



**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

Licenciatura em Engenharia Informática

## **Comunicações por Computador**

Ano Letivo de 2022/2023

### **TP2 Parte A - Implementação de um sistema DNS**

Grupo 7

Orlando José da Cunha Palmeira, A97755

16 de novembro de 2022

cc

## **TP2 Parte A - Implementação de um sistema DNS**

Orlando José da Cunha Palmeira, A97755

16 de novembro de 2022

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Arquitectura do sistema</b>	<b>2</b>
<b>3</b>	<b>Modelo comunicativo do sistema</b>	<b>5</b>
3.1	Transferência de Zona . . . . .	5
3.1.1	Cenário completo . . . . .	5
3.1.2	Cenário de recusa da transferência de zona . . . . .	7
3.1.3	Cenário de transferência de zona desnecessária . . . . .	8
3.1.4	Cenário de ocorrência de erros . . . . .	8
3.1.5	Especificação dos PDU's utilizados na transferência de zona . . . . .	9
3.2	<i>Queries</i> . . . . .	9
3.2.1	Transacções de <i>queries</i> entre os componentes . . . . .	9
3.2.2	Cenário de pedido cliente-servidor primário/secundário . . . . .	9
3.2.3	Cenário de pedido cliente-servidor de resolução (iterativo) . . . . .	10
3.2.4	Cenário de pedido cliente-servidor de resolução (recursivo) . . . . .	11
3.2.5	Especificação dos PDU's utilizados nas <i>Queries</i> . . . . .	12
<b>4</b>	<b>Implementação dos elementos</b>	<b>15</b>
4.1	Cliente . . . . .	15
4.1.1	O seu funcionamento . . . . .	15
4.1.2	O modo <i>debug</i> . . . . .	15
4.2	Servidor Primário . . . . .	15
4.2.1	O seu funcionamento . . . . .	16
4.2.2	O modo <i>debug</i> . . . . .	16
4.2.3	Actividades reportadas nos <i>logs</i> . . . . .	16
4.3	Servidor Secundário . . . . .	16
4.3.1	O seu funcionamento . . . . .	17
4.3.2	O modo <i>debug</i> . . . . .	17
4.3.3	Actividades reportadas nos <i>logs</i> . . . . .	17
<b>5</b>	<b>Análise de testes do sistema</b>	<b>18</b>
	<b>Anexos</b>	<b>29</b>
	Anexo 1 - Manual de utilização do cliente . . . . .	29
	Anexo 2 - Manual de utilização do servidor . . . . .	29
	Anexo 3 - Conteúdo do ficheiro de configuração do servidor de topo primário . . . . .	30

Anexo 4 - Conteúdo do ficheiro de configuração do servidor de topo secundário . . .	30
Anexo 5 - Conteúdo do ficheiro de configuração do servidor de domínio de topo primário do domínio .xpto . . . . .	31
Anexo 6 - Conteúdo do ficheiro de configuração do servidor de domínio de topo secundário do domínio .xpto . . . . .	31
Anexo 7 - Conteúdo do ficheiro de configuração do servidor de domínio de topo primário do domínio .otpx . . . . .	32
Anexo 8 - Conteúdo do ficheiro de configuração do servidor de domínio de topo secundário do domínio .otpx . . . . .	32
Anexo 9 - Conteúdo do ficheiro de configuração do servidor primário do domínio abcd.xpto . . . . .	33
Anexo 10 - Conteúdo do ficheiro de configuração do primeiro servidor secundário do domínio abcd.xpto . . . . .	33
Anexo 11 - Conteúdo do ficheiro de configuração do segundo servidor secundário do domínio abcd.xpto . . . . .	34
Anexo 12 - Conteúdo do ficheiro de configuração do servidor primário do domínio dcba.otpx . . . . .	34
Anexo 13 - Conteúdo do ficheiro de configuração do primeiro servidor secundário do domínio dcba.otpx . . . . .	35
Anexo 14 - Conteúdo do ficheiro de configuração do segundo servidor secundário do domínio dcba.otpx . . . . .	35
Anexo 15 - Conteúdo do ficheiro de base de dados do servidor de topo primário . .	36
Anexo 16 - Conteúdo do ficheiro de base de dados do servidor de domínio de topo primário do domínio .xpto . . . . .	37
Anexo 17 - Conteúdo do ficheiro de base de dados do servidor de domínio de topo primário do domínio .otpx . . . . .	38
Anexo 18 - Conteúdo do ficheiro de base de dados do servidor primário do domínio abcd.xpto . . . . .	39
Anexo 19 - Conteúdo do ficheiro de base de dados do servidor primário do domínio dcba.otpx . . . . .	40

# Lista de Figuras

2.1	Topologia utilizada na implementação do sistema . . . . .	2
2.2	Esquema hierárquico dos domínios do sistema . . . . .	2
3.1	Transferência de zona no cenário completo . . . . .	5
3.2	Transferência de zona no cenário de recusa . . . . .	7
3.3	Transferência de zona desnecessária . . . . .	8
3.4	Pedido cliente-servidor primário/secundário . . . . .	9
3.5	Pedido cliente-servidor de resolução . . . . .	10
3.6	Pedido cliente-servidor de resolução . . . . .	11
5.1	Transferência de zona - Servidor Primário . . . . .	19
5.2	Transferência de zona - Servidor Secundário . . . . .	19
5.3	<i>Query</i> ao servidor primário - Cliente . . . . .	20
5.4	<i>Query</i> ao servidor primário - Servidor Primário . . . . .	20
5.5	<i>Query</i> ao servidor secundário - Cliente . . . . .	21
5.6	<i>Query</i> ao servidor secundário - Servidor Secundário . . . . .	21
5.7	Transferência de zona - Servidor Primário . . . . .	22
5.8	Transferência de zona - Servidor Secundário . . . . .	22
5.9	<i>Query</i> ao servidor primário - Cliente . . . . .	23
5.10	<i>Query</i> ao servidor primário - Servidor Primário . . . . .	23
5.11	<i>Query</i> ao servidor secundário - Cliente . . . . .	24
5.12	<i>Query</i> ao servidor secundário - Servidor Secundário . . . . .	24
5.13	Transferência de zona - Servidor Primário . . . . .	25
5.14	Transferência de zona - Servidor Secundário . . . . .	25
5.15	<i>Query</i> ao servidor primário - Cliente . . . . .	26
5.16	<i>Query</i> ao servidor primário - Servidor Primário . . . . .	26
5.17	<i>Query</i> ao servidor secundário - Cliente . . . . .	27
5.18	<i>Query</i> ao servidor secundário - Servidor Secundário . . . . .	27
5.19	Primeira versão da base de dados . . . . .	28
5.20	Segunda versão da base de dados . . . . .	28

# Lista de Tabelas

2.1	Funcionalidades de cada componente da topologia . . . . .	3
-----	---	---

# 1 Introdução

No âmbito da unidade curricular de Comunicações por Computador, foi proposto que se desenvolvesse um pequeno sistema DNS.

Neste relatório são abordados tópicos do enunciado que foram implementados para esta primeira fase bem como algumas propostas de implementação de tópicos da segunda fase. Para a primeira fase, está implementado o cliente, os servidores primários e os servidores secundários. O cliente consegue enviar pedidos aos servidores primários e secundários e os servidores secundários conseguem efectuar pedidos de transferência de zona aos servidores primários. Quanto aos tópicos da segunda fase, é abordado um possível funcionamento do servidor de resolução, é apresentada uma proposta de codificação binária para as *queries* e são explicados os possíveis modos de comunicação entre os servidores (iterativo ou recursivo).





Cada um dos elementos da topologia exerce um papel sobre o seu domínio, se aplicável. As funcionalidades de cada componente são apresentadas a seguir.

(A designação de cada componente é a que foi utilizada na topologia da figura 2.1)

Componente	Domínio	Papel desempenhado
client	Não aplicável	Enviar e receber <i>queries</i> dos servidores.
STp	root	Servidor de topo primário
STs	root	Servidor de topo secundário
SP-XPTO	.xpto	Servidor de domínio de topo primário
SS-XPTO	.xpto	Servidor de domínio de topo secundário
SP-OTPX	.otpx	Servidor de domínio de topo primário
SS-OTPX	.otpx	Servidor de domínio de topo secundário
SP-ABCD-XPTO	abcd.xpto	Servidor primário
SSA-ABCD-XPTO	abcd.xpto	Servidor secundário
SSB-ABCD-XPTO	abcd.xpto	Servidor secundário
SP-DCBA-OTPX	dcba.otpx	Servidor primário
SSA-DCBA-OTPX	dcba.otpx	Servidor secundário
SSB-DCBA-OTPX	dcba.otpx	Servidor secundário
SR-ABCD-XPTO	abcd.xpto	Servidor de resolução
SP-reverse	.reverse	Servidor primário do domínio de DNS reverso

Tabela 2.1: Funcionalidades de cada componente da topologia

O cliente deve apenas enviar *queries* (e receber as respostas das mesmas) aos servidores, tendo em conta que é da sua responsabilidade enviar um nome correcto para um servidor correcto sob "pena" de nunca receber uma resposta se isso não acontecer. Se a *query* for correcta, então o cliente apresenta os respectivos resultados obtidos através do servidor que contactou.

Para evitar uma situação como a que foi descrita anteriormente, existe o servidor de resolução (DNS *resolver*). O servidor de resolução deve ser o contacto preferencial do cliente uma vez que ele (i) vai ter a informação que o cliente pretende na sua cache, aumentando o desempenho, ou (ii) vai fazer um contacto iterativo (ou recursivo) com todos os servidores para obter a resposta desejada e não fará apenas um contacto, tal como o cliente faria.

Os servidores de topo possuem a informação acerca do domínio *root* bem como dos domínios de topo. Este tipo de servidores é contactado maioritariamente pelos servidores de resolução para obter respostas a pedidos dos clientes. Neste trabalho existirão dois servidores de topo, um primário e um secundário. Deste modo será possível fazer com que a base de dados seja transferida automaticamente do servidor primário para o secundário sem necessidade de leitura do ficheiro em cada servidor.

Os servidores de domínios de topo serão responsáveis pelos domínios .xpto e .otpx (domínios de topo no contexto deste trabalho) e possuem informações acerca dos servidores autoritativos respectivos subdomínios (abcd.xpto e dcba.otpx). Tal como nos servidores de topo, para cada

domínio de topo existirão dois servidores, um primário e um secundário. Estes servidores são muito semelhantes aos servidores de topo.

Os servidores primários "normais", isto é, responsáveis pelos subdomínios dos domínios de topo, terão informações e irão responder a *queries* acerca do seu domínio, bem como proporcionar aos respectivos servidores secundários o serviço de transferência de zona (a transferência de zona é explicada mais à frente).

Os servidores secundários, tal como os primários, terão informações e irão responder a *queries* acerca do seu domínio. Estes servidores, no seu arranque, devem pedir de imediato ao seu servidor primário a transferência de zona por forma a poder ter acesso à base de dados para poder responder a *queries* sobre o seu domínio, auxiliando o servidor primário.

O processo de transferência de zona ocorre em duas situações: (i) no arranque de um servidor secundário e (ii) quando o tempo de vida da réplica da base de dados de um servidor secundário (*SOAEXPIRE*) termina.

A transferência de zona é sempre solicitada por um servidor secundário ao servidor primário. No início do processo, o servidor efectua o pedido (pede o número de série) ao servidor primário e este último aprova, ou não, o respectivo pedido, dependendo das características do servidor que fez a solicitação. Se o pedido for aceite pelo servidor primário, este envia ao servidor secundário o número de entradas da sua base de dados. Após o servidor secundário aceitar o número de entradas informado, inicia a transferência de zona propriamente dita, isto é, o servidor primário envia uma entrada devidamente indexada de cada vez e, a cada envio, o servidor secundário verifica se o índice da entrada recebida é o esperado. Este processo repete-se até ao servidor primário informar que a transferência de zona terminou.

No caso em que o servidor secundário recebe uma entrada cujo índice não é o expectável, este desiste da transferência de zona, fecha a conexão TCP com o servidor primário e aguarda durante um intervalo de tempo (*SOARETRY*) para voltar a tentar efectuar a transferência de zona.

O servidor de DNS reverso é responsável pelo domínio *.reverse*. Neste domínio é possível obter informações dos nomes dos domínios "normais" (todos os que não estão associados ao *.reverse*) através dos endereços IP dos seus servidores.

O servidor de DNS reverso funciona da mesma maneira que qualquer outro servidor. A única diferença é que, nas *queries* que ele recebe, irá um endereço IP de um servidor de um certo domínio e ele responde com o respectivo nome associado ao IP.

Neste sistema, o cliente comunica com qualquer tipo de servidor enviando datagramas UDP. Os servidores podem comunicar entre si com uma conexão TCP, mas apenas na transferência de zona. No caso de *queries*, os servidores comunicam uns com os outros com UDP.

Nota: Todos os ficheiros de configuração e bases de dados encontram-se nos anexos deste relatório.

## 3 Modelo comunicativo do sistema

Neste capítulo são abordados os diversos casos de interação entre componentes bem como é explicada a estrutura dos pacotes enviados entre cada componente.

### 3.1 Transferência de Zona

#### 3.1.1 Cenário completo

O cenário completo é aquele em que a transferência de zona é efectivamente feita, isto é, o servidor secundário está autorizado a fazer a solicitação, a versão da base de dados do servidor secundário é mais antiga do que a do servidor primário (ou pode ser também que o servidor secundário esteja a arrancar) e não há ocorrência de erros na transmissão.

Apresenta-se abaixo um esquema da comunicação entre o servidor primário e secundário neste cenário.

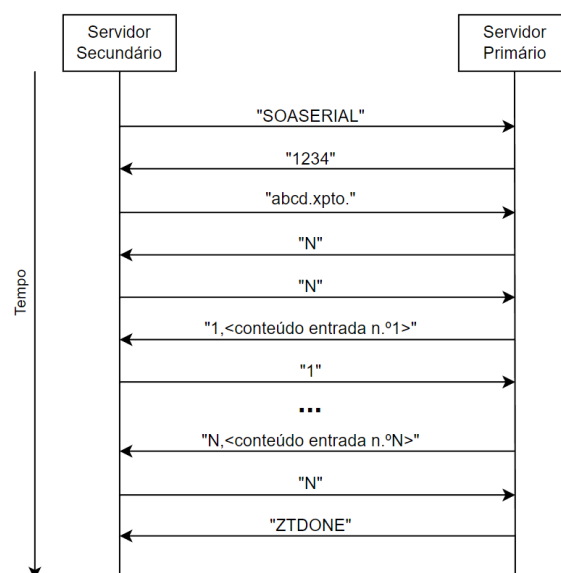


Figura 3.1: Transferência de zona no cenário completo

As interações apresentadas no esquema da figura 3.1 podem ser explicadas do seguinte modo:

1. O servidor secundário envia a *string* "SOASERIAL" ao servidor primário, indicando que pretende saber a sua versão actual da base de dados.
2. O servidor primário responde enviando o número de série da sua base de dados.
3. O servidor secundário verifica que a versão que tem é mais antiga do que a que o servidor primário lhe comunicou. Deste modo, o servidor secundário irá solicitar ao servidor primário que lhe envie uma réplica da base de dados indicando-lhe o domínio que quer receber.
4. O servidor primário verifica que o domínio é válido e que o servidor que efectuou o pedido está autorizado a receber a réplica. Assim, o servidor primário responde com o número de entradas que a sua base de dados possui.
5. O servidor secundário aceita receber esse número de entradas, pelo que responde ao servidor primário com o mesmo número.
6. O servidor primário envia uma entrada com o respectivo índice.
7. O servidor primário verifica que o índice da entrada recebida é o esperado e informa o servidor secundário enviando-lhe o índice recebido.
8. Os dois itens anteriores repetem-se até não haver mais entradas a enviar.
9. O servidor primário verifica que chegou ao fim das entradas da base de dados e informa o servidor secundário dessa situação enviando-lhe a *string* "ZTDONE"

### 3.1.2 Cenário de recusa da transferência de zona

O cenário de recusa da transferência de zona aborda o caso em que o servidor primário recusa fornecer a um servidor secundário a réplica da sua base de dados. Isto acontece porque o servidor secundário não tem autorização para receber a réplica ou porque o domínio que o servidor secundário pretende receber é inválido/inexistente. Apresenta-se abaixo um esquema da comunicação entre o servidor primário e secundário neste cenário.

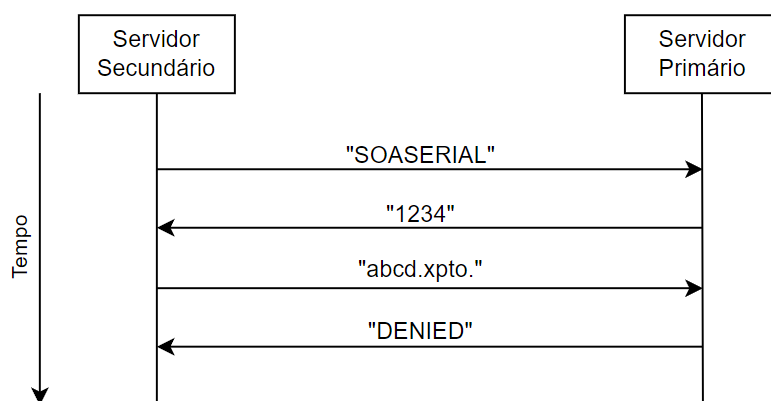


Figura 3.2: Transferência de zona no cenário de recusa

As interações apresentadas no esquema da figura 3.2 podem ser explicadas do seguinte modo:

1. O servidor secundário envia a *string* "SOASERIAL" ao servidor primário, indicando que pretende saber a sua versão actual da base de dados.
2. O servidor primário responde enviando o número de série da sua base de dados.
3. O servidor secundário verifica que a versão que tem é mais antiga do que a que o servidor primário lhe comunicou. Deste modo, o servidor secundário irá solicitar ao servidor primário que lhe envie uma réplica da base de dados indicando-lhe o domínio que quer receber.
4. O servidor primário verifica que ocorreu uma das seguintes situações: (i) o servidor secundário não está autorizado a receber a réplica da base de dados ou (ii) o domínio pretendido pelo servidor secundário é inválido. Deste modo, o servidor primário recusa a transferência de zona enviando ao servidor secundário a *string* "DENIED".

### 3.1.3 Cenário de transferência de zona desnecessária

O cenário de transferência de zona desnecessária aborda o caso em que o servidor secundário verifica que a versão da sua base de dados é igual à do servidor primário. Apresenta-se abaixo um esquema da comunicação entre o servidor primário e secundário neste cenário.

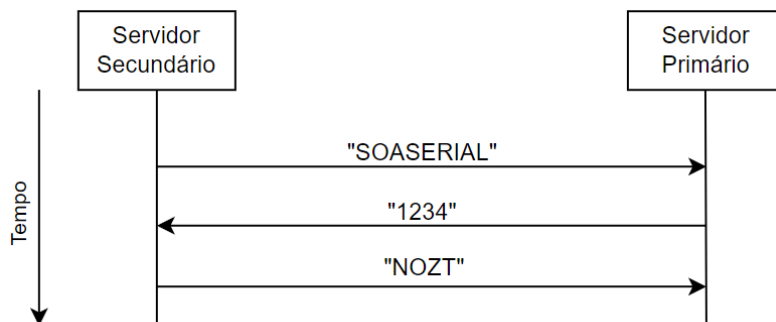


Figura 3.3: Transferência de zona desnecessária

As interações apresentadas no esquema da figura 3.3 podem ser explicadas do seguinte modo:

1. O servidor secundário envia a *string* "SOASERIAL" ao servidor primário, indicando que pretende saber a sua versão actual da base de dados.
2. O servidor primário responde enviando o número de série da sua base de dados.
3. O servidor secundário verifica que a sua versão da base de dados é igual à que o servidor primário lhe comunicou. Deste modo, não é necessário fazer a transferência de zona e o servidor secundário envia ao servidor primário a *string* "NOZT"

### 3.1.4 Cenário de ocorrência de erros

Na transferência de zona podem ocorrer diversos erros (rede, mau funcionamento do componente, componente indisponível para comunicação, etc...). Quando há a ocorrência de um erro nalgum dos componentes envolvidos na transferência de zona, o respectivo componente desiste da transferência de zona e fecha a conexão TCP.

Não foi colocado um esquema gráfico a representar este cenário uma vez que um erro pode ocorrer em qualquer altura da transferência de zona.

### 3.1.5 Especificação dos PDU's utilizados na transferência de zona

Os PDU's utilizados na transferência de zona neste trabalho não têm uma estrutura específica. Todos os dados enviados entre os componentes envolvidos são apenas as *strings* presentes nos esquemas das figuras 3.1, 3.2 e 3.3 enviadas em segmentos TCP.

## 3.2 Queries

### 3.2.1 Transacções de *queries* entre os componentes

O envio de *queries* funciona da mesma maneira entre todos os componentes do sistema. Todas as *queries* são enviadas em datagramas UDP.

Nos casos em que há erros na decodificação de *queries*, o receptor responde ao remetente com uma *query* "vazia" com o campo "*responde code*" igual a 3 (os campos da estrutura da *query* são explicados mais à frente).

### 3.2.2 Cenário de pedido cliente-servidor primário/secundário

Este cenário aborda o caso em que um cliente efectua uma *query* directamente a um servidor primário/secundário e o servidor, para além de decodificar a *query* correctamente, consegue fornecer uma resposta ao pedido recebido.

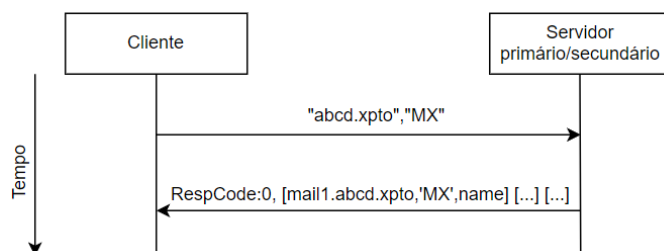


Figura 3.4: Pedido cliente-servidor primário/secundário

As interações apresentadas no esquema da figura 3.4 podem ser explicadas do seguinte modo:

1. O cliente envia ao servidor uma *query* indicando que pretende saber quais são os servidores de *e-mail* ('MX') do domínio 'abcd.xpto'.
2. O servidor decodifica a *query* correctamente e envia a respectiva resposta.

Neste cenário ainda existem as hipóteses de o servidor não ter a informação completa que o cliente solicitou (*response code* = 1) ou o domínio sobre o qual o cliente deseja receber informações não existe na base de dados ou cache do servidor (*response code* = 2).

### 3.2.3 Cenário de pedido cliente-servidor de resolução (iterativo)

Este cenário aborda o caso em que o cliente envia uma *query* a um servidor de resolução no modo iterativo.

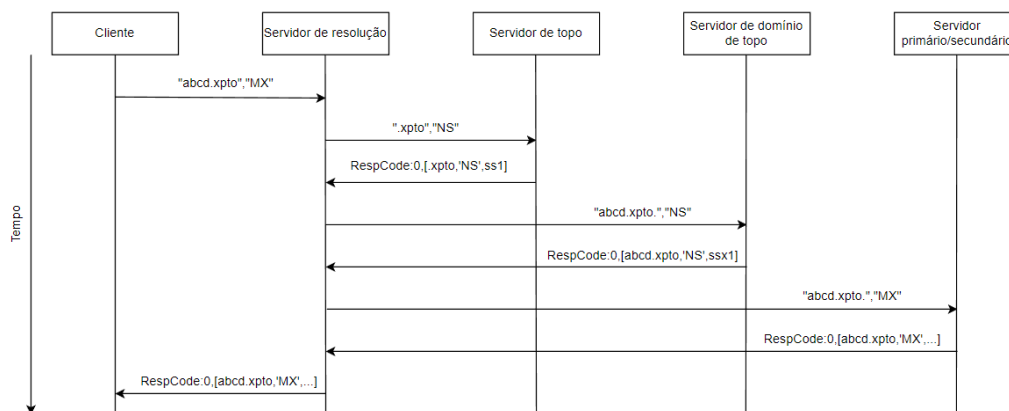


Figura 3.5: Pedido cliente-servidor de resolução

As interações apresentadas no esquema da figura 3.5 podem ser explicadas do seguinte modo:

1. O cliente envia ao servidor uma *query* indicando que pretende saber quais são os servidores de *e-mail* ('MX') do domínio 'abcd.xpto'.
2. O servidor de resolução solicita ao servidor de topo informações acerca das autoridades do domínio ".xpto".
3. O servidor de topo envia ao servidor de resolução as informações das autoridades do domínio ".xpto", isto é, envia informações sobre os servidores de domínio de topo do domínio ".xpto".
4. O servidor de resolução envia ao servidor de domínio de topo do domínio ".xpto" um pedido de informações acerca das autoridades do domínio "abcd.xpto".
5. O servidor de domínio de topo do domínio ".xpto" envia ao servidor de resolução as informações das autoridades do domínio "abcd.xpto".
6. O servidor de resolução envia ao servidor autoritativo do domínio "abcd.xpto" um pedido de informações acerca dos servidores de *e-mail* desse mesmo domínio.



7. O servidor autoritativo do domínio "abcd.xpto" responde enviando as informações dos servidores de *e-mail* desse domínio.
8. O servidor de resolução finalmente obtém a resposta desejada e encaminha-a para o cliente.

É importante referir que, se o servidor de resolução tiver a informação pretendida na sua cache, este envia a resposta directamente ao cliente sem contactar todos os outros servidores.

### 3.2.4 Cenário de pedido cliente-servidor de resolução (recursivo)

Este cenário aborda o caso em que o cliente envia uma *query* a um servidor de resolução no modo recursivo.

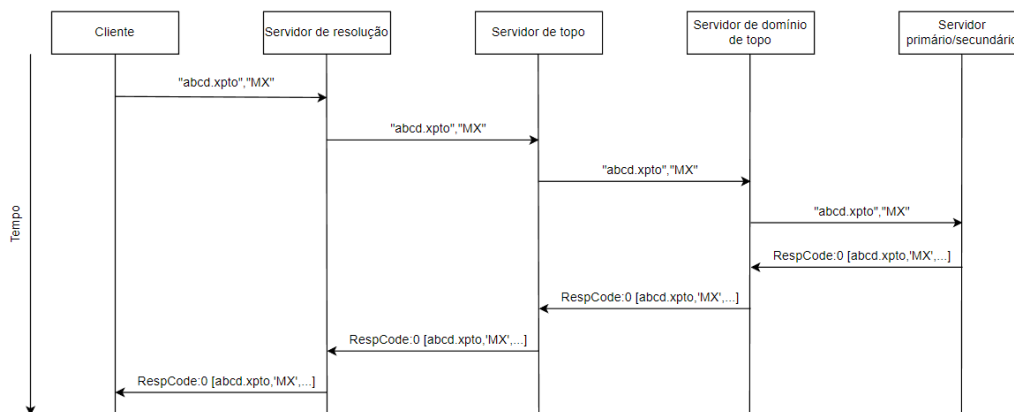


Figura 3.6: Pedido cliente-servidor de resolução

As interações apresentadas no esquema da figura 3.6 podem ser explicadas do seguinte modo:

1. O cliente envia ao servidor uma *query* indicando que pretende saber quais são os servidores de *e-mail* ('MX') do domínio 'abcd.xpto'.
2. O servidor de resolução solicita ao servidor de topo que obtenha informações dos servidores de *e-mail* do domínio "abcd.xpto"
3. O servidor de topo solicita ao servidor de domínio de topo do domínio ".xpto" que obtenha informações dos servidores de *e-mail* do domínio "abcd.xpto".
4. O servidor de domínio de topo do domínio ".xpto" solicita ao servidor autoritativo do domínio "abcd.xpto" informações sobre os seus servidores de *e-mail*.
5. O servidor autoritativo do domínio "abcd.xpto" responde ao servidor de domínio de topo do domínio ".xpto" enviando os servidores de *e-mail* do domínio "abcd.xpto".

6. O servidor de domínio de topo do domínio ".xpto" envia a informação que obteve ao servidor de topo.
7. O servidor de topo envia a informação que obteve ao servidor de resolução.
8. O servidor de resolução finalmente obtém a resposta desejada e encaminha-a para o cliente.

Tal como no modo iterativo, se o servidor de resolução tiver a informação pretendida na sua cache, este envia a resposta directamente ao cliente sem contactar todos os outros servidores.

### 3.2.5 Especificação dos PDU's utilizados nas *Queries*

Os PDU's das *queries* possuem os seguintes campos:

1. MESSAGE ID - Indica o *id* da mensagem. Serve para o remetente saber que recebeu a resposta ao seu pedido. Este campo é um valor entre 0 e 65535 e ocupa 2 *bytes*.
2. FLAGS - Existem três *flags*: Q, R e A. A flag 'Q' indica que a *query* é uma pergunta, senão é uma resposta. A flag 'R' indica se a comunicação para obter a resposta ao pedido deve ser feita em modo recursivo, senão deve ser feita em modo iterativo. A flag 'A' indica se a resposta enviada na *query* é proveniente de uma fonte autoritativa. Os valores deste campo variam de 0 a 7 e correspondem à conversão de valores binários para decimais, sendo os valores binários a representação da ativação das *flags*. Este campo ocupa 1 *byte*.
3. RESPONSE CODE - Este campo identifica possíveis erros na resposta a uma *query*. Se o seu valor for 0, significa que não houve qualquer erro. Caso o seu valor seja 1, significa que não existe resposta directa ao pedido mas o domínio incluído na *query* existe na fonte da resposta. No caso do valor ser 2, significa que a fonte da resposta desconhece o domínio incluído na *query*, isto é, o domínio não existe. Finalmente, se o valor for 3, significa que houve um erro a decodificar a *query*. Este campo ocupa 1 *byte*.
4. NUMBER OF VALUES - Este campo indica o número de entradas na resposta que respondem directamente ao pedido. Este campo ocupa 1 *byte*.
5. NUMBER OF AUTHORITIES - Indica o número de servidores autoritativos para o domínio mencionado na *query*.
6. NUMBER OF EXTRA VALUES - Indica o número de entradas com informação adicional relacionada com os resultados da *query* ou com os servidores da lista de autoridades. Este campo ocupa 1 *byte*.

7. NAME - Indica o domínio sobre o qual se pretende obter informação.  
Este campo tem um tamanho variável.
8. TYPE OF VALUE - Indica o tipo de valor que se pretende obter: NS, A, MX ou PTR.  
Como os tipos de valor são poucos e numeráveis, podemos atribuir-lhes números inteiros permitindo que este campo ocupe 1 *byte*.
9. RESPONSE VALUES - Este campo armazena os valores que correspondem à resposta directa à *query*. Este campo tem um tamanho variável.
10. AUTHORITIES VALUES - Este campo armazena os nomes dos servidores cujos domínios são iguais ao domínio do campo NAME.  
Este campo tem um tamanho variável.
11. EXTRA VALUES - Este campo armazena os valores do tipo 'A' cujos nomes dos servidores estão no campo RESPONSE VALUES ou AUTHORITIES VALUES.  
Este campo tem um tamanho variável.

Segue-se uma proposta de codificação binária de uma *query*. Os campos são inseridos no datagrama UDP conforme a ordem que aparecem a seguir.

- MESSAGE ID - Número inteiro limitado a 1 *byte*.
- FLAGS - Número inteiro limitado a 1 *byte*.
- RESPONSE CODE - Número inteiro limitado a 1 *byte*.
- NUMBER OF VALUES - Número inteiro limitado a 1 *byte*
- Tamanho do campo NAME - Número inteiro limitado a 1 *byte*.
- NAME - *String* de tamanho variável.
- TYPE OF VALUE - Número inteiro limitado a 1 *byte*.
- Tamanho do campo RESPONSE VALUES - Número inteiro limitado a 2 *bytes*.
- RESPONSE VALUES - *String* de tamanho variável
- Tamanho do campo AUTHORITIES VALUES - Número inteiro limitado a 2 *bytes*.
- AUTHORITIES VALUES - *String* de tamanho variável
- Tamanho do campo EXTRA VALUES - Número inteiro limitado a 2 *bytes*.
- EXTRA VALUES - *String* de tamanho variável

Nesta proposta de codificação, tendo em conta que é previsível que o campo NAME transporte uma *string* pequena, 1 *byte* será suficiente para representar o tamanho do campo NAME.

Já os campos RESPONSE VALUES, AUTHORITIES VALUES e EXTRA VALUES podem transportar uma quantidade considerável de dados. Deste modo, deverá ser necessário, no mínimo, alocar 2 *bytes* para informar o tamanho de cada um dos referidos campos.

## 4 Implementação dos elementos

### 4.1 Cliente

A implementação do cliente é muito simples. Neste trabalho, o cliente abre um *socket*, utiliza-o para enviar os dados e aguarda por uma resposta a esses dados.

#### 4.1.1 O seu funcionamento

Neste trabalho, o cliente apenas envia uma *query* ao servidor e aguarda pela sua resposta. O comportamento do cliente é influenciado pelos argumentos que lhe são fornecidos na linha de comandos, sendo que há três argumentos que são obrigatórios fornecer e outros três que são opcionais. Os argumentos obrigatórios são o IP do servidor ao qual se vai fazer a *query*, o domínio sobre o qual se pretende obter informações e o tipo de valor ('NS','A','MX','PTR') que se pretende obter na *query*. Os argumentos opcionais são a personalização do tempo que o cliente aguarda pela resposta (*timeout*), a desactivação do modo *debug* e a activação do modo de comunicação recursivo.

#### 4.1.2 O modo *debug*

O modo *debug* do cliente é muito simples. Este modo apenas permite que o cliente envie as suas *queries* numa *string* relativamente legível que depois possa ser lida por uma pessoa. Assim, ao utilizar um *sniffer* (por exemplo, o *Wireshark*) podemos ver facilmente o conteúdo enviado pelo cliente aos servidores.

### 4.2 Servidor Primário

O servidor primário, nesta fase, possui três funcionalidades. Ele começa por abrir um *socket* TCP para receber e atender pedidos de transferência de zona, depois abre um *socket* UDP para receber e responder a *queries*. Finalmente, o servidor primário ainda executa uma *thread* que "vigia" a sua cache, isto é, verifica o TTL de cada uma das suas entradas e quando. Quando

o TTL de uma entrada da cache termina, a *thread* "vigilante" altera o estado dessa entrada de VALID (1) para FREE (0).

### 4.2.1 O seu funcionamento

O servidor primário recebe e atende pedidos. Este tipo de servidor responde a *queries* (na porta 5300) e a pedidos de transferência de zona (na porta 5200). O comportamento do servidor primário é influenciado pelos argumentos que lhe são fornecidos pela linha de comandos, sendo que há quatro argumentos que são obrigatórios fornecer e um que é opcional. Os argumentos obrigatórios são o IP do próprio servidor, o tipo de papel que ele vai ter (primário, secundário ou resolução), o nome completo do domínio ao qual ele vai pertencer e o caminho para o seu ficheiro de configuração. O argumento opcional é a desactivação do modo *debug*.

### 4.2.2 O modo *debug*

O modo *debug* do servidor primário tem duas funcionalidades. Uma permite que o servidor envie as suas respostas numa *string* relativamente legível para ser facilmente visualizada num *sniffer*. A outra permite que as atividades reportadas no ficheiro de *logs* também sejam impressas no *standard output*.

### 4.2.3 Actividades reportadas nos *logs*

Nesta fase, o servidor primário reporta os seguintes eventos nos seus *logs*.

- Inexistência do seu ficheiro de configuração/base de dados.
- Incoerências/ incorreções na leitura do ficheiro de configuração/base de dados.
- Recepção de *queries*.
- Respostas a *queries*.
- Início e conclusão de uma transferência de zona.
- Erros na execução de uma transferência de zona.

## 4.3 Servidor Secundário

O servidor secundário, nesta fase, possui três funcionalidades. Ele começa por abrir um *socket* UDP (na porta 5300) para receber e responder a *queries* (após realizar a primeira transferência

de zona), depois executa uma *thread* que vai avaliando a passagem do tempo para saber quando deve solicitar a versão da base de dados ao servidor primário (e, se necessário, esta *thread* efectua o pedido da réplica da base de dados ao servidor primário). Finalmente, tal como o servidor primário, este servidor tem uma *thread* que "vigia" a cache para verificar o TTL das suas entradas.

### 4.3.1 O seu funcionamento

O servidor secundário, tal como o primário, recebe e atende pedidos. Este tipo de servidor apenas responde a *queries* e efectua pedidos de transferência de zona. O seu comportamento é influenciado pelos argumentos que lhe são fornecidos na linha de comandos, sendo que há quatro argumentos obrigatórios fornecer e um opcional. Os argumentos obrigatórios são o IP do próprio servidor, o tipo de papel que ele vai ter (primário, secundário, ou resolução), o nome completo do domínio ao qual vai pertencer e o caminho para o ficheiro de configuração. O argumento opcional é a desactivação do modo debug.

### 4.3.2 O modo *debug*

O modo *debug* do servidor secundário tem duas funcionalidades. Uma permite que o servidor envie as suas respostas numa *string* relativamente legível para ser facilmente visualizada num *sniffer*. A outra permite que as atividades reportadas no ficheiro de *logs* também sejam impressas no *standard output*

### 4.3.3 Actividades reportadas nos *logs*

Nesta fase, o servidor secundário reporta os seguintes eventos nos seus *logs*.

- Inexistência do seu ficheiro de configuração.
- Incoerências/incorreções na leitura do ficheiro de configuração.
- Recepção de *queries*.
- Respostas a *queries*.
- Início e conclusão de uma transferência de zona.
- Erros na execução de uma transferência de zona.

## 5 Análise de testes do sistema

A realização de testes aos componentes do sistema é essencial para verificar o seu correcto funcionamento.

Para isso, nesta primeira fase, foram realizados os seguintes testes para verificar o funcionamento do sistema:

1. Realização de uma transferência de zona no servidor de topo secundário no seu arranque.
2. Realização de uma *query* do cliente ao servidor de topo primário.
3. Realização de uma *query* do cliente ao servidor de topo secundário.
4. Realização de uma transferência de zona no servidor de domínio de topo secundário do domínio .xpto no seu arranque.
5. Realização de uma *query* ao servidor de domínio de topo primário do domínio .xpto.
6. Realização de uma *query* ao servidor de domínio de topo secundário do domínio .xpto.
7. Realização de uma transferência de zona no servidor secundário do domínio abcd.xpto no seu arranque.
8. Realização de uma *query* do cliente ao servidor primário do domínio abcd.xpto.
9. Realização de uma *query* do cliente ao servidor secundário do domínio abcd.xpto.
10. Realização de uma transferência de zona no servidor secundário após o servidor primário actualizar a base de dados.



## 1.º Teste - Transferência de zona no servidor de topo secundário no seu arranque

Este teste pretende verificar se o servidor de topo secundário, no seu arranque, consegue fazer uma transferência de zona do servidor de topo primário.

Obtiveram-se os seguintes resultados:

```
<server.py "10.0.0.10" -sp "('.', './files/stp.conf')"  
14/11/2022 20:23:08 EV 127.0.0.1 conf-file-read './files/stp.conf'  
14/11/2022 20:23:08 EV 127.0.0.1 db-file './files/root.db' was read'  
14/11/2022 20:24:43 ZT 10.0.1.10 SP  
□
```

Figura 5.1: Transferência de zona - Servidor Primário

```
</TP2# python3 server.py "10.0.1.10" -ss "('.', './files/sts.conf')"  
14/11/2022 20:24:43 ZT 10.0.0.10 SS  
TTL DEFAULT 86400  
DAY DEFAULT 86400  
@ DEFAULT .  
. SOA SP servp. TTL  
. SOA SERIAL 1 TTL  
. SOA REFRESH 100 TTL  
. SOA RETRY 10 TTL  
. SOA EXPIRE 1000 TTL  
. NS servs. TTL 1  
. NS servp. TTL 2  
xpto. NS xss.xpto. TTL 1  
xpto. NS xsp.xpto. TTL 2  
otpx. NS oss.otpx. TTL 1  
otpx. NS osp.otpx. TTL 2  
servs. A 10.0.1.10 DAY  
servp. A 10.0.0.10 DAY  
xsp.xpto. A 10.0.2.10 DAY  
xss.xpto. A 10.0.3.10 DAY  
osp.otpx. A 10.0.2.11 DAY  
oss.otpx. A 10.0.3.11 DAY  
sp CNAME servp DAY  
ss CNAME servs DAY
```

Figura 5.2: Transferência de zona - Servidor Secundário

Neste teste, o primeiro servidor a arrancar foi o primário. Posteriormente, iniciou-se o servidor secundário e o mesmo efectuou de imediato a transferência de zona imprimindo no seu ecrã o conteúdo que recebeu do servidor primário.

## 2.º Teste - *Query* do cliente ao servidor de topo primário.

Este teste pretende verificar se o cliente consegue fazer *queries* ao servidor primário. Obtiveram-se os seguintes resultados:

```
root@client:/home/core/Desktop/TP2#  
<op/TP2# python3 client.py "10.0.0.10" "xpto." "NS"  
# Header  
MESSAGE-ID = 40613, FLAGS = A, RESPONSE-CODE = 0,  
N-VALUES = 2, N-AUTHORITIES = 2, N-EXTRA-VALUES = 2;  
# Data: Query Info  
QUERY-INFO.NAME = xpto., QUERY-INFO.TYPE = NS;  
# Data: List of Response, Authorities and Extra Values  
RESPONSE-VALUES = xpto. NS xss,xpto. 86395 1,  
RESPONSE-VALUES = xpto. NS xsp,xpto. 86395 2;  
AUTHORITIES-VALUES = xpto. NS xss,xpto. 86395 1,  
AUTHORITIES-VALUES = xpto. NS xsp,xpto. 86395 2;  
EXTRA-VALUES = xsp,xpto. A 10.0.2.10 86395,  
EXTRA-VALUES = xss,xpto. A 10.0.3.10 86395;
```

Figura 5.3: *Query* ao servidor primário - Cliente

```
root@sp:/home/core/Desktop/TP2#  
Server.py "10.0.0.10" -sp "('.', './files/stp.conf')"  
14/11/2022 20:54:33 EV 127.0.0.1 conf-file-read './files/stp.conf'  
14/11/2022 20:54:33 EV 127.0.0.1 db-file './files/root.db' was read  
14/11/2022 20:54:38 QR 10.0.12.20:34220 40613,0,0,0,0;xpto.,NS;  
14/11/2022 20:54:38 QE 10.0.12.20:34220 40613,A,0,2,2;xpto.,NS;xpto. NS xss,xp  
to. 86395 1,xpto. NS xsp,xpto. 86395 2;xpto. NS xss,xpto. 86395 1;xpto. NS xsp,x  
pto. 86395 2;xsp,xpto. A 10.0.2.10 86395,xss,xpto. A 10.0.3.10 86395;
```

Figura 5.4: *Query* ao servidor primário - Servidor Primário

Neste teste, o cliente efectua uma *query* ao servidor de topo primário solicitando informações sobre os servidores autoritativos do domínio .xpto.

### 3.º Teste - *Query* do cliente ao servidor de topo secundário.

Este teste pretende verificar se o cliente consegue fazer *queries* ao servidor secundário. Obtiveram-se os seguintes resultados:

```
root@client:/home/core/Desktop/TP2# python3 client.py "10.0.1.10" "xpto." "NS"
# Header
MESSAGE-ID = 60874, FLAGS = A, RESPONSE-CODE = 0,
N-VALUES = 2, N-AUTHORITIES = 2, N-EXTRA-VALUES = 2;
# Data: Query Info
QUERY-INFO.NAME = xpto., QUERY-INFO.TYPE = NS;
# Data: List of Response, Authorities and Extra Values
RESPONSE-VALUES = xpto. NS xss.xpto. 86384 1;
RESPONSE-VALUES = xpto. NS xsp.xpto. 86384 2;
AUTHORITIES-VALUES = xpto. NS xss.xpto. 86384 1;
AUTHORITIES-VALUES = xpto. NS xsp.xpto. 86384 2;
EXTRA-VALUES = xsp.xpto. A 10.0.2.10 86384;
EXTRA-VALUES = xss.xpto. A 10.0.3.10 86384;
```

Figura 5.5: *Query* ao servidor secundário - Cliente

```
14/11/2022 20:59:17 QR 10.0.12.20:45343 60874.0.0.0.0:xpto.,NS;
14/11/2022 20:59:17 QE 10.0.12.20:45343 60874.A.0.2.2.2:xpto.,NS;xpto. NS xss.xp
to. 86384 1;xpto. NS xsp.xpto. 86384 2;xpto. NS xss.xpto. 86384 1;xpto. NS xsp.x
pto. 86384 2;xsp.xpto. A 10.0.2.10 86384;xss.xpto. A 10.0.3.10 86384;
```

Figura 5.6: *Query* ao servidor secundário - Servidor Secundário

Neste teste, o cliente efectua uma *query* ao servidor de topo secundário solicitando informações sobre os servidores autoritativos do domínio .xpto.

#### 4.º Teste - Transferência de zona no servidor de domínio de topo secundário do domínio .xpto no seu arranque.

Este teste pretende verificar se o servidor de domínio de topo secundário, no seu arranque, consegue fazer uma transferência de zona do servidor de domínio de topo primário.

Obtiveram-se os seguintes resultados:

```
root@SP-XPT0:/home/core/Desktop/TP2#  
< "10.0.2.10" -sp "{'xpto.', './files/sp_xpto.conf'}"  
14/11/2022 21:13:37 EV 127.0.0.1 conf-file-read './files/sp_xpto.conf'  
14/11/2022 21:13:37 EV 127.0.0.1 db-file './files/xpto.db' was read'  
14/11/2022 21:13:44 ZT 10.0.3.10 SP  
□
```

Figura 5.7: Transferência de zona - Servidor Primário

```
root@SS-XPT0:/home/core/Desktop/TP2#  
< "10.0.3.10" -ss "{'xpto.', './files/ss_xpto.conf'}"  
14/11/2022 21:13:44 ZT 10.0.2.10 SS  
TTL DEFAULT 86400  
@ DEFAULT xpto.  
@ SOA servp.xpto. TTL  
@ SOA SERIAL 1 TTL  
@ SOA REFRESH 100 TTL  
@ SOA RETRY 10 TTL  
@ SOA EXPIRE 10 TTL  
@ NS servs.xpto. TTL 1  
@ NS servp.xpto. TTL 2  
abcd NS xssa.abcd.xpto. TTL 1  
abcd NS xssb.abcd.xpto. TTL 2  
abcd NS xsp.abcd.xpto. TTL 3  
servs A 10.0.3.10 TTL  
servp A 10.0.2.10 TTL  
xsp.abcd A 10.0.8.10 TTL  
xssa.abcd A 10.0.9.10 TTL  
xssb.abcd A 10.0.10.10 TTL  
sp CNAME servp TTL  
ss CNAME servs TTL
```

Figura 5.8: Transferência de zona - Servidor Secundário

Neste teste, o primeiro servidor a arrancar foi o primário. Posteriormente, iniciou-se o servidor secundário e o mesmo efectuou de imediato a transferência de zona imprimindo no seu ecrã o conteúdo que recebeu do servidor primário.

## 5.º Teste - *Query* do cliente ao servidor de domínio de topo primário.

Este teste pretende verificar se o cliente consegue fazer *queries* ao servidor primário.

Obtiveram-se os seguintes resultados:

```
root@client:/tmp/pycore.34967/client.conf# !cd
cd /home/core/Desktop/TP2/
<2# python3 client.py "10.0.2.10" "abcd.xpto." "NS"
# Header
MESSAGE-ID = 6893, FLAGS = A, RESPONSE-CODE = 0,
N-VALUES = 3, N-AUTHORITIES = 3, N-EXTRA-VALUES = 3;
# Data: Query Info
QUERY-INFO.NAME = abcd.xpto., QUERY-INFO.TYPE = NS;
# Data: List of Response, Authorities and Extra Values
RESPONSE-VALUES = abcd.xpto. NS xssa.abcd.xpto. 86362 1,
RESPONSE-VALUES = abcd.xpto. NS xssb.abcd.xpto. 86362 2,
RESPONSE-VALUES = abcd.xpto. NS xsp.abcd.xpto. 86362 3;
AUTHORITIES-VALUES = abcd.xpto. NS xssa.abcd.xpto. 86362 1,
AUTHORITIES-VALUES = abcd.xpto. NS xssb.abcd.xpto. 86362 2,
AUTHORITIES-VALUES = abcd.xpto. NS xsp.abcd.xpto. 86362 3;
EXTRA-VALUES = xsp.abcd.xpto. A 10.0.8.10 86362,
EXTRA-VALUES = xssa.abcd.xpto. A 10.0.9.10 86362,
EXTRA-VALUES = xssb.abcd.xpto. A 10.0.10.10 86362;
```

Figura 5.9: *Query* ao servidor primário - Cliente

```
root@SP-XPT0:/home/core/Desktop/TP2#
< "10.0.2.10" -sp "(xpto.',./files/sp_xpto.conf)"
14/11/2022 21:20:49 EV 127.0.0.1 conf-file-read './files/sp_xpto.conf'
14/11/2022 21:20:49 EV 127.0.0.1 db-file './files/xpto.db' was read'
14/11/2022 21:20:52 ZT 10.0.3.10 SP
14/11/2022 21:21:27 QR 10.0.12.20:35275 6893,0.0.0.0:abcd.xpto.,NS;
14/11/2022 21:21:27 QE 10.0.12.20:35275 6893,A,0,3,3:abcd.xpto.,NS:abcd.xpto,
NS xssa.abcd.xpto. 86362 1,abcd.xpto. NS xssb.abcd.xpto. 86362 2,abcd.xpto. NS x
sp.abcd.xpto. 86362 3:abcd.xpto. NS xssa.abcd.xpto. 86362 1,abcd.xpto. NS xssb,a
bcd.xpto. 86362 2,abcd.xpto. NS xsp.abcd.xpto. 86362 3;xsp.abcd.xpto. A 10.0.8.1
0 86362,xssa.abcd.xpto. A 10.0.9.10 86362,xssb.abcd.xpto. A 10.0.10.10 86362;
```

Figura 5.10: *Query* ao servidor primário - Servidor Primário

Neste teste, o cliente efectua uma *query* ao servidor de domínio de topo primário solicitando informações sobre os servidores autoritativos do domínio abcd.xpto.

## 6.º Teste - *Query* do cliente ao servidor de domínio de topo secundário.

Este teste pretende verificar se o cliente consegue fazer *queries* ao servidor secundário.

Obtiveram-se os seguintes resultados:

```
root@client:/home/core/Desktop/TP2# python3 client.py "10,0,3,10" "abcd,xpto.">
# Header
MESSAGE-ID = 33531, FLAGS = A, RESPONSE-CODE = 0,
N-VALUES = 3, N-AUTHORITIES = 3, N-EXTRA-VALUES = 3;
# Data: Query Info
QUERY-INFO.NAME = abcd,xpto,, QUERY-INFO.TYPE = NS;
# Data: List of Response, Authorities and Extra Values
RESPONSE-VALUES = abcd,xpto, NS xssa,abcd,xpto, 86348 1,
RESPONSE-VALUES = abcd,xpto, NS xssb,abcd,xpto, 86348 2,
RESPONSE-VALUES = abcd,xpto, NS xsp,abcd,xpto, 86348 3;
AUTHORITIES-VALUES = abcd,xpto, NS xssa,abcd,xpto, 86348 1,
AUTHORITIES-VALUES = abcd,xpto, NS xssb,abcd,xpto, 86348 2,
AUTHORITIES-VALUES = abcd,xpto, NS xsp,abcd,xpto, 86348 3;
EXTRA-VALUES = xsp,abcd,xpto, A 10,0,8,10 86348,
EXTRA-VALUES = xssa,abcd,xpto, A 10,0,9,10 86348,
EXTRA-VALUES = xssb,abcd,xpto, A 10,0,10,10 86348;
```

Figura 5.11: *Query* ao servidor secundário - Cliente

```
14/11/2022 21:21:43 OR 10,0,12,20:45205 33531,0,0,0,0:abcd,xpto,,NS;
14/11/2022 21:21:43 OE 10,0,12,20:45205 33531,A,0,3,3,3:abcd,xpto,,NS:abcd,xpto,
NS xssa,abcd,xpto, 86348 1,abcd,xpto, NS xssb,abcd,xpto, 86348 2,abcd,xpto, NS
xsp,abcd,xpto, 86348 3;abcd,xpto, NS xssa,abcd,xpto, 86348 1,abcd,xpto, NS xssb,
abcd,xpto, 86348 2,abcd,xpto, NS xsp,abcd,xpto, 86348 3;xsp,abcd,xpto, A 10,0,8,
10 86348,xssa,abcd,xpto, A 10,0,9,10 86348,xssb,abcd,xpto, A 10,0,10,10 86348;
```

Figura 5.12: *Query* ao servidor secundário - Servidor Secundário

Neste teste, o cliente efectua uma *query* ao servidor de domínio de topo secundário solicitando informações sobre os servidores autoritativos do domínio abcd.xpto.

## 7.º Teste - Transferência de zona no servidor secundário do domínio abdc.xpto no seu arranque.

Este teste pretende verificar se o servidor secundário, no seu arranque, consegue fazer uma transferência de zona do servido primário.

Obtiveram-se os seguintes resultados:

```
cd /home/core/Desktop/TP2/
<ver.py "10.0.8.10" -sp "('./files/sp_abcd_xpto.conf')">
14/11/2022 21:50:18 EV 127.0.0.1 conf-file-read './files/sp_abcd_xpto.conf'
14/11/2022 21:50:18 EV 127.0.0.1 db-file './files/abcd-xpto.db' was read'
14/11/2022 21:51:41 ZT 10.0.9.10 SP
```

Figura 5.13: Transferência de zona - Servidor Primário

```
root@SSA-ABCD-XPT0:/home/core/Desktop/TP2# python3 server.py "10.0.9.10" -ss ">
14/11/2022 21:51:41 ZT 10.0.8.10 SS
TTL DEFAULT 86400
@ DEFAULT abcd.xpto.
DAY DEFAULT 86400
@ SOA SP servp.abcd.xpto. TTL
@ SOA ADMIN abcd\,admin.abcd.xpto. TTL
@ SOA SERIAL 1 TTL
@ SOA REFRESH 100 TTL
@ SOA RETRY 10 TTL
@ SOA EXPIRE 100 TTL
@ NS servsa.abcd.xpto. TTL 1
@ NS servsb.abcd.xpto. TTL 2
@ NS servp.abcd.xpto. TTL 3
servp A 10.0.8.10 TTL
servsa A 10.0.9.10 TTL
servsb A 10.0.10.10 TTL
mail1 A 200.0.1.1 TTL
mail2 A 200.0.1.2 TTL
mail3 A 200.0.1.3 TTL
www A 200.0.1.4 TTL
sp CNAME servp TTL
ssa CNAME servsa TTL
```

Figura 5.14: Transferência de zona - Servidor Secundário

Neste teste, o primeiro servidor a arrancar foi o primário. Posteriormente, iniciou-se o servidor secundário e o mesmo efectuou de imediato a transferência de zona imprimindo no seu ecrã o conteúdo que recebeu do servidor primário.

## 8.º Teste - *Query* do cliente ao servidor primário do domínio abcd.xpto.

Este teste pretende verificar se o cliente consegue fazer *queries* ao servidor primário.

Obtiveram-se os seguintes resultados:

```
cd /home/core/Desktop/TP2/
<top/TP2# python3 client.py "10.0.8.10" "abcd.xpto." "NS"
# Header
MESSAGE-ID = 48706, FLAGS = A, RESPONSE-CODE = 0,
N-VALUES = 3, N-AUTHORITIES = 3, N-EXTRA-VALUES = 3;
# Data: Query Info
QUERY-INFO.NAME = abcd.xpto., QUERY-INFO.TYPE = NS;
# Data: List of Response, Authorities and Extra Values
RESPONSE-VALUES = abcd.xpto. NS servsa,abcd.xpto. 86307 1,
RESPONSE-VALUES = abcd.xpto. NS servsb,abcd.xpto. 86307 2,
RESPONSE-VALUES = abcd.xpto. NS servp,abcd.xpto. 86307 3;
AUTHORITIES-VALUES = abcd.xpto. NS servsa,abcd.xpto. 86307 1,
AUTHORITIES-VALUES = abcd.xpto. NS servsb,abcd.xpto. 86307 2,
AUTHORITIES-VALUES = abcd.xpto. NS servp,abcd.xpto. 86307 3;
EXTRA-VALUES = servp,abcd.xpto. A 10.0.8.10 86307,
EXTRA-VALUES = servsa,abcd.xpto. A 10.0.9.10 86307,
EXTRA-VALUES = servsb,abcd.xpto. A 10.0.10.10 86307;
```

Figura 5.15: *Query* ao servidor primário - Cliente

```
root@SP-ABCD-XPT0:/home/core/Desktop/TP2#
<0" -sp "('abcd.xpto.', './files/sp_abcd_xpto.conf')")
14/11/2022 22:05:49 EV 127.0.0.1 conf-file-read './Files/sp_abcd_xpto.conf'
14/11/2022 22:05:49 EV 127.0.0.1 db-file './files/abcd-xpto.db' was read'
14/11/2022 22:05:56 ZI 10.0.9.10 SP
14/11/2022 22:07:22 QR 10.0.12.20:55595 48706,0.0.0.0:abcd.xpto.,NS;
14/11/2022 22:07:22 DE 10.0.12.20:55595 48706,A.0.3.3:abcd.xpto.,NS;abcd.xpto.
NS servsa,abcd.xpto. 86307 1,abcd.xpto. NS servsb,abcd.xpto. 86307 2,abcd.xpto.
NS servp,abcd.xpto. 86307 3;abcd.xpto. NS servsa,abcd.xpto. 86307 1,abcd.xpto.
NS servsb,abcd.xpto. 86307 2,abcd.xpto. NS servp,abcd.xpto. 86307 3;servp,abcd.x
pto. A 10.0.8.10 86307,servsa,abcd.xpto. A 10.0.9.10 86307,servsb,abcd.xpto. A 1
0.0.10.10 86307;
```

Figura 5.16: *Query* ao servidor primário - Servidor Primário

Neste teste, o cliente efectua uma *query* ao servidor primário do domínio abcd.xpto solicitando informações sobre os servidores autoritativos do domínio abcd.xpto.



## 9.º Teste - *Query* do cliente ao servidor secundário do domínio abcd.xpto.

Este teste pretende verificar se o cliente consegue fazer *queries* ao servidor secundário.

Obtiveram-se os seguintes resultados:

```
root@client:/home/core/Desktop/TP2# python3 client.py "10,0,9,10" "abcd.xpto.">
# Header
MESSAGE-ID = 32503, FLAGS = A, RESPONSE-CODE = 0,
N-VALUES = 3, N-AUTHORITIES = 3, N-EXTRA-VALUES = 3;
# Data: Query Info
QUERY-INFO.NAME = abcd.xpto., QUERY-INFO.TYPE = NS;
# Data: List of Response, Authorities and Extra Values
RESPONSE-VALUES = abcd.xpto. NS servsa,abcd.xpto. 86297 1,
RESPONSE-VALUES = abcd.xpto. NS servsb,abcd.xpto. 86297 2,
RESPONSE-VALUES = abcd.xpto. NS servp,abcd.xpto. 86297 3;
AUTHORITIES-VALUES = abcd.xpto. NS servsa,abcd.xpto. 86297 1,
AUTHORITIES-VALUES = abcd.xpto. NS servsb,abcd.xpto. 86297 2,
AUTHORITIES-VALUES = abcd.xpto. NS servp,abcd.xpto. 86297 3;
EXTRA-VALUES = servp,abcd.xpto. A 10,0,8,10 86297,
EXTRA-VALUES = servsa,abcd.xpto. A 10,0,9,10 86297,
EXTRA-VALUES = servsb,abcd.xpto. A 10,0,10,10 86297;
```

Figura 5.17: *Query* ao servidor secundário - Cliente

```
14/11/2022 22:07:38 QR 10,0,12,20:40842 32503,Q,0,0,0,0:abcd.xpto.,NS:
14/11/2022 22:07:38 QE 10,0,12,20:40842 32503,A,0,3,3,3:abcd.xpto.,NS:abcd.xpto.
NS servsa,abcd.xpto. 86297 1;abcd.xpto. NS servsb,abcd.xpto. 86297 2;abcd.xpto.
NS servp,abcd.xpto. 86297 3;abcd.xpto. NS servsa,abcd.xpto. 86297 1;abcd.xpto.
NS servsb,abcd.xpto. 86297 2;abcd.xpto. NS servp,abcd.xpto. 86297 3;servp,abcd.x
pto. A 10,0,8,10 86297,servsa,abcd.xpto. A 10,0,9,10 86297,servsb,abcd.xpto. A 1
0,0,10,10 86297;
```

Figura 5.18: *Query* ao servidor secundário - Servidor Secundário

Neste teste, o cliente efectua uma *query* ao servidor secundário do domínio abcd.xpto solicitando informações sobre os servidores autoritativos do domínio abcd.xpto.

## 10.º Teste - Transferência de zona no servidor secundário após o servidor primário actualizar a base de dados

Este teste pretende verificar se, ao fim do intervalo de tempo *SOAEXPIRE*, o servidor secundário consegue actualizar a sua base de dados para uma versão mais recente se o servidor primário tiver.

Obtiveram-se os seguintes resultados:

```
<" -ss "('abcd,xpto.','',/files/ssa_abcd_xpto.conf')"  
15/11/2022 00:10:47 ZT 10.0.8.10 SS  
TTL DEFAULT 86400  
@ DEFAULT abcd.xpto.  
DAY DEFAULT 86400  
@ SOASP servp.abcd.xpto. TTL  
@ SOADMIN abcd\,admin,abcd.xpto. TTL  
@ SOASERIAL 1 TTL  
@ SOAREFRESH 100 TTL  
@ SOARETRY 10 TTL  
@ SOAEXPIRE 80 TTL  
@ NS servsa.abcd.xpto. TTL 1  
@ NS servsb.abcd.xpto. TTL 2  
@ NS servp.abcd.xpto. TTL 3  
servp A 10.0.8.10 TTL  
servsa A 10.0.9.10 TTL  
servsb A 10.0.10.10 TTL  
mail1 A 200.0.1.1 TTL  
mail2 A 200.0.1.2 TTL  
mail3 A 200.0.1.3 TTL  
www A 200.0.1.4 TTL  
sp CNAME servp TTL
```

Figura 5.19: Primeira versão da base de dados

```
15/11/2022 00:12:07 ZT 10.0.8.10 SS  
15/11/2022 00:12:07 EV 127.0.0.1 db-file 'zone_transfer' was read'  
TTL DEFAULT 86400  
@ DEFAULT abcd.xpto.  
DAY DEFAULT 86400  
@ SOASP servp.abcd.xpto. TTL  
@ SOADMIN abcd\,admin,abcd.xpto. TTL  
@ SOASERIAL 2 TTL  
@ SOAREFRESH 100 TTL  
@ SOARETRY 10 TTL  
@ SOAEXPIRE 80 TTL  
@ NS servsa.abcd.xpto. TTL 1  
@ NS servsb.abcd.xpto. TTL 2  
@ NS servp.abcd.xpto. TTL 3  
servp A 10.0.8.10 TTL  
servsa A 10.0.9.10 TTL  
servsb A 10.0.10.10 TTL  
mail1 A 200.0.1.1 TTL  
mail2 A 200.0.1.2 TTL  
mail3 A 200.0.1.3 TTL  
www A 200.0.1.4 TTL  
sp CNAME servp TTL  
ssa CNAME servsa TTL
```

Figura 5.20: Segunda versão da base de dados

No início, o valor *SOASERIAL* da base de dados do servidor secundário é igual a 1. Passado um certo tempo, o servidor secundário solicita uma transferência de zona e a sua base de dados para a ter uma versão (*SOASERIAL*) igual a 2.

# Anexos

## Anexo 1 - Manual de utilização do cliente

O cliente tem de receber, no mínimo, os seguintes argumentos:

```
python3 client.py <ip_a_quem_pergunta> <domínio_completo> <tipo_de_valor>
```

Os argumentos acima devem ser fornecidos pela ordem em que constam.

Podem ser passados, opcionalmente, os seguintes argumentos:

```
python3 client.py -t <timeout> --no-debug -r
```

- `-t <timeout>`: Personaliza o tempo que o cliente espera pela sua resposta
- `--no-debug`: Indica que o cliente não funcionará em modo *debug*
- `-r`: Indica que a comunicação entre os servidores para responder à *query* deve ser feita em modo recursivo.

## Anexo 2 - Manual de utilização do servidor

O servidor tem de receber, no mínimo, os seguintes argumentos:

```
python3 server.py <ip_próprio> <papel> "(<domínio_completo>','<path_ficheiro_configuração>')"
```

Os argumentos acima devem ser fornecidos pela ordem em que constam.

Nota: O `<papel>` pode assumir um dos seguintes valores: `-sp`, `-ss` ou `-sr`.

Podem ser passados, opcionalmente, os seguintes argumentos:

- `--no-debug`: Indica que o servidor não funcionará em modo *debug*

### Anexo 3 - Conteúdo do ficheiro de configuração do servidor de topo primário

```
# primary top server config

. DB ./files/root.db
. SS 10.0.1.10
. DD 127.0.0.1
otpx. DD 127.0.0.1
xpto. DD 127.0.0.1
. LG ./logs/sp_root.log
all LG ./logs/sp_root_all.log
root ST ./files/st_servers.db
```

### Anexo 4 - Conteúdo do ficheiro de configuração do servidor de topo secundário

```
# secondary top server config

. SP 10.0.0.10
. DD 127.0.0.1
xpto. DD 127.0.0.1
otpx. DD 127.0.0.1
. LG ./logs/ss_root.log
all LG ./logs/ss_root_all.log
root ST ./files/st_servers.db
```

## Anexo 5 - Conteúdo do ficheiro de configuração do servidor de domínio de topo primário do domínio .xpto

```
# primary server config for top domain xpto.  
  
xpto. DB ./files/xpto.db  
xpto. SS 10.0.3.10  
xpto. DD 127.0.0.1  
abcd.xpto. DD 127.0.0.1  
xpto. LG ./logs/sp_xpto.log  
all LG ./logs/sp_xpto_all.log  
root ST ./files/st_servers.db
```

## Anexo 6 - Conteúdo do ficheiro de configuração do servidor de domínio de topo secundário do domínio .xpto

```
# secondary server config for top domain xpto.  
  
xpto. SP 10.0.2.10  
xpto. DD 127.0.0.1  
abcd.xpto. DD 127.0.0.1  
xpto. LG ./logs/ss_xpto.log  
all LG ./logs/ss_xpto_all.log  
root ST ./files/st_servers.db
```

## Anexo 7 - Conteúdo do ficheiro de configuração do servidor de domínio de topo primário do domínio .otpx

```
# primary server config for top domain otpx.  
  
otpx. DB ./files/otpx.db  
otpx. SS 10.0.3.11  
otpx. DD 127.0.0.1  
dcba.otpx. DD 127.0.0.1  
otpx. LG ./logs/sp_otpx.log  
all LG ./logs/sp_otpx_all.log  
root ST ./files/st_servers.db
```

## Anexo 8 - Conteúdo do ficheiro de configuração do servidor de domínio de topo secundário do domínio .otpx

```
# secondary server config for top domain otpx.  
  
otpx. SP 10.0.2.11  
otpx. DD 127.0.0.1  
abcd.otpx. DD 127.0.0.1  
otpx. LG ./logs/ss_otpx.log  
all LG ./logs/ss_otpx_all.log  
root ST ./files/st_servers.db
```

## Anexo 9 - Conteúdo do ficheiro de configuração do servidor primário do domínio abcd.xpto

```
# primary server config for top domain abcd.xpto.  
  
abcd.xpto. DB ./files/abcd-xpto.db  
abcd.xpto. SS 10.0.9.10  
abcd.xpto. SS 10.0.10.10  
abcd.xpto. DD 127.0.0.1  
abcd.xpto. LG ./logs/sp_abcd_xpto.log  
all LG ./logs/sp_abcd_xpto_all.log  
root ST ./files/st_servers.db
```

## Anexo 10 - Conteúdo do ficheiro de configuração do primeiro servidor secundário do domínio abcd.xpto

```
# secondary server config for top domain abcd.xpto.  
  
abcd.xpto. SP 10.0.8.10  
abcd.xpto. DD 127.0.0.1  
abcd.xpto. LG ./logs/ssa_abcd_xpto.log  
all LG ./logs/ssa_abcd_xpto_all.log  
root ST ./files/st_servers.db
```

## Anexo 11 - Conteúdo do ficheiro de configuração do segundo servidor secundário do domínio abcd.xpto

```
# secondary server config for top domain abcd.xpto.  
  
abcd.xpto. SP 10.0.8.10  
abcd.xpto. DD 127.0.0.1  
abcd.xpto. LG ./logs/ssb_abcd_xpto.log  
all LG ./logs/ssb_abcd_xpto_all.log  
root ST ./files/st_servers.db
```

## Anexo 12 - Conteúdo do ficheiro de configuração do servidor primário do domínio dcba.otpx

```
# primary server config for top domain dcba.otpx.  
  
dcba.otpx. DB ./files/dcba-otpx.db  
dcba.otpx. SS 10.0.9.11  
dcba.otpx. SS 10.0.10.11  
dcba.otpx. DD 127.0.0.1  
dcba.otpx. LG ./logs/sp_dcba_otpx.log  
all LG ./logs/sp_dcba_otpx_all.log  
root ST ./files/st_servers.db
```



## Anexo 13 - Conteúdo do ficheiro de configuração do primeiro servidor secundário do domínio dcba.otpx

```
# secondary server config for top domain dcba.otpx.  
  
dcba.otpx. SP 10.0.8.11  
dcba.otpx. DD 127.0.0.1  
dcba.otpx. LG ./logs/ssa_dcba_otpx.log  
all LG ./logs/ssa_dcba_otpx_all.log  
root ST ./files/st_servers.db
```

## Anexo 14 - Conteúdo do ficheiro de configuração do segundo servidor secundário do domínio dcba.otpx

```
# secondary server config for top domain dcba.otpx.  
  
dcba.otpx. SP 10.0.8.11  
dcba.otpx. DD 127.0.0.1  
dcba.otpx. LG ./logs/ssb_dcba_otpx.log  
all LG ./logs/ssb_dcba_otpx_all.log  
root ST ./files/st_servers.db
```

## Anexo 15 - Conteúdo do ficheiro de base de dados do servidor de topo primário

```
# root database

TTL DEFAULT 86400
DAY DEFAULT 86400
@ DEFAULT .

. SOASP servp. TTL
. SOASERIAL 1 TTL
. SOAREFRESH 100 TTL
. SOARETRY 10 TTL
. SOAEXPIRE 1000 TTL

# My domain
. NS servs. TTL 1
. NS servp. TTL 2

# My subdomains
xpto. NS xss.xpto. TTL 1
xpto. NS xsp.xpto. TTL 2
otpx. NS oss.otpx. TTL 1
otpx. NS osp.otpx. TTL 2

# SERVER IP's
servs. A 10.0.1.10 DAY
servp. A 10.0.0.10 DAY

xsp.xpto. A 10.0.2.10 DAY
xss.xpto. A 10.0.3.10 DAY
osp.otpx. A 10.0.2.11 DAY
oss.otpx. A 10.0.3.11 DAY

# Canonic names
sp CNAME servp DAY
ss CNAME servs DAY
```

## Anexo 16 - Conteúdo do ficheiro de base de dados do servidor de domínio de topo primário do domínio .xpto

```
# xpto top domain database

TTL DEFAULT 86400
@ DEFAULT xpto.
DAY DEFAULT 86400

@ SOASP servp.xpto. TTL
@ SOASERIAL 1 TTL
@ SOAREFRESH 100 TTL
@ SOARETRY 10 TTL
@ SOAEXPIRE 10 TTL

# My domain
@ NS servs.xpto. TTL 1
@ NS servp.xpto. TTL 2

# My subdomains
abcd NS xssa.abcd.xpto. TTL 1
abcd NS xssb.abcd.xpto. TTL 2
abcd NS xsp.abcd.xpto. TTL 3

# servers ip's
servs A 10.0.3.10 TTL
servp A 10.0.2.10 TTL

xsp.abcd A 10.0.8.10 TTL
xssa.abcd A 10.0.9.10 TTL
xssb.abcd A 10.0.10.10 TTL

# Canonics
sp CNAME servp TTL
ss CNAME servs TTL
sp.abcd CNAME xsp.abcd TTL
ssa.abcd CNAME xssa.abcd TTL
ssb.abcd CNAME xssb.abcd TTL
```

## Anexo 17 - Conteúdo do ficheiro de base de dados do servidor de domínio de topo primário do domínio .otpx

```
# otpx top domain database

TTL DEFAULT 86400
@ DEFAULT otpx.
DAY DEFAULT 86400

@ SOASP servp.otpx. TTL
@ SOASERIAL 1 TTL
@ SOAREFRESH 100 TTL
@ SOARETRY 10 TTL
@ SOAEXPIRE 10 TTL

# My domain
@ NS servs.otpx. TTL 1
@ NS servp.otpx. TTL 2

# My subdomains
dcba NS ossa.dcba.otpx. TTL 1
dcba NS ossb.dcba.otpx. TTL 2
dcba NS osp.dcba.otpx. TTL 3

# server ip's
servs A 10.0.3.11 TTL
servp A 10.0.2.11 TTL

osp.dcba A 10.0.8.11 TTL
ossa.dcba A 10.0.9.11 TTL
ossb.dcba A 10.0.10.11 TTL

# Canonics
sp CNAME servp TTL
ss CNAME servs TTL
sp.dcba CNAME osp.dcba TTL
ssa.dcba CNAME ossa.dcba TTL
ssb.dcba CNAME ossb.dcba TTL
```

## Anexo 18 - Conteúdo do ficheiro de base de dados do servidor primário do domínio abcd.xpto

```
# abcd.xpto. subdomain database

TTL DEFAULT 86400
@ DEFAULT abcd.xpto.
DAY DEFAULT 86400

@ SOASP servp.abcd.xpto. TTL
@ SOADMIN abcd\admin.abcd.xpto. TTL
@ SOASERIAL 2 TTL
@ SOAREFRESH 100 TTL
@ SOARETRY 10 TTL
@ SOAEXPIRE 80 TTL

# My domain
@ NS servsa.abcd.xpto. TTL 1
@ NS servsb.abcd.xpto. TTL 2
@ NS servp.abcd.xpto. TTL 3

# servers ip's
servp A 10.0.8.10 TTL
servsa A 10.0.9.10 TTL
servsb A 10.0.10.10 TTL
mail1 A 200.0.1.1 TTL
mail2 A 200.0.1.2 TTL
mail3 A 200.0.1.3 TTL
www A 200.0.1.4 TTL

# Canonics
sp CNAME servp TTL
ssa CNAME servsa TTL
ssb CNAME servsb TTL

# EMAILS
@ MX mail1.abcd.xpto. TTL
@ MX mail2.abcd.xpto. TTL
@ MX mail3.abcd.xpto. TTL
```

## Anexo 19 - Conteúdo do ficheiro de base de dados do servidor primário do domínio dcba.otpx

```
# dcba.otpx. subdomain database

TTL DEFAULT 86400
@ DEFAULT dcba.otpx.
DAY DEFAULT 86400

@ SOASP servp.abcd.xpto. TTL
@ SOADMIN dcba\.admin.dcba.otpx. TTL
@ SOASERIAL 1 TTL
@ SOAREFRESH 100 TTL
@ SOARETRY 10 TTL
@ SOAEXPIRE 100 TTL

# My domain
@ NS servsa.dcba.otpx. TTL 1
@ NS servsb.dcba.otpx. TTL 2
@ NS servp.dcba.otpx. TTL 3

# servers ip's
servp A 10.0.8.11 TTL
servsa A 10.0.9.11 TTL
servsb A 10.0.10.11 TTL
mail1 A 200.0.2.1 TTL
mail2 A 200.0.2.2 TTL
mail3 A 200.0.2.3 TTL
www A 200.0.2.4 TTL

# Canonics
sp CNAME servp TTL
ssa CNAME servsa TTL
ssb CNAME servsb TTL

# EMAILS
@ MX mail1.abcd.xpto. TTL
@ MX mail2.abcd.xpto. TTL
@ MX mail3.abcd.xpto. TTL
```