



Universidade do Minho
Escola de Engenharia

Comunicações por Computador

Trabalho Prático 3

Luís Pedro Oliveira de Castro Vieira A89601
José Pedro de Castro Ferreira A89572
Luís Enes Sousa A89597



A89601



A89572



A89597

Conteúdo

1	Parte I	1
1.1	Questões e Respostas	1
1.1.1	Pergunta A	1
1.1.2	Pergunta B	1
1.1.3	Pergunta C	2
1.1.4	Pergunta D	3
1.1.5	Pergunta E	3
1.1.6	Pergunta F	4
1.1.7	Pergunta G	4
1.1.8	Pergunta H	5
1.1.9	Pergunta I	5
1.1.10	Pergunta J	6
2	Parte II	8
2.1	Domínio CC.PT	8
2.2	Demonstração	11
3	Conclusão	12

1 Parte I

1.1 Questões e Respostas

1.1.1 Pergunta A

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

O ficheiro `/etc/resolv.conf` contém informação sobre o nome do servidor DNS local e respetivos IPs para os servidores associados. Esta informação é variável, uma vez que depende da rede em que o host se encontra. De notar que, quando um utilizador quer aceder a um domínio, o nome do servidor é o primeiro a ser interrogado, procurando pelo mesmo nos registos.

```
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
```

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

1.1.2 Pergunta B

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

Apenas o servidor `www.ubuntu.com` possui endereços IPv6, sendo eles:

- `www.ubuntu.com` - 2001:67c:1360:8001::2c
- `www.ubuntu.com` - 2001:67c:1360:8001::2b

```
core@core-VirtualBox:~$ nslookup
> set q=AAAA
> www.uminho.pt.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
*** Can't find www.uminho.pt.: No answer
> www.ubuntu.com.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.ubuntu.com
Address: 2001:67c:1360:8001::2c
Name:   www.ubuntu.com
Address: 2001:67c:1360:8001::2b
```

Figura 2: Endereços IPv6 do servidor `www.ubuntu.com`

```
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 91.189.88.181
www.ubuntu.com has address 91.189.88.180
www.ubuntu.com has IPv6 address 2001:67c:1360:8001::2c
www.ubuntu.com has IPv6 address 2001:67c:1360:8001::2b
```

Figura 3: Endereços IPv6 do servidor `www.uminho.pt`

1.1.3 Pergunta 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`, `ns2.sapo.pt`, `dns01.sapo.pt`, `dns02.sapo.pt`.

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

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

Figura 4: Servidores associados ao domínio ‘`sapo.pt.`’

Já ao domínio `pt.` estão associados 9 servidores, sendo eles:

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

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

Figura 5: Servidores associados ao domínio ‘`pt.`’

Por fim, relativamente ao domínio `.` estão definidos 13 servidores.

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

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

Figura 6: Servidores associados ao domínio ‘`.`’

1.1.4 Pergunta D

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

Sim, o domínio *open.money* existe, uma vez que existe correspondência para este. No entanto não se trata de um host mas sim de um serviço de email. Isto pode ser comprovado através do uso do comando *host open.money*, sendo que o resultado do mesmo nos indica que o domínio se trata de um *alias* para um servidor de email.

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

Figura 7: Execução do comando "host *open.money*."

1.1.5 Pergunta 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 definido para o domínio *un.org* é o servidor *ns1.un.org*. Conseguimos confirmar esta informação ao analisar o campo *origin*.

```
> set q=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:
```

Figura 8: Execução do comando "nslookup" com uma querye do tipo *SOA* a *un.org*.

Quando executamos o comando *dig un.org*, obtemos as seguintes informações:

```
core@core-VirtualBox:/var/cache/bind$ dig ns1.un.org  
  
;<<> Dig 9.16.1-Ubuntu <<> ns1.un.org  
;; global options: +cmd  
;; Got answer:  
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 22102  
;; 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. 249 IN A 157.150.185.28  
  
;; Query time: 56 msec  
;; SERVER: 127.0.0.53#53(127.0.0.53)  
;; WHEN: ter mai 04 20:07:15 WEST 2021  
;; MSG SIZE rcvd: 55
```

Figura 9: Execução do comando "dig *un.org*."

A partir daqui podemos afirmar que este servidor primário aceita queries recursivas uma vez que possui a flag *ra*, 'Recursion Available'.

1.1.6 Pergunta F

Obtenha uma resposta “autoritativa” para a questão anterior.

Com o intuito de obter uma resposta “autoritativa”, executamos o comando “nslookup” com uma query do tipo *SOA*, obtendo o seguinte resultado:

```
core@core-VirtualBox:~$ nslookup
> set q=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 = 2021042900
    refresh = 1200
    retry = 3600
    expire = 1209600
    minimum = 300

Authoritative answers can be found from:
>
```

Figura 10: Comando “nslookup” com uma query do tipo SOA a “un.org.”

O passo seguinte seria questionar os servidores autoritativos obtidos, que sabemos ser, pelo menos um, *ns1.un.org*, executando “server ns1.un.org” dentro do “nslookup” e procurando novamente por “un.org”. No entanto devido às medidas de segurança da rede, não é possível realizar o mesmo pois estaria a pôr em risco os servidores existentes.

O expectado seria que, ao questionar novamente por “un.org” tendo como servidor “ns1.un.org” obtivéssemos uma resposta do tipo “*connection timed out*” uma vez que apenas um servidor local permite responder de forma recursiva ao cliente.

1.1.7 Pergunta G

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

Para obter a informação acerca do destino das mensagens de correio eletrónico temos que executar os comandos descritos abaixo:

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

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

Authoritative answers can be found from:
>
```

Figura 11: Comando “nslookup” como uma query do tipo MX

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

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

Authoritative answers can be found from:
>
```

Figura 12: Comando “nslookup” como uma query do tipo MX

Assim sendo, sabemos que as mensagens enviadas a *presidencia@2021portugal.eu*. irão ser entregues em *mxg.eu.mpssec.net*..


```

core@core-VirtualBox:~$ nslookup
> set q=soa
> fccn.pt.
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
fccn.pt
  origin = ns01.fccn.pt
  mail addr = hostmaster.fccn.pt
  serial = 2021041901
  refresh = 21600
  retry = 7200
  expire = 1209600
  minimum = 300

Authoritative answers can be found from:
> set q=mx
> fccn.pt
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
fccn.pt mail exchanger = 10 smtp02.fccn.pt.
fccn.pt mail exchanger = 10 smtp01.fccn.pt.

```

Figura 15: Comando "nslookup" com uma query do tipo SOA

Avaliando então o resultado obtido, podemos concluir que o responsável pelo endereço IPv6 apresentado no enunciado é *ns01.fccn.pt*.

1.1.10 Pergunta J

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

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

Isto é, um cliente solicita uma transferência de dados de um servidor primário para um secundário, sendo a parte da base de dados replicada designada por zona.

Através dos parâmetros definidos no registo do tipo SOA, e fazendo uso do domínio *cc.pt* criado na topologia virtual, podemos visualizar as seguintes informações:

```

;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA     Server1.cc.pt. PL04G20.cc.pt. (
                                3          ; Serial
                                604800    ; Refresh
                                86400     ; Retry
                                2419200   ; Expire
                                604800    ; Negative Cache TTL
                                )

```

Figura 16: Conteúdo do registo do tipo SOA *bd.cc.pt*

- **Serial** - Corresponde ao número de série da zona. Se o servidor secundário associado a este verificar uma alteração neste, assume que a zona está desatualizada e inicia-se uma transferência de zona.
- **Refresh** - Número de segundos após o qual o servidor secundário contactará o servidor primário para atualizar informações, de modo a detetar possíveis alterações na zona.

- **Retry** - Corresponde ao número de segundos que o servidor secundário deve esperar após uma falha para se reconectar novamente ao servidor primário, sendo que este deve ser sempre inferior ao valor de *refresh*.
- **Expire** - Número de segundos após o qual o servidor secundário deve parar de fazer solicitações para a zona específica, caso o servidor primário não responda. Este valor deve ser superior à soma do valor do *refresh* e do *retry*.

Em suma, o servidor secundário deverá contactar o primário para se atualizar após 604800 segundos terem passado; deve esperar 86400 segundos até poder tentar uma nova conexão com o servidor primário após ter falhado inicialmente; caso o servidor primário não responda, até 2419200 segundos depois, o servidor secundário deixa de tentar conectar-se.

2 Parte II

2.1 Domínio CC.PT

Durante a execução de grande parte das instruções presentes no enunciado, limitámo-nos a seguir as mesmas em diversas alturas uma vez que não havia mais alternativas relativamente às mesmas. No entanto houve vezes em que o grupo teve de tomar certas decisões e refletir sobre as mesmas, e estas serão os principais focos de discussão e explicação no presente relatório.

A primeira decisão vem no seguimento da instrução presente no enunciado referente ao ficheiro *"named.conf"*. Neste ficámos de incluir as diferentes zonas presentes na topologia core apresentada para o ano de 2021. No entanto no enunciado apenas fazia menção à zona *"cc.pt"* e à zona *"1.1.10.in-addr.arpa"*, mas a topologia apresenta 4 redes LAN diferentes, o que levou o grupo a incluir também as zonas de procura inversa para as redes que faltavam, nomeadamente: *"2.2.10.in-addr.arpa"*, *"3.3.10.in-addr.arpa"* e *"4.4.10.in-addr.arpa"*. O ficheiro ficou assim com 5 zonas.

Em cada zona definida no ficheiro *"named.conf"* tivemos que definir as mesmas com o tipo *master*, uma vez que fazem parte do servidor DNS principal. Tivemos também que adicionar uma cláusula *"allow-transfer {10.2.2.2;}"* que permitisse ao servidor secundário transferir e guardar a informação sobre o servidor primário na sua 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 17: Conteúdo do ficheiro *named.conf*

Depois de termos as zonas definidas e criadas seguiu-se a criação e configuração do ficheiro *"db.cc.pt"*. O primeiro passo neste processo foi a configuração do SOA (Start of Authority), para o qual escolhemos *Server1.cc.pt* para a posição do DNS principal, visto se tratar do servidor principal, e como administrador, como referido no enunciado, estabelecemos que seria *PL04G20.cc.pt*.

```

;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      Server1.cc.pt. PL04G20.cc.pt. (
                                3          ; Serial
                                604800    ; Refresh
                                86400     ; Retry
                                2419200   ; Expire
                                604800    ; Negative Cache TTL
                                )

```

Figura 18: Configuração SOA do ficheiro *db.cc.pt*

Após termos estabelecido o SOA, resta-nos introduzir o resto da informação necessária para o bom funcionamento do nosso servidor DNS. Começámos por introduzir os *nameservers*, nomeadamente *Server1* e *Mercurio* fazendo uso da cláusula *NS* e os servidores de email *Server2*, como prioritário, e *Server3.cc.pt* como secundário, como referido no enunciado, desta vez com a cláusula *MX*.

De seguida, para todos os elementos inserimos os seus nomes mencionados na topologia bem como o seu endereço IP, fazendo uso da cláusula *A*, resolvendo assim os requisitos Marte.cc.pt, Mercurio.cc.pt e Venus.cc.pt. Adicionamos ainda *alias* para alguns dos elementos como requisitado no enunciado, tais como *ns* e *ns2* para os servidores primário e secundário respetivamente.

Foi ainda necessário definir os serviços requisitados no enunciado como o servidor web e servidor e-mail, presentes em **Server2**, e servidor pop e imap, presentes em *Server3*.

```

@         IN      NS       Server1
@         IN      NS       Mercurio
@         IN      MX       10      Server2
@         IN      MX       20      Server3.cc.pt.

Server1   IN      A        10.1.1.1
ns        IN      CNAME     Server1
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
ns2       IN      CNAME     Mercurio
Venus     IN      A        10.2.2.3
mail      IN      CNAME     Server2
pop       IN      CNAME     Server3
imap      IN      CNAME     Server3
www       IN      CNAME     Server2
Laptop1   IN      A        10.4.4.1
g20.cc.pt. IN      CNAME     Laptop1
Laptop2   IN      A        10.4.4.2
Laptop3   IN      A        10.4.4.3

```

Figura 19: Configuração da base de dados cc.pt

Após o término da configuração da base de dados cc.pt, procedemos à configuração dos restantes ficheiros que permitirão a procura inversa. E, embora havendo 4 redes diferentes, o processo é semelhante para todos, acabando por ter a mesma configuração SOA e a adição dos dois *nameservers*: *Server1* e *Mercurio*.

Sendo assim, resta-nos apenas introduzir o conhecimento necessário para podermos fazer o *reverse mapping*, o qual fazemos introduzindo o endereço da máquina em questão e o seu nome, acompanhados da cláusula *PTR*.

```

;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      Server1.cc.pt.  PL04G20.cc.pt. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       Server1.cc.pt.
@         IN      NS       Mercurio.cc.pt.

1         IN      PTR      Pico.cc.pt.
2         IN      PTR      Faial.cc.pt.
3         IN      PTR      Corvo.cc.pt.

```

Figura 20: Exemplo do reverse mapping para a rede 10.3.3.0/24

Com a conclusão da configuração das bases de dados para os domínios reversos, podemos dar por concluída a configuração do servidor DNS principal.

Na configuração do servidor secundário foi apenas necessário alterar o ficheiro *named.conf*, adicionando as zonas existentes no servidor primário com algumas alterações nas cláusulas previamente apresentadas. O *type* que antes era *master* agora passa a ser *slave*, e a cláusula *allow-transfer* foi substituída pela cláusula *masters {10.1.1.1;}*, sendo também o path para o ficheiro atualizado.

```

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

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

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

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

zone "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 21: Configuração do ficheiro *named.conf* do servidor secundário

2.2 Demonstração

Abaixo seguem exemplos da demonstração de comandos que comprovam a funcionalidade na totalidade dos servidores, quer primário, quer secundário.

```
root@Laptop1:/tmp/pycore.40163/Laptop1.conf# nslookup - 10.1.1.1
> set q=soa
> cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

cc.pt
    origin = Server1.cc.pt
    mail addr = PL04G20.cc.pt
    serial = 3
    refresh = 604800
    retry = 86400
    expire = 2419200
    minimum = 604800
> set q=ptr
> 10.4.4.3
Server:      10.1.1.1
Address:     10.1.1.1#53

3.4.4.10.in-addr.arpa  name = Laptop3.cc.pt.
> set q=ptr
> g20.cc.pt
Server:      10.1.1.1
Address:     10.1.1.1#53

g20.cc.pt      canonical name = Laptop1.cc.pt.
```

Figura 22: Demonstração do funcionamento do servidor primário

```
> server 10.2.2.2
Default server: 10.2.2.2
Address: 10.2.2.2#53
> set q=soa
> cc.pt
Server:      10.2.2.2
Address:     10.2.2.2#53

cc.pt
    origin = Server1.cc.pt
    mail addr = PL04G20.cc.pt
    serial = 3
    refresh = 604800
    retry = 86400
    expire = 2419200
    minimum = 604800
> set q=ptr
> 10.3.3.1
Server:      10.2.2.2
Address:     10.2.2.2#53

1.3.3.10.in-addr.arpa  name = Pico.cc.pt.
```

Figura 23: Demonstração do funcionamento do servidor secundário

3 Conclusão

Tendo terminado o terceiro trabalho prático da Unidade Curricular de Comunicações por Computador, conseguimos afirmar com certeza que fomos capazes de aprofundar em pleno o nosso conhecimento, pondo em prática os conteúdos lecionados nas aulas teóricas.

Assim sendo, somos capazes de apresentar como resultado final deste trabalho, ambos os servidores, primário e secundário, funcionais na sua capacidade total. Apesar das dificuldades encontradas durante a resolução do projeto, o grupo conseguiu superá-las e todas as indicações propostas foram completadas com sucesso, apresentando um trabalho consistente e devidamente estruturado.