov.pt.                                IN      ANY

;; ANSWER SECTION:
gov.pt. 599 IN SOA  dnssec.gov.pt. dns.ceger.gov.pt. 2019071842 18000 7200 2419200 86400
gov.pt. 599 IN RRSIG SOA 10 2 600 20210520143218 20210506143218 57803 gov.pt. Zf0F8b5z8X5JwQJad6XTUAA50HiCV2aW0r00Z/HmAAG0mmZfFb5pGvxC. K4m/6U9yvYpvl5Cpm7yfeD1I7Bd3s/
eTDr by0anTBHgZK+bPnXyzMXDS2TdrzNObmmb+8cUVrfz68wHTEono9sldi E7dwmZIC/lxagiYAR9kgLzXtbshhEgfsAx3iickUok1y6sW+qIIVBye //MdPY5jYSD3ihvI33etAZt+KbSSGMQJh2CallndHy8WvrVuKH0+90 faNg10==
gov.pt. 599 IN RRSIG SOA 10 2 600 20210520143218 20210506143218 60971 gov.pt. cuz4H020rweUSMjQsmFTcgs3FlvQZ5VjCdLPZ1ze/48nuhN8W2ZYmx7t IwkppCeh0rXv/QRUevd0hFww4W38Fm
YXmv BTLXaSLGwb11bvORufyWP70RthQfrItCm6T20T3kF0ea/Gm3W0pa1C5J U/yNSCzLk2JVPX7mdJsfojNMY8Dn7Jwaj8Za0oGI0HDX+hE242qLP/a4 6xmVs1hjmttTYcNzfWErqLLxGoAy15HTJUq1Tf7GivFaHY2Rfn14FMT fXfobg==

;; Query time: 135 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: sex mai 07 20:11:05 WEST 2021
;; MSG SIZE rcvd: 676

core@core-VirtualBox:~$

```

Figura 12: Execução do comando "dig gov.pt ANY"

## I Consegue interrogar o DNS sobre o endereço IPv6 `2001:690:2080:8005::38` usando algum dos clientes DNS? Que informação consegue obter? Supondo que teve problemas com esse endereço, consegue obter um contacto do responsável por esse IPv6?

Usando o comando `nslookup` do tipo PTR de forma a fazer *reverse mapping*, obtemos o domínio associado ao endereço `2001:690:2080:8005::38`.



```

core@core-VirtualBox:~$ nslookup
> set type=PTR
> 2001:690:2080:8005::38
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
8.3.0.0.0.0.0.0.0.0.0.0.0.0.5.0.0.8.0.8.0.2.0.9.6.0.1.0.0.2.ip6.arpa      name = smtp01.fccn.pt.

Authoritative answers can be found from:
0.9.6.0.1.0.0.2.ip6.arpa      nameserver = ns02.fccn.pt.
0.9.6.0.1.0.0.2.ip6.arpa      nameserver = ns03.fccn.pt.
0.9.6.0.1.0.0.2.ip6.arpa      nameserver = ns01.fccn.pt.
ns03.fccn.pt      internet address = 138.246.255.249
ns02.fccn.pt      internet address = 193.136.2.228
ns01.fccn.pt      internet address = 193.136.192.40
ns03.fccn.pt      has AAAA address 2001:4ca0:106:0:250:56ff:fea9:3fd
ns02.fccn.pt      has AAAA address 2001:690:a80:4001::200
ns01.fccn.pt      has AAAA address 2001:690:a00:4001::200
>

```

Figura 13: Execução do comando "nslookup" do tipo PTR

Sendo, então, **smtp01.fccn.pt.** o domínio que equivale a este endereço IP.

Realizamos, ainda, outra query do tipo SOA, para identificar os seus contactos responsáveis. Obtendo a seguinte figura:

```

> smtp01.fccn.pt.
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
*** Can't find smtp01.fccn.pt.: No answer

Authoritative answers can be found from:
fccn.pt
    origin = ns01.fccn.pt
    mail addr = hostmaster.fccn.pt
    serial = 2021042201
    refresh = 21600
    retry = 7200
    expire = 1209600
    minimum = 300
>

```

Figura 14: Servidor responsável

Concluindo que o servidor responsável é *ns01.fccn.pt.*

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).

O mecanismo denominado por “Transferência de zona” é um tipo de transação disponível de DNS.

A transferência de zona é um tipo de transação entre um cliente e um servidor, usando TCP, e é normalmente utilizada por admins para replicar as suas bases de dados.

Desta forma, é necessário entender todos os seus campos, uma vez que os parâmetros definidos no Record do tipo SOA do domínio se tornam fundamentais.

- **Serial** - Representa o número de série da zona. Caso o servidor secundário que lhe está associado note um decréscimo deste valor, este entende que a zona se encontra desatualizada, e consequentemente executa uma transferência de zona.
- **Refresh** - Após a deteção de alterações da zona, o servidor secundário deve esperar estes segundos até contactar o primário e atualizar
- **Retry** - Caso uma tentativa tenha falhado, este é o número de segundos a esperar para uma segunda tentativa de contacto com o primário. Uma vez que o parâmetro refresh também representa o tempo de espera, este tem de ser inferior.
- **Expire** - Caso não obtenha resposta, este é o número de segundos após o qual o servidor secundário deve parar de solicitar o primário. Desta forma, este valor tem de ser maior que os dois anteriores.

Posto isto, temos como exemplo os parâmetros do domínio cc.pt. que foi criado na topologia virtual:

```
$TTL      604800
@         IN      SOA      ns.cc.pt. PL01G05.cc.pt. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
```

Figura 15

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

### 2.1 Objetivos

Pretende-se que crie um domínio CC.PT para a topologia de rede que estamos a usar nas aulas práticas (CC-Topo2021.imn), de modo a que se possam usar os nomes em vez dos endereços IP. No final deve, por exemplo, poder fazer-se “ping Marte.cc.pt” ou mesmo apenas “ping Marte” ou “ping Marte.cc.pt.” em vez de “ping 10.2.2.1.

### 2.2 Requisitos

- Criação do domínio cc.pt em que o Server1 (10.1.1.1) seja o primário e que o servidor Mercúrio (1.2.2.2) seja o secundário.
- Criação pelo menos do domínio reverso 1.1.10.in-addr.arpa com os mesmos servidores.
- O servidor primário do domínio é o “Server1” com endereço 10.1.1.1, também designado por ns.cc.pt, tendo como servidor secundário o “Mercurio” de endereço 10.2.2.2 e alias ns2.cc.pt. O administrador do domínio é o PLYYGXX@cc.pt (onde YY é o número do turno PL e XX é o número do grupo).
- O domínio tem também um servidor Web (www.cc.pt) e um servidor de e-mail principal (mail.cc.pt) no Servidor2. O servidor pop e imap é o Servidor3, que é também servidor secundário do e-mail para o domínio.
- Sem prejuízo de outros registos que se possam considerar, devem estar registados também o Laptop1.cc.pt com alias gXX.cc.pt, onde XX é o número do grupo, e Marte.cc.pt, Mercurio.cc.pt e Venus.cc.pt no domínio de nomes e no domínio reverso.

### 2.3 Preparativos para o ambiente CORE

#### Passo 1) Replicar ficheiros de configuração

Para não criar conflitos, visto que todos os nós da topologia CORE usam o mesmo filesystem, o primeiro passo é copiar os ficheiros de configuração para uma nova pasta. Segure-se que se use, por exemplo, **\$HOME/primario**, para o primário e **\$HOME/secundario** para o servidor secundário, onde **\$HOME** é a diretoria *default* do utilizador (no nosso caso será /home/core/):

```

core@core-VirtualBox:~$ rsync -av /etc/bind/ ~/primario/
sending incremental file list
created directory /home/core/primario
./
bind.keys
db.0
db.127
db.255
db.empty
db.local
named.conf
named.conf.default-zones
named.conf.local
named.conf.options
rsync: send_files failed to open "/etc/bind/rndc.key": Permission denied (13)
zones.rfc1918

sent 7,399 bytes  received 289 bytes  5,125.33 bytes/sec
total size is 6,748  speedup is 0.88
rsync error: some files/attrs were not transferred (see previous errors) (code 23) at main.c(1207)
[sender=3.1.3]

```

Figura 16: Cópia dos ficheiros para a pasta primario

```

core@core-VirtualBox:~$ rsync -av /etc/bind/ ~/secundario/
sending incremental file list
created directory /home/core/secundario
./
bind.keys
db.0
db.127
db.255
db.empty
db.local
named.conf
named.conf.default-zones
named.conf.local
named.conf.options
rsync: send_files failed to open "/etc/bind/rndc.key": Permission denied (13)
zones.rfc1918

sent 7,399 bytes  received 291 bytes  15,380.00 bytes/sec
total size is 6,748  speedup is 0.88
rsync error: some files/attrs were not transferred (see previous errors) (code 23) at main.c(1207)
[sender=3.1.3]

```

Figura 17: Cópia dos ficheiros para a pasta secundario

## Passo 2) Parar o servidor DNS pré-instalado

De forma a podermos testar a nossa configuração, foi necessário parar o servidor DNS pré-instalado, executando os comandos presentes na seguinte figura:

```

core@core-VirtualBox:~$ sudo systemctl stop bind9.service
[sudo] password for core:
core@core-VirtualBox:~$ sudo systemctl status bind9.service
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sat 2021-05-01 11:57:21 WEST; 3 days ago
     Docs: man:named(8)
    Main PID: 60239 (code=exited, status=0/SUCCESS)

mai 01 11:57:10 core-VirtualBox named[60239]: no longer listening on 127.0.0.1#53
mai 01 11:57:10 core-VirtualBox named[60239]: no longer listening on 10.0.2.15#53
mai 01 11:57:10 core-VirtualBox named[60239]: no longer listening on ::1#53
mai 01 11:57:10 core-VirtualBox named[60239]: no longer listening on fe80::9d95:d407:c7a8:51d1%25
mai 01 11:57:10 core-VirtualBox named[60239]: shutting down: flushing changes
mai 01 11:57:10 core-VirtualBox named[60239]: stopping command channel on 127.0.0.1#953
mai 01 11:57:10 core-VirtualBox named[60239]: stopping command channel on ::1#953
mai 01 11:57:18 core-VirtualBox named[60239]: exiting
mai 01 11:57:21 core-VirtualBox systemd[1]: named.service: Succeeded.
mai 01 11:57:21 core-VirtualBox systemd[1]: Stopped BIND Domain Name Server.
lines 1-16/16 (END)

```

Figura 18: Comando executado para parar o servodpr dns pré-instalado

### Passo 3) Reconfigurar apparmor para permitir que `/usr/sbin/named` aceda a ficheiros noutros locais

O kernel Linux inclui um sistema de proteção para evitar que alguns programas acessem a ficheiros que não devem. Para isso, deve-se verificar se o daemon respetivo (`named`) consta na lista de perfis controlados pelo apparmor.

Neste caso, vamos precisar de reconfigurar essas permissões para que o `/usr/sbin/named` possa ler as novas diretorias. Para tal, adicionamos as novas pastas a este ficheiro e reiniciamos o app armor:

```
# /etc/bind should be read-only for bind
# /var/lib/bind is for dynamically updated zone (and journal) files.
# /var/cache/bind is for slave/stub data, since we're not the origin of it.
# See /usr/share/doc/bind9/README.Debian.gz
/etc/bind/** r,
/home/core/primario/** r,
/home/core/secundario/** r,
/var/lib/bind/** rw,
/var/lib/bind/ rw,
/var/cache/bind/** lrw,
/var/cache/bind/ rw,
```

Figura 19: Pastas adicionadas ao `/etc/apparmor.d/usr.sbin.named`

## 2.4 Configuração de Servidores

As configurações a fazer devem respeitar as regras anteriormente mencionadas.

Inicialmente, tivemos de editar o ficheiro `/etc/hosts` para incluir os registos 10.1.1.1 Server1 ns.cc.pt do primário e 10.2.2.2 Mercurio ns2.cc.pt do secundário, de modo a que os servidores de DNS se identifiquem corretamente.

```
core@core-VirtualBox:~$ cat /etc/hosts
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
fe80::      ip6-localhost
```

Figura 20: Ficheiro `/etc/hosts`

De seguida editamos o ficheiro `/primario/named.conf.options` por forma a incluir os servidores do DI como forwarders:

```
forwarders {
    193.136.9.240;
    193.136.19.1;
};
```

Figura 21: Ficheiro `primario/named.conf.options`

Depois de analisarmos a topologia CC-Topo-2021, reparamos que esta tem 4 LAN's diferentes, e, como tal, decidimos, ao invés de criar apenas as zonas `cc.pt` e `1.1.10.in-addr.arpa`, criar uma zona para cada rede LAN, ficando assim com 5 zonas: as duas mencionadas anteriormente mais a `2.2.10.in-addr.arpa`, `3.3.10.in-addr.arpa` e `4.4.10.in-addr.arpa`.

Assim sendo, colocamos no ficheiro `primario/named.conf` cada uma destas zonas nas quais foi necessário usar o `type master`, visto que estamos a mexer nas zonas do servidor DNS principal. A esta informação também adicionamos a cláusula `allow-transfer {10.2.2.2;}` de modo a permitir a transferência para o servidor secundário e adicionar o ficheiro na respetiva base de dados.

```
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 "3.3.10.in-addr.arpa" {
    type master;
    file "/home/core/primario/db.3-3-10.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;};
};
```

Figura 22: Ficheiro `primario/named.conf`

Após configurar as zonas, seguiu-se a criação e configuração do ficheiro `db.cc.pt`. O primeiro passo foi a configuração do Start of Authority(SOA). Escolhemos o `ns.cc.pt` para a posição do DNS principal desta zona, visto que este é o servidor principal e para o lugar do administrador usamos o `PL01G05.cc.pt`, por ser um dos requisitos mencionados no enunciado.

```
$TTL      604800
@         IN      SOA     ns.cc.pt. PL01G05.cc.pt. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
```

Figura 23: Introdução do SOA na base de dados

Feito isto, introduzimos os nameservers `ns.cc.pt` e `ns2.cc.pt` utilizando o `type NS`. A estes seguiram-se os servidores de e-mail `email.cc.pt` e `Server3.cc.pt`, usando desta vez o

type MX, colocando também um número que representa a prioridade do servidor e, visto que o email.cc.pt é o que tem maior prioridade, colocamos um número menor.

```
@      IN      NS      ns.cc.pt.
@      IN      NS      ns2.cc.pt.
@      IN  MX  10  mail.cc.pt.
@      IN      MX  20  Server3.cc.pt.
```

Figura 24: Introdução dos nameservers e dos servidores de email

Para todos os elementos na rede, inserimos um registo com o nome do elemento na topologia e o respetivo endereço recorrendo ao tipo A permitindo, deste modo, executar o comando ping (Nome na Topologia).cc.pt para todos os elementos presentes na topologia.

Também foi necessário fazer o mesmo para os serviços necessários. Por isso inserimos um registo com o nome do elemento no Serviço (mail,pop,imap,www,ns,n2) e o respetivo endereço IP.

Por fim, era necessário estabelecer o alias entre Laptop1.cc.pt e g05.cc.pt usando uma cláusula com o tipo CNAME.

```
mail    IN      A      10.1.1.2
pop     IN      CNAME  Server3
imap    IN      CNAME  Server3
www     In      A      10.1.1.2
ns      In      A      10.1.1.1
ns2     In      A      10.2.2.2
Marte   IN      A      10.2.2.1
Mercurio      IN      A      10.2.2.2
Venus   IN      A      10.2.2.3
Server1 IN      A      10.1.1.1
Server2 IN      A      10.1.1.2
Server3 IN      A      10.1.1.3
Laptop1 IN      A      10.4.4.1
Laptop2 IN      A      10.4.4.2
Laptop3 IN      A      10.4.4.3
Pico    IN      A      10.3.3.1
Faial   IN      A      10.3.3.2
Corvo   IN      A      10.3.3.3
g05     IN      CNAME  Laptop1
```

Figura 25: Mapping para todos os elementos e serviços da topologia

Feita a configuração da db.cc.pt, é necessário configurar todos os ficheiros correspondentes aos domínios reversos. Embora fosse necessário configurar 4 ficheiros diferentes, este processo era muito semelhante entre eles. Definimos o SOA de modo semelhante ao do db.cc.pt, mudando apenas o nome para o respetivo domínio reverso, por exemplo , 1.1.10.in-addr.arpa. Adicionamos apenas os 2 nameservers existentes, ns.cc.pt e ns2.cc.pt e a cláusula \$ORIGIN para definir o nome da zona.

Para todos os elementos presentes em cada LAN foi adicionada uma entrada na base de dados no ficheiro da rede correspondente, indicando o valor da interface de rede e o nome do elemento na topologia seguido de .cc.pt.. A estas entradas foi ainda necessário adicionar o reverse mapping para os serviços e para todos os alias.

```

$TTL 604800
1.1.10.in-addr.arpa. IN SOA ns.cc.pt. PL01G05.cc.pt. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
IN NS ns.cc.pt.
IN NS ns2.cc.pt.

$ORIGIN 1.1.10.in-addr.arpa.
2 IN PTR mail.cc.pt.
1 IN PTR Server1.cc.pt.
2 IN PTR Server2.cc.pt.
3 IN PTR Server3.cc.pt.
2 IN PTR www.cc.pt.
1 IN PTR ns.cc.pt.
3 IN PTR pop.cc.pt.
3 IN PTR imap.cc.pt.

```

Figura 26: Exemplo do reverse mapping para a rede 10.1.1.0/24

Após definir as 4 bases de dados para o domínio reverso, a configuração do servidor DNS primário ficou concluída.

O próximo passo era configurar o servidor secundário, onde apenas foi necessário adicionar as zonas definidas no servidor primário, com algumas alterações nas cláusulas usadas no ficheiro *named.conf*. O type, que no servidor primário tinha valor *master*, passar a ter o valor de *slave* e o allow-transfer foi substituído por *masters{10.1.1.1;}*. O path para o ficheiro também foi atualizado.

```

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/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 "3.3.10.in-addr.arpa" {
    type slave;
    file "/var/cache/bind/db.3-3-10.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;};
};

```

Figura 27: Ficheiro secundario/named.conf



## 2.5 Demonstração do domínio

Após configurar tanto o servidor primário como o servidor secundário, é necessário testar as funcionalidades destes e para tal fizemos diferentes tipos de teste.

O primeiro consistiu em iniciar o servidor secundário, sem ter iniciado primeiro o servidor primário e testar o serviço DNS, e, tal como era esperado, não obtivemos qualquer resposta, visto que ainda não havia informação que tornasse a resposta possível.

```
04-May-2021 13:41:47.229 zone 1.1.10.in-addr.arpa/IN: refresh; retry limit for master 10.1.1.1#53 exceeded (source 0.0.0.0#0)
04-May-2021 13:41:47.229 zone 1.1.10.in-addr.arpa/IN: zone transfer deferred due to quota
04-May-2021 13:41:47.229 transfer of '3.3.10.in-addr.arpa/IN' from 10.1.1.1#53: failed to connect: connection refused
04-May-2021 13:41:47.229 zone cc.pt/IN: Transfer started.
04-May-2021 13:41:47.229 transfer of '3.3.10.in-addr.arpa/IN' from 10.1.1.1#53: Transfer status: connection refused
04-May-2021 13:41:47.229 transfer of '3.3.10.in-addr.arpa/IN' from 10.1.1.1#53: Transfer completed: 0 messages, 0 records, 0 bytes, 0.001 secs (0 bytes/sec)
04-May-2021 13:41:47.229 transfer of '2.2.10.in-addr.arpa/IN' from 10.1.1.1#53: failed to connect: connection refused
04-May-2021 13:41:47.229 zone 1.1.10.in-addr.arpa/IN: Transfer started.
04-May-2021 13:41:47.229 transfer of '2.2.10.in-addr.arpa/IN' from 10.1.1.1#53: Transfer status: connection refused
04-May-2021 13:41:47.229 transfer of '2.2.10.in-addr.arpa/IN' from 10.1.1.1#53: Transfer completed: 0 messages, 0 records, 0 bytes, 0.001 secs (0 bytes/sec)
04-May-2021 13:41:47.229 zone cc.pt/IN: got transfer_quota; skipping zone transfer as master 10.1.1.1#53 (source 0.0.0.0#0) is unreachable (cached)
04-May-2021 13:41:47.229 zone 1.1.10.in-addr.arpa/IN: got transfer_quota; skipping zone transfer as master 10.1.1.1#53 (source 0.0.0.0#0) is unreachable (cached)
[]
```

```
root@Laptop1:/tmp/pycore.40595/Laptop1.conf# nslookup www.cc.pt 10.2.2.2
Server:      10.2.2.2
Address:     10.2.2.2#53

** server can't find www.cc.pt: SERVFAIL

root@Laptop1:/tmp/pycore.40595/Laptop1.conf#
```

Figura 28: Teste com apenas o servidor secundário ligado, sem ter ligado o primário anteriormente

Feito isto, e para de facto testarmos o domínio de nomes cc.pt, arrancamos os dois servidores, o primário e o secundário e fizemos uma query a cada um, obtendo assim os seguintes resultados:

```
04-May-2021 14:33:33.229 zone 127.in-addr.arpa/IN: loaded serial 1
04-May-2021 14:33:33.229 zone cc.pt/IN: loaded serial 2
04-May-2021 14:33:33.229 all zones loaded
04-May-2021 14:33:33.229 running
04-May-2021 14:33:33.229 zone 4.4.10.in-addr.arpa/IN: sending notifies (serial 2)
04-May-2021 14:33:33.229 zone 1.1.10.in-addr.arpa/IN: sending notifies (serial 2)
04-May-2021 14:33:33.253 zone 2.2.10.in-addr.arpa/IN: sending notifies (serial 2)
04-May-2021 14:33:33.253 zone cc.pt/IN: sending notifies (serial 2)
04-May-2021 14:33:33.253 zone 3.3.10.in-addr.arpa/IN: sending notifies (serial 2)
04-May-2021 14:33:34.433 timed out resolving './DNSKEY/IN': 183.136.19.1#53
04-May-2021 14:33:35.633 timed out resolving './DNSKEY/IN': 183.136.9.240#53
04-May-2021 14:33:43.233 managed-keys-zone: Unable to fetch DNSKEY set '': timed out
04-May-2021 14:33:43.233 resolver priming query complete
04-May-2021 14:33:43.233 managed-keys.bind.jnl: open: permission denied
04-May-2021 14:33:43.233 managed-keys-zone: keyfetch_done:dns_journal_open -> unexpected error
04-May-2021 14:33:43.233 managed-keys-zone: error during managed-keys processing (unexpected error): DNSSEC validation may be at risk
[]
```

```
root@Laptop1:/tmp/pycore.40595/Laptop1.conf# nslookup www.cc.pt 10.1.1.1
Server:      10.1.1.1
Address:     10.1.1.1#53

Name: www.cc.pt
Address: 10.1.1.2

root@Laptop1:/tmp/pycore.40595/Laptop1.conf#
```

```
04-May-2021 14:35:04.789 zone 127.in-addr.arpa/IN: loaded serial 1
04-May-2021 14:35:04.789 zone 255.in-addr.arpa/IN: loaded serial 1
04-May-2021 14:35:04.809 zone cc.pt/IN: loaded serial 2
04-May-2021 14:35:04.813 zone localhost/IN: loaded serial 2
04-May-2021 14:35:04.813 all zones loaded
04-May-2021 14:35:04.817 running
04-May-2021 14:35:04.817 zone 3.3.10.in-addr.arpa/IN: sending notifies (serial 2)
04-May-2021 14:35:04.821 zone cc.pt/IN: sending notifies (serial 2)
04-May-2021 14:35:04.821 zone 2.2.10.in-addr.arpa/IN: sending notifies (serial 2)
04-May-2021 14:35:04.821 zone 1.1.10.in-addr.arpa/IN: sending notifies (serial 2)
04-May-2021 14:35:04.821 zone 4.4.10.in-addr.arpa/IN: sending notifies (serial 2)
04-May-2021 14:35:14.821 managed-keys-zone: Unable to fetch DNSKEY set '': timed out
04-May-2021 14:35:14.821 managed-keys.bind.jnl: open: permission denied
04-May-2021 14:35:14.821 managed-keys-zone: keyfetch_done:dns_journal_open -> unexpected error
04-May-2021 14:35:14.821 managed-keys-zone: error during managed-keys processing (unexpected error): DNSSEC validation may be at risk
04-May-2021 14:35:14.821 resolver priming query complete
[]
```

```
root@Laptop3:/tmp/pycore.40595/Laptop3.conf# nslookup Marte.cc.pt 10.2.2.2
Server:      10.2.2.2
Address:     10.2.2.2#53

Name: Marte.cc.pt
Address: 10.2.2.1

root@Laptop3:/tmp/pycore.40595/Laptop3.conf#
```

Figura 29: Teste com ambos os servidores ligados

Como podemos ver pela imagem acima, ambos os servidores respondem à query feita, obtendo assim o IP 10.1.1.2 para o domain name www.cc.pt e o IP 10.2.2.1 para o domain

Para testar a funcionalidade do servidor secundário, desligamos primeiro o servidor primário, e fizemos as mesmas queries que fizemos no teste anterior e obtivemos os seguintes resultados:

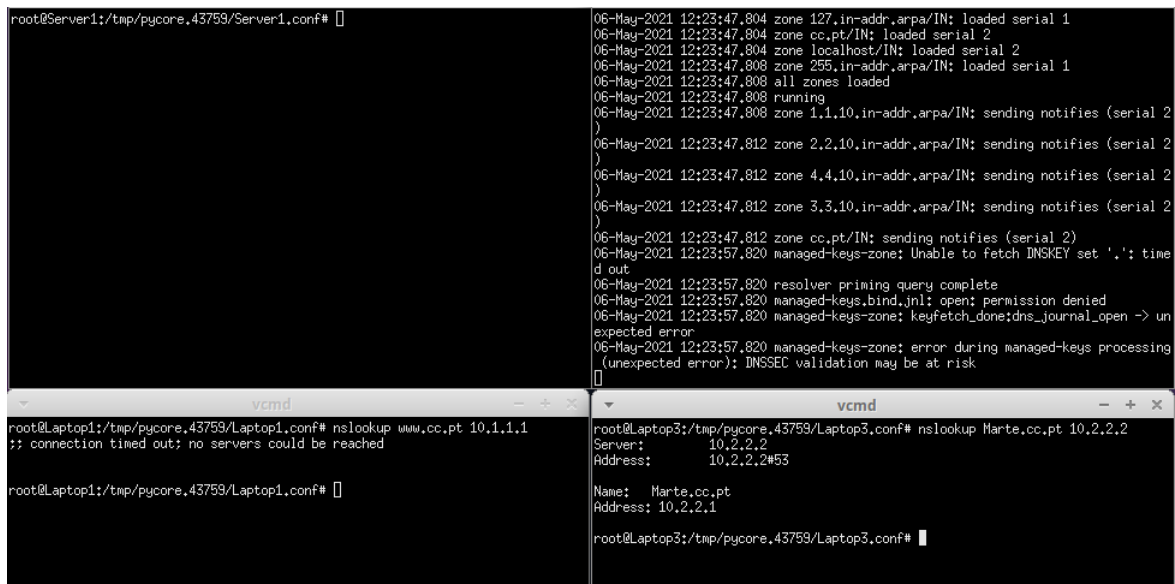


Figura 30: Teste com apenas o servidor secundário

Como podemos observar, a query feita ao servidor primário deu timeout, visto que este se encontra desligado, no entanto, o servidor secundário respondeu à query de forma correta, demonstrando assim que, mesmo que o servidor primário não esteja disponível, ainda é possível obter resultados das queries através do servidor secundário.

Para testarmos os domínios reversos, arrancamos os dois servidores e utilizamos o dig com a opção -x para obter os seguintes resultados:

```

root@Laptop1:/tmp/pycore.40595/Laptop1.conf# dig @10.1.1.1 -x 10.3.3.3

;<> DiG 9.16.1-Ubuntu <> @10.1.1.1 -x 10.3.3.3
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26729
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 072aad4c8ccbf90c010000006091504af163db92d50ded57 (good)
;; QUESTION SECTION:
;3.3.3.10.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
3.3.3.10.in-addr.arpa. 604800 IN      PTR      Corvo.cc.pt.

;; Query time: 4 msec
;; SERVER: 10.1.1.1#53(10.1.1.1)
;; WHEN: ter mai 04 14:46:50 WEST 2021
;; MSG SIZE rcvd: 103

```

Figura 31: Reverse com o servidor primário

```

;<> DiG 9.16.1-Ubuntu <> @10.2.2.2 -x 10.2.2.3
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26739
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 508f9b473776dc760100000060914f26e6f122383d31642c (good)
;; QUESTION SECTION:
;3.2.2.10.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
3.2.2.10.in-addr.arpa. 604800 IN      PTR      Venus.cc.pt.

;; Query time: 4 msec
;; SERVER: 10.2.2.2#53(10.2.2.2)
;; WHEN: ter mai 04 14:41:58 WEST 2021
;; MSG SIZE rcvd: 103

root@Laptop1:/tmp/pycore.40595/Laptop1.conf#

```

Figura 32: Reverse com o servidor secundário

Podemos ver pelas imagens que as queries funcionaram e também podemos reparar que, na *ANSWER SECTION*, temos o resultado da query e também o domínio reverso responsável pela resposta.

Por fim queríamos testar o domínio reverso com algum componente que tivesse um alias, por exemplo, o Laptop1 e obtivemos o seguinte resultado.

```

^Croot@Pico:/tmp/pycore.43759/Pico.conf# dig @10.1.1.1 -x 10.4.4.1

; <<> DiG 9.16.1-Ubuntu <<> @10.1.1.1 -x 10.4.4.1
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59381
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 4096
; COOKIE: 9397fc7922a791e3010000006093cf2d2f420ef3fd3f7fc4 (good)
;; QUESTION SECTION:
;1.4.4.10.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
1.4.4.10.in-addr.arpa.  604800 IN      PTR      Laptop1.cc.pt.
1.4.4.10.in-addr.arpa.  604800 IN      PTR      g05.cc.pt.

;; Query time: 20 msec
;; SERVER: 10.1.1.1#53(10.1.1.1)
;; WHEN: qui mai 06 12:12:45 WEST 2021
;; MSG SIZE rcvd: 123

```

Figura 33: Reverse DNS lookup do Laptop1

### 3 Conclusão

Durante a realização deste terceiro projeto da UC de Comunicações por Computador, foi-nos possível aprofundar e aprimorar conhecimentos adquiridos em aulas, tendo posto em prática o aprendido em aulas teóricas.

Apesar da atenção e dedicação em aulas e partilha de informação e conhecimento entre elementos do grupo, tivemos dificuldade em obter algumas informações mesmo executando os comandos indicados para cada questão, como por exemplo as respostas "autoritativas" e mesmo a secção "answer". Com o auxílio do professor, acabamos por optar pelo comando "sudo nano /etc/resolv.conf" para editar o resultado do comando "cat /etc/resolv.conf" de forma a resolver a situação.

Independentemente das adversidades, o grupo conseguiu superá-las alcançando os objetivos do projeto. Apresentamos, então, ambos os servidores propostos devidamente funcionais e acreditamos responder corretamente a todas as questões, considerando, por isso, ter resolvido com sucesso o proposto.