

# **Projeto Temático em Redes de Computadores**

## **Relatório Final**

### **Rede na Empresa**

#### **NANKAJ**

Grupo 3:

Ana Castro (Nº 87908)

João Santos (Nº 88007)

Luís Batista (Nº 78387)

Luís Tiago (Nº 80897)

Rafael Faustino (Nº 80914)

Tiago Silva (Nº 87913)

# **Projeto Temático em Redes de Computadores**

## **Relatório Final**

### **Rede na Empresa**

#### **NANKAJ**

Grupo 3:

Ana Castro (Nº 87908)

\_\_\_\_\_

João Santos (Nº 88007)

\_\_\_\_\_

Luís Batista (Nº 78387)

\_\_\_\_\_

Luís Tiago (Nº 80897)

\_\_\_\_\_

Rafael Faustino (Nº 80914)

\_\_\_\_\_

Tiago Silva (Nº 87913)

\_\_\_\_\_

Orientador:

Ricardo Marau

## **Resumo**

O presente relatório surge no âmbito da Unidade Curricular (UC) Projeto Temático em Redes de Computadores, orientada pelo docente Ricardo Marau, do segundo semestre do primeiro ano da Licenciatura em Tecnologias da Informação da Escola Superior de Tecnologia e Gestão de Águeda - Universidade de Aveiro (ESTGA - UA).

Este projeto visa explorar mecanismos e formas de implementar a estrutura da intranet, conectividade, endereçamento e serviços de rede com o intuito de criar uma rede bem sucedida, respeitando o objetivo esperado. Com recurso a algumas ferramentas, tais como: sustentabilidade do planeamento de projetos, emulador de software de rede, processos de virtualização, entre outros, é possível, assim, reunir o conjunto de características ideais para o desenvolver de uma rede com uma boa estrutura, tanto a trocar informações como a partilhar recursos.

O trabalho desenvolvido tem como principal objetivo a apresentação do projeto, que tem como finalidade a criação da rede de uma empresa chamada NANKAJ e a sua simulação. De uma forma objetiva, o trabalho baseia-se na descrição das etapas desenvolvidas, no que respeita ao planeamento do projeto, planificação das tarefas, desenvolvimento da rede e das configurações dos diferentes serviços. Consiste também na abordagem aos softwares utilizados no decurso do desenvolvimento da rede, explicitando-os. Predominam também as dificuldades surgidas no desenvolvimento da rede e a superação, ou não, desses constrangimentos. Não tomando como menos relevante, o domínio e a distribuição das tarefas, os requisitos funcionais e não funcionais e a descrição das etapas realizadas nas diversas configurações são também referidos na estrutura do relatório, não desvalorizando qualquer aspeto que tenha sido crucial na elaboração do trabalho.

# Índice

Índice de tabelas.....	ii
Índice de figuras.....	iii
1. Introdução .....	1
2. Planificação do trabalho .....	3
2.1. Mapa de Gantt previsto .....	3
2.2. Mapa de Gantt executado.....	3
2.3. Levantamento de requisitos.....	4
2.3.1. Requisitos funcionais e não funcionais .....	4
2.4. Orçamento .....	7
3. Descrição das atividades desenvolvidas .....	9
3.1. Solução proposta.....	9
3.2. Softwares utilizados .....	11
3.2.1. MS Project .....	11
3.2.2. GNS3.....	13
3.2.3. VirtualBox.....	14
3.2.4. DHCPD.....	14
3.2.5. BIND .....	14
3.2.6. Squid.....	15
3.3. Configurações .....	15
3.3.1. DHCP .....	15
3.3.2. OSPF .....	17
3.3.3. DNS .....	19
3.3.4. NAT.....	31
3.3.5. TFTP.....	32
3.3.6. VPN.....	34
3.3.7. Proxy.....	37
3.3.8. Firewall .....	39
3.3.9. WebServer.....	40
4. Análise dos resultados .....	43
5. Reflexão crítica e conclusões .....	47
6. Bibliografia .....	49
Anexos.....	a

## Índice de tabelas

Tabela 1 - Requisitos funcionais .....	5
Tabela 2 - Requisitos não funcionais .....	5
Tabela 3 - Orçamento da implementação da rede.....	7
Tabela 4 - Número de máquinas (hosts) por departamento.....	7
Tabela 5 - Nomes internos propostos .....	21
Tabela 6 - Análise do cumprimento dos requisitos; RF - requisito funcional.....	43
Tabela 7 - Análise do cumprimento dos requisitos; RNF - requisito não funcional .....	44

## Índice de figuras

Figura 1 – Diagrama da rede proposta.....	9
Figura 2 - Modelos incorporados do MS Project .....	12
Figura 3 – Planeamento de projetos no MS Project .....	12
Figura 4 - Linha cronológica .....	13
Figura 5 - Atribuição de recursos (Folha de Recursos) .....	13
Figura 6 - Logótipo do GNS3.....	14
Figura 7 - Logótipo do VirtualBox.....	14
Figura 8 - Configuração do dhcpd.....	16
Figura 9 - Main (caminhos).....	18
Figura 10 - DC1 (caminhos).....	18
Figura 11 – ED3 (caminhos) .....	18
Figura 12 - Hierarquia DNS (Fonte: Slides da Aula DNS) .....	19
Figura 13 - Funcionamento Recursivo (Nunes, Resolução de Nomes - Recursiva) .....	20
Figura 14 - Funcionamento Iterativo (Nunes, Resolução de Nomes - Interativa) .....	20
Figura 15 - As várias features dos vários softwares de DNS (Comparison of DNS server software, 2018) .....	22
Figura 16 – Configurações adicionadas ao ficheiro named.conf.options .....	23
Figura 17 - Zona de pesquisa direta.....	24
Figura 18 - Zona de pesquisa inversa .....	25
Figura 19 - Estrutura de um ficheiro de zona direta.....	25
Figura 20 - Configuração da zona direta "datacenter" .....	26
Figura 21 - Configuração da zona direta "comercial" .....	26
Figura 22 - Configuração da zona direta "engenharia" .....	27
Figura 23 - Configuração da zona direta "Gestão".....	27
Figura 24 - Configuração da zona direta "nankaj.com" .....	28
Figura 25 - Estrutura de um ficheiro de zona inversa .....	28
Figura 26 - Configuração da zona inversa "datacenter" .....	29
Figura 27 - Configuração da zona inversa "Comercial" .....	29
Figura 28 - Configuração da zona inversa "engenharia".....	30
Figura 29 - Configuração da zona inversa "Gestão".....	30
Figura 30 - Configuração do serviço de TFTP no servidor .....	33
Figura 31 - Configuração básica dos backups automáticos nos routers.....	33
Figura 32 - Fluxograma do Script.....	34
Figura 33 - Exemplo de bloqueio de um website .....	38
Figura 34 - Definições da ligação.....	39
Figura 35 - Firewall (Regras) .....	39

Figura 36 - Esquema do funcionamento básico de um webserver (Fonte: <a href="https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/o_que_e_um_web_server">https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/o_que_e_um_web_server</a> ) .....	40
Figura 37 - Market share do Apache em Abril de 2018 (Fonte: <a href="https://news.netcraft.com/archives/2018/04/26/april-2018-web-server-survey.html">https://news.netcraft.com/archives/2018/04/26/april-2018-web-server-survey.html</a> ) .....	41

## 1. Introdução

No âmbito do Projeto Temático em Redes de Computadores, foi proposto desenvolver uma nova rede informática e um Datacenter para um cliente, neste caso a empresa NANKAJ, a instalar numa nova delegação na cidade de Águeda. Para tal adquiriu novas instalações que se encontram repartidas por três edifícios, dois de um piso e um com dois pisos. A empresa possui três departamentos: o departamento de gestão, que ocupa o edifício 2, o departamento comercial e o departamento de engenharia, que ocupam, respetivamente, o piso 1 e 2 do edifício 3. O Datacenter, onde são mantidos todos os servidores da empresa, encontra-se no edifício 1. Foi também atribuído um IP ao grupo (192.168.229.213) que representa o endereço IP público da rede criada para a NANKAJ.

Neste contexto, o cliente apresentou uma lista de requisitos ajustada às capacidades do grupo, tornando o trabalho em prática realizável, dos quais constam a estrutura da intranet (construção da rede interna), conectividade (ligação entre os edifícios), endereçamento (atribuição de IP às redes e máquinas), serviços de rede, serviços de virtualização e opcionais (bloqueio de sites/serviços da intranet)

Cada empresa tem as suas necessidades, sendo que, no caso da NANKAJ, estas estão relacionadas com a comunicação na empresa, nomeadamente entre edifícios e departamentos, e com a forma de acesso a ficheiros remotamente, de forma a poderem trabalhar e lucrar. Com uma infraestrutura própria de Data center pretende-se garantir a privacidade dos dados dos clientes.

Neste relatório serão apresentados os procedimentos realizados ao longo do planeamento e desenvolvimento do projeto, começando com a planificação do trabalho (tarefas desenvolvidas ao longo do projeto, esquematização das mesmas e das atividades do projeto em cronogramas, tipos de requisitos associados ao desenvolvimento da rede e folha de recursos), seguida de uma descrição pormenorizada das atividades desenvolvidas (tipos de softwares utilizados, fases/etapas durante os mesmos e onde está presente também uma descrição da rede em si) que nos permitirá analisar os resultados e, por fim, refletir criticamente e concluir sobre o projeto.





## 2. Planificação do trabalho

A fase de planeamento deve contemplar várias etapas onde se definem os objetivos gerais e de segunda ordem, o tipo de tarefas, os recursos necessários à concretização dos objetivos e das tarefas, e a pesquisa. É na fase de execução/implementação que se aplicam todas as tarefas e requisitos estabelecidos na fase de planeamento.

É na fase de execução/implementação que se aplicam todas as tarefas e requisitos estabelecidos na fase de planeamento. Nesta fase desenvolve-se, neste caso, a rede informática utilizando os softwares GNS3, VirtualBox, Squid, BIND e DHCPD. Nela também se efetuam as atualizações das tarefas recorrendo a ajustes consoante o decorrer do projeto em si. Por fim, na fase de finalização recorre-se a uma verificação geral de todo o projeto, validando-o e procedendo à realização do relatório final. Neste trabalho, na fase de planeamento, mais concretamente durante o planeamento de tarefas e elaboração de cronogramas, foi necessário recorrer ao software Microsoft Project - software de gestão de projetos produzido pela Microsoft.

### 2.1. Mapa de Gantt previsto

No âmbito da fase de planeamento, a elaboração de um cronograma é uma forma de organizar e gerir as tarefas durante a execução de um projeto. Consiste na determinação da melhor forma de posicionar as tarefas ao longo do tempo de acordo com a duração das mesmas, das relações de precedência entre elas e dos prazos a cumprir. Com base nisto, foi elaborado um cronograma correspondente a este projeto no qual estão presentes as atividades previstas bem como o tempo previsto de execução de cada uma delas (anexo).

### 2.2. Mapa de Gantt executado

Esta fase consistia em aplicar todas as tarefas estabelecidas nas fases iniciais do planeamento. À medida que se iam realizando e implementando as

tarefas, observou-se que eram necessários ajustes relativamente às mesmas. Em anexo estão as tarefas atualizadas de acordo com imprevistos que iam decorrendo ao longo do projeto.

## 2.3. Levantamento de requisitos

Um requisito é uma condição necessária e indispensável colocada sobre um serviço ou sistema. O levantamento de requisitos está associado ao processo de descobrir quais são as operações que o sistema deve realizar e quais são as restrições que existem sobre essas mesmas operações.

### 2.3.1. Requisitos funcionais e não funcionais

Os requisitos tanto podem ser funcionais como não funcionais. Os primeiros são definidos como as funcionalidades ou atividades que um sistema/software deve realizar, já os não funcionais devem conter elementos específicos, tais como: descrição da tarefa a ser executada pelo software, origem do requisito e o seu utilizador, relação da passagem de informação entre o software e o utilizador e, se existirem, algumas restrições lógicas associadas à tarefa. Dentro dos requisitos não funcionais estão incluídos os requisitos de desempenho, de interface, operacionais, de recurso, de verificação e de aceitação, entre outros.

Os requisitos não funcionais estão relacionados com os requisitos funcionais e indicam como o sistema/software deve ser feito e como deve funcionar, ou seja, são os critérios que qualificam os requisitos funcionais.

Em seguida estão apresentadas tabelas que contêm os requisitos funcionais e não funcionais alusivos ao desenvolvimento da rede (Tabela 1 e 2, respetivamente). Todos os requisitos não funcionais mencionados na Tabela 2 estão relacionados com os requisitos funcionais explícitos na Tabela 1.

*Tabela 1 - Requisitos funcionais*

Requisitos Funcionais	
<b>RF1</b>	A intranet deverá ser segmentada e composta por várias redes IP interligadas entre si
<b>RF2</b>	Deverá existir uma rede IP associada a cada departamento
<b>RF3</b>	Deverá existir uma rede IP que abranja todas as redes IP existentes num determinado edifício
<b>RF4</b>	Todas as redes IP deverão utilizar tecnologia cablada Ethernet
<b>RF5</b>	Para além da rede IP cablada, deverá também existir uma rede IP wireless nos departamentos comercial e de Engenharia
<b>RF6</b>	A interligação da Intranet com redes externas (e.g. ISP) deverá ser realizada apenas através do edifício 1
<b>RF7</b>	Todos os edifícios e departamentos deverão ser interligado e possuir conectividade entre si
<b>RF8</b>	Por motivos de segurança, utilizadores ligados à rede Wi-Fi dos departamentos Comercial e de Engenharia deverão apenas ter acesso à rede interna do próprio departamento
<b>RF9</b>	A intranet deverá ser baseada em redes IP classe C
<b>RF10</b>	Funcionários da empresa deverão conseguir aceder à intranet através do exterior
<b>RF11</b>	Deverão ser instalados serviços de DNS que efetuem a resolução direta e inversa dos servidores do datacenter

*Tabela 2 - Requisitos não funcionais*

Requisitos não funcionais	
<b>RNF1</b>	Os endereços de rede atribuídos às diversas redes IP deverão refletir e identificar um determinado edifício e departamento
<b>RNF2</b>	Deverão ser fornecidos endereços IP dinâmicos a todas as máquinas cliente
<b>RNF3</b>	É desejável a utilização de encaminhamento dinâmico na intranet
<b>RNF4</b>	Foi contratado um único endereço IP público (ISP_IP1) ao ISP
<b>RNF5</b>	Deverá ser implementado um serviço de proxy
<b>RNF6</b>	Os web browsers deverão detetar de forma automática a configuração do proxy

<b>RNF7</b>	A rede empresarial deverá estar protegida por firewall, mantendo abertos os portos estritamente essenciais e necessários ao funcionamento dos diversos serviços de rede
<b>RNF8</b>	Serviço de Trivial File Transfer Protocol (TFTP), para fazer o backup automático das configurações dos routers
<b>RNF9</b>	Visualizador central de alertas, utilizando o Simple Network Management Protocol (SNMP), para alertar o gestor da rede sempre que haja eventos anómalos nos equipamentos de rede
<b>RNF10</b>	Estes serviços devem existir numa máquina dedicada à recolha, armazenamento e visualização desta informação
<b>RNF11</b>	As máquinas e routers sob consulta devem estar elencados num ficheiro de configuração csv, carregado automaticamente, com o seguinte formato: Nome, ip, período_em_segundos
<b>RNF12</b>	Os backups devem ser guardados num diretório para cada router (diretório criado automaticamente com o nome do router) e os ficheiros devem ter o formato AAAAMMDD.bck (AAAA-ano MM-mês DD-dia)
<b>RNF13</b>	Sempre que haja um novo registo, este só deve ser guardado se houver de facto uma alteração na configuração do router. Caso contrário mantém a configuração anterior como a mais recente
<b>RNF14</b>	O serviço de anomalias deve executar um daemon e um visualizador ligados por sockets
<b>RNF15</b>	O daemon fica encarregue de obter periodicamente o estado das máquinas, registar esse estado num ficheiro de log e enviar uma notificação ao visualizador sempre que haja uma mudança no estado de uma máquina
<b>RNF16</b>	O daemon também deverá ser capaz de receber um pedido do visualizador para que lhe seja enviado o estado de todas as máquinas
<b>RNF17</b>	O visualizador deve apresentar no ecrã (terminal ou GUI) o estado atual de todas as máquinas. Para isso tem de pedir esse estado no arranque e manter o ecrã atualizado quando chegam atualizações

## 2.4. Orçamento

O orçamento previsto para a montagem da nova rede é de 16,497.5€, não estando incluída a mão de obra nem a cablagem estruturada. A tabela 3 demonstra a distribuição do custo em relação ao produto escolhido, tendo em conta o número de máquinas por departamento que terá de ser suportado, discriminado na tabela 4.

Tabela 3 - Orçamento da implementação da rede

Categoria	Nome do produto	Preço Unitário (€)	Quantidade	Preço (€)
Router	Cisco ISR 4431-Router-Rack Mountable	3,900€	3	11,700€
Switch	Cisco Small Business 500 Series Stackable Managed Switch SG500X-48	1,299.75€	2	2,599.5€
Switch	Cisco Small Business 300 Series Managed Switch SG300-52MP	1,099€	2	2,198€
			<b>Total</b>	<b>16,497.5€</b>

Tabela 4 - Número de máquinas (hosts) por departamento

Departamento	Número de hosts
Edifício 1 – Datacenter	20
Edifício 2 – Dep. Gestão	10
Edifício 3 – Dep. Comercial (Cablada)	20
Edifício 3 – Dep. Comercial (Wireless)	20
Edifício 3 – Dep. Engenharia (Cablada)	50
Edifício 3 – Dep. Engenharia (Wireless)	50



### 3. Descrição das atividades desenvolvidas

#### 3.1. Solução proposta

Muitos serviços necessários ao bom funcionamento da rede podem ser configurados tanto nos routers como nos servidores. A vantagem de fazer nos routers é que o grupo já possui os conhecimentos para realizar estas tarefas, pois estes foram adquiridos na unidade curricular de Tecnologias de Redes de Computadores. No entanto configurar os serviços em servidores, faz com que seja mais fácil gerir os mesmos, além de os centralizar, faz com que sejam mais facilmente acedidos à distância pelo administrador de redes, o que faz com que não seja necessária a presença do mesmo fisicamente. Além do mais estamos a distribuir a carga da rede, para os servidores, fazendo assim com que o tempo de vida do router aumente.

Optámos então por utilizar servidores pelas vantagens referidas acima e pelo facto de querermos aprender a configurar os serviços em ambiente de servidor uma vez que já sabemos em ambiente de router. A figura 1 representa a solução proposta para a criação da nova rede informática da NANKAJ.

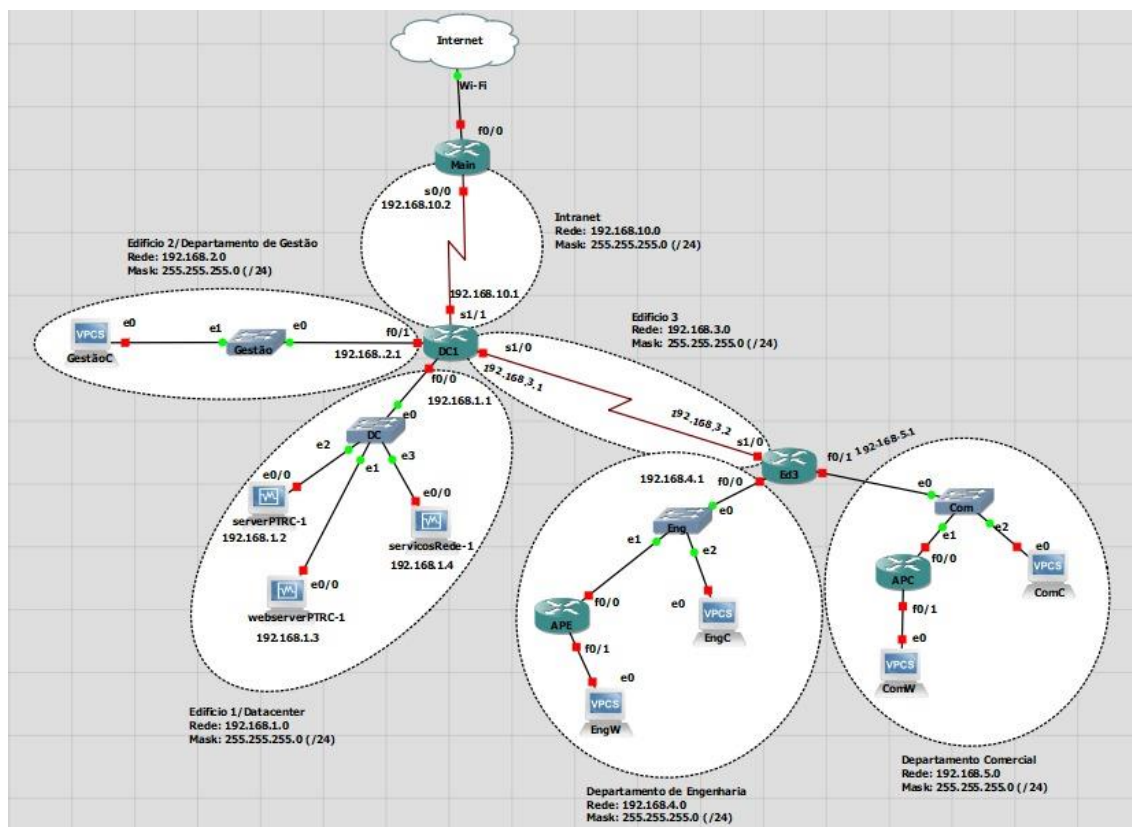


Figura 1 – Diagrama da rede proposta



Posto isto, foi necessário decidir que sistema operativo usar - Windows Server ou Linux.

O Windows Server tem como vantagens a existência de *costumer support*, o facto de os serviços e protocolos já virem instalados e de ser muito *user-friendly*, e como desvantagens a questão de ser pago, de existir menos documentação disponível do que em Linux e de consumir mais recursos do que o mesmo. Já o Linux é gratuito, disponibiliza de muita documentação, principalmente online, é mais seguro pois um utilizador normal não tem acesso a todos os ficheiros inicialmente, e consome menos recursos do que o Windows Server. Por outro lado, é menos user-friendly do que o Windows Server - onde existe GUI para a maior parte dos programas - pois o trabalho é feito em linha de comandos. Para além disso não existe *costumer support* e é necessário instalar os softwares e protocolos.

Desejamos que todos os servidores tenham IPs estáticos. Para configurar um IP estático nos nossos servidores, que estão a correr Ubuntu 16.04, é necessário editar o ficheiro `/etc/network/interfaces`.

No “server” adicionar a este ficheiro:

```
auto enp0s3  
  
iface enp0s3 inet static  
  
address 192.168.1.2  
  
netmask 255.255.255.0  
  
network 192.168.1.0  
  
gateway 192.168.1.1  
  
dns-nameservers 192.168.1.2
```

No “webserver” adicionar a este ficheiro:

```
auto enp0s3  
  
iface enp0s3 inet static  
  
address 192.168.1.3
```

```
netmask 255.255.255.0  
network 192.168.1.0  
gateway 192.168.1.1  
dns-nameservers 192.168.1.2
```

No “servicosRede” adicionar a este ficheiro:

```
auto enp0s3  
iface enp0s3 inet static  
address 192.168.1.2  
netmask 255.255.255.0  
network 192.168.1.0  
gateway 192.168.1.1  
dns-nameservers 192.168.1.2
```

## 3.2. Softwares utilizados

### 3.2.1. MS Project

Os modelos incorporados e personalizados (figura 2) utilizam as práticas recomendadas da indústria para ajudar a começar no caminho certo, não sendo necessário criar planos a partir do zero. Neste caso o grupo decidiu começar um projeto em branco e desenvolver a partir desse projeto.

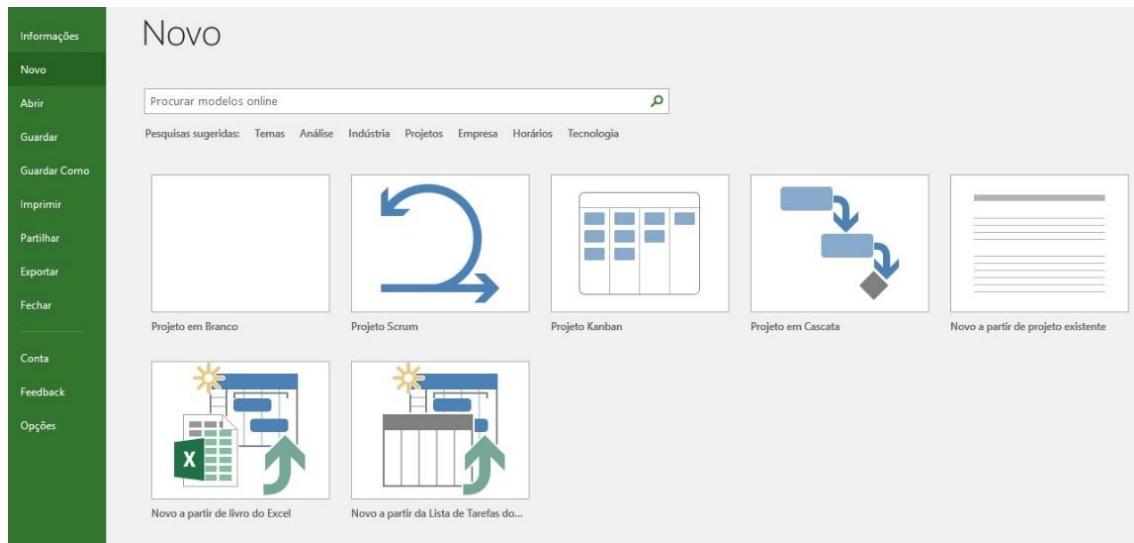


Figura 2 - Modelos incorporados do MS Project

Depois de limitar o projeto a nível de prazos, criou-se uma tarefa principal, à qual foi atribuída um calendário com horários de trabalho semelhantes ao de um trabalhador por conta de outrem. Foram-se adicionando mais tarefas, bem como a respetiva duração, agendadas automaticamente, de forma a que as mesmas fossem ajustadas à concretização do objetivo do jogo. Pode-se observar o resultado no cronograma da figura 3.

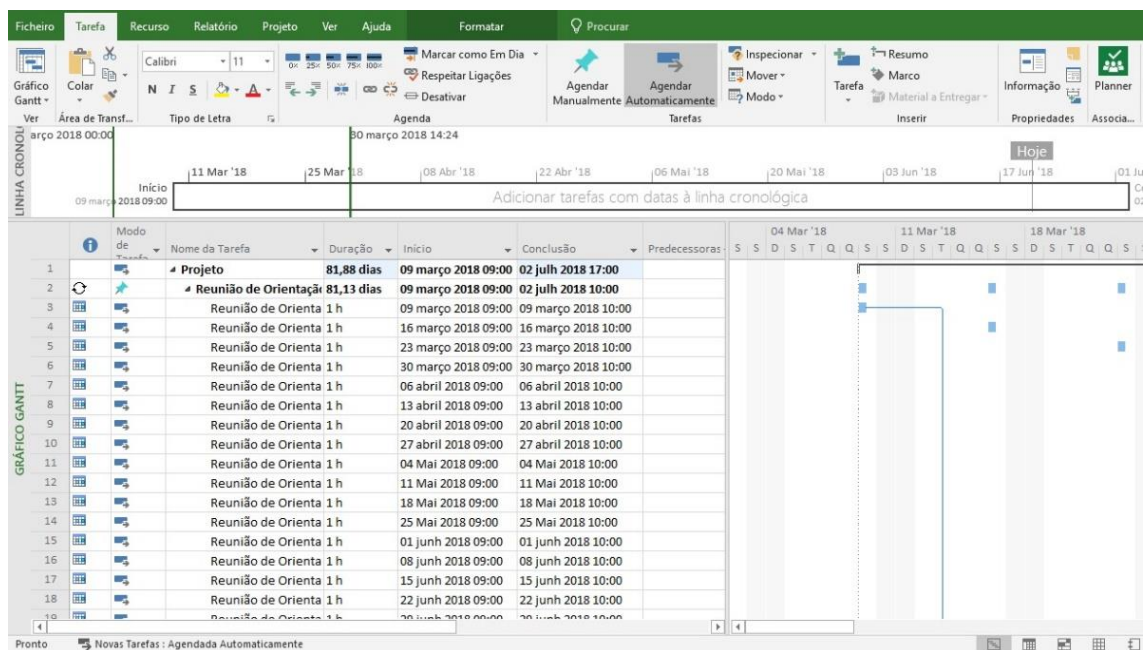


Figura 3 – Planeamento de projetos no MS Project

As linhas cronológicas (Figura 4) permitem conferir rapidamente todas as atividades do projeto, desde tarefas a marcos futuros. Podem ser personalizadas para apresentar dados específicos e partilhar as mesmas facilmente com os intervenientes do projeto.



Figura 4 - Linha cronológica

Posteriormente à implementação de predecessoras nas tarefas, para que fosse seguida uma sequência de execução das mesmas, foram-lhes atribuídas certos recursos, previamente definidos na Folha de Recursos (Figura 5).

Alocar os recursos às equipas de projeto adequadas é fundamental para uma gestão de recursos eficaz. É possível criar e editar perfis de recurso com detalhes importantes, como o tipo de funcionário, a função principal, habilidades e experiência, exibir, editar e acompanhar alocações e carga de trabalho de recursos, e comparar a capacidade com a demanda do recurso.

		Nome do Recurso	Tipo	Rótulo Material	Iniciais	Grupo	Unidade Máx.	Taxa Normal	Taxa Trab.	Custo/Utiliz.	Imputar Em	Calendário Base	Código	Adicionar Nova Color.
FOLHA DE RECURSOS	1	Luís Ricardo	Trabalho		L		100%	0,00 €/h	0,00 €/h	0,00 €	Rateado	Padrão		
	2	Tiago Silva	Trabalho		T		100%	0,00 €/h	0,00 €/h	0,00 €	Rateado	Padrão		
	3	Rafael Faustino	Trabalho		R		100%	0,00 €/h	0,00 €/h	0,00 €	Rateado	Padrão		
	4	Ana Castro	Trabalho		A		100%	0,00 €/h	0,00 €/h	0,00 €	Rateado	Padrão		
	5	João Santos	Trabalho		J		100%	0,00 €/h	0,00 €/h	0,00 €	Rateado	Padrão		
	6	Luís Tiago	Trabalho		L		100%	0,00 €/h	0,00 €/h	0,00 €	Rateado	Padrão		
	7	PC	Material		P			0,00 €		0,00 €	Rateado			
	8	GNS3	Material		G			0,00 €		0,00 €	Rateado			

Figura 5 - Atribuição de recursos (Folha de Recursos)

### 3.2.2. GNS3

O GNS3 (Figura 6) é um simulador de redes gráfico, completamente grátis, que emula os mais diversos equipamentos ativos de uma rede, tal como routers, switches, PCs, firewalls, etc. Considerando por exemplo um router, o GNS3 permite-nos emular o IOS (sistema operativo dos equipamentos Cisco) de um router real e proceder às respetivas configurações.



*Figura 6 - Logótipo do GNS3*

### 3.2.3. VirtualBox

O VirtualBox (Figura 7) é um software de virtualização que, como o VMware Workstation, visa criar ambientes para instalação de sistemas distintos. Ele permite a instalação e utilização de um sistema operativo dentro de outro, assim como os respetivos softwares, como dois ou mais computadores independentes, mas compartilhando fisicamente o mesmo hardware.



*Figura 7 - Logótipo do VirtualBox*

### 3.2.4. DHCPD

O ISC DHCP, também conhecido como DHCPD, é um software que opera como daemon num servidor para fornecer o serviço de *Dynamic Host Configuration Protocol* a uma rede. Suporta tanto DHCPv4 como DHCPv6. A distribuição inclui um cliente, um servidor e *relay*.

### 3.2.5. BIND

BIND é um *software open source* que permite a publicação do próprio *Domain Name System* (DNS) na internet e permite a resolução de *queries* para os utilizadores. O nome BIND é acrónimo para “Berkeley Internet Name Domain”, porque o *software* foi criado no início no início de 1980 na Universidade da Califórnia.

BIND é o software de DNS mais usado na Internet, providenciado uma plataforma estável e compatível com os DNS *standards*.

#### 3.2.6. Squid

O Squid é um dos melhores servidores proxy para Linux que suporta HTTP, HTTPS, FTP e outros protocolos. Reduz a utilização da conexão e melhora os tempos de resposta ao fazer a cache de requisições frequentes de páginas web numa rede de computadores, para além disso é um software completamente grátis. Permite que as outras máquinas cliente acessem a páginas web mesmo que não tenham conexão direta com a Internet, tudo o que eles precisam é do acesso ao próprio servidor onde o Squid foi configurado.

### 3.3. Configurações

#### 3.3.1. DHCP

Nesta secção do presente relatório pretende-se abordar o *Dynamic Host Configuration Protocol* (DHCP) *SERVER*, fazendo referência primeiramente a uma breve introdução sobre o mesmo, e de seguida ao modo como foi implementado no presente projeto.

O protocolo DHCP pode ser definido como um serviço da rede que utiliza um modelo cliente-servidor para atribuir dinamicamente um IP a cada máquina.

O servidor DHCP é, então, o meio necessário para a gestão da atribuição de endereços numa rede, possuindo assim um intervalo(*pool*) de endereços mediante o departamento em que se encontra, e à medida que vai sendo feita a solicitação de conexão com a rede.

O endereço alocado a cada computador é um empréstimo(*lease*), ou seja, após um determinado intervalo de tempo definido pelo administrador do servidor da rede, o cliente terá de voltar a fazer um novo pedido e será atribuído um novo endereço. Na empresa *NANKAJ* o empréstimo por defeito (*default-lease-time*) terá um tempo de oito horas, enquanto que o tempo máximo permitido (*max-lease-time*) será de 24 horas.

### Configuração:

No presente projeto, para a implementação do serviço de DHCP foi primeiramente garantido que o servidor teria um IP estático.

De seguida, foi instalado o *dhcpcd* através do comando `sudo apt-get install isc-dhcp-server` para a configuração do DHCP.

O passo seguinte foi a configuração do mesmo através do editor de texto usando o comando **`sudo gedit /etc/dhcp/dhcpd.conf`** que está representado na figura 8.

```
ddns-update-style none;

option domain-name "datacenter";
option domain-name-servers server;

default-lease-time 28800;
max-lease-time 86400;

authoritative;

log-facility local7;

subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option domain-name-servers 192.168.1.2;
    range dynamic-bootp 192.168.1.5 192.168.1.254;
}

subnet 192.168.2.0 netmask 255.255.255.0 {
    option routers 192.168.2.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.2.255;
    option domain-name-servers 192.168.1.2;
    range dynamic-bootp 192.168.2.2 192.168.2.254;
}

subnet 192.168.4.0 netmask 255.255.255.0 {
    option routers 192.168.4.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.4.255;
    option domain-name-servers 192.168.1.2;
    range dynamic-bootp 192.168.4.2 192.168.4.254;
}

subnet 192.168.5.0 netmask 255.255.255.0 {
    option routers 192.168.5.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.5.255;
    option domain-name-servers 192.168.1.2;
    range dynamic-bootp 192.168.5.2 192.168.5.254;
}
```

Figura 8 - Configuração do *dhcpcd*

No fim da configuração do DHCP é preciso reiniciar através do comando **sudo service isc-dhcp-server restart** para que as novas configurações sejam ativadas.

Para finalizar, foi configurado um *relay agent* nos routers DC1 e Ed3, que tem como função realizar o encaminhamento dos pedidos de atribuição de endereços das máquinas para o servidor DHCP.

### 3.3.2. OSPF

Nesta secção do presente relatório pretende-se abordar o protocolo *Open Shortest Path First* (OSPF), fazendo primeiramente referência ao porquê da utilização deste protocolo de roteamento dinâmico em vez de outro protocolo, e de seguida uma breve introdução acerca do mesmo, e para finalizar o modo como foi implementado no projeto apresentado.

O protocolo RIP (*Routing Information Protocol*) é um protocolo de *distance-vector* baseado no número de saltos sendo no máximo utilizados 16 saltos. É caracterizado pelo pouco consumo de memória, a sua fácil configuração e a sua facilidade em encontrar solução para os problemas. A versão 2 do RIP é, hoje em dia, a mais utilizada em pequenas redes.

O protocolo OSPF é um protocolo especialmente projetado para o ambiente TCP/IP com o objetivo de ser usado internamente em sistemas autónomos, também conhecidos como protocolos de roteamento internos – Interior Gateway Protocol (IGP).

O protocolo OSPF baseia-se na transmissão de informação em pacotes *Link State Advertisements* (LSAs), baseado no protocolo *Link-State*, que tem por base o cálculo do caminho mais curto, utilizando o algoritmo *Short Path First* (SPF), e trocando informações sobre os estados de enlaces de comunicação ligados às portas dos routers.

Um dos seus princípios de funcionamento é a utilização de Áreas, que faz com uma rede seja dividida de forma hierárquica, com o intuito de diminuir a complexidade e minimizar a comunicação entre roteadores. Neste caso, deve existir sempre uma área central chamada *Backbone Area* (Área 0), que serve



como elo de ligação entre as demais áreas existentes, devendo ser feita obrigatoriamente através dela mesmo.

Ao comparar ambos os protocolos, existem definitivamente mais vantagens em utilizar o protocolo OSPF pois, a convergência é mais rápida, a utilização de caminhos múltiplos de forma a uma maior eficiência não implicando uma sobrecarga, é utilizado para interligações de redes grandes, o tráfego de informação é muito menor do que no protocolo RIP e utilização de diferentes mecanismos de autenticação entre os routers fazem com que o OSPF para este projeto seja o mais indicado.

No presente projeto, o encaminhamento entre departamentos é realizado através do encaminhamento *OSPF*. Na figura 9, 10 e 11 estão representados os caminhos que o router Main, DC1 (*Datacenter*) e ED3 (Edifício 3) conhecem, respetivamente.

```
router ospf 1
log-adjacency-changes
network 192.168.10.0 0.0.0.255 area 0
default-information originate
```

*Figura 9 - Main (caminhos)*

```
router ospf 1
log-adjacency-changes
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 1
network 192.168.3.0 0.0.0.255 area 0
network 192.168.10.0 0.0.0.255 area 0
```

*Figura 10 - DC1 (caminhos)*

```
router ospf 1
log-adjacency-changes
network 192.168.3.0 0.0.0.255 area 0
network 192.168.4.0 0.0.0.255 area 1
network 192.168.5.0 0.0.0.255 area 1
```

*Figura 11 – ED3 (caminhos)*

### 3.3.3. DNS

Nesta secção do presente relatório irá abordar-se o *Domain Name System* (DNS), fazendo primeiramente referência a uma breve introdução acerca do mesmo e, de seguida, ao modo como foi implementado no projeto apresentado.

O DNS é uma base de dados distribuída para associar nomes a endereços de IP e vice-versa. Existe, porque, para nós humanos é mais fácil memorizar um nome para um website do que o seu endereço de IP. Funciona de modo hierárquico e conforme se vai descendo na hierarquia, mais específicos são os nomes (Figura 12).

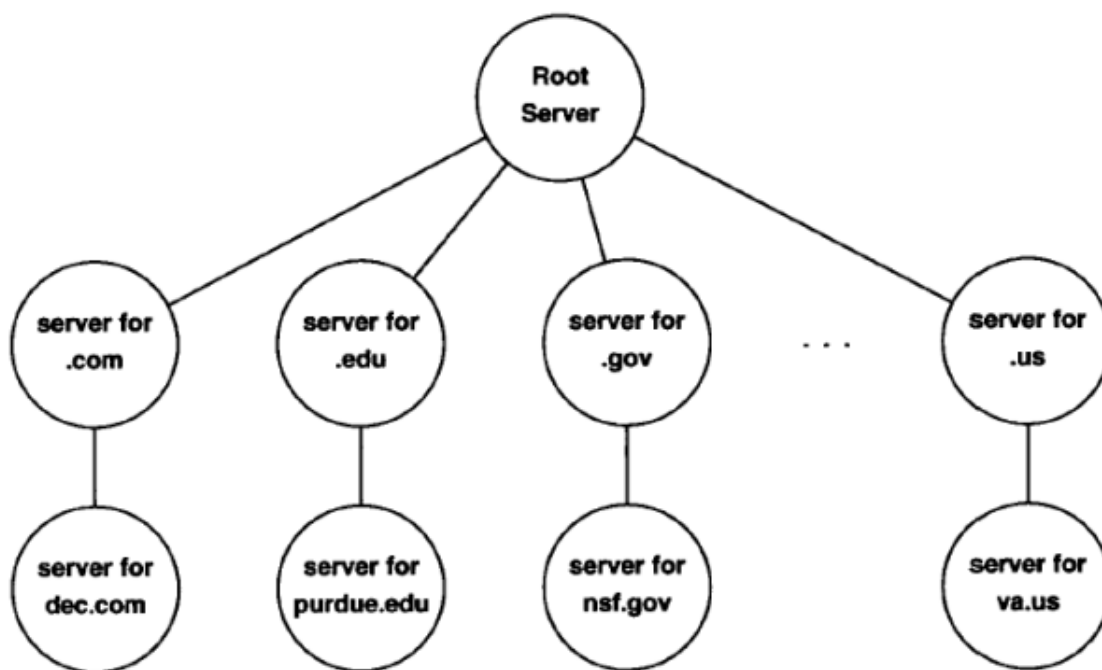


Figura 12 - Hierarquia DNS (Fonte: Slides da Aula DNS)

Existem, principalmente, dois tipos de funcionamento de um servidor de DNS: o recursivo e o iterativo.

No funcionamento recursivo (Figura 13), quando se pergunta para resolver um nome, se o servidor for *authoritative* para essa zona, ou seja, se for ele que contém as informações da zona questionada, então devolve logo a

resposta ou erro se não a encontrar. Se o servidor não for *authoritative* para a zona questionada então ele passa a ser *forwarder* e vai questionar outros servidores de DNS até obter a informação solicitada ou um erro.

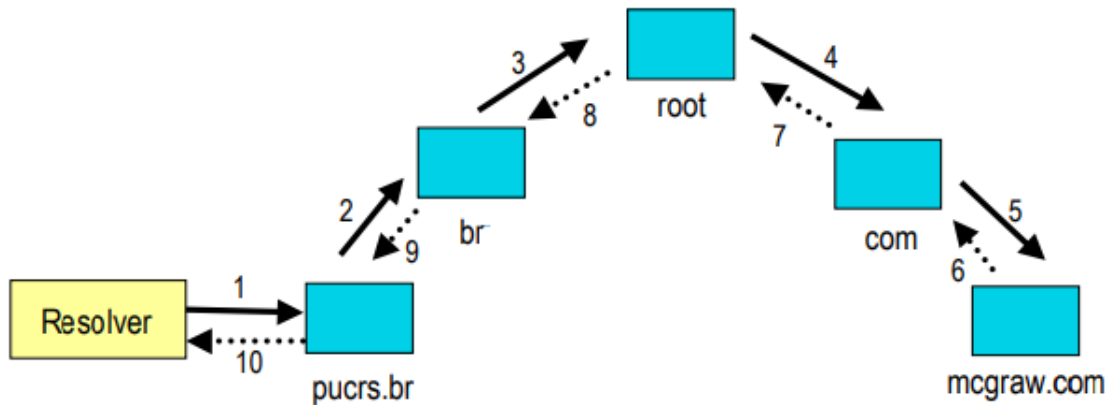


Figura 13 - Funcionamento Recursivo (Nunes, Resolução de Nomes - Recursiva)

No funcionamento iterativo (Figura 14), quando se pergunta para resolver um nome, o servidor vê as zonas para o qual é *authoritative* e a sua cache, se a resposta não estiver em nenhum dos dois, em vez de fazer o *forward* para outros servidores e devolver a resposta à pergunta, devolve antes um *referral*, ou seja, indica ao cliente um outro servidor DNS a quem perguntar.

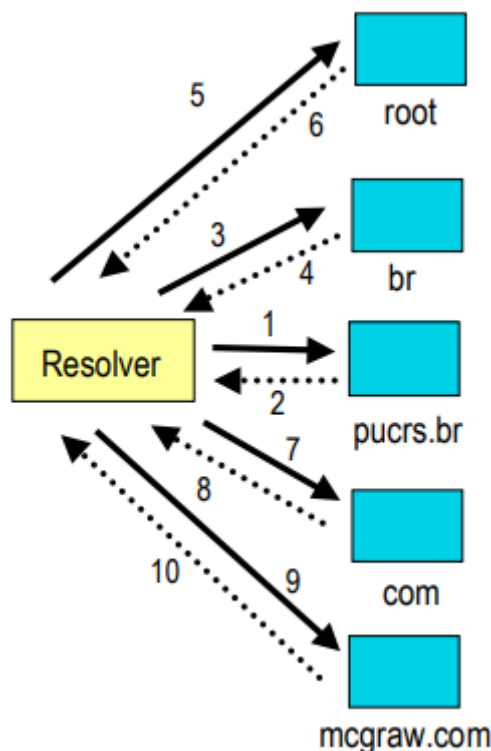


Figura 14 - Funcionamento Iterativo (Nunes, Resolução de Nomes - Iterativa)

Com os conhecimentos adquiridos na unidade curricular de Tecnologias de Redes de Computadores, e posteriores pesquisas, decidiu-se criar um servidor de DNS com funcionamento recursivo que resolvesse internamente todos os endereços IP estáticos de cada departamento (Tabela 5), e que fizesse o *forward* dos restantes nomes. Os nomes internos serão para uma utilização mais técnica, são mais direcionados para um administrador de redes do que para outro tipo de funcionário, com exceção do nome “www.nankaj.com”, esse sim direcionado a todos os empregados, pois acede ao *webserver* da empresa.

*Tabela 5 - Nomes internos propostos*

Host	Endereço IP
gw.comercial	192.168.5.1
gw.engenharia	192.168.4.1
gw.gestao	192.168.2.1
gw.datacenter	192.168.1.1
server.datacenter	192.168.1.2
webserver.datacenter	192.168.1.3
www.nankaj.com	192.168.1.3
servicosRede.datacenter	192.168.1.4

Esta solução pode ser executada de duas maneiras: no router ou numa máquina. Ambas as maneiras possuem vantagens e desvantagens. As vantagens de se implementar a solução no router é que é mais simples e rápido. Já a vantagem de se implementar numa máquina é que oferece a possibilidade de uma configuração mais detalhada. A desvantagem de se implementar a solução no router é de não possível fazer uma configuração detalhada, enquanto

a de se implementar numa máquina é que existe alguma curva de aprendizagem, pois em aula, na parte prática, foi feita a configuração no router e é mais complicado realizar backups da máquina.

Posto isto decidiu-se configurar o DNS numa máquina, uma vez que, se for necessário, pode ser feita uma configuração muito detalhada de cada zona e como em aula foi praticado a configuração de DNS em routers decidimos desafiar-nos e configurar numa máquina.

Quanto ao *software* utilizado para configurar o DNS, decidimos usar o BIND9, pois como é o *software* mais usado para este propósito, por ser o mais completo (como demonstra a Figura 15), a maior parte dos administradores de rede tem algum conhecimento ou experiência com ele e existem muitos recursos online sobre como o configurar e usar.

Server	Authoritative	Recursive	Recursion ACL	Slave mode	Caching	DNSSSEC	TSIG	IPv6	Wildcard	Free Software	Interface	split horizon
AnswerX	No	Yes	Yes	No	Yes	Yes	Yes	Yes	No	No	API, command line	Yes
BIND	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes (since 9.x)	Yes (since 4.x)	Yes	Web <sup>[Note 1]</sup> , command line	Yes
PowerDNS	Yes	Yes	Yes	Yes <sup>[Note 2]</sup>	Yes	Yes (since 3.0) (Note 3)	Yes (since 3.0)	Yes <sup>[Note 2]</sup>	Yes	Yes	Web <sup>[Note 4]</sup> , command line	Partial <sup>[Note 5]</sup>
djbdns	Yes	Yes	Yes	Yes <sup>[Note 6]</sup>	Yes	Partial <sup>[Note 7]</sup>	No	Partial via generic records. (12) <sup>[8]</sup>	Partial <sup>[Note 8]</sup>	Yes	command line and web (VegaDNS & NoTool <sup>[9]</sup> ) <sup>[14]</sup>	Yes <sup>[Note 9]</sup>
dbndns	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Partial	Yes	command line and web	Yes
pdnsd	Partial	Yes	Partial	Partial	Yes	No <sup>[15]</sup>	Partial	Yes	Yes	Yes	command line, pdnsd-cli program	Partial
MaraDNS	Yes	Yes	Yes	Partial <sup>[Note 10]</sup>	Yes	No	No	Partial	Yes	Yes	command line	No
Posadis	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	command line, API	No
Unbound	Partial	Yes	Yes	N/A	Yes	Yes	No	Yes	N/A	Yes	command line, API	No
Dnsmasq	Partial <sup>[Note 11]</sup>	No	No	No	Yes	Yes (since 2.66) (Note 12)	No	Yes	Yes	Yes	command line	Partial <sup>[Note 13]</sup>
NSD	Yes	No	N/A	Yes	N/A	Yes	Yes	Yes	Yes	Yes	command line	No
Knot DNS	Yes	No	N/A	Yes	N/A	Yes	Yes	Yes	Yes	Yes	command line	No
dnrd	No	Yes	No	No	Yes	No	No	?	?	Yes	command line	No
gdnsd	Yes	No	No	No	No	No	No	Yes	Yes	Yes	command line	Yes
YADIFA	Yes	No	N/A	Yes	N/A	Yes	Yes	Yes	Yes	Yes	command line	No
yaku-ns	Yes	?	?	Yes	?	No	No	No	Yes	Yes	command line	?
Microsoft DNS	Yes	Yes	Yes <sup>[Note 14]</sup>	Yes	Yes	Yes <sup>[Note 15]</sup>	Yes <sup>[Note 16]</sup>	Yes <sup>[Note 17]</sup>	Yes	No	GUI, command line, API <sup>[Note 18]</sup> , WM <sup>[Note 19]</sup> , RPC <sup>[Note 20]</sup>	Yes <sup>[Note 14]</sup>
Simple DNS Plus	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	GUI, Web, command line	Yes <sup>[Note 21]</sup>
Nominum ANS	Yes	No	N/A	Yes	No	Yes	Yes	Yes	Yes	No	command line, api, SOAP interface, SNMP	Yes
Nominum Vantio	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	command line, api, SOAP interface, SNMP	Yes
DNS Blast	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	CLI, SOAP, REST, SNMP, DNSTAP	Yes
Secure64 DNS Authority	Yes	No	No	Yes	No	Yes	Yes	Yes	Yes	No	Command Line or Web GUI	Yes
Secure64 DNS Cache	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Command Line or Web GUI	Yes
Technitium DNS Server	Yes	Yes	No	No	Yes	No	No	Yes	Yes	Yes	Command Line, Web GUI, or REST API	No
Server	Authoritative	Recursive	Recursion ACL	Slave mode	Caching	DNSSSEC	TSIG	IPv6	Wildcard	Free Software	Interface	split horizon

Figura 15 - As várias features dos vários softwares de DNS (Comparison of DNS server software, 2018)

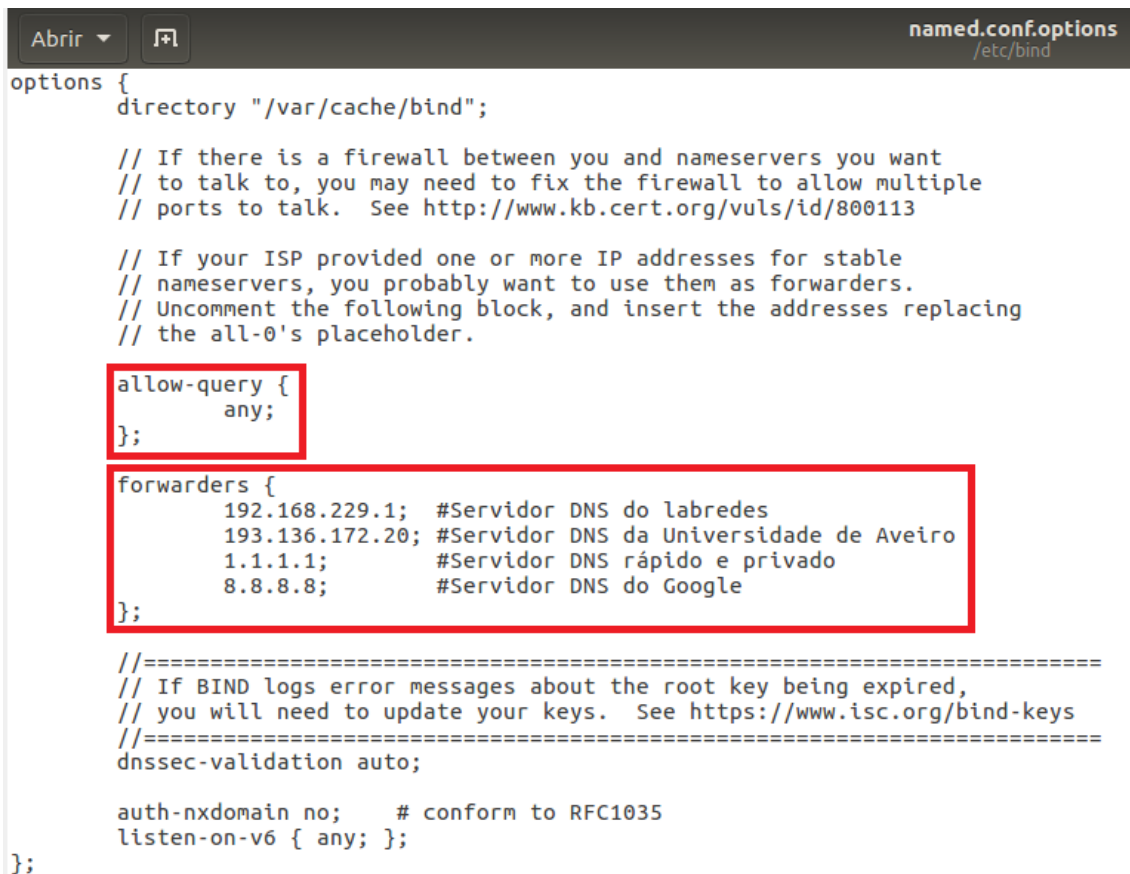
**1º** - Garantir que o servidor tem um IP estático.

**2º** - (TERMINAL) `sudo apt install bind9`

Comando para instalar o BIND9, que fica instalado no diretório “/etc/bind/”.

**3º** - Adicionar ao ficheiro “named.conf.options” as configurações destacadas na figura 16.

O comando inserido no primeiro retângulo permite que sejam feitas perguntas ao servidor por parte dos clientes. O segundo retângulo configura os servidores de DNS para os quais os pedidos, que não consigam ser resolvidos por este servidor, serão encaminhados.



```
named.conf.options
/etc/bind

options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    allow-query {
        any;
    };

    forwarders {
        192.168.229.1; #Servidor DNS do labredes
        193.136.172.20; #Servidor DNS da Universidade de Aveiro
        1.1.1.1; #Servidor DNS rápido e privado
        8.8.8.8; #Servidor DNS do Google
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

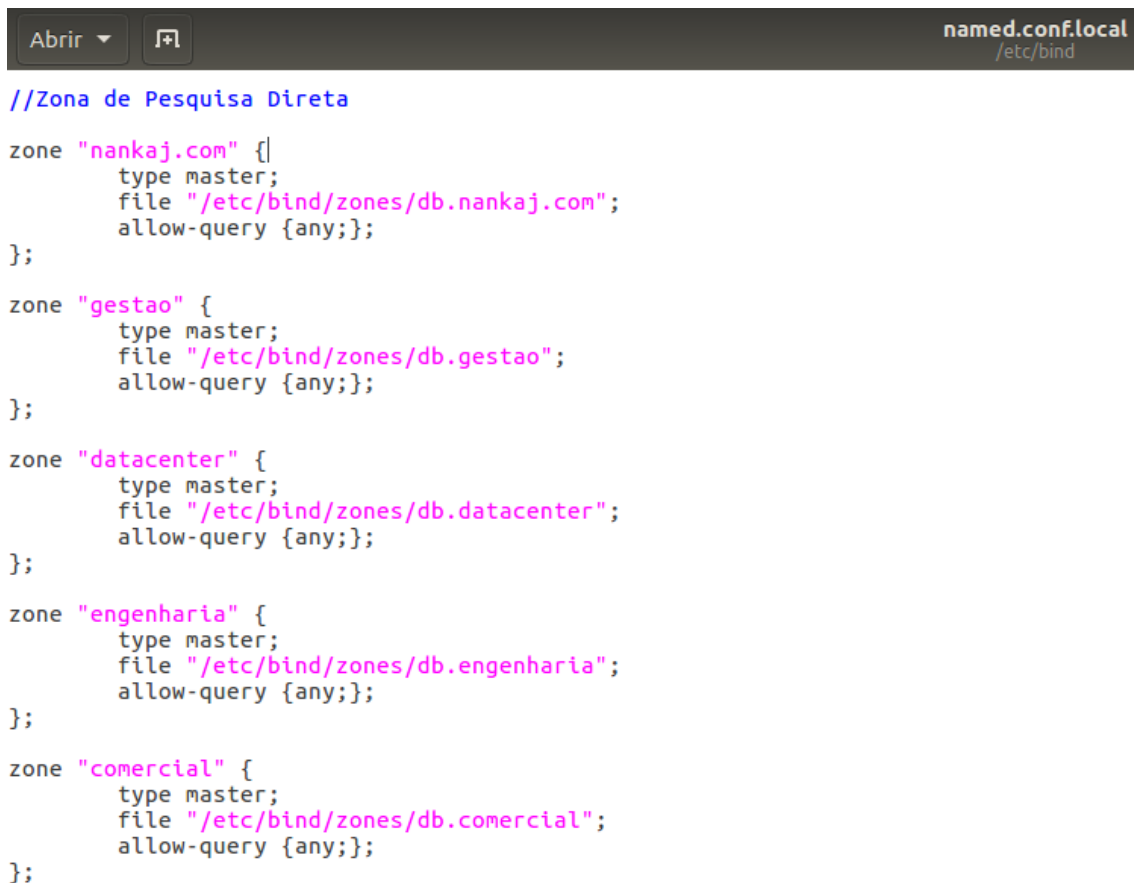
    auth-nxdomain no; # conform to RFC1035
    listen-on-v6 { any; };
};
```

Figura 16 – Configurações adicionadas ao ficheiro named.conf.options

**4º** - Adicionar ao ficheiro “named.conf.local” as configurações das figuras 17 e 18.

Neste ficheiro são indicadas as zonas de pesquisa direta e inversa.

Na figura 17 é definida a zona de pesquisa direta que consta no nome da zona (ex: comercial), indica-se que este servidor é *authoritative* para a zona (type master;), o ficheiro onde está localizada a configuração da zona (file “...”) e permite-se que esta zona seja pesquisada por todos os clientes (allow-query {any;};).



The image shows a text editor window with a dark theme. The title bar at the top right says "named.conf.local" and "/etc/bind". On the left, there are two buttons: "Abrir" with a dropdown arrow and a file icon. The main area contains a configuration file for BIND. It starts with a comment "//Zona de Pesquisa Direta" in blue. Then, there are five zone definitions, each for a different domain: "nankaj.com", "gestao", "datacenter", "engenharia", and "comercial". Each zone is configured as a master zone, with its file path set to "/etc/bind/zones/db.[domain]" and "allow-query {any;}".

```
//Zona de Pesquisa Direta

zone "nankaj.com" {
    type master;
    file "/etc/bind/zones/db.nankaj.com";
    allow-query {any;};
};

zone "gestao" {
    type master;
    file "/etc/bind/zones/db.gestao";
    allow-query {any;};
};

zone "datacenter" {
    type master;
    file "/etc/bind/zones/db.datacenter";
    allow-query {any;};
};

zone "engenharia" {
    type master;
    file "/etc/bind/zones/db.engenharia";
    allow-query {any;};
};

zone "comercial" {
    type master;
    file "/etc/bind/zones/db.comercial";
    allow-query {any;};
};
```

Figura 17 - Zona de pesquisa direta

Na figura 18 é definida a zona de pesquisa inversa que é feita através do endereço da rede de cada departamento, ao contrário (ex: rede de comercial = 192.168.5.0, fica 5.168.192), indica-se que este servidor é *authoritative* para a zona (type master;) e o ficheiro onde está localizada a configuração inversa da zona (file "...").

The screenshot shows a text editor window with a dark theme. The title bar at the top right indicates the file is `*named.conf.local` located in `/etc/bind`. The editor contains the following configuration for reverse lookup zones:

```
//Zona de Pesquisa Inversa

zone "2.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/zones/db.gestao.rev";
};

zone "1.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/zones/db.datacenter.rev";
};

zone "4.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/zones/db.engenharia.rev";
};

zone "5.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/zones/db.comercial.rev";
};
```

Figura 18 - Zona de pesquisa inversa

## 5º - Criação do diretório “/etc/bind/zones/” e ficheiros de zonas.

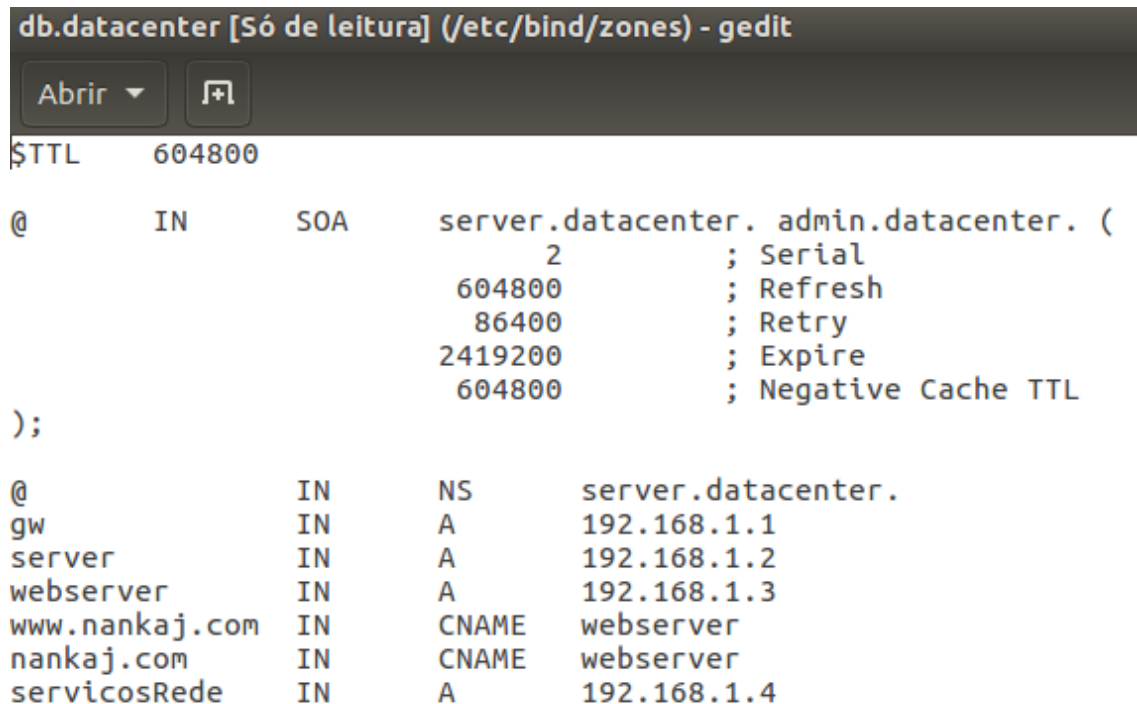
Pela figura 19 é possível observar a estrutura de um ficheiro de zona direta.

<code>\$TTL</code>	Tempo para viver em cache
<code>@ IN SOA &lt;servidor-dns&gt; &lt;admin-email&gt; (</code>	Indica o nome do servidor de DNS que tem autoridade sobre a zona e o e-mail da pessoa responsável sobre a zona
<code>&lt;serial-number&gt;</code>	
<code>&lt;time-to-refresh&gt;</code>	
<code>&lt;time-to-retry&gt;</code>	
<code>&lt;time-to-expire&gt;</code>	
<code>&lt;minimum-TTL&gt;</code>	Tempo para viver em cache
<code>);</code>	
<code>@ IN NS &lt;servidor-dns&gt;</code>	Indica o nome do servidor de DNS que tem autoridade sobre a zona
<code>&lt;host&gt; IN A &lt;endereço-IP&gt;</code>	Liga um nome a um endereço de IP
<code>&lt;alias&gt; CNAME &lt;host&gt;</code>	Liga um nome a outro já existente

Figura 19 - Estrutura de um ficheiro de zona direta



Pelas figuras 20, 21, 22, 23 e 24 é possível observar a configuração da zona direta “datacenter”, “comercial”, “engenharia”, “gestão” e “nankaj.com”, respectivamente.



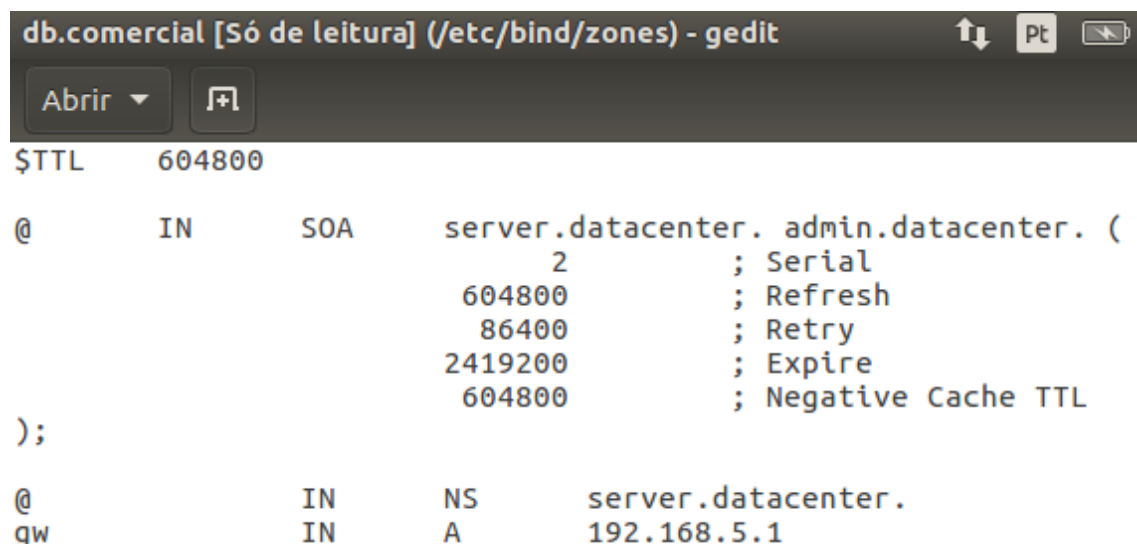
```
db.datacenter [Só de leitura] (/etc/bind/zones) - gedit
Abrir ▾ [+]

$TTL      604800

@          IN      SOA      server.datacenter. admin.datacenter. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

@          IN      NS       server.datacenter.
gw         IN      A        192.168.1.1
server     IN      A        192.168.1.2
webserver  IN      A        192.168.1.3
www.nankaj.com IN      CNAME  webserver
nankaj.com IN      CNAME  webserver
servicosRede IN      A        192.168.1.4
```

Figura 20 - Configuração da zona direta "datacenter"



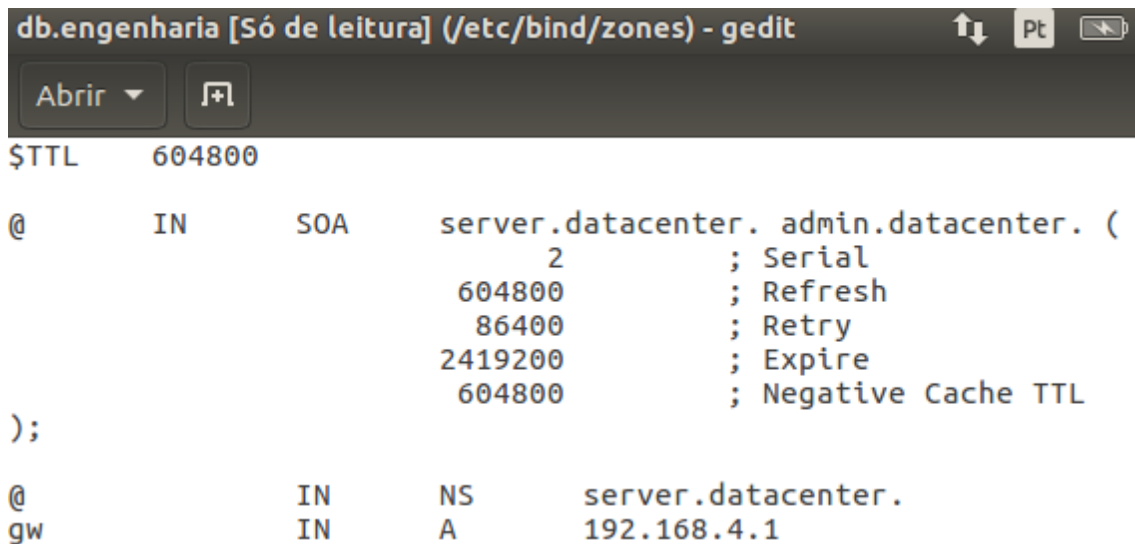
```
db.comercial [Só de leitura] (/etc/bind/zones) - gedit
Abrir ▾ [+]

$TTL      604800

@          IN      SOA      server.datacenter. admin.datacenter. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

@          IN      NS       server.datacenter.
gw         IN      A        192.168.5.1
```

Figura 21 - Configuração da zona direta "comercial"

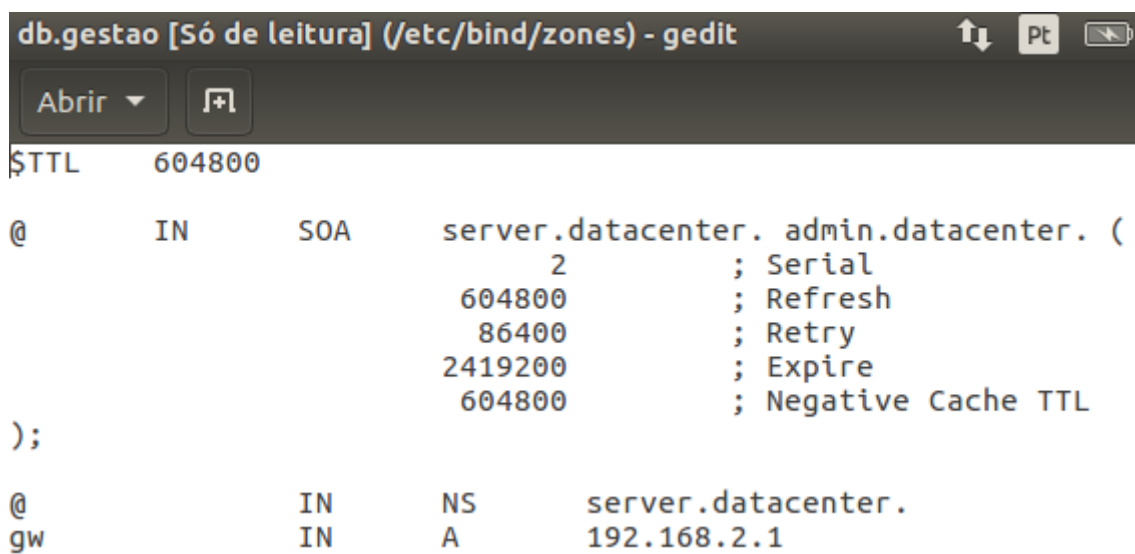


```
db.engenharia [Só de leitura] (/etc/bind/zones) - gedit
$TTL      604800

@          IN      SOA      server.datacenter. admin.datacenter. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

@          IN      NS       server.datacenter.
gw         IN      A        192.168.4.1
```

Figura 22 - Configuração da zona direta "engenharia"



```
db.gestao [Só de leitura] (/etc/bind/zones) - gedit
$TTL      604800

@          IN      SOA      server.datacenter. admin.datacenter. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

@          IN      NS       server.datacenter.
gw         IN      A        192.168.2.1
```

Figura 23 - Configuração da zona direta "Gestão"

```

db.nankaj.com [Só de leitura] (/etc/bind/zones) - gedit
Abrir  [icon]

$TTL      604800

@         IN      SOA      server.datacenter. admin.datacenter. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

@         IN      NS       server.datacenter.
@         IN      A        192.168.1.3
www       IN      A        192.168.1.3

```

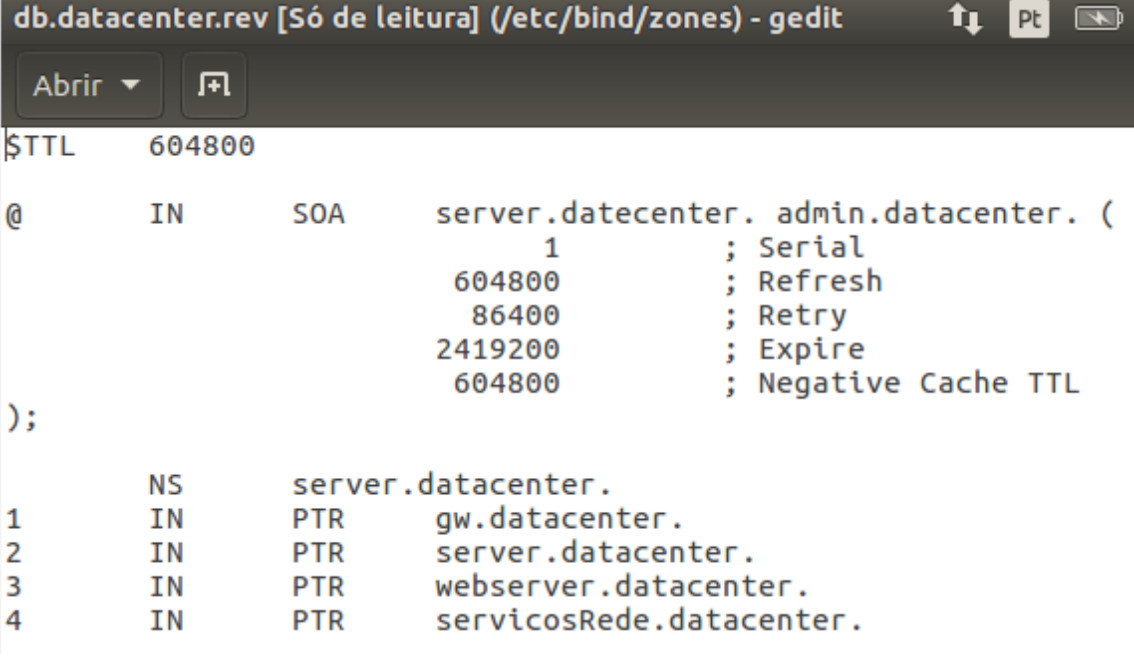
Figura 24 - Configuração da zona direta "nankaj.com"

Pela figura 25 é possível observar a estrutura de um ficheiro de zona inversa.

\$TTL	Tempo para viver em cache
@ IN SOA <servidor-dns> <admin-email> (	Indica o nome do servidor de DNS que tem autoridade sobre a zona e o e-mail da pessoa responsável sobre a zona
<serial-number>	
<time-to-refresh>	
<time-to-retry>	
<time-to-expire>	
<minimum-TTL>	Tempo para viver em cache
);	
@ IN NS <servidor-dns>	Indica o nome do servidor de DNS que tem autoridade sobre a zona
<ultimo numero> IN PTR <host>	<ultimo numero> corresponde ao ultimo numero de um endereço de IP da rede especificada no nome da zona
	<host> nome que corresponde ao endereço IP. Tem que ser igual ao nome dado no ficheiro de pesquisa direta da zona

Figura 25 - Estrutura de um ficheiro de zona inversa

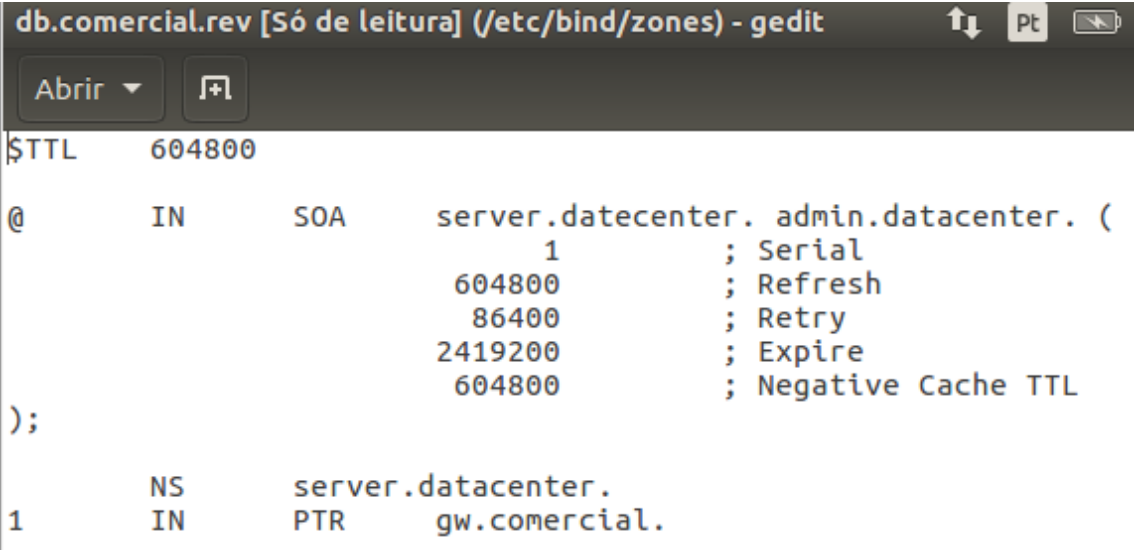
Pelas figuras 26 e 27, 28 e 29 é possível observar a configuração da zona direta “datacenter”, “comercial”, “engenharia” e “gestão” respectivamente.



```
db.datacenter.rev [Só de leitura] (/etc/bind/zones) - gedit
Abrir ▾
$TTL      604800
@         IN      SOA      server.datecenter. admin.datacenter. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

         NS       server.datecenter.
1        IN      PTR      gw.datacenter.
2        IN      PTR      server.datacenter.
3        IN      PTR      webserver.datacenter.
4        IN      PTR      servicosRede.datacenter.
```

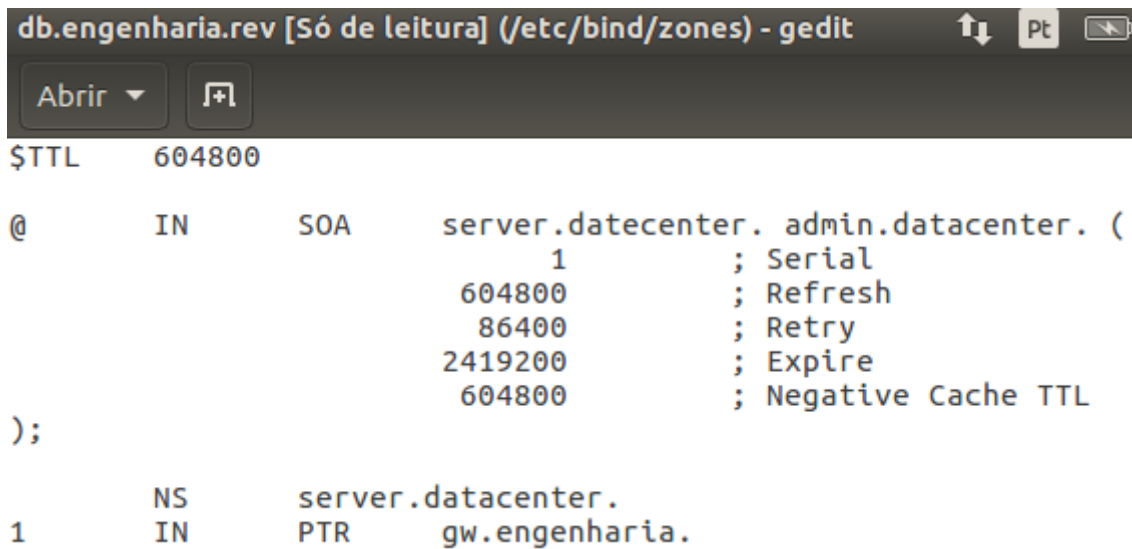
Figura 26 - Configuração da zona inversa "datacenter"



```
db.comercial.rev [Só de leitura] (/etc/bind/zones) - gedit
Abrir ▾
$TTL      604800
@         IN      SOA      server.datecenter. admin.datacenter. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

         NS       server.datecenter.
1        IN      PTR      gw.comercial.
```

Figura 27 - Configuração da zona inversa "Comercial"



db.engenharia.rev [Só de leitura] (/etc/bind/zones) - gedit

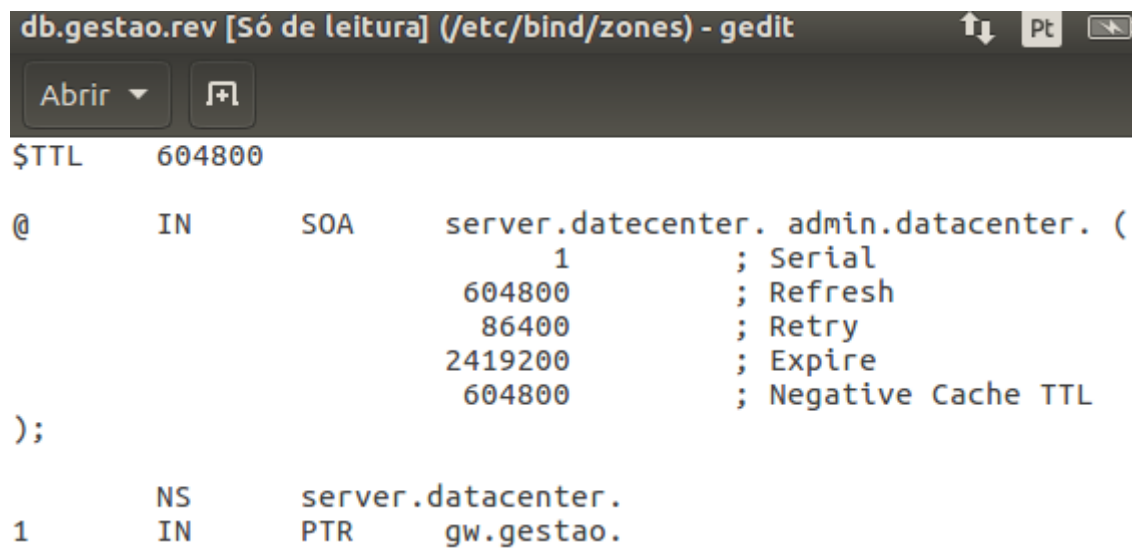
Abrir ▾ [ícone]

```
$TTL      604800

@          IN      SOA      server.datecenter. admin.datecenter. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

1          NS      server.datecenter.
           IN      PTR      gw.engenharia.
```

Figura 28 - Configuração da zona inversa "engenharia"



db.gestao.rev [Só de leitura] (/etc/bind/zones) - gedit

Abrir ▾ [ícone]

```
$TTL      604800

@          IN      SOA      server.datecenter. admin.datecenter. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
);

1          NS      server.datecenter.
           IN      PTR      gw.gestao.
```

Figura 29 - Configuração da zona inversa "Gestão"

### 3.3.4. NAT

Sabendo que os IPs públicos IPv4 são um recurso limitado e atualmente escasso, a NAT (*Network Address Translation*) tem como objetivo poupar o espaço de endereçamento público, recorrendo a endereços de IP privados.

Os endereços públicos são geridos por uma entidade reguladora, são pagos, e permitem identificar univocamente uma máquina (PC, routers, etc.) na Internet. Por outro lado, os endereços privados apenas fazem sentido num domínio local e não são conhecidos (encaminháveis) na Internet, sendo que uma máquina configurada com um IP privado terá de sair para a Internet através de um IP público. A tradução de um endereço privado num endereço público é então definida como NAT.

Existem ao todo três tipos de NAT, e para o nosso projeto optámos por usar o PAT (*Port Address Translation*) que considerámos ser o mais eficiente e adequado ao que o projeto pretendia.

PAT - (NAT Overload) é certamente a técnica mais usada. Um exemplo de PAT é quando temos um único endereço público e por ele conseguimos fazer sair várias máquinas. Este processo é conseguido uma vez que o equipamento que faz PAT utiliza portas que identificam univocamente cada pedido das máquinas locais para o exterior.

#### **Configuração:**

```
Main>enable
```

```
Main#configure terminal
```

```
Main(config)#ip nat inside source list 1 interface FastEthernet0/0 overload
```

```
Main(config)#access-list 1 permit any
```

```
Main(config)#interface s0/0
```

```
Main(config-if)#ip nat inside
```

```
Main(config-if)#exit
```

```
Main(config)#interface f0/0
```

```
Main(config-if)#ip nat outside
```

```
Main(config-if)#exit
```

```
Main(config)#ip nat inside source static tcp 192.168.1.2 interface fastethernet  
0/0 1723
```

```
Main(config)#ip nat inside source static tcp 192.168.1.3 interface fastethernet  
0/0 80
```

### 3.3.5. TFTP

Nesta secção do presente relatório pretende-se abordar o *Trivial File Transfer Protocol* (TFTP), fazendo primeiramente referência a uma breve introdução acerca do mesmo, e de seguida ao modo como foi implementado no projeto apresentado.

O TFTP (Trivial File Transfer Protocol) é um protocolo de transferência de ficheiros, semelhante ao FTP. É geralmente utilizado para transferir pequenos ficheiros entre *hosts* numa rede, como por exemplo o ficheiro de configuração de um router. Toda a transferência começa com um pedido de leitura ou escrita de um arquivo, que sirva também para pedir uma conexão. Se o servidor conceder o pedido, a conexão está aberta e o arquivo é emitido em blocos fixos do comprimento de 512 bytes.

Iremos utilizar este protocolo para realizar backups automáticos das configurações dos routers para o nosso servidor. Para tal teremos que configurar um serviço de TFTP no nosso servidor para guardar os backups dos routers, configurar os routers para realizarem backups automáticos e escrever um script (anexo) que mude o nome dos ficheiros das configurações dos routers para o formato “AAAAMMDD” e guardá-los na pasta apropriada tal como é pedido nos requisitos bem como apenas guardar o backup se este fosse diferente do backup mais recente.

Depois de realizarmos uma pesquisa sobre os *softwares* de TFTP disponíveis para Linux concluímos que os dois mais populares são o “atftpd” e o “tftpd-hpa”. Não havendo diferenças significativas entre os dois poderíamos optar por qualquer um e optámos pelo primeiro.

## Configuração

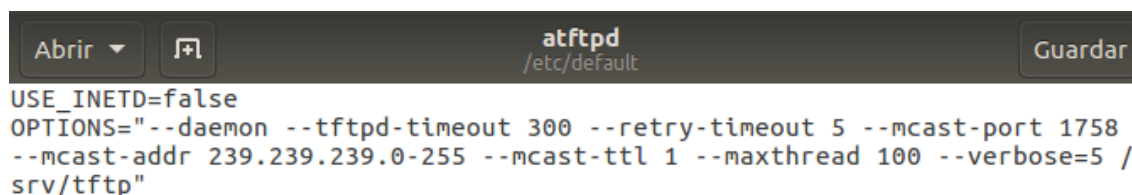
1º - Garantir que o servidor tem um IP estático.

### 2º - (TERMINAL) `sudo apt install atftpd`

Comando para instalar o atftpd, que fica instalado no diretório “/etc/default/atftpd” e guarda os backups “/srv/tftp”.

**3º Passo - No ficheiro “/etc/default/atftpd” alterar a variável `USE_INETD` para false e adicionar a `OPTIONS` o comando `--daemon`**

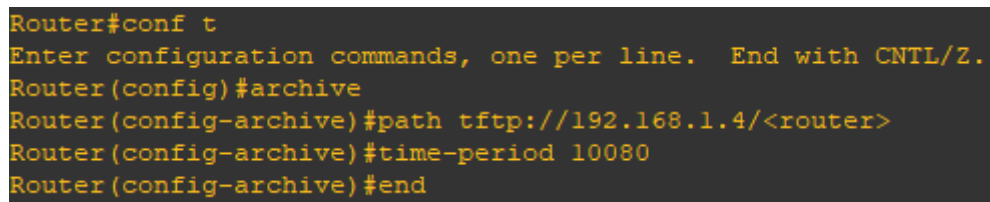
Esta configuração (Figura 30) faz com que o atftpd corra como *daemon*, ou seja como no *background*.

A screenshot of a text editor window titled 'atftpd /etc/default'. The window has buttons for 'Abrir', '+', and 'Guardar'. The text inside the editor is: 

```
USE_INETD=false
OPTIONS="--daemon --tftpd-timeout 300 --retry-timeout 5 --mcast-port 1758
--mcast-addr 239.239.239.0-255 --mcast-ttl 1 --maxthread 100 --verbose=5 /
srv/tftp"
```

Figura 30 - Configuração do serviço de TFTP no servidor

**4º - Configurar nos routers os comandos para se realizar os backups automáticos** como demonstrado pela figura 31.

A screenshot of a terminal window showing the configuration of a router. The text is: 

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#archive
Router(config-archive)#path tftp://192.168.1.4/<router>
Router(config-archive)#time-period 10080
Router(config-archive)#end
```

Figura 31 - Configuração básica dos backups automáticos nos routers

Onde <router> deve ser substituído por “main” no router Main, “datacenter” no router DC1 e “edificio3” no router Ed3

### 5º - Escrever o script

Escrever um script (anexo) que mude o nome dos ficheiros das configurações dos routers para o formato “AAAAMMDD” e os guarde na pasta



apropriada. Segue-se um fluxograma na Figura 32 que esquematiza a sua estrutura.

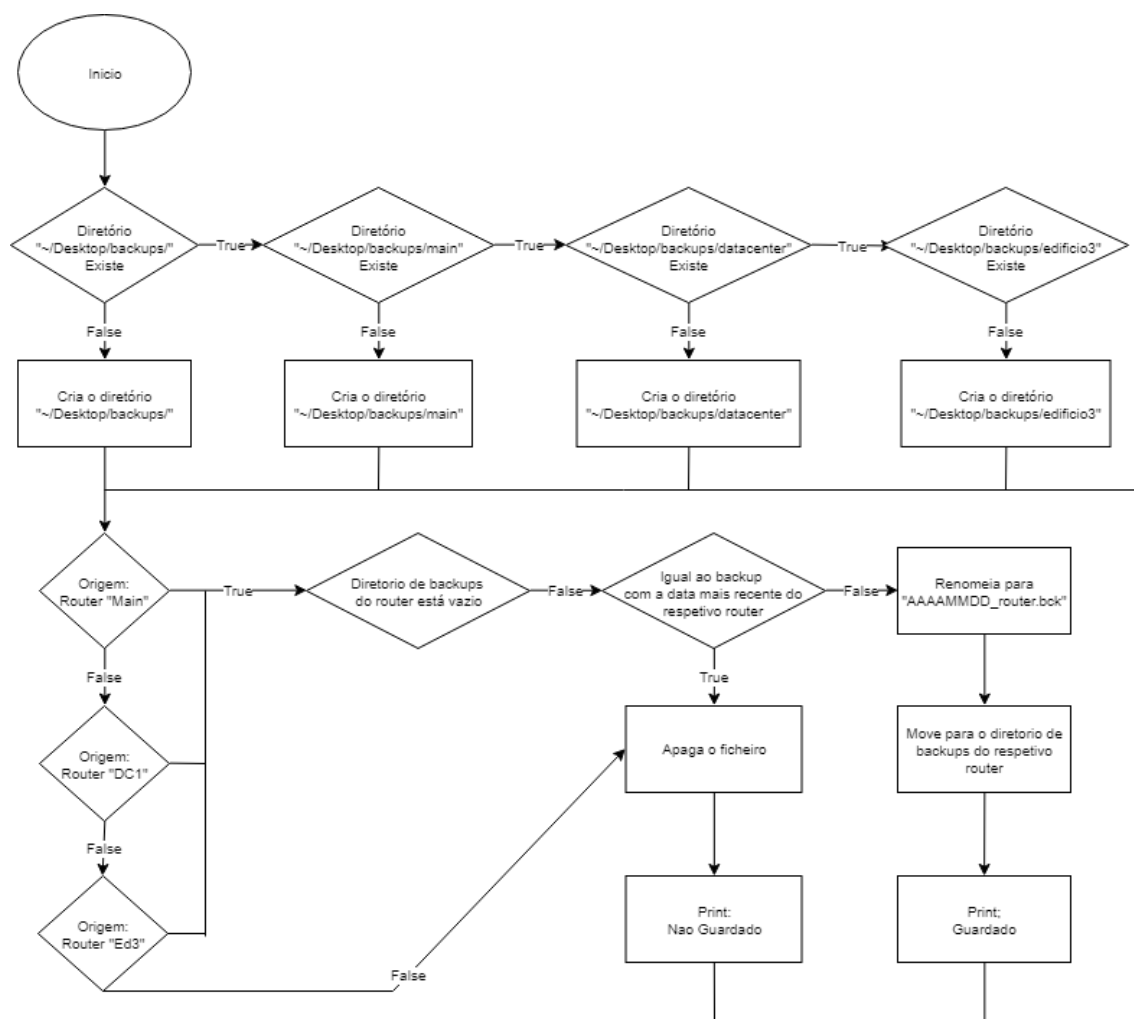


Figura 32 - Fluxograma do Script

### 3.3.6. VPN

Uma Rede privada virtual (*Virtual Private Network*, VPN) é uma rede de comunicações privada construída sobre uma rede de comunicações pública. O tráfego de dados é levado pela rede pública utilizando protocolos padrões. Cria uma conexão segura e criptografada, que pode ser considerada como um túnel, entre o computador e um servidor configurado com o serviço VPN.

Uma VPN é uma conexão estabelecida sobre uma infraestrutura pública ou compartilhada, usando tecnologias de tuneis e criptografia para manter

seguros os dados trafegados. Redes privadas virtuais seguras usam protocolos de criptografia por tuneis que fornecem a confidencialidade, autenticação e integridade necessárias para garantir a privacidade das comunicações requeridas. Alguns dos protocolos normalmente usados na implementação de uma VPN são o Point-to-Point Tunneling Protocol (PPTP), o IP Security Protocol (IPsec) e o OpenVPN. Aquando da configuração adequada da VPN, estes protocolos podem assegurar comunicações seguras através de redes inseguras.

Para o desenvolvimento da VPN na rede proposta, o grupo optou pela escolha do protocolo Point-to-Point Tunneling Protocol (PPTP). Apesar de ser um protocolo um pouco desatualizado e com alguns problemas de segurança, observamos que a Universidade de Aveiro ainda usa este tipo de protocolo de VPN e que a sua implementação é relativamente rápida.

Configuração:

1º - (terminal) **sudo apt-get install pptpd**

Este comando significa que ao selecionar o servidor no qual será implementado a VPN, este ficará responsável pelo tratamento de IPs e pela autenticação de todos os seus servidores na sua VPN.

2º - (terminal) **sudo gedit /etc/pptp.conf**

**Localip 192.168.20.1**

**Remoteip 192.168.20.100-200**

Com este comando irá se editar os IPs atribuídos ao LocalIP e remotelP. LocalIP será o Ip do servidor e remotelP serão os IPs que serão atribuídos aos clientes que se pretendam conectar a este servidor.

3º - (terminal) **sudo gedit /etc/ppp/chap-secrets**

Este comando irá permitir a criação do username e da password. Foi usada a seguinte nomenclatura: user pptpd user \*. O username e password foi atribuido "user" o pptpd é o servidor em questão e o \* significa que um cliente pode ter acesso a este servidor a partir de qualquer lado.

4º - (terminal) **sudo gedit /etc/ppp/pptpd-options**

**ms-dns 192.168.1.2**

Com este comando estamos a adicionar os dns's da Google

5º - (terminal) **sudo ptpd restart**

Como este comando vamos reiniciar o servidor ptp

6º - (terminal) **sudo gedit /etc/sysctl.conf**

**Net.ipv4.ip\_forward = 1**

Este comando irá permitir encaminhar pacotes entre IPs públicos e privados que são definidos com PPTP.

7º - (terminal) **sudo sysctl -p**

Este comando irá tornar ativo o comando anterior

8º - (terminal) **sudo iptables --table nat --append POSTROUTING --out-interface ppp0 -j MASQUERADE**

9º - (terminal) **sudo iptables -I INPUT -s 192.168.20.0/24 -i ppp0 -j**

**ACCEPT**

10º - (terminal) **sudo iptables --append FORWARD --in-interface enp0s3 -j ACCEPT**

Os comandos 8,9 e 10 têm como objetivo a permissão dos clientes PPTP falarem uns com os outros.

11º - (terminal) **sudo iptables-save > save**

Este comando serve para que as iptables fiquem guardadas e não seja preciso, sempre que se reiniciar o servidor, escrever os comandos 8,9 e 10. Posto isto, sempre que se reiniciar o servidor deve ser feito o seguinte comando no terminal: **sudo iptables-restore < save**. O grupo optou por gravar o ficheiro onde ficam guardadas as iptables de "save", no entanto, este nome não é obrigatório, ou seja, a escolha do nome pode ser facultativa.

### 3.3.7. Proxy

Um servidor proxy é um sistema de computadores, ou uma aplicação que atua como intermediário entre os pedidos de recursos de outros servidores, efetuados pelos clientes. Este simula um novo cliente, com o próprio endereço IP e efetua a transferência dos pacotes. Quando os pacotes chegam ao servidor proxy, estes são redirecionados para o endereço original que efetuou o pedido em primeiro lugar.

#### **Configurações:**

1º - (terminal) **sudo apt-get install squid**

Este comando instala o servidor Squid na máquina.

2º - (terminal) **cd /etc/squid**

3º - (terminal) **sudo cp squid.conf squid.conf.backup**

Criar uma cópia do ficheiro de configurações que servirá de backup do original

4º - (terminal) **sudo gedit squid.conf &**

Editar o ficheiro de configurações

5º - (squid.conf)

linha 971

descomentar **acl localnet src 10.0.0.0/8**

linha 976

**acl localnet src 192.168.1.0/24**

linha 991

**acl block\_websites dstdomain .msn.com .espn.com #http**

**http\_access deny block\_websites**

Estes comandos criam uma lista de websites com protocolo http que irá ser bloqueada, bem como o acesso aos mesmos.

**acl block\_websites2 dstdomain .facebook.com .twitter.com**

**#https**

**http\_reply\_access deny block\_websites2**

**http\_access deny CONNECT block\_websites2**

Estes comandos criam uma lista de websites com protocolo https que irá ser bloqueada, bem como o acesso aos mesmos.

A figura 33 exemplifica o bloqueio de uma ligação.



*Figura 33 - Exemplo de bloqueio de um website*

linha 1194

descomentar **http\_access allow localnet**

7º - (terminal) **sudo pkill -9 squid**

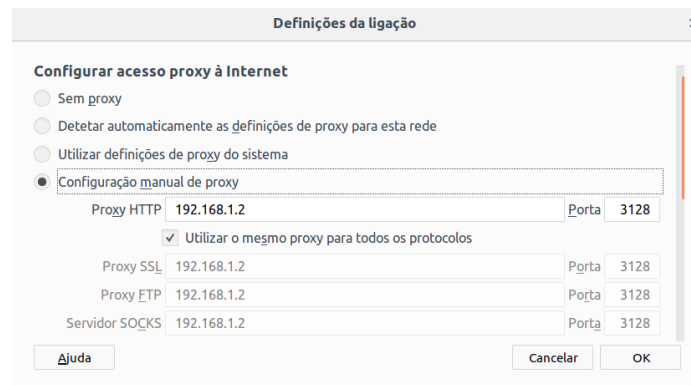
8º - (terminal) **sudo /usr/sbin/squid**

9º - Como a Figura 34 demonstra, seguir os seguintes passos:

(Firefox) **Abrir menu -> Preferências -> Proxy de rede (Definições) ->**

**Configuração manual de proxy**

**-> Proxy HTTP 192.168.1.2 (ip serverPTRC) Porta 3128 -> Ativar:  
Utilizar o mesmo proxy para todos os protocolos; Desativar outras opções -  
> [OK]**



*Figura 34 - Definições da ligação*

### 3.3.8. Firewall

Nesta secção do presente relatório pretende-se abordar a Firewall, fazendo primeiramente referência a uma breve introdução acerca do mesmo e o porquê da sua utilização em vez de outro, e para finalizar o modo como foi implementado no projeto apresentado.

Uma firewall é um sistema de segurança que monitoriza e controla o tráfego numa rede de computadores. Pode ser implementada via hardware ou software. Para este projeto foi escolhida a última opção, através de ACL's.

Decidimos usar ACL'S pois temos mais conhecimentos em relação a esse método de trabalho, e guiões fornecidos pelo docente da unidade curricular Tecnologias de Redes de Computadores.

Para criar a firewall foi necessário criar regras que permitissem passar certos tipos de serviços para fora da rede. (Figura 35).

```
Main#access-list 110 permit tcp any any eq 80 #http
Main#access-list 110 permit tcp any any eq 53 #DNS
Main#access-list 110 permit tcp any any eq 21 #FTP
Main#access-list 110 permit tcp any any eq 25 #SMTP
Main#access-list 110 permit tcp any any eq 23 #telnet
Main#access-list 110 permit tcp any any eq 143 #IMAP
Main#access-list 110 permit tcp any any eq 1723 #vpn
Main#access-list 110 permit ospf any any
Main#access-list 110 permit icmp any any
Main#access-list 110 deny ip any any
```

*Figura 35 - Firewall (Regras)*

Para testar esta configuração, realizámos o envio de tráfego de cada tipo de serviço pelas diferentes zonas da rede para o exterior, e verificámos que a comunicação era executada com sucesso.

### 3.3.9. WebServer

Nesta secção do presente relatório pretende-se abordar o *Webserver*, fazendo primeiramente referência a uma breve introdução acerca do mesmo, e de seguida ao modo como foi implementado no projeto apresentado.

Um *webserver* é, em termos de *hardware*, uma máquina que armazena arquivos que compõem os websites (por exemplo, documentos HTML, imagens, folhas de estilo, e arquivos JavaScript) e em termos de software, controla como os clientes acedem aos arquivos hospedados. Isto é, um *webserver* serve principalmente para armazenar, processar e entregar páginas web aos clientes (Figura 36).

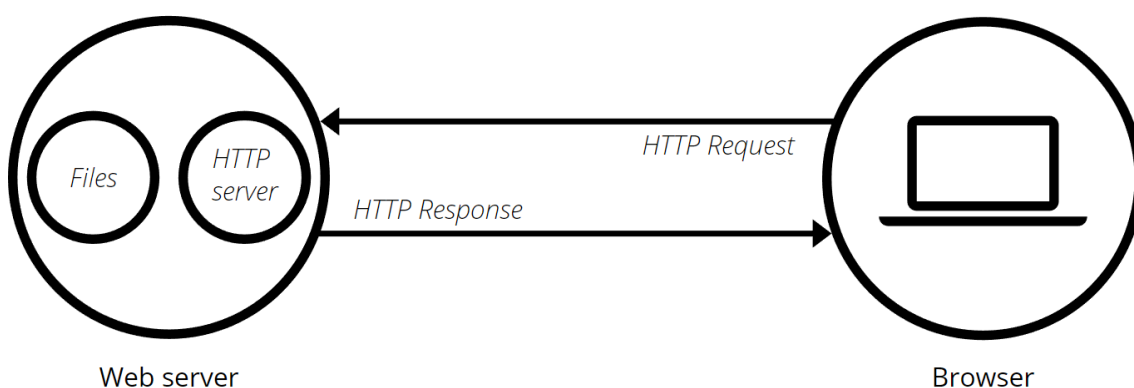
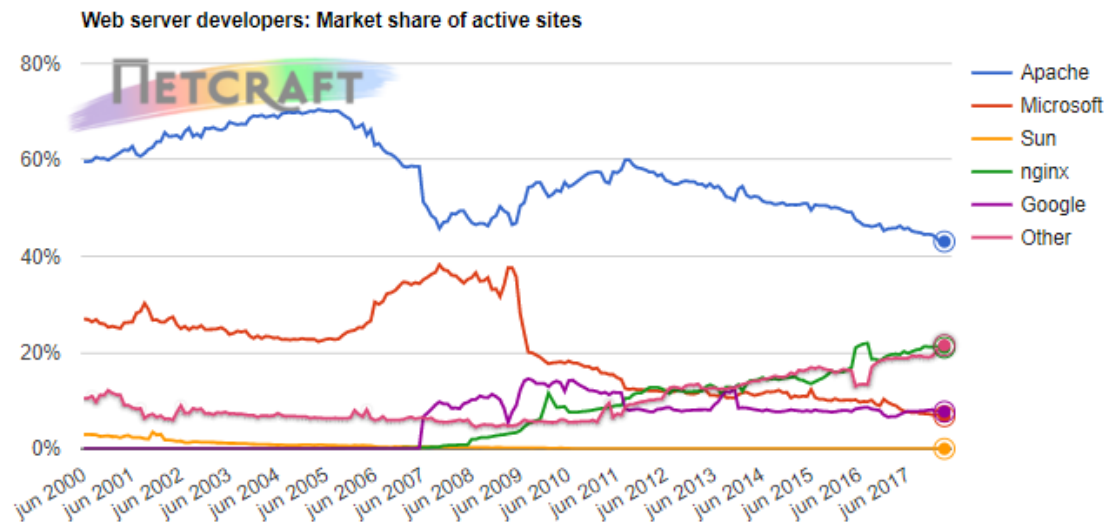


Figura 36 - Esquema do funcionamento básico de um webserver (Fonte: [https://developer.mozilla.org/pt-BR/docs/Learn/Common\\_questions/o\\_que\\_e\\_um\\_web\\_server](https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/o_que_e_um_web_server))

Optámos por utilizar o Apache HTTP Server como o nosso *webserver*, pois é o “*industry standard*”, muito completo, grátis e o mais popular e utilizado, o que leva a que exista muita documentação sobre o mesmo (Figura 37).



Developer	March 2018	Percent	April 2018	Percent	Change
Apache	76,398,184	43.03%	75,298,051	42.41%	-0.62
nginx	37,321,104	21.02%	37,478,429	21.11%	0.09
Google	13,684,777	7.71%	14,159,867	7.97%	0.27
Microsoft	11,986,413	6.75%	11,935,138	6.72%	-0.03

Figura 37 - Market share do Apache em Abril de 2018 (Fonte: <https://news.netcraft.com/archives/2018/04/26/april-2018-web-server-survey.html>)

## Configuração

**1º Passo** - Garantir que o servidor tem um IP estático.

**2º Passo** - (TERMINAL) `sudo apt install apache2`

Comando para instalar o Apache

**3º Passo** - Inserir o website desejado no diretório `/var/www/html`





## 4. Análise dos resultados

Nas tabelas abaixo, Tabelas 6 e 7, estão indicados os requisitos que foram cumpridos (representados com o símbolo "+") e os que não foram cumpridos (representados com o símbolo "-").

*Tabela 6 - Análise do cumprimento dos requisitos; RF - requisito funcional*

<b>Tipo de Requisitos</b>	<b>Descrição</b>	<b>Análise de Resultados</b>
<b>RF1</b>	A intranet deverá ser segmentada e composta por várias redes IP interligadas entre si	<b>+</b>
<b>RF2</b>	Deverá existir uma rede IP associada a cada departamento	<b>+</b>
<b>RF3</b>	Deverá existir uma rede IP que abranja todas as redes IP existentes num determinado edifício	<b>+</b>
<b>RF4</b>	Todas as redes IP deverão utilizar tecnologia cablada Ethernet	<b>+</b>
<b>RF5</b>	Para além da rede IP cablada, deverá também existir uma rede IP wireless nos departamentos comercial e de Engenharia	<b>-</b>
<b>RF6</b>	A interligação da Intranet com redes externas (e.g. ISP) deverá ser realizada apenas através do edifício 1	<b>+</b>
<b>RF7</b>	Todos os edifícios e departamentos deverão ser interligado e possuir conectividade entre si	<b>+</b>
<b>RF8</b>	Por motivos de segurança, utilizadores ligados à rede Wi-Fi dos departamentos Comercial e de Engenharia deverão apenas ter acesso à rede interna do próprio departamento	<b>-</b>
<b>RF9</b>	A intranet deverá ser baseada em redes IP classe C	<b>+</b>
<b>RF10</b>	Funcionários da empresa deverão conseguir aceder à intranet através do exterior	<b>+/-</b>
<b>RF11</b>	Deverão ser instalados serviços de DNS que efetuem a resolução direta e inversa dos servidores do datacenter	<b>+</b>

Tabela 7 - Análise do cumprimento dos requisitos; RNF - requisito não funcional

Tipo de Requisitos	Descrição	Análise de Resultados
<b>RNF1</b>	Os endereços de rede atribuídos às diversas redes IP deverão refletir e identificar um determinado edifício e departamento	+
<b>RNF2</b>	Deverão ser fornecidos endereços IP dinâmicos a todas as máquinas cliente	+
<b>RNF3</b>	É desejável a utilização de encaminhamento dinâmico na intranet	+
<b>RNF4</b>	Foi contratado um único endereço IP público (ISP_IP1) ao ISP	+
<b>RNF5</b>	Deverá ser implementado um serviço de proxy	+
<b>RNF6</b>	Os web browsers deverão detetar de forma automática a configuração do proxy	-
<b>RNF7</b>	A rede empresarial deverá estar protegida por firewall, mantendo abertos os portos estritamente essenciais e necessários ao funcionamento dos diversos serviços de rede	+
<b>RNF8</b>	Serviço de Trivial File Transfer Protocol (TFTP), para fazer o backup automático das configurações dos routers	+
<b>RNF9</b>	Visualizador central de alertas, utilizando o Simple Network Management Protocol (SNMP), para alertar o gestor da rede sempre que haja eventos anómalos nos equipamentos de rede	-
<b>RNF10</b>	Estes serviços devem existir numa máquina dedicada à recolha, armazenamento e visualização desta informação	+
<b>RNF11</b>	As máquinas e routers sob consulta devem estar elencados num ficheiro de configuração csv, carregado automaticamente, com o seguinte formato: Nome, ip, período_em_segundos	-
<b>RNF12</b>	Os backups devem ser guardados num diretório para cada router (diretório criado automaticamente com o nome do router) e os ficheiros devem ter o formato AAAAMMDD.bck (AAAA-ano MM-mês DD-dia)	+
<b>RNF13</b>	Sempre que haja um novo registo, este só deve ser guardado se houver de facto uma alteração na configuração do router. Caso contrário mantém a configuração anterior como a mais recente	+

<b>RNF14</b>	O serviço de anomalias deve executar um daemon e um visualizador ligados por sockets	-
<b>RNF15</b>	O daemon fica encarregue de obter periodicamente o estado das máquinas, registar esse estado num ficheiro de log e enviar uma notificação ao visualizador sempre que haja uma mudança no estado de uma máquina	-
<b>RNF16</b>	O daemon também deverá ser capaz de receber um pedido do visualizador para que lhe seja enviado o estado de todas as máquinas	-
<b>RNF17</b>	O visualizador deve apresentar no ecrã (terminal ou GUI) o estado atual de todas as máquinas. Para isso tem de pedir esse estado no arranque e manter o ecrã atualizado quando chegam atualizações	-



## 5. Reflexão crítica e conclusões

No decorrer do presente relatório, pretendeu-se descrever a estrutura do projeto durante o seu desenvolvimento, bem como das etapas inerentes à criação da rede. No planeamento previsto, como o próprio nome indica, prevê-se o rumo do projeto, não se tendo uma perceção total do tipo de imprevistos que podem ocorrer ao longo da realização do mesmo. As tarefas têm uma duração estimada, mas no fim acabam por durar, geralmente, mais tempo, como aconteceu com o desenvolvimento da rede. Assim, o planeamento inicial serve essencialmente para organizar as tarefas e a sua distribuição, pois a duração não é certa. Contudo no final do projeto, consideramos que se conseguiu cumprir as tarefas e os requisitos, no geral, propostos.

Relativamente ao software de gestão de projetos, Microsoft Project, este é de fácil manuseamento, permitindo ao utilizador organizar as suas tarefas de acordo com as suas relações de precedência e durações (colocando data de começo e finalização da tarefa). No caso de projetos com múltiplas tarefas, ajuda na observação da evolução de cada uma delas e consequente evolução do projeto, podendo serem efetuados ajustes à medida que determinados problemas venham a ocorrer.

Quando feita a questão ao nosso orientador sobre qual sistema operativo utilizar, o mesmo respondeu que para o âmbito do projeto qualquer um serviria. No entanto também disse que seria mais desafiante utilizar Linux. Depois de ponderadas todas as vantagens e desvantagens de cada sistema operativo, e como forma de expandirmos os nossos conhecimentos, decidimos aceitar o desafio e correr nos nossos servidores uma distribuição de Linux - Ubuntu 16.04.

Relativamente ao servidor DNS, concluiu-se que, apesar de funcional, aquando a realização da pesquisa inversa do endereço 192.168.1.3, correspondente ao endereço IP do webserver, é devolvido o nome “webserver.datacenter” em vez de “www.nankaj.com”. Para além dos nomes internos já realizados, seria possível ainda expandir e adicionar nomes para os edifícios associando-os à *gateway* dos mesmos

Quanto ao protocolo TFTP, a principal dificuldade desta secção de trabalho residiu na parte de *scripting*, onde as permissões das pastas e dos

ficheiros foram um desafio. Ainda nesta parte, alguns aspetos podem ser melhorados, tais como algumas validações que ainda podem ser realizadas de forma a torná-lo mais eficiente. O comando archive apenas funciona em routers que estejam a correr uma versão do Cisco IOS que seja igual ou superior à 12.4.

No que respeita ao dispositivo de segurança de rede, Firewall, notou-se que inicialmente houve certos problemas em saber quais os serviços essenciais para elaborar as Access lists (ACLs), dificuldade esta que foi ultrapassada.

Relativamente ao servidor Proxy, surgiram inicialmente alguns problemas em tentar cumprir um dos requisitos, que os web browsers detetassem automaticamente a sua configuração, o que provocou que assim fosse realizada, mas manualmente.

## 6. Bibliografia

- Comparison of DNS server software.* (2018). Obtido de Wikipedia:  
[https://en.wikipedia.org/wiki/Comparison\\_of\\_DNS\\_server\\_software](https://en.wikipedia.org/wiki/Comparison_of_DNS_server_software)
- ISC DHCP.* (7 de Setembro de 2017). Obtido de Internet Systems Consortium:  
<https://www.isc.org/downloads/dhcp/>
- Nunes, C. (s.d.). Resolução de Nomes - Interativa. Obtido de  
[https://www.inf.pucrs.br/~cnunes/redes/aulas/DNS\\_6.pdf](https://www.inf.pucrs.br/~cnunes/redes/aulas/DNS_6.pdf)
- Nunes, C. (s.d.). Resolução de Nomes - Recursiva. Obtido de  
[https://www.inf.pucrs.br/~cnunes/redes/aulas/DNS\\_6.pdf](https://www.inf.pucrs.br/~cnunes/redes/aulas/DNS_6.pdf)
- Proxies o que são?* (12 de Outubro de 2009). Obtido de PPLWARE:  
<https://pplware.sapo.pt/informacao/proxies-o-que-sao/>



## Anexos

### Script python

```
#!/usr/bin/env python
# coding=utf-8

import os, time

#Definicao de Funcoes
def dirExists( dir ):
    dir = os.path.expanduser(dir)
    if os.path.isdir(dir) == False:
        os.mkdir(dir)
        os.chmod(dir, 0o777)
    return

def discardLastBck( router ):
    dir = os.path.expanduser("~/Desktop/backups/" + router + "/")
    if len(os.listdir(dir)) == 0:
        return False
    else:
        mostRecentBckDate = None
        mostRecentBckName = None
        for bck in os.listdir(dir):
            bckDate = bck.split("_")
            bckDate = int(bckDate[0])
            if bckDate >= mostRecentBckDate or mostRecentBckDate
            == None:
                mostRecentBckDate = bckDate
                mostRecentBckName = dir + bck

        f1 = open(file, "r")
        fileContent = f1.read()
        f1.close()
        f2 = open(mostRecentBckName, "r")
```

```

        mostRecentBckContent = f2.read()
        f2.close()

        if fileContent == mostRecentBckContent:
            return True
        else:
            return False

def deleteLastBck( router ):
    os.remove(file)
    data = time.strftime("%d/%m/%Y")
    print "Router=> {0}\t\tData=> {1} Backup=> Nao
Guardado".format(router, data)

def moveToFolder( router ):
    data = time.strftime("%Y%m%d")
    newName = os.path.expanduser("~/Desktop/backups/" + router + "/" ) +
data + "_" + router + ".bck"
    os.rename(file, newName)
    os.chmod(newName, 0o777)
    data = time.strftime("%d/%m/%Y")
    print "Router=> {0}\t\tData=> {1} Backup=> Guardado".format(router,
data)

#Verifica se os diretorios existem
dirExists("~/Desktop/backups/")
dirExists("~/Desktop/backups/main/")
dirExists("~/Desktop/backups/datacenter/")
dirExists("~/Desktop/backups/edificio3/")

#Corre o programa
while True:
    for file in os.listdir("/srv/tftp/"):
        if "main" in file:
            file = "/srv/tftp/" + file
            if discardLastBck("main") == True:
                deleteLastBck("main")
            else:
                moveToFolder("main")
        elif "datacenter" in file:

```

```
file = "/srv/tftp/" + file
if discardLastBck("datacenter") == True:
    deleteLastBck("datacenter")
else:
    moveToFolder("datacenter")
elif "edificio3" in file:
    file = "/srv/tftp/" + file
    if discardLastBck("edificio3") == True:
        deleteLastBck("edificio3")
    else:
        moveToFolder("edificio3")
else:
    deleteLastBck("Desconhecido")
time.sleep(5)
```