

Comunicações por Computador

Relatório TP3:
Serviço de Resolução de Nomes

João Nunes (A82300) Luís Braga (A82088)

Luís Martins (A82298)

Grupo 52

12 de Abril 2019

Conteúdo

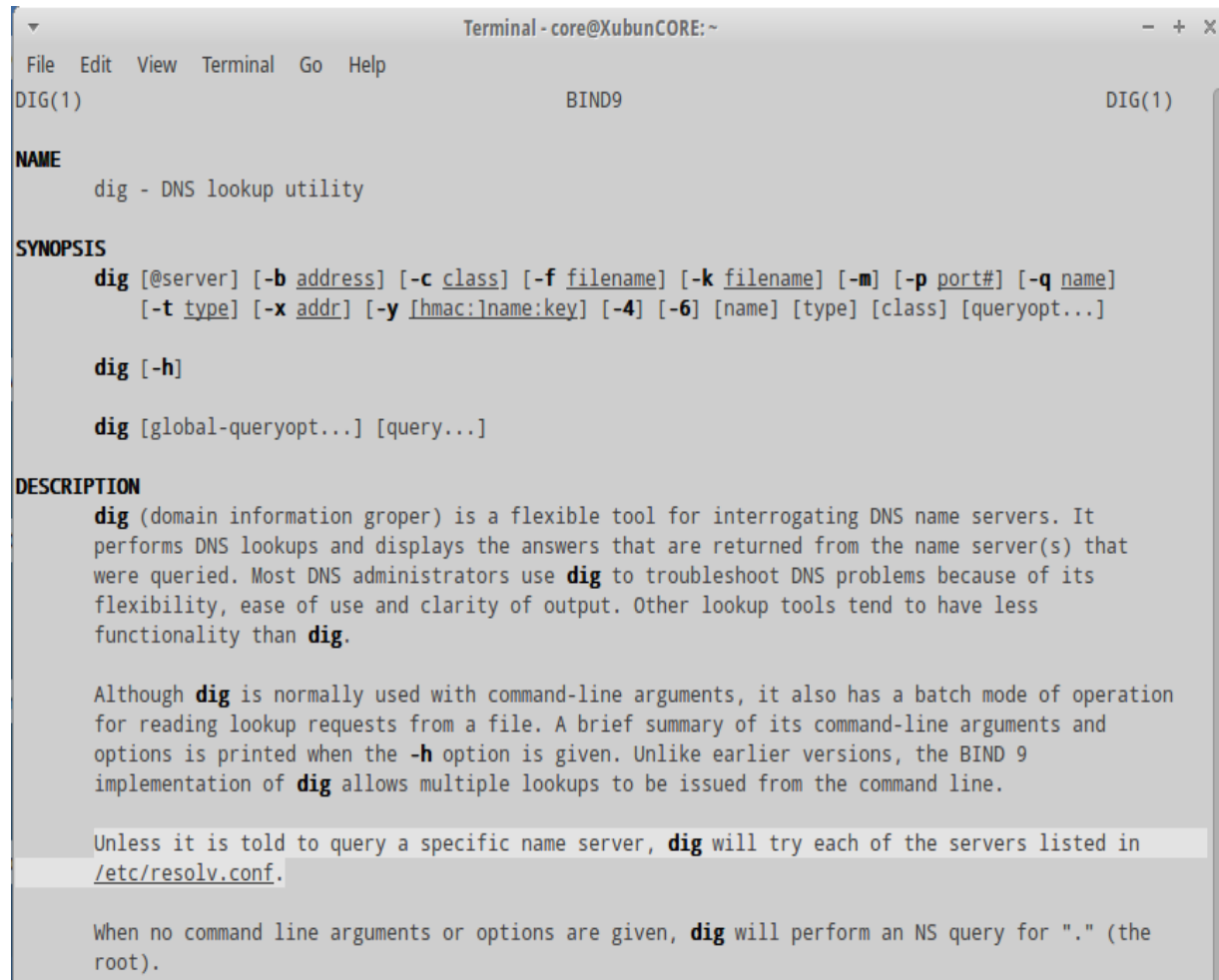
1	Questões e Respostas	4
1.1	Parte I: Consultas ao serviço de nomes DNS	4
1.2	Parte II: Instalação, configuração e teste de um domínio CC.PT . .	19
1.2.1	Preparativos especiais para ambiente Core.	19
1.2.2	Configuração do servidor primário.	25
1.2.3	Configuração do cliente e teste do primário.	36
1.2.4	Configuração do servidor secundário.	37
2	Conclusão	42

Lista de Figuras

1	Conteúdo do ficheiro resolv.conf.	4
2	Execução do comando man dig.	5
3	Resultado do comando nslookup no servidor www.google.pt	6
4	Resultado do comando nslookup no servidor www.google.com	6
5	Resultado do comando nslookup para o domínio ccg.pt.	6
6	Resultado do comando nslookup para o domínio pt.	7
7	Resultado do comando nslookup para o domínio	8
8	Resultado do comando nslookup com a query anterior.	9
9	Resultado do comando nslookup com a query SOA.	9
10	Resultado do comando nslookup com a query SOA.	10
11	Resultado do comando dig sobre o servidor primário da ami.	11
12	Resultado do comando nslookup no domínio presidencia.pt com a query MX	12
13	Resultado do comando nslookup no domínio onu.org com a query MX	12
14	Resultado do comando nslookup com os diferentes record types anteriores.	13
15	Resultado do comando nslookup no endereço IPv6.	14
16	Resultado do comando nslookup com o record type AAAA no domínio "fcen.pt".	15
17	Resultado do comando nslookup com o record type SOA aplicado ao "fcen.pt".	16
18	Resultado da query "di.uminho.pt.".	18
19	Execução do comando \$ rsync -av /etc/bind/ ~/primario/	20
20	Execução do comando \$ rsync -av /etc/bind/ /secundario/	20
21	Execução do comando \$ cd ~/primario; ls;	21
22	Execução do comando \$ cd ~/secundario; ls;	21
23	Execução dos comando \$ sudo /etc/init.d/bind9 status e \$ sudo /etc/init.d/bind9 stop.	21
24	Execução do comando \$ sudo /etc/init.d/apparmor status.	22
25	Aspecto geral do ficheiro \$ /etc/apparmor.d/usr.sbin.named.	23
26	Execução dos comandos \$ sudo /etc/init.d/apparmor stop e \$ sudo /etc/init.d/apparmor start.	24
27	Execução do comando \$ sudo /etc/init.d/apparmor status.	25
28	Aspecto geral do ficheiro /etc/hosts.	26
29	Aspecto geral do ficheiro primario/named.conf.options.	27
30	Inclusões no ficheiro primario/named.conf.	27
31	Definição da zona cc.pt.	28
32	Definição da zona 1.1.10.in-addr.arpa.	28
33	Definição da zona 2.2.10.in-addr.arpa.	28

34	Definição da zona 3.3.10.in-addr.arpa.	28
35	Definição da zona 4.4.10.in-addr.arpa.	29
36	Aspecto geral do ficheiro primario/db.cc.pt.	30
37	Aspecto geral do ficheiro primario/db.1-1-10.rev.	31
38	Aspecto geral do ficheiro primario/db.2-2-10.rev.	32
39	Aspecto geral do ficheiro primario/db.3-3-10.rev.	32
40	Aspecto geral do ficheiro primario/db.4-4-10.rev.	33
41	Verificação de toda a configuração e carregamento dos ficheiros de dados.	33
42	Verificação individual do ficheiro de dados de zona cc.pt.	34
43	Verificação individual do ficheiro de dados de zona 1.1.10.in-addr.arpa.	34
44	Verificação individual do ficheiro de dados de zona 2.2.10.in-addr.arpa.	34
45	Verificação individual do ficheiro de dados de zona 3.3.10.in-addr.arpa.	35
46	Verificação individual do ficheiro de dados de zona 4.4.10.in-addr.arpa.	35
47	Servidor primário em funcionamento na linha de comandos.	35
48	Servidor primário em funcionamento na bash do Servidor1, com um teste simples recorrendo ao nslookup.	36
49	Modificação do ficheiro /etc/resolv.conf.	37
50	Teste recorrendo ao nslookup no nó do Cliente1.	37
51	Inclusão dos endereços IP no named.conf.options.	38
52	Definição das zonas no named.conf.new-zones.	39
53	Verificação da configuração elaborada.	39
54	Servidor primário e secundário em funcionamento.	40
55	Resultado do comando nslookup.	41

Na figura abaixo pode-se ver uma descrição da utilização deste ficheiro na informação dada pelo comando **man dig**.



```
Terminal - core@XubunCORE: ~
File Edit View Terminal Go Help
DIG(1) BIND9 DIG(1)

NAME
    dig - DNS lookup utility

SYNOPSIS
    dig [@server] [-b address] [-c class] [-f filename] [-k filename] [-m] [-p port#] [-q name]
    [-t type] [-x addr] [-y hmac:name:key] [-4] [-6] [name] [type] [class] [queryopt...]

    dig [-h]

    dig [global-queryopt...] [query...]

DESCRIPTION
    dig (domain information groper) is a flexible tool for interrogating DNS name servers. It
    performs DNS lookups and displays the answers that are returned from the name server(s) that
    were queried. Most DNS administrators use dig to troubleshoot DNS problems because of its
    flexibility, ease of use and clarity of output. Other lookup tools tend to have less
    functionality than dig.

    Although dig is normally used with command-line arguments, it also has a batch mode of operation
    for reading lookup requests from a file. A brief summary of its command-line arguments and
    options is printed when the -h option is given. Unlike earlier versions, the BIND 9
    implementation of dig allows multiple lookups to be issued from the command line.

    Unless it is told to query a specific name server, dig will try each of the servers listed in
    /etc/resolv.conf.

    When no command line arguments or options are given, dig will perform an NS query for "." (the
    root).
```

Figura 2: Execução do comando man dig.

b. Os servidores **www.google.pt.** e **www.google.com.** têm endereços IPv6? Se sim, quais?

Sim possuem endereços IPv6, uma vez que igualando a query ao *record type* AAAA com o comando nslookup, utilizando o seguinte comando:

set q=AAAA

Com a query anterior, é então possível retornar o endereço IPv6 caso exista dos servidores, como se pode verificar nas seguintes figuras:

```

> set q=AAAA
> www.google.pt.
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
www.google.pt  has AAAA address 2a00:1450:4003:80a::2003

```

Figura 3: Resultado do comando nslookup no servidor www.google.pt

```

core@XubunCORE:~/secundario$ nslookup
> set q=AAAA
> www.google.com.
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
www.google.com has AAAA address 2a00:1450:4003:80a::2004

```

Figura 4: Resultado do comando nslookup no servidor www.google.com

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

Novamente, recorreu-se ao serviço do nslookup, apenas alterando o *record type* para o seguinte:

$$\text{set } q = NS$$

Portanto, é possível observar os seguintes resultados:

```

> set q=NS
> cpg.pt
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
cpg.pt  nameserver = ns1.cpg.pt.
cpg.pt  nameserver = ns3.cpg.pt.

Authoritative answers can be found from:
ns1.cpg.pt  internet address = 193.136.11.201
ns3.cpg.pt  internet address = 193.136.11.203

```

Figura 5: Resultado do comando nslookup para o domínio cpg.pt.

```

> set q=NS
> pt.
Server:          193.137.16.65
Address:         193.137.16.65#53

Non-authoritative answer:
pt      nameserver = ns2.nic.fr.
pt      nameserver = e.dns.pt.
pt      nameserver = b.dns.pt.
pt      nameserver = ns.dns.br.
pt      nameserver = g.dns.pt.
pt      nameserver = d.dns.pt.
pt      nameserver = sns-pb.isc.org.
pt      nameserver = f.dns.pt.
pt      nameserver = c.dns.pt.
pt      nameserver = a.dns.pt.

Authoritative answers can be found from:
g.dns.pt      internet address = 193.136.2.226
sns-pb.isc.org internet address = 192.5.4.1
f.dns.pt      internet address = 162.88.45.1
ns.dns.br     internet address = 200.160.0.5
ns2.nic.fr    internet address = 192.93.0.4
a.dns.pt      internet address = 185.39.208.1
c.dns.pt      internet address = 204.61.216.105
b.dns.pt      internet address = 194.0.25.23
e.dns.pt      internet address = 193.136.192.64
d.dns.pt      internet address = 185.39.210.1
g.dns.pt      has AAAA address 2001:690:a80:4001::100
sns-pb.isc.org has AAAA address 2001:500:2e::1
f.dns.pt      has AAAA address 2600:2000:3009::1
ns.dns.br     has AAAA address 2001:12ff:0:a20::5
ns2.nic.fr    has AAAA address 2001:660:3005:1::1:2

```

Figura 6: Resultado do comando nslookup para o domínio pt.


```

> set q=NS
> .
Server:      193.137.16.65
Address:     193.137.16.65#53

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

Authoritative answers can be found from:
j.root-servers.net      internet address = 192.58.128.30
k.root-servers.net      internet address = 193.0.14.129
e.root-servers.net      internet address = 192.203.230.10
f.root-servers.net      internet address = 192.5.5.241
d.root-servers.net      internet address = 199.7.91.13
i.root-servers.net      internet address = 192.36.148.17
h.root-servers.net      internet address = 198.97.190.53
c.root-servers.net      internet address = 192.33.4.12
a.root-servers.net      internet address = 198.41.0.4
g.root-servers.net      internet address = 192.112.36.4
l.root-servers.net      internet address = 199.7.83.42
m.root-servers.net      internet address = 202.12.27.33
b.root-servers.net      internet address = 199.9.14.201
j.root-servers.net      has AAAA address 2001:503:c27::2:30
k.root-servers.net      has AAAA address 2001:7fd::1

```

Figura 7: Resultado do comando nslookup para o domínio . .

d. Existe o domínio eureka.software.? Será que eureka.software. é um host?

De modo a responder a esta pergunta, foi necessário alterar de novo o *record type* da query de tal forma que, desta vez, seja retornado o endereço IPv4 de modo a averiguar ou não se é um host e ao mesmo tempo verificar a existência do domínio, portanto:

set q=A

Desta maneira, os resultados verificados seguem-se:

```

> set q=A
> eureka.software.
Server:          127.0.0.1
Address:         127.0.0.1#53

Non-authoritative answer:
Name:   eureka.software
Address: 34.214.90.141

```

Figura 8: Resultado do comando nslookup com a query anterior.

Portanto, existe então um domínio porque está associado a um endereço, e é um host uma vez que possui endereço IP (34.214.90.141).

e. Qual é o servidor DNS primário definido para o domínio ami.pt. ? Este servidor primário (master) aceita queries recursivas? Porquê?

Novamente, foi necessário alterar o *record type*, neste caso, foi utilizado o *SOA* (*Start of Authority*) que é utilizado para obter informação autoritativa, portanto fez-se a seguinte query, recorrendo novamente ao nslookup:

set q=SOA

Através deste processo, obteve-se os seguintes resultados:

```

> set q=SOA
> ami.pt.
Server:          127.0.0.1
Address:         127.0.0.1#53

Non-authoritative answer:
ami.pt
  origin = ns1.dot2web.com
  mail addr = dc.dot2web.pt
  serial = 2019021301
  refresh = 3600
  retry = 7200
  expire = 1209600
  minimum = 86400

```

Figura 9: Resultado do comando nslookup com a query SOA.

Portanto, através de uma análise da figura anterior, é possível verificar que o servidor primário é o *ns1.dot2web.com*.

De modo verificar se este servidor aceita queries recursivas, pode-se admitir o seguinte processo:

1. Mudar para o servidor primário;
2. Realizar uma query que não seja sobre o domínio do servidor.

Para tal, foi necessário executar os seguintes comandos:

```
> server ns1.dot2web.com 1º Passo
Default server: ns1.dot2web.com
Address: 80.172.230.28#53
> ccg.pt. 2º Passo
Server:      ns1.dot2web.com
Address:     80.172.230.28#53
** server can't find ccg.pt: REFUSED
```

Figura 10: Resultado do comando nslookup com a query SOA.

No primeiro passo especifica-se o servidor que é interrogado, e é de notar que este é o servidor primário do domínio **ami.pt.** .

De seguida, fez-se uma query sobre um domínio de uma alínea anterior que possui a individualidade de não pertencer ao domínio **ami.pt.**, cujo resultado podemos ver rodeado a vermelho como "REFUSED". Assim, podemos concluir que este domínio não aceita interrogações recursivas.

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

De seguida, temos uma resposta autoritativa para a pergunta anterior.

```
luis@luis-VirtualBox ~ $ dig @ns1.dot2web.com. ami.pt. SOA

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns1.dot2web.com. ami.pt. SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 44540
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ami.pt. IN SOA

;; ANSWER SECTION:
ami.pt. 86400 IN SOA ns1.dot2web.com. dc.dot2web.pt. 2019040100 3600 7200 1209600 86400

;; AUTHORITY SECTION:
ami.pt. 86400 IN NS ns1.dot2web.com.
ami.pt. 86400 IN NS ns2.dot2web.com.

;; ADDITIONAL SECTION:
ns2.dot2web.com. 14400 IN A 54.36.137.213
ns1.dot2web.com. 14400 IN A 80.172.230.28

;; Query time: 21 msec
;; SERVER: 80.172.230.28#53(80.172.230.28)
;; WHEN: Sat Apr 06 19:00:04 WEST 2019
;; MSG SIZE rcvd: 161
```

Figura 11: Resultado do comando dig sobre o servidor primário da ami.

g. Onde são entregues as mensagens dirigidas a `marcelo@presidencia.pt` ? E a `guterres@onu.org` ?

Partindo novamente do comando *nslookup*, e tal como nas perguntas supracitadas, foi necessário mudar o record type para:

$$set\ q=MX$$

Uma vez que, com este record type, é possível obter informação referente a que servidor de mail é responsável por aceitar os emails.

Tendo isto em conta, é possível observar o seguinte comportamento de ambos os domínios:

```

core@XubunCORE:~$ nslookup
> set q=MX
> presidencia.pt
Server:      192.168.1.1
Address:     192.168.1.1#53

Non-authoritative answer:
presidencia.pt mail exchanger = 10 mail2.presidencia.pt.
presidencia.pt mail exchanger = 50 mail1.presidencia.pt.

Authoritative answers can be found from:
presidencia.pt nameserver = ns02.fccn.pt.
presidencia.pt nameserver = ns2.presidencia.pt.
presidencia.pt nameserver = ns1.presidencia.pt.
mail2.presidencia.pt internet address = 192.162.17.32
mail1.presidencia.pt internet address = 192.162.17.31
ns1.presidencia.pt internet address = 192.162.17.5
ns2.presidencia.pt internet address = 192.162.17.6
ns02.fccn.pt internet address = 193.136.2.228
ns02.fccn.pt has AAAA address 2001:690:a80:4001::200

```

Figura 12: Resultado do comando nslookup no domínio presidencia.pt com a query MX

```

core@XubunCORE:~$ nslookup
> set q=MX
> onu.org
Server:      192.168.1.1
Address:     192.168.1.1#53

Non-authoritative answer:
onu.org mail exchanger = 10 mail.onu.org.

Authoritative answers can be found from:
onu.org nameserver = ns20.semillasl.com.
onu.org nameserver = cp.semillasl.com.
mail.onu.org internet address = 188.165.87.178
cp.semillasl.com internet address = 188.165.87.178

```

Figura 13: Resultado do comando nslookup no domínio onu.org com a query MX

Portanto, tendo em conta a figura 11 e 12, as mensagens dirigidas a marcelo@presidencia.pt são entregues ao "mail2.presidencia.pt." e "mail1.presidencia.pt.".

No que toca as mensagens dirigidas a guterres@onu.org, estas são entregues ao "mail.onu.org.".

h. Que informação é possível obter acerca de **www.whitehouse.gov**? Qual é o endereço IPv4 associado?

Relativamente ao *whitehouse.gov*, foi possível retirar alguma informação, nomeadamente o endereço IPv4 e IPv6 desse domínio, como tal, foi necessário estabelecer dois tipos de queries:

set q=A

De modo a ser retornada informação sobre o endereço IPv4, e também:

set q=AAAA

Para retornar o conjunto de informação referente ao endereço IPv6 desse domínio.

Como tal, foi utilizado novamente o comando *nslookup*, tendo sido utilizadas as queries supracitadas, tendo se obtido o seguinte conjunto de informação:

```
core@XubunCORE:~$ nslookup
> set q=A
> whitehouse.gov
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
Name:   whitehouse.gov
Address: 172.227.167.182 ← Endereço IPv4
> set q=AAAA
> whitehouse.gov.
Server:      193.137.16.145
Address:     193.137.16.145#53 ← Endereços IPv6

Non-authoritative answer:
whitehouse.gov has AAAA address 2a02:26f0:dd:28e::2add
whitehouse.gov has AAAA address 2a02:26f0:dd:28b::2add

Authoritative answers can be found from:
whitehouse.gov nameserver = a22-66.akam.net.
whitehouse.gov nameserver = zc.akam.net.
whitehouse.gov nameserver = use6.akam.net.
whitehouse.gov nameserver = a12-64.akam.net.
whitehouse.gov nameserver = asia9.akam.net.
whitehouse.gov nameserver = a20-65.akam.net.
whitehouse.gov nameserver = a5-64.akam.net.
whitehouse.gov nameserver = usw1.akam.net.
whitehouse.gov nameserver = ns1-176.akam.net.
whitehouse.gov nameserver = ns1-145.akam.net.
whitehouse.gov nameserver = a1-61.akam.net.
whitehouse.gov nameserver = a3-67.akam.net.
zc.akam.net internet address = 184.26.160.66
usw1.akam.net internet address = 23.61.199.66
a3-67.akam.net internet address = 96.7.49.67
a5-64.akam.net internet address = 95.100.168.64
a12-64.akam.net internet address = 184.26.160.64
a20-65.akam.net internet address = 95.100.175.65
ns1-145.akam.net internet address = 193.108.91.145
zc.akam.net has AAAA address 2600:1401:1::40
a20-65.akam.net has AAAA address 2a02:26f0:67::41
```

Figura 14: Resultado do comando *nslookup* com os diferentes record types anteriores.

i. Consegue interrogar o DNS sobre o endereço IPv6 2001:690:a00:1036:1113::247 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?

De modo a interrogar o endereço IPv6 disponibilizado, foi utilizado o comando nslookup sem nenhum tipo de record type, pelo que foi retornado de seguida, um todo conjunto de informação útil no que toca a esse endereço.

```
core@XubunCORE:~$ nslookup
> 2001:690:a00:1036:1113::247
Server:      193.137.16.145
Address:     193.137.16.145#53

Non-authoritative answer:
7.4.2.0.0.0.0.0.0.0.0.0.3.1.1.1.6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa    name = www.fccn.pt.

Authoritative answers can be found from:
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.
0.9.6.0.1.0.0.2.ip6.arpa      nameserver = ns02.fccn.pt.
ns01.fccn.pt      internet address = 193.136.192.40
ns03.fccn.pt      internet address = 138.246.255.249
ns02.fccn.pt      internet address = 193.136.2.228
ns01.fccn.pt      has AAAA address 2001:690:a00:4001::200
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
```

Figura 15: Resultado do comando nslookup no endereço IPv6.

Com os resultados anteriores, é possível determinar o domínio associado a esse endereço IPv6, sendo esse o "*fccn.pt.*", tratando-se portanto de um website.

De modo a confirmar que o endereço IPv6 do "*fccn.pt.*" corresponde com o endereço disponibilizado, foi também elaborada uma nova interrogação:

set q=AAAA

Desta vez, aplicado ao domínio anterior, de tal forma que:

```

core@XubunCORE:~$ nslookup
> set q=AAAA
> fccn.pt.
Server:          193.137.16.65
Address:         193.137.16.65#53
Endereço IPv6
↑
Non-authoritative answer:
fccn.pt has AAAA address 2001:690:a00:1036:1113::247

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

```

Figura 16: Resultado do comando nslookup com o record type AAAA no domínio "fccn.pt".

Portanto, desta forma, está confirmado que o endereço IPv6 disponibilizado corresponde ao domínio do "fccn.pt.", uma vez que são iguais.

De modo a obter um contacto de um responsável por esse IPv6 é necessário interrogar o domínio associado ou seja, o "fccn.pt.", sendo utilizado então a seguinte query:

set q=SOA

De modo a obter informação a seguinte informação administrativa:


```

> set q=SOA
> fccn.pt.
Server:      193.137.16.65
Address:     193.137.16.65#53

Non-authoritative answer:
fccn.pt
  origin = ns01.fccn.pt
  mail addr = hostmaster.fccn.pt
  serial = 2019040904
  refresh = 21600
  retry = 7200
  expire = 1209600
  minimum = 14400

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

```

Figura 17: Resultado do comando nslookup com o record type SOA aplicado ao "fccn.pt".

Portanto, o contacto do responsável do endereço IPv6 é o "*hostmaster.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 serviço de DNS é um dos sistemas mais utilizados e é responsável por manter a Internet a funcionar e, por isso, é necessário que este sistema esteja disponível ininterruptamente, sendo considerado um serviço crítico.

Os servidores DNS hospedam o que pode ser conhecido como zona DNS. Uma zona DNS é uma porção do domínio que é servida por um servidor DNS e que contém vários *records*, que são os valores dados como resposta a perguntas que um cliente poderá, eventualmente, fazer a um servidor.

No entanto, um dos servidores responsáveis por uma zona poderá sofrer uma avaria. Portanto, todos os servidores que hospedam uma zona têm de se manter actualizados, fazendo assim o que é chamado uma transferência de zona DNS.

Uma transferência de zona DNS é um de vários métodos de que um administrador de redes dispõe para replicar bases de dados através dos servidores DNS.

Também com grande importância podemos referir os parâmetros do tipo SOA do domínio que são os seguintes e possuem os seguintes significados:

- **Serial** - Representa o número de série da zona. Se o servidor secundário que lhe está associado sentir que houve alguma diminuição deste valor, ele irá assumir que a zona foi actualizada e irá iniciar uma transferência de zona.
- **Refresh** - Número de segundos que um servidor secundário deve demorar até pedir novamente, ao servidor primário, os registos SOA.
- **Retry** - Número de segundos que um servidor secundário deve demorar até pedir, ao servidor primário, o número de série, se este não estiver a responder.
- **Expire** - Número de segundos que um servidor secundário deve demorar a parar de pedir informação ao primário se este não estiver a responder.
- **TTL** - Referente ao tempo que a informação será mantida em cache (em segundos).

A título de exemplo considere-se a figura abaixo, na qual está retratada a query *Start Of Authority* (SOA) do domínio *di.uminho.pt.* .

Neste domínio o servidor primário é o destacado a vermelho (na figura), sendo os servidores secundários os destacados a azul. Portanto, pode-se dizer que todos estes servidores estão na mesma zona e que todos devem estar actualizados.

```

> set q=SOA
> di.uminho.pt.
Server:  dns3.uminho.pt
Address: 193.137.16.65

di.uminho.pt
  primary name server = dns.di.uminho.pt
  responsible mail addr = dnsadmin.di.uminho.pt
  serial = 2019040803
  refresh = 28800 (8 hours)
  retry = 7200 (2 hours)
  expire = 28800 (8 hours)
  default TTL = 43200 (12 hours)
di.uminho.pt  nameserver = dns2.uminho.pt
di.uminho.pt  nameserver = dns.uminho.pt
di.uminho.pt  nameserver = dns.di.uminho.pt
di.uminho.pt  nameserver = alfa.di.uminho.pt
di.uminho.pt  nameserver = marco.uminho.pt
di.uminho.pt  nameserver = dns2.di.uminho.pt
di.uminho.pt  nameserver = ns1.eurotux.com
di.uminho.pt  nameserver = dns3.uminho.pt
di.uminho.pt  nameserver = ns3.eurotux.com
dns.di.uminho.pt  internet address = 193.136.19.1
dns.uminho.pt  internet address = 193.137.16.75
ns1.eurotux.com internet address = 194.107.127.1
ns3.eurotux.com internet address = 216.75.63.6
alfa.di.uminho.pt  internet address = 193.136.19.3
dns2.di.uminho.pt  internet address = 193.136.19.2
dns2.uminho.pt  internet address = 193.137.16.145
dns3.uminho.pt  internet address = 193.137.16.65
marco.uminho.pt internet address = 193.136.9.240
dns.di.uminho.pt  AAAA IPv6 address = 2001:690:2280:28::1
dns.uminho.pt  AAAA IPv6 address = 2001:690:2280:1::75
dns2.di.uminho.pt  AAAA IPv6 address = 2001:690:2280:28::2
dns2.uminho.pt  AAAA IPv6 address = 2001:690:2280:801::145

```

Figura 18: Resultado da query "di.uminho.pt".

Podemos analisar a figura e retirar dos domínios os valores referidos anteriormente e concluir que:

- **Serial** - 2019040803
- **Refresh** - 28800 (8 horas)
- **Retry** - 7200 (2 horas)
- **Expire** - 28800 (8 horas)
- **TTL** - 43200 (12 horas)

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

A segunda fase do presente enunciado consiste em instalar e configurar um domínio para realizar futuros testes, a fim de melhor entender o funcionamento de um domínio e dos mecanismos que estão por detrás de um sistema deste género. Cumprindo os seguintes requisitos:

- Criação do domínio cc.pt co servidor primário em Servidor1 10.1.1.1 e secundário em Urano 10.2.2.3;
- Criação do domínio reverso 1.1.10.in-addr.arpa com os mesmos servidores;
- O servidor primário do domínio é o “Servidor1” com endereço 10.1.1.1, também designado por dns.cc.pt, tendo como secundário o “Urano” com endereço 10.2.2.3, com alias dns2.cc.pt. O administrador do domínio é o grupoXX@cc.pt (onde 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) em Servidor3. O servidor pop e imap é o Servidor2, 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 Cliente1.cc.pt com alias GrupoXX.cc.pt onde XX é o número do grupo, e Alfa.cc.pt, Beta.cc.pt e Gama.cc.pt no domínio de nomes e no domínio reverso.

Para o efeito, foi necessário realizar uma sequência de passos, cuja prova de realização se encontra abaixo descrita.

1.2.1 Preparativos especiais para ambiente Core.

Passo 1) Replicar ficheiros de configuração

Conforme é recomendado no enunciado: foram executados os comandos que se encontram nas duas próximas figuras.

```
core@XubunCORE:~$ rsync -av /etc/bind/ ~/primario/
sending incremental file list
./
bind.keys
db.0
db.127
db.255
db.empty
db.local
db.root
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 10869 bytes  received 262 bytes  22262.00 bytes/sec
total size is 10153  speedup is 0.91
rsync error: some files/attrs were not transferred (see previous errors) (code 23) at main.c(1070) [sender=3.0.9]
```

Figura 19: Execução do comando `$ rsync -av /etc/bind/ ~/primario/`.

```
core@XubunCORE:~$ rsync -av /etc/bind/ ~/secundario/
sending incremental file list
./
bind.keys
db.0
db.127
db.255
db.empty
db.local
db.root
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 10869 bytes  received 262 bytes  22262.00 bytes/sec
total size is 10153  speedup is 0.91
rsync error: some files/attrs were not transferred (see previous errors) (code 23) at main.c(1070) [sender=3.0.9]
```

Figura 20: Execução do comando `$ rsync -av /etc/bind/ /secundario/`.

Para confirmar a correta execução destes comandos, isto é, se os ficheiros foram

realmente replicados, foram executados os comandos das duas próximas figuras.

```
core@XubunCORE:~$ cd ~/primario; ls;  
bind.keys db.127 db.empty db.root named.conf.default-zones named.conf.options  
db.0 db.255 db.local named.conf named.conf.local zones.rfc1918
```

Figura 21: Execução do comando \$ cd ~/primario; ls;

```
core@XubunCORE:~$ cd ~/secundario; ls;  
bind.keys db.127 db.empty db.root named.conf.default-zones named.conf.options  
db.0 db.255 db.local named.conf named.conf.local zones.rfc1918
```

Figura 22: Execução do comando \$ cd ~/secundario; ls;

Depois de visualizar as figuras pode-se concluir que tudo está a decorrer normalmente.

Passo 2) Parar o servidor DNS pré-instalado

Também foi necessário parar o servidor DNS pré-instalado para se conseguirem realizar os passos de forma a atingir o objectivo final.

```
core@XubunCORE:~$ sudo /etc/init.d/bind9 status  
[sudo] password for core:  
* bind9 is running  
core@XubunCORE:~$ sudo /etc/init.d/bind9 stop  
* Stopping domain name service... bind9
```

Figura 23: Execução dos comando \$ sudo /etc/init.d/bind9 status e \$ sudo /etc/init.d/bind9 stop.

Passo 3) reconfigurar apparmor para permitir que /usr/sbin/named aceda a ficheiros noutros locais.

De forma a permitir que o "/usr/sbin/named"aceda a ficheiros noutro locais é necessário dar-lhe tais permissões. Para isso é necessário parar e reiniciar o apparmor, que é um módulo de segurança do sistema operativo linux, que permite ao administrador do sistema restringir as capacidades de um programa.

Verifica-se, então se o daemon respectivo (named) consta na lista de perfis controlados pelo apparmor e executa-se o comando seguinte:

```
core@XubunCORE:~$ sudo /etc/init.d/apparmor status
apparmor module is loaded.
17 profiles are loaded.
17 profiles are in enforce mode.
  /sbin/dhclient
  /usr/bin/evince
  /usr/bin/evince-previewer
  /usr/bin/evince-previewer//launchpad_integration
  /usr/bin/evince-previewer//sanitized_helper
  /usr/bin/evince-thumbnailer
  /usr/bin/evince-thumbnailer//sanitized_helper
  /usr/bin/evince//launchpad_integration
  /usr/bin/evince//sanitized_helper
  /usr/lib/NetworkManager/nm-dhcp-client.action
  /usr/lib/connman/scripts/dhclient-script
  /usr/lib/cups/backend/cups-pdf
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper//chromium_browser
  /usr/sbin/cupsd
  /usr/sbin/named ←
  /usr/sbin/tcpdump
0 profiles are in complain mode.
2 processes have profiles defined.
2 processes are in enforce mode.
  /sbin/dhclient (2127)
  /usr/sbin/cupsd (570)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

Figura 24: Execução do comando `$ sudo /etc/init.d/apparmor status`.

Podemos ver que sim, ele está na lista de perfis. Acrescentam-se, então, as seguintes linhas ao ficheiro `/etc/apparmor.d/usr.sbin.named`

```
/home/core/primario/** r,
/home/core/secundario/** r,
```

Ficando ele com o seguinte aspecto:

```

/usr/sbin/named {
    #include <abstractions/base>
    #include <abstractions/nameservice>

    capability net_bind_service,
    capability setgid,
    capability setuid,
    capability sys_chroot,
    capability sys_resource,

    # /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/** rw,
    /var/cache/bind/ rw,

```

Figura 25: Aspecto geral do ficheiro \$ /etc/apparmor.d/usr.sbin.named.

Depois de adicionadas as linhas tem de se parar e reiniciar o programa, executando os comandos apresentados seguidamente


```

core@XubunCORE:~$ sudo /etc/init.d/apparmor stop
* Clearing AppArmor profiles cache
All profile caches have been cleared, but no profiles have been unloaded.
Unloading profiles will leave already running processes permanently
unconfined, which can lead to unexpected situations.

To set a process to complain mode, use the command line tool
'aa-complain'. To really tear down all profiles, run the init script
with the 'teardown' option."
core@XubunCORE:~$ sudo /etc/init.d/apparmor start
* Starting AppArmor profiles
Skipping profile in /etc/apparmor.d/disable: usr.bin.firefox
Skipping profile in /etc/apparmor.d/disable: usr.sbin.rsyslogd

core@XubunCORE:~$ sudo /etc/init.d/apparmor status
apparmor module is loaded.
17 profiles are loaded.
17 profiles are in enforce mode.
  /sbin/dhclient
  /usr/bin/evince
  /usr/bin/evince-previewer
  /usr/bin/evince-previewer//launchpad_integration
  /usr/bin/evince-previewer//sanitized_helper
  /usr/bin/evince-thumbnailer
  /usr/bin/evince-thumbnailer//sanitized_helper
  /usr/bin/evince//launchpad_integration
  /usr/bin/evince//sanitized_helper
  /usr/lib/NetworkManager/nm-dhcp-client.action
  /usr/lib/connman/scripts/dhclient-script
  /usr/lib/cups/backend/cups-pdf
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper//chromium_browser
  /usr/sbin/cupsd
  /usr/sbin/named
  /usr/sbin/tcpdump
0 profiles are in complain mode.
2 processes have profiles defined.
2 processes are in enforce mode.
  /sbin/dhclient (2127)
  /usr/sbin/cupsd (570)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.

```

Figura 26: Execução dos comandos `$ sudo /etc/init.d/apparmor stop` e `$ sudo /etc/init.d/apparmor start`.

E verifica-se novamente se o daemon respectivo consta na lista de perfis controlados pelo apparmor.

```

core@XubunCORE:~$ sudo /etc/init.d/apparmor status
apparmor module is loaded.
17 profiles are loaded.
17 profiles are in enforce mode.
  /sbin/dhclient
  /usr/bin/evince
  /usr/bin/evince-previewer
  /usr/bin/evince-previewer//launchpad_integration
  /usr/bin/evince-previewer//sanitized_helper
  /usr/bin/evince-thumbnailer
  /usr/bin/evince-thumbnailer//sanitized_helper
  /usr/bin/evince//launchpad_integration
  /usr/bin/evince//sanitized_helper
  /usr/lib/NetworkManager/nm-dhcp-client.action
  /usr/lib/connman/scripts/dhclient-script
  /usr/lib/cups/backend/cups-pdf
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper
  /usr/lib/lightdm/lightdm/lightdm-guest-session-wrapper//chromium_browser
  /usr/sbin/cupsd
  /usr/sbin/named ←
  /usr/sbin/tcpdump
0 profiles are in complain mode.
2 processes have profiles defined.
2 processes are in enforce mode.
  /sbin/dhclient (2127)
  /usr/sbin/cupsd (570)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.

```

Figura 27: Execução do comando `$ sudo /etc/init.d/apparmor status`.

Podendo confirmar-se que este se encontra na lista.

1.2.2 Configuração do servidor primário.

Inicia-se agora a configuração do servidor primário propriamente dita, onde se configuraram os diferentes ficheiros da directoria "*primario*", de acordo com as indicações dadas no enunciado.

Passo 1) Editar o ficheiro `/etc/hosts` para incluir os registos *10.1.1.1 Servidor1 dns.cc.pt* do primário e *10.2.2.3 Urano dns2.cc.pt* do secundário; este passo é obrigatório para que os servidores DNS se identifiquem correctamente a si próprios.

Começou-se por se escrever as seguintes linhas no ficheiro `/etc/hosts`.

10.1.1.1	Servidor1	dns.cc.pt
10.2.2.3	Urano	dns2.cc.pt

Ficando o ficheiro com o seguinte aspecto:

```
127.0.0.1      localhost
127.0.1.1      XubunCORE

# 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
### begin CORE auto-generated hosts entries
10.0.0.1      A0
10.0.0.2      A1
10.0.0.3      A2
10.0.0.4      A3
10.0.0.5      A4
10.0.0.6      A5
10.0.0.7      A6
10.0.0.8      A7
10.0.0.9      A8
10.0.0.10     A9
10.0.0.11     A10
10.0.0.12     A11
10.0.0.13     A12
10.0.0.14     A13
10.0.0.15     A14
10.0.0.16     A15
10.0.0.17     A16
10.1.1.1      Servidor1      dns.cc.pt
10.2.2.3      Urano          dns2.cc.pt
### end CORE auto-generated hosts entries
```

Figura 28: Aspecto geral do ficheiro `/etc/hosts`.

Passo 2) Editar o ficheiro *primario/named.conf.options* por forma a incluir os servidores 193.136.9.240 e 193.136.19.1 (servidores do DI) como forwarders.

Portanto, acrescentou-se os IP's no ficheiro *primario/named.conf.options*, ficando ele com o seguinte aspecto.

```

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;

    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
};

```

Figura 29: Aspecto geral do ficheiro `primario/named.conf.options`.

Passo 3) Editar o ficheiro `primario/named.conf` para incluir a indicação das novas zonas “cc.pt”, “1.1.10.in-addr.arpa” etc (ver exemplos em `named.conf.default-zones`) e corrigir os nomes das diretorias, substituindo `/etc/bind/` pela nova diretoria (`/home/core/primario` ou outra equivalente). Não esquecer de incluir uma cláusula “allow-transfer” a dar permissão de transferência da base de dados ao servidor secundário.

Acrescentou-se a linha abaixo ao ficheiro `primario/named.conf` e substituiu-se `/etc/bind/` por `/home/core/primario`:

```
include "home/core/primario/named.conf.new-zones;"
```

```

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

```

Figura 30: Inclusões no ficheiro `primario/named.conf`.

Onde no ficheiro *primario/named.conf.new-zones* estão definidas as zonas do domínio. Abaixo encontram-se as definições destas zonas.

```
zone "cc.pt" {  
    type master;  
    file "/home/core/primario/db.cc.pt";  
    allow-transfer {10.2.2.3};  
};
```

Figura 31: Definição da zona cc.pt.

```
zone "1.1.10.in-addr.arpa" {  
    type master;  
    file "/home/core/primario/db.1-1-10.rev";  
    allow-transfer {10.2.2.3};  
};
```

Figura 32: Definição da zona 1.1.10.in-addr.arpa.

```
zone "2.2.10.in-addr.arpa" {  
    type master;  
    file "/home/core/primario/db.2-2-10.rev";  
    allow-transfer {10.2.2.3};  
};
```

Figura 33: Definição da zona 2.2.10.in-addr.arpa.

```
zone "3.3.10.in-addr.arpa" {  
    type master;  
    file "/home/core/primario/db.3-3-10.rev";  
    allow-transfer {10.2.2.3};  
};
```

Figura 34: Definição da zona 3.3.10.in-addr.arpa.

```
zone "4.4.10.in-addr.arpa" {  
    type master;  
    file "/home/core/primario/db.4-4-10.rev";  
    allow-transfer {10.2.2.3};  
};
```

Figura 35: Definição da zona 4.4.10.in-addr.arpa.

Passo 4) Baseando-se por exemplo no conteúdo do ficheiro *primario/db.local*, procure criar o ficheiro de dados do domínio de nomes: *primario/db.cc.pt* (incluir a informação de acordo com as regras definidas acima).

Procedeu-se, então, à configuração do ficheiro *db.cc.pt*. Assimilou-se primeiro a sintaxe do ficheiro *db.local* como também se fez uma pesquisa para melhor entender os campos necessário para realizar correctamente a tarefa em mão.

Depois deste processo de aprendizagem editou-se o ficheiro *db.cc.pt* e criaram-se os campos na figura abaixo, que se encontram destacados.

```

$TTL      604800
@         IN      SOA      Servidor1.cc.pt. grupo52.cc.pt. (
                                1          ;Serial
                                604800     ;Refresh
                                86400      ;Retry
                                2419200    ;Expire
                                604800) ;Negative Cache TTL
;

@         IN      NS       Servidor1.cc.pt.
@         IN      NS       Urano.cc.pt.

@         IN      MX       10    Servidor3.cc.pt.
@         IN      MX       20    Servidor2.cc.pt.

Servidor1 IN      A        10.1.1.1
Servidor2 IN      A        10.1.1.2
Servidor3 IN      A        10.1.1.3
Urano     IN      A        10.2.2.3
PLutao   IN      A        10.2.2.1
Neptuno  IN      A        10.2.2.2
Alfa     IN      A        10.3.3.1
Beta     IN      A        10.3.3.2
Gama     IN      A        10.3.3.3
Cliente1 IN      A        10.4.4.1
Cliente2 IN      A        10.4.4.2
Cliente3 IN      A        10.4.4.3

pop       IN      CNAME    Servidor2
imap      IN      CNAME    Servidor2
Grupo52   IN      CNAME    Cliente1
dns       IN      CNAME    Servidor1
dns2      IN      CNAME    Urano
mail      IN      CNAME    Servidor3
www       IN      CNAME    Servidor3

```

Figura 36: Aspecto geral do ficheiro primario/db.cc.pt.

A **vermelho** temos o campo *Start of Authority* (SOA), onde se pode ver qual o servidor primário para esta zona *dns.cc.pt*, o administrador da mesma como *grupo52.cc.pt*, o número de série, assim como os tempos de "*Refresh*", "*Retry*", "*Expire*" e "*TTL*" (para informações mais detalhadas ver página 13).

A **azul** estão identificados os *name servers* (NS).

A **roxo** estão identificados *mail exchangers* (MX) e a sua respectiva prioridade. Sendo que o que tem o valor mais baixo o que tem mais prioridade, isto é, a primeira de tentativa de comunicação com um servidor de e-mail será sempre com o Servidor3 e, se este não responder tentar-se-á comunicar com o Servidor2.

A **castanho** estão definidas os *address records* (A) dos diferentes servidores e máquinas do domínio.

A **cor-de-rosa** estão definidos os *canonical name records* (CNAME).

É de salientar que, mais uma vez, toda esta configuração foi feita de acordo com as indicações dadas no enunciado.

Passo 5) Baseando-se por exemplo no conteúdo do ficheiro *primario/db.127* procure criar o ficheiro de dados do(s) domínio(s) de reverse: *primario/db.1-1-10.rev* (ou outros, de acordo com as regras definidas acima).

É definido, neste passo, o domínio de dados reverso, que foi editado de forma semelhante ao ficheiro *db.127*.

```
$TTL      604800
@         IN      SOA     Servidor1.cc.pt. grupo52.cc.pt. (
                                1          ;Serial
                                604800     ;Refresh
                                86400      ;Retry
                                2419200    ;Expire
                                604800)    ;Negative Cache TTL
;
@         IN      NS      Servidor1.cc.pt.
@         IN      NS      Urano.cc.pt.
1         IN      PTR      Servidor1.cc.pt.
2         IN      PTR      Servidor2.cc.pt.
3         IN      PTR      Servidor3.cc.pt.
```

Figura 37: Aspecto geral do ficheiro *primario/db.1-1-10.rev*.

Acima encontra-se a definição do domínio reverso da zona **10.1.1.0/24**. Onde se definiu um tipo de record que nunca antes tinha aparecido, destacado a **vermelho**: o *POINTER* (PTR), que serve como apontador para um determinado nome canónico. É também importante referir que é devido a este "/24" que só se escreveu:

```
1 ...
2 ...
3 ..
```


De forma semelhante ao ficheiro *db.1-1-10.rev* criaram-se os restantes ficheiros, cujo conteúdo se encontra nas figuras que se seguem.

```
$TTL      604800
@         IN      SOA      Servidor1.cc.pt. grupo52.cc.pt. (
                                1          ;Serial
                                604800     ;Refresh
                                86400      ;Retry
                                2419200    ;Expire
                                604800)    ;Negative Cache TTL
;

@         IN      NS       Servidor1.cc.pt.
@         IN      NS       Urano.cc.pt.

3         IN      PTR      Urano.cc.pt.
1         IN      PTR      Plutao.cc.pt.
2         IN      PTR      Neptuno.cc.pt.
```

Figura 38: Aspecto geral do ficheiro primario/db.2-2-10.rev.

```
$TTL      604800
@         IN      SOA      Servidor1.cc.pt. grupo52.cc.pt. (
                                1          ;Serial
                                604800     ;Refresh
                                86400      ;Retry
                                2419200    ;Expire
                                604800)    ;Negative Cache TTL
;

@         IN      NS       Servidor1.cc.pt.
@         IN      NS       Urano.cc.pt.

1         IN      PTR      Alfa.cc.pt.
2         IN      PTR      Beta.cc.pt.
3         IN      PTR      Gama.cc.pt.
```

Figura 39: Aspecto geral do ficheiro primario/db.3-3-10.rev.

```

$TTL      604800
@         IN      SOA      Servidor1.cc.pt. grupo52.cc.pt. (
                                1          ;Serial
                                604800     ;Refresh
                                86400      ;Retry
                                2419200    ;Expire
                                604800)    ;Negative Cache TTL
;
@         IN      NS       Servidor1.cc.pt.
@         IN      NS       Urano.cc.pt.

1         IN      PTR      Cliente1.cc.pt.
2         IN      PTR      Cliente2.cc.pt.
3         IN      PTR      Cliente3.cc.pt.
~

```

Figura 40: Aspecto geral do ficheiro primario/db.4-4-10.rev.

Passo 6) Testar as configurações e os ficheiros de dados com auxílio de algumas ferramentas.

Finda a fase anterior testaram-se as configurações e os ficheiros de dados.

Para testar a configuração executou-se:

```
\$ /usr/sbin/named-checkconf -z /home/core/primario/named.conf -g
```

Cujo resultado se encontra a seguir:

```

core@XubunCORE:~/primario$ /usr/sbin/named-checkconf -z /home/core/primario/named.conf
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
zone cc.pt/IN: loaded serial 1
zone 1.1.10.in-addr.arpa/IN: loaded serial 1
zone 2.2.10.in-addr.arpa/IN: loaded serial 1
zone 3.3.10.in-addr.arpa/IN: loaded serial 1
zone 4.4.10.in-addr.arpa/IN: loaded serial 1

```

Figura 41: Verificação de toda a configuração e carregamento dos ficheiros de dados.

Testaram-se depois os ficheiros de zona. Para a zona *cc.pt* e introduziu-se o seguinte comando:

```
\$ /usr/sbin/named-checkzone cc.pt /home/core/primario  
/db.cc.pt
```

Cujo resultado é:

```
core@XubunCORE:~/primario$ /usr/sbin/named-checkzone cc.pt /home/core/primario/db.cc.pt  
zone cc.pt/IN: loaded serial 1  
OK
```

Figura 42: Verificação individual do ficheiro de dados de zona cc.pt.

Para as restantes zonas executaram-se comandos semelhantes mudando apenas a zona e o ficheiro. Comandos esses que apresentam o seguinte aspecto:

```
\$ /usr/sbin/named-checkzone 1.1.10.in-addr.arpa /home/core  
/primario/db.1-1-10.rev  
.  
.  
.  
\$ /usr/sbin/named-checkzone 4.4.10.in-addr.arpa /home/core  
/primario/db.4-4-10.rev
```

Os resultados encontram-se a seguir pela respectiva ordem dos comandos acima.

```
core@XubunCORE:~/primario$ /usr/sbin/named-checkzone 1.1.10.in-addr.arpa /home/core/primario/d  
b.1-1-10.rev  
zone 1.1.10.in-addr.arpa/IN: loaded serial 1  
OK
```

Figura 43: Verificação individual do ficheiro de dados de zona 1.1.10.in-addr.arpa.

```
core@XubunCORE:~/primario$ /usr/sbin/named-checkzone 2.2.10.in-addr.arpa /home/core/primario/db.2-2-10.rev  
zone 2.2.10.in-addr.arpa/IN: loaded serial 1  
OK
```

Figura 44: Verificação individual do ficheiro de dados de zona 2.2.10.in-addr.arpa.

```
core@XubunCORE:~/primario$ /usr/sbin/named-checkzone 3.3.10.in-addr.arpa /home/core/primario/db.3-3-10.rev
zone 3.3.10.in-addr.arpa/IN: loaded serial 1
OK
```

Figura 45: Verificação individual do ficheiro de dados de zona 3.3.10.in-addr.arpa.

```
core@XubunCORE:~/primario$ /usr/sbin/named-checkzone 4.4.10.in-addr.arpa /home/core/primario/db.4-4-10.rev
zone 4.4.10.in-addr.arpa/IN: loaded serial 1
OK
```

Figura 46: Verificação individual do ficheiro de dados de zona 4.4.10.in-addr.arpa.

Todos os testes deram OK, portanto seguiu-se para a seguinte passo.

Passo 7) Executar o servidor, na linha de comando.

Testou-se o servidor, com o comando:

```
\$ /usr/sbin/named -c /home/core/primario/named.conf -g
```

E obteve-se o resultado:

```
core@XubunCORE:~/primario$ /usr/sbin/named -c /home/core/primario/named.conf -g
11-Apr-2019 18:58:08.866 starting BIND 9.8.1-P1 -c /home/core/primario/named.conf -g
11-Apr-2019 18:58:08.866 built with '--prefix=/usr' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--sysconfdir=/etc/bind' '--localstatedir=/var' '--enable-threads' '--enable-largefile' '--with-libtool' '--enable-shared' '--enable-static' '--with-openssl=/usr' '--with-gssapi=/usr' '--with-gnu-ld' '--with-geoip=/usr' '--enable-ipv6' 'CFLAGS=-fno-strict-aliasing -DDIG_SIGCHASE -O2' 'LDFLAGS=-Wl,-Bsymbolic-functions -Wl,-z,relro' 'CPPFLAGS=-D_FORTIFY_SOURCE=2'
11-Apr-2019 18:58:08.866 found 1 CPU, using 1 worker thread
11-Apr-2019 18:58:08.866 using up to 4096 sockets
11-Apr-2019 18:58:08.871 loading configuration from '/home/core/primario/named.conf'
11-Apr-2019 18:58:08.871 reading built-in trusted keys from file '/etc/bind/bind.keys'
11-Apr-2019 18:58:08.871 using default UDP/IPv4 port range: [1024, 65535]
11-Apr-2019 18:58:08.871 using default UDP/IPv6 port range: [1024, 65535]
```

Figura 47: Servidor primário em funcionamento na linha de comandos.

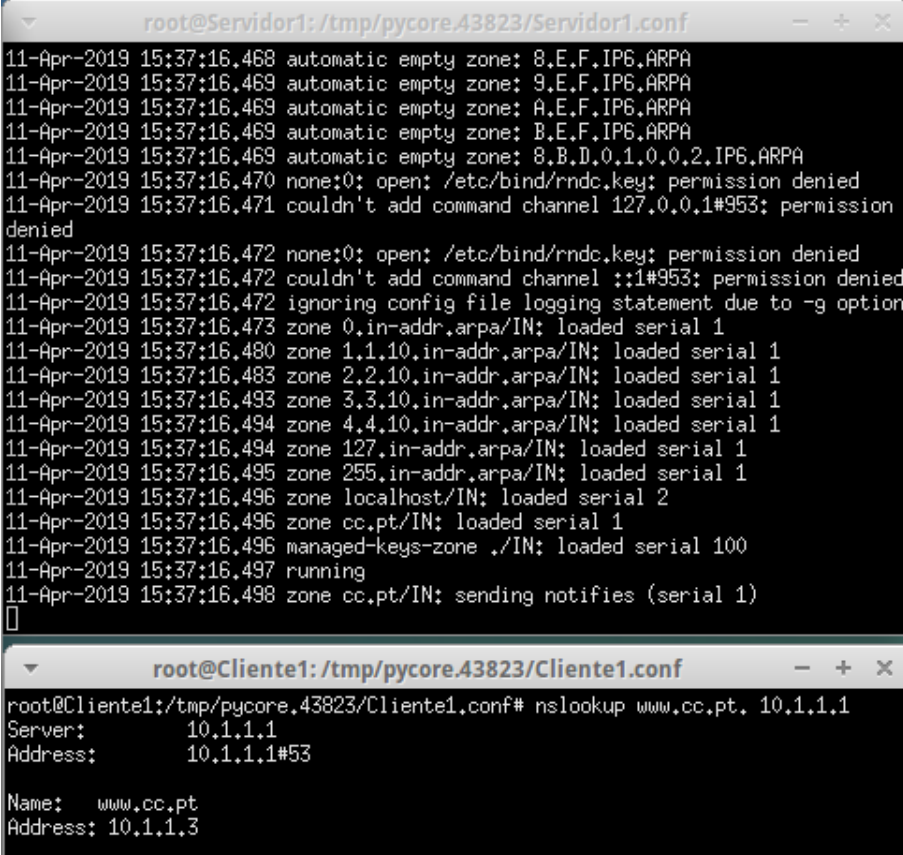
Como se pode ver decorreu tudo normalmente.

1.2.3 Configuração do cliente e teste do primário.

De seguida, executou-se, no Servidor1, o servidor primário na topologia e, no Cliente1, testou-se o seguinte comando:

```
\$ nslookup www.cc.pt 10.1.1.1
```

Na figura que se segue podem-se ver ambos os terminais (Servidor1 e Cliente1). No Servidor1 está a correr o servidor primário e no Cliente1 foi executada a ordem anteriormente descrita.



```
root@Servidor1: /tmp/pycore.43823/Servidor1.conf
11-Apr-2019 15:37:16.468 automatic empty zone: 8.E.F.IP6.ARPA
11-Apr-2019 15:37:16.469 automatic empty zone: 9.E.F.IP6.ARPA
11-Apr-2019 15:37:16.469 automatic empty zone: A.E.F.IP6.ARPA
11-Apr-2019 15:37:16.469 automatic empty zone: B.E.F.IP6.ARPA
11-Apr-2019 15:37:16.469 automatic empty zone: 8.B.D.0.1.0.0.2.IP6.ARPA
11-Apr-2019 15:37:16.470 none:0; open: /etc/bind/rndc.key; permission denied
11-Apr-2019 15:37:16.471 couldn't add command channel 127.0.0.1#953; permission
denied
11-Apr-2019 15:37:16.472 none:0; open: /etc/bind/rndc.key; permission denied
11-Apr-2019 15:37:16.472 couldn't add command channel ::1#953; permission denied
11-Apr-2019 15:37:16.472 ignoring config file logging statement due to -g option
11-Apr-2019 15:37:16.473 zone 0.in-addr.arpa/IN: loaded serial 1
11-Apr-2019 15:37:16.480 zone 1.1.10.in-addr.arpa/IN: loaded serial 1
11-Apr-2019 15:37:16.483 zone 2.2.10.in-addr.arpa/IN: loaded serial 1
11-Apr-2019 15:37:16.493 zone 3.3.10.in-addr.arpa/IN: loaded serial 1
11-Apr-2019 15:37:16.494 zone 4.4.10.in-addr.arpa/IN: loaded serial 1
11-Apr-2019 15:37:16.494 zone 127.in-addr.arpa/IN: loaded serial 1
11-Apr-2019 15:37:16.495 zone 255.in-addr.arpa/IN: loaded serial 1
11-Apr-2019 15:37:16.496 zone localhost/IN: loaded serial 2
11-Apr-2019 15:37:16.496 zone cc.pt/IN: loaded serial 1
11-Apr-2019 15:37:16.496 managed-keys-zone ./IN: loaded serial 100
11-Apr-2019 15:37:16.497 running
11-Apr-2019 15:37:16.498 zone cc.pt/IN: sending notifies (serial 1)
[]

root@Cliente1: /tmp/pycore.43823/Cliente1.conf
root@Cliente1: /tmp/pycore.43823/Cliente1.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: 10.1.1.3
```

Figura 48: Servidor primário em funcionamento na bash do Servidor1, com um teste simples recorrendo ao nslookup.

Como se pode confirmar obtivemos a resposta correcta.

Visto isto, procedeu-se à modificação do ficheiro de */etc/resolv.conf* como recomendado.

```
core@XubunCORE:~/primario$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
#nameserver 193.137.16.65
#nameserver 193.137.16.145
#nameserver 193.137.16.75
#search eduroam.uminho.pt

nameserver 10.1.1.1
domain cc.pt
search cc.pt
```

Figura 49: Modificação do ficheiro */etc/resolv.conf*.

E testou-se novamente no Cliente1.

```
root@Cliente1: /tmp/pycore.43830/Cliente1.conf
root@Cliente1:/tmp/pycore.43830/Cliente1.conf# nslookup www.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

Name:   www.cc.pt
Address: 10.1.1.3

root@Cliente1:/tmp/pycore.43830/Cliente1.conf#
```

Figura 50: Teste recorrendo ao nslookup no nó do Cliente1.

1.2.4 Configuração do servidor secundário.

Após a configuração do servidor primário, segue-se a configuração do servidor secundário, de acordo com os seguintes passos.

Passo 1) Editar o ficheiro *secundario/named.conf.options* por forma a incluir os servidores 193.136.9.240 e 193.136.19.1 (servidores do DI) como *fowarders*.

Tal como foi feito para o servidor primário, foi necessário acrescentar os endereços IP ao ficheiro *secundarios/named.conf.options*, tal como se pode observar na seguinte figura.

A terminal window titled "Terminal - core@XubunCORE: ~/secundario" showing the configuration of the named.conf.options file. The configuration includes a directory path for bind, firewall instructions, and forwarders for 193.136.9.240 and 193.136.19.1.

```
File Edit View Terminal Go Help
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;
    };
};
```

Figura 51: Inclusão dos endereços IP no named.conf.options.

Passo 2) Editar o ficheiro *secundario/named.conf* para incluir a indicação das novas zonas "cc.pt", "1.1.10.in-addr.arpa"etc, mas desta vez apenas como zonas do tipo *slave* (ver manual ou exemplos). Não se esqueça de cláusulas "*masters*" adequada. Assegure-se que os ficheiros de dados das zonas vão para */var/cache/bind/ ...db.cc.pt*, *db.1-1-10.rev*, etc. por causa das permissões de escrita! Não os copie! O servidor secundário terá de os transferir e atualizar automaticamente!

De forma análoga ao que foi feito no servidor primário, foi criado um novo ficheiro *named.conf.new-zones* que contém as zonas identificadas anteriormente. Contudo, ocorreram algumas alterações no que toca a este ficheiro, o tipo da zona é *slave* (*type slave*) uma vez que estamos a tratar do servidor secundário. Devido a esta alteração agora o ficheiro possui a indicação do servidor primário (*masters {10.1.1.1;}*).

```

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 52: Definição das zonas no `named.conf.new-zones`.

Após ser criado este ficheiro, este foi adicionado como uma nova inclusão no ficheiro `secundario/named.conf`.

Passo 3) Testar as configurações e os ficheiros de dados com auxílio de algumas ferramentas.

```
$ /usr/sbin/named-checkconf -z /home/core/secundario/named.conf
```

De modo a averiguar as configurações elaboradas, foi executado o comando supracitado, sendo possível então, concluir que o passo anterior foi completado com sucesso.

```

core@XubunCORE:~$ sudo /usr/sbin/named-checkconf -z /home/core/secundario/named.conf
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1

```

Figura 53: Verificação da configuração elaborada.

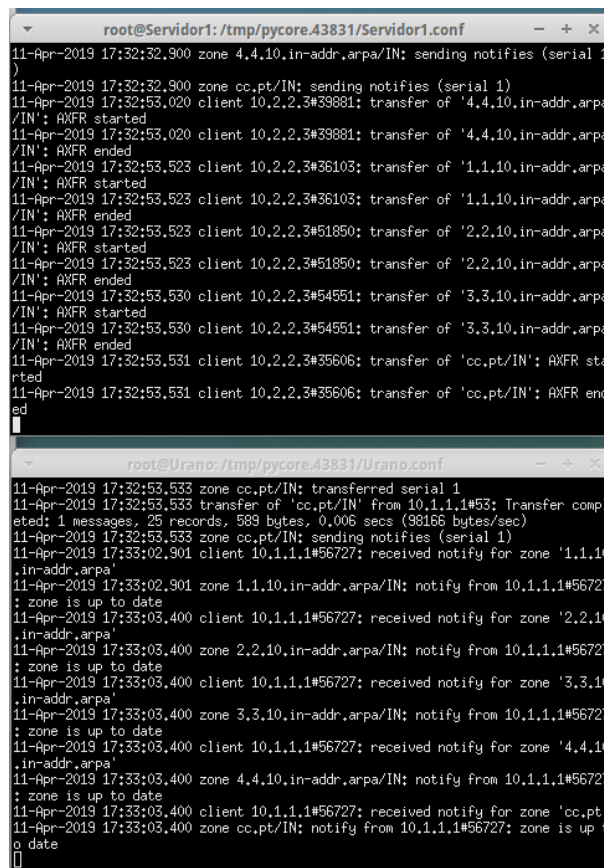
Passo 4) Executar o core e abrir um bash no nó Urano. Executar o servidor, na linha de comando, fazendo por exemplo:

```
$ sudo /usr/sbin/named -c /home/core/primario/named.conf -g
```

Neste passo, de modo a permitir a transferência do ficheiro de dados do servidor primário para o secundário, primeiramente foi necessário inicializar o servidor primário recorrendo ao seguinte comando numa bash do Servidor1:

```
$ sudo /usr/sbin/named -c /home/core/secundario/named.conf -g
```

De seguida, foi necessário correr o servidor secundário no nó de Urano, pelo que foi utilizado o comando dado no próprio passo, pelo que o resultado da inicialização se encontra na figura 52, sendo possível observar a transferência dos ficheiros de dados do servidor primário para o secundário.



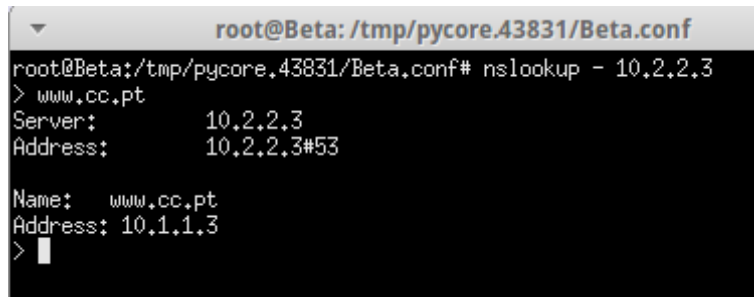
```
root@Servidor1: /tmp/pycore.43831/Servidor1.conf
11-Apr-2019 17:32:32.900 zone 4.4.10.in-addr.arpa/IN: sending notifies (serial 1)
11-Apr-2019 17:32:32.900 zone cc.pt/IN: sending notifies (serial 1)
11-Apr-2019 17:32:53.020 client 10.2.2.3#39881: transfer of '4.4.10.in-addr.arpa/IN': AXFR started
11-Apr-2019 17:32:53.020 client 10.2.2.3#39881: transfer of '4.4.10.in-addr.arpa/IN': AXFR ended
11-Apr-2019 17:32:53.523 client 10.2.2.3#36103: transfer of '1.1.10.in-addr.arpa/IN': AXFR started
11-Apr-2019 17:32:53.523 client 10.2.2.3#36103: transfer of '1.1.10.in-addr.arpa/IN': AXFR ended
11-Apr-2019 17:32:53.523 client 10.2.2.3#51850: transfer of '2.2.10.in-addr.arpa/IN': AXFR started
11-Apr-2019 17:32:53.523 client 10.2.2.3#51850: transfer of '2.2.10.in-addr.arpa/IN': AXFR ended
11-Apr-2019 17:32:53.530 client 10.2.2.3#54551: transfer of '3.3.10.in-addr.arpa/IN': AXFR started
11-Apr-2019 17:32:53.530 client 10.2.2.3#54551: transfer of '3.3.10.in-addr.arpa/IN': AXFR ended
11-Apr-2019 17:32:53.531 client 10.2.2.3#35606: transfer of 'cc.pt/IN': AXFR started
11-Apr-2019 17:32:53.531 client 10.2.2.3#35606: transfer of 'cc.pt/IN': AXFR ended

root@Urano: /tmp/pycore.43831/Urano.conf
11-Apr-2019 17:32:53.533 zone cc.pt/IN: transferred serial 1
11-Apr-2019 17:32:53.533 transfer of 'cc.pt/IN' from 10.1.1.1#53: Transfer completed: 1 messages, 25 records, 589 bytes, 0.006 secs (98166 bytes/sec)
11-Apr-2019 17:32:53.533 zone cc.pt/IN: sending notifies (serial 1)
11-Apr-2019 17:33:02.901 client 10.1.1.1#56727: received notify for zone '1.1.10.in-addr.arpa'
11-Apr-2019 17:33:02.901 zone 1.1.10.in-addr.arpa/IN: notify from 10.1.1.1#56727: zone is up to date
11-Apr-2019 17:33:03.400 client 10.1.1.1#56727: received notify for zone '2.2.10.in-addr.arpa'
11-Apr-2019 17:33:03.400 zone 2.2.10.in-addr.arpa/IN: notify from 10.1.1.1#56727: zone is up to date
11-Apr-2019 17:33:03.400 client 10.1.1.1#56727: received notify for zone '3.3.10.in-addr.arpa'
11-Apr-2019 17:33:03.400 zone 3.3.10.in-addr.arpa/IN: notify from 10.1.1.1#56727: zone is up to date
11-Apr-2019 17:33:03.400 client 10.1.1.1#56727: received notify for zone '4.4.10.in-addr.arpa'
11-Apr-2019 17:33:03.400 zone 4.4.10.in-addr.arpa/IN: notify from 10.1.1.1#56727: zone is up to date
11-Apr-2019 17:33:03.400 client 10.1.1.1#56727: received notify for zone 'cc.pt'
11-Apr-2019 17:33:03.400 zone cc.pt/IN: notify from 10.1.1.1#56727: zone is up to date
```

Figura 54: Servidor primário e secundário em funcionamento.

Passo 5) Teste simples com o nslookup, em qualquer nó da topologia

Após inicializados os servidores, de modo a averiguar e testar o estado destes mesmos, foi elaborada um simples interrogação no nó Beta da topologia, pelo que analisando os resultados da figura 53, está de acordo com o que, na prática, seria expectável retornar, uma vez que o servidor web encontra-se alojado no Servidor3 da topologia, que possui o endereço IP "10.1.1.3".

A terminal window titled 'root@Beta: /tmp/pycore.43831/Beta.conf' displays the output of the 'nslookup - 10.2.2.3' command. The user enters 'www.cc.pt' at the prompt. The terminal shows the server used (10.2.2.3) and the address (10.2.2.3#53). Below this, it shows the lookup results: 'Name: www.cc.pt' and 'Address: 10.1.1.3'. The prompt returns to the user.

```
root@Beta: /tmp/pycore.43831/Beta.conf
root@Beta:/tmp/pycore.43831/Beta.conf# nslookup - 10.2.2.3
> www.cc.pt
Server:      10.2.2.3
Address:     10.2.2.3#53

Name:   www.cc.pt
Address: 10.1.1.3
> 
```

Figura 55: Resultado do comando nslookup.

2 Conclusão

Posto fim ao terceiro trabalho prático, que se revelou como sendo crucial de modo a melhor entender e consolidar um serviço importantíssimo para um bom funcionamento da Internet o DNS. Para tal, foi desenvolvido um domínio, o cc.pt, onde foi necessário aplicar os conhecimentos sobre o DNS, de modo a conseguir implementar os dois servidores, de uma maneira fidedigna. Para além do domínio, foi também necessário também responder às questões iniciais, sendo bastante útil no sentido de melhor entender os diferentes tipos de *record types* do DNS, e o seu significado.

Contudo, é de referir que ocorreram durante a elaboração deste trabalho prático alguns *stumbling blocks* no que toca à aprendizagem da sintaxe e formato dos ficheiros de dados do domínio de nomes como também quanto à configuração desses mesmos devido a questões mais técnicas como "que registos deve conter o ficheiro?", "quantas e quais as zonas a criar?", "aquando do preenchimento de um campo que máquinas/endereços/nomes devem ser considerados?". No entanto, depois de períodos de reflexão e pesquisa o grupo superou todas estas dificuldades.