Serviço de Resolução de Nomes (DNS)

Trabalho realizado por:

Bruno Filipe de Sousa Dias

Francisco Alves Andrade

Paulo Silva Sousa

A89513
Francisco Andrade

A89583
Bruno Dias

A89465
Paulo Sousa

Grupo 14
Comunicações por Computador 2020/2021
Universidade do Minho

Índice

1 Questões e Respostas

1.1 Parte 1 - Consultas ao serviço de nomes DNS

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

O ficheiro */etc/resolv.conf* contém informações sobre o nome do servidor de DNS e sobre o endereço de IP a este associado.

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

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

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

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

O servidor *www.uminho.pt.* não contém endereço IPv6.

Por outro lado, o servidor *www.ubuntu.com.* contém os seguintes endereços IPv6: 2001:67c:1360:8001::2b e 2001:67c:1360:8001::2c.

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

Figure 2:

```
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::2b
www.ubuntu.com has IPv6 address 2001:67c:1360:8001::2c
```

Figure 3:

c) Quais os servidores de nomes definidos para os domínios: *"sapo.pt."*, *"pt."* e *"."*?

```
core@core-VirtualBox:~$ nslookup
> set type=NS
>
```

Figure 4:

Para o servidor *sapo.pt.* estão definidos os seguintes servidores de nome: *ns2.sapo.pt.*, *dns02.sapo.pt.*, *ns.sapo.pt.* e *dns01.sapo.pt.*.

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

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

Authoritative answers can be found from:
```

Figure 5:

Para o servidor *pt.* podemos encontrar 9 servidores de nome, como podemos verificar na seguinte figura:

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

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

Authoritative answers can be found from:
```

Figure 6:

Para o servidor . podemos encontrar 13 servidores de nome, como podemos verificar na seguinte figura:

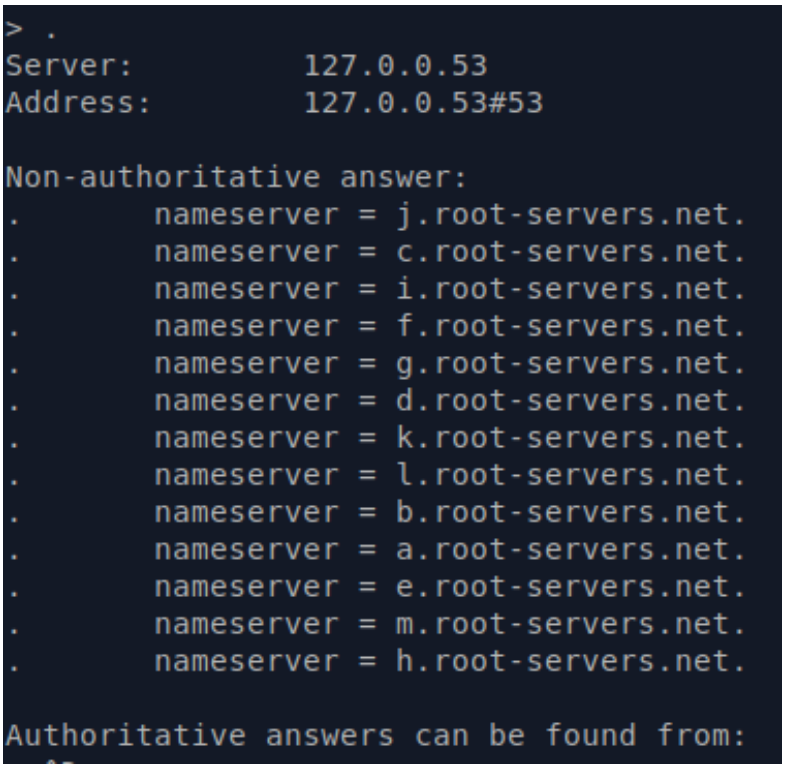

Figure 7:

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

O domínio *open.money.* existe, uma vez que ao executarmos e comando host open.money. é estabeleciada uma conexão.

No entanto, como podemos ver na figura seguinte, o domínio *open.money.* não é um host, mas sim um servidor de mail.

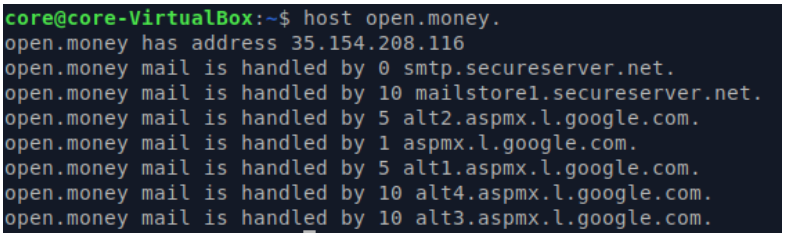

Figure 8:

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

O servidor DNS primário para o domínio *un.org* é o servidor *"ns1.un.org"*. Como forma de validarmos a resposta dada podemos observar a secção *"Non-authoritative answer"* e a sub-secção *"origin"*.

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

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

Authoritative answers can be found from:
```

Figure 9:

O servidor primário em questão aceita *queries* recursivas, já que, ao ser executado o comando "dig ns1.un.org", deparamo-nos com o seguinte resultado:

```
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: 18590
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
ns1.un.org.             45      IN      A       157.150.185.28

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: ter abr 20 12:18:29 WEST 2021
;; MSG SIZE  rcvd: 55
```

Figure 10:

Através da imagem acima conseguimos ver que o servidor contém uma flag *ra* (*"Recursion Available"*), o que indica que o suporte de consulta recursiva está disponível, existindo assim recursividade neste servidor.

f) Obtenha uma resposta *"autoritativa"* para a questão anterior.

Com o intuito de obter uma resposta "autoritativa" para a questão anterior, foi executado o comando *nslookup* do tipo *SOA*, que nos forneceu o seguinte resultado:

```
> 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 = ns3.un.org.
un.org  nameserver = ns2.un.org.
ns1.un.org      internet address = 157.150.185.28
ns3.un.org      internet address = 157.150.241.25
ns2.un.org      internet address = 157.150.34.57
> server ns1.un.org.
Default server: ns1.un.org.
Address: 157.150.185.28#53
> un.org.
;; connection timed out; no servers could be reached

> un.org.
;; connection timed out; no servers could be reached
```

Figure 11:

Assim, obtivemos os seguintes endereços autoritativos: *ns1.un.org ns2.un.org.* Para obtermos uma resposta autoritativa, é necessário questionar esses mesmos endereços. Para isso foi executado *"server ns1.un.org"*.

Por outro lado quando questionamos o servidor com *"un.org"*, obtemos como resposta *"connection timed out; no servers could be reached"*, uma vez que apenas um servidor local consegue responder de forma recursiva ao cliente.

g) Onde são entregues as mensagens de correio eletrónico dirigidas *apresidency@eu.eu* ou *presidencia@2021portugal.eu*?

Para conseguirmos obter informações relacionadas com os os servidores responsáveis pela gestão de e-mails referidos acima, temos de executar os seguintes comandos:

```
> 2021portugal.eu
Server:         127.0.0.53
Address:        127.0.0.53#53

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

Authoritative answers can be found from:
```

Figure 12:

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

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

Authoritative answers can be found from:
```

Figure 13:

Assim, podemos verificar que as mensagens enviadas para *"presidencia@2021portugal.eu"* são entregues em *mxg.eu.mpssec.net.*

Por outro lado, nas mensagens enviadas para apresidency@eu.eu existem duas opções distintas: o valor 20 (smtp02.level.be.) e o valor 10 (smtp01.level.be.). O segundo tem o valor mais baixo, logo será este o servidor primário.

Tendo isso em conta, sabemos que a mensagem será entregue em *smtp01.level.be.* e, caso este não esteja disponível, esta tentará ser entregue em *smtp02.level.be.*.

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

Executando o comando *"dig gov.pt"*, obtemos os seguintes resultados:

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

; <<>> DiG 9.16.1-Ubuntu <<>> gov.pt
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52668
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

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

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: ter abr 27 11:46:24 WEST 2021
;; MSG SIZE  rcvd: 35
```

Figure 14:

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?

Conseguimos, utilizando o comando *"nslookup"* com o tipo PTR, realiza-se um *"reverse mapping"* e obtemos o domínio associado ao endereço IP fornecido: 2001:690:2080:8005::38.

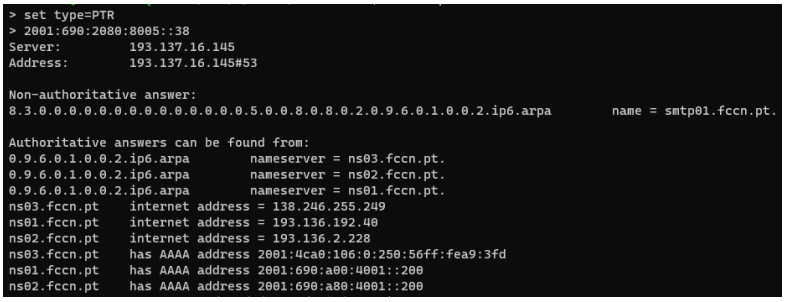

Figure 15:

Assim, coseguimos verificar que o domínio equivalente ao endereço IP em questão é "smtp01.fccn.pt.". Com a informação obtida realizamos outra query sobre o mesmo domínio, do tipo SOA, onde identificamos os contactos responsáveis. O resultado obtido encontra-se abbaixo apresentado, que demonstra que os servidor responsável é "ns01.fccn.pt".

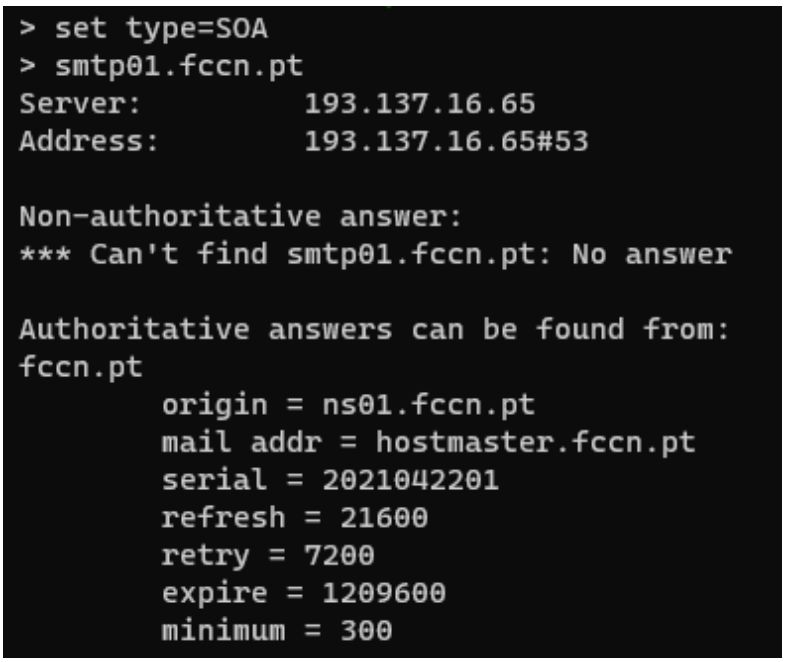

Figure 16:

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

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

O cliente solicita uma transferência de dados de um servidor primário para um servidor secundário. À parte da base de dados que é replicada dá-se o nome de zona. Fazendo uso do domínio cc.pt e do ficheiro db.cc.pt. que foi criado na topologia virtual e utilizada no exercício 2, podemos obter as seguintes informações:

```
$TTL    604800
@       IN      SOA     ns.cc.pt        PL06G14.cc.pt. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
```

Figure 17:

- **Serial:** número de série da zona, utilizada para verificar atualizações. Se um servidor secundário ligado a este verificar uma decrementação deste número, então percebe que zona está desatualizada e atualiza-a, iniciando-se uma transferência de zona.
- **Refresh:** intervalo de segundos que servidor secundário deve respeitar para comunicar com o servidor primário, atualizando informações de forma a verificar se existem alterações de zona (como explicada anteriormente).
- **Retry:** quando um servidor falha uma tentativa de conexão, deve esperar este número de segundos, de modo a poder tentar estabelecer uma nova reconexão com o servidor primário. É importante realçar que este tempo deve ser obrigatoriamente inferior ao tempo de Refresh enunciado acima.
- **Expire:** segundos após o qual servidor seucndário deve parar de fazer solicitações para uma zona quando um servidor primário não responde. É importante realçar que este tempo deve ser maior do que a soma dos tempos dos parâmetros Refresh e de Retry.

Resumindo, o servidor secundário deve contactar com o servidor primário para verificar existência de atualizações a cada 604800 segundos, ou seja, 7 dias. Se por ventura uma tentatia de conexão for falhada, o servidor secundário deve tentar estabelecer uma reconexão após 86400 segundos, ou seja, 1 dia. Eventualmente, se o servidor primário deixar de responder, o servidor secundário ainda tenta obter respontas e contactos, no entanto, se isso não acontecer num periódo de 2419200 segundos, ou seja, 28 dais, a base de dados do servidor secundário deixa de tentar conectar-se à do primário.

2 Domínio de Nomes CC.PT

Esta fase do projeto passava em grande parte por seguir as indicações e instruções dadas pelo enunciado. No entanto, foi necessário tomar algumas decisões e ter em conta diversas interpertações quanto ao que era proposto, de modo a obtermos o melhor sucesso possível. Para melhor compreensão, iremos explicar apenas os passos que achamos mais relavantes (todos aqueles em que for apenas necessário introduzir simples elementos não serão aqui mencionados).

Começamos então no caso primário, onde atualizamos o ficheiro *primario/named.conf*. Neste caso colocamos a zona *cc.pt* como enunciado. Dado que a topologia usada como objeto de estudo possui 4 redes LAN distintas, acrescentamos ainda mais 4 zonas: *1.1.10.in-addr.arpa*, *2.2.10.in-addr.arpa*, *3.3.10.in-addr.arpa* e *4.4.10.in-addr.arpa*. Ficamos então, com um total de 5 zonas definidas.

Para além de acrescentar as zonas, foi necessário definir certos aspetos dentro das mesmas. Todas elas ficaram com o *type master* dado que estamos a operar em zonas do servidor DNS principal/primário. Adicionamos ainda a cláusula *allow-transfer10.2.2.2;* de modo a permitir a transferência da base de dados ao servidor secundáio. Para além disso, foram ainda atualizadas as diretorias e os ficheiros onde os dados de cada zona devem ser e/ou se encontram armazenados.

```
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 "4.4.10.in-addr.arpa" {
        type master;
        file "/home/core/primario/db.4-4-10.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 "2.2.10.in-addr.arpa" {
        type master;
        file "/home/core/primario/db.2-2-10.rev";
        allow-transfer {10.2.2.2;};
};
```

Figure 18:

Seguidamente passamos à configuração das zonas, começando pela *cc.pt*. Decidimos no nosso caso utilizar o *ns.cc.pt* como posição de DNS principal para esta zona em estudo, dado ser o servidor principal/mais superior. Respeitando as regras enunciadas, decidimos utilizar como administrador do domínio *PL06G14.cc.pt*.

Seguidamente, e guiando-nos pelo ficheiro *db.local*, colocamos os *nameservers* em estudo através do type NS. Seguidamente, colocamos os servidores de e-mail pedidos no enunciado, utilizando agora o tipo *MX*. É importante perceber nesta última parte que o número que segue o *MX* clarifica a prioridade, ou seja, quanto menor o número, maior a prioridade. Assim, o que possuir um menor número será o servidor de e-mail principal, e o outro, o servidor de e-mail secundário.

Posto isto, passamos a introduzir todos os elementos da Topologia em estudo, com o seu endereço, recorrendo agora ao tipo A. Faremos então o seguinte passo para todos os elementos:

```
(Elemento da Topologia) IN A (IP Address)
```

Neste momento, precisamos ainda de introduzir os serviços fornecidos. Assim, utilizaremos um passo análogo ao anterior de modo a mapearmos todos os seviços utilizando para isso o seguinte passo:

```
(Nome do elemento no Serviço) IN A (IP Adress)
```

No meio ainda é ainda definido um *alias* pedido no enunciado para com o *Laptop1.cc.pt* e *g14.cc.pt*, através do *CNAME*.

Posto isto, ficamos com o seguinte resultado final:

```
$TTL    604800
@       IN      SOA     ns.cc.pt        PL06G14.cc.pt. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;

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

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

Server1 IN      A       10.1.1.1
Server2 IN      A       10.1.1.2
Server3 IN      A       10.1.1.3

Marte   IN      A       10.2.2.1
Mercurio IN     A       10.2.2.2
Venus   IN      A       10.2.2.3

Pico    IN      A       10.3.3.1
Corvo   IN      A       10.3.3.2
Faial   IN      A       10.3.3.3

Laptop1 IN      A       10.4.4.1
g14     IN      CNAME   Laptop1
Laptop2 IN      A       10.4.4.2
Laptop3 IN      A       10.4.4.3
```

Figure 19:

Estando configurado o ficheiro *primario/db.cc.pt*, passamos agora a definir os 4 ficheiros do domínio reverse. Todos foram feitos de maneira análoga, utilizando o mesmo SOA que em *primario/db.cc.pt*.

Aqui, para todos os elmentos de uma LAN adicionamos uma entrada na base de dados, através do tipo PTR, utilizando o seguinte passo:

`(Interface da Rede) IN PTR (Elemento da Topologia).cc.pt`

Ficamos com algo como no seguinte exemplo:

```
$TTL    604800
@       IN      SOA     ns.cc.pt        PL06G14.cc.pt. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.cc.pt.
@       IN      NS      ns2.cc.pt.

1       IN      PTR     Marte.cc.pt.
1       IN      PTR     ns2.cc.pt.
2       IN      PTR     Mercurio.cc.pt.
3       IN      PTR     Venus.cc.pt.
```

Figure 20:

Ficando assim concluida a configuração do servidor DNS primário/principal, passamos à configuração do servidor DNS secundário, onde apenas apresentamos no ficheiro *secundario/named.conf* as zonas que indicamos no servidor primário. Estas zonas, no entanto, vão ter algumas definições diferentes. O seu tipo passa agora a ser *slave*, a cláusula *allow-transfer* foi substituida por *masters 10.1.1.1;* e os ficheiros/diretorias de destino foram ainda substituídas. Ficamos então com a seguinte configuração final:

```
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 "4.4.10.in-addr.arpa" {
        type slave;
        file "/var/cache/bind/db.4-4-10.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 "2.2.10.in-addr.arpa" {
        type slave;
        file "/var/cache/bind/db.2-2-10.rev";
        masters {10.1.1.1;};
};
```

Figure 21:

3 Conclusões

Após a finalização desta fase do trabalho prático da UC de Comunicações por Computador, atribuímos um balanço positivo ao desempenho do grupo.

Este projeto permitiu uma aplicação produtiva dos conhecimentos adquiridos nas aulas e, também se revelou um desafio, na medida em que nos forçou a expandir os nossos horizontes extra-aula.

O resultado final consiste na apresentação de ambos os servidores propostos, primário e secundário, totalmente funcionais. Continuamos focados no desenvolimento da fase dois do trabalho, com vista a apresentar bons resultados.