

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

**Soluções *Open-source* para os
Serviços de Fax e *VPN*
numa Rede Empresarial**

Victor Braga

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major em Telecomunicações

Orientador: (Prof. Doutor) José Magalhães Cruz

Orientador: (Eng.) Fernando Vieira

Janeiro, 2012

A Dissertação intitulada

“Soluções Open-Source para os Serviços de Fax e VPN numa Rede Empresarial”

foi aprovada em provas realizadas em 17-02-2012


o júri



Presidente Professor Doutor Ricardo Santos Morla
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor José Carlos Leite Ramalho
Professor Auxiliar Departamento de Informática da Universidade do Minho



Professor Doutor José Manuel Magalhães Cruz
Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de
Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - Vitor Manuel Braga da Silva

Faculdade de Engenharia da Universidade do Porto

Resumo

O crescente movimento "*Open-Source*" tem produzido soluções de valor para diferentes áreas mas a sua implementação em ambiente empresarial ainda é reduzida.

As principais vantagens passam pelas sinergias, em termos de construção de soluções integradas e altamente personalizadas e adaptadas à lógica do negócio que se quer desenvolver. Além disso factores como desempenho, escalabilidade e resiliência conferem a estas plataformas uma vantagem competitiva no mercado empresarial.

Este projecto visa explorar e implementar essas mesmas soluções, nomeadamente nas áreas de *VoIP* e Redes Privadas Virtuais, colocando-as em produção, num ambiente com um patamar de exigência tão elevado como o de uma empresa.

Abstract

The growing *Open-Source* movement has produced value solutions for different areas but its implementation in business environment is still very low.

The main advantages are the synergies, in terms of building integrated solutions and highly personalized and tailored to the business logic that you want to develop. In addition factors such as performance, scalability and resilience of these platforms provide a competitive advantage in the business market.

This project aims to explore and implement those solutions, particularly in the areas of *VoIP* and Virtual Private Networks, putting them into production in an environment with a requirement level as high as such of a enterprise.

Agradecimentos

Em primeiro lugar gostaria de agradecer ao meu co-orientador, Eng. Fernando Vieira, pelo desafio que me lançou, pelo apoio e pela paciência que me dispensou ao longo deste percurso, que em muito me fez crescer...

Ao orientador, Professor José Magalhães, quero deixar o meu sincero agradecimento pela forma como orientou o meu trabalho, pelas críticas e sugestões realizadas no sentido de alcançar o êxito neste trabalho. Agradeço também a leitura e revisão efectuadas deste documento.

Quero também agradecer aos meus pais, Victor e Irene, pelo apoio que me deram e pelo esforço que realizaram para eu poder chegar até aqui.

Agradeço aos meus companheiros de curso, Paulo Vaz, Ricardo Faria, Francisco Barbosa e Jorge Carvalho e aos meus colegas na *UQ*, Gabriel Santos, Rui Lino, Miguel Duarte, Paulo Jorge Ribeiro, Edgar Gonçalves e Osvaldo Gomes pelo apoio que deram ao longo deste trajecto.

Uma última palavra de reconhecimento por todo o seu amor e compreensão para a Vera.

O Autor

“Science can amuse and fascinate us all, but it is engineering that changes the world.”

Isaac Asimov

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Descrição do Problema	3
1.3	Estratégia de Resolução	6
2	Estado da Arte	9
2.1	VoIP	9
2.1.1	SIP	9
2.1.2	RTP	13
2.1.3	IAX	14
2.1.4	Asterisk	16
2.1.5	Fax via <i>VoIP</i>	18
2.2	VPN - Redes Privadas Virtuais	20
2.2.1	OpenVPN	21
2.2.2	IPsec	22
2.2.3	Distribuições de <i>Firewall/Router</i>	24
2.2.3.1	PfSense	24
2.2.3.2	Endian	24
2.2.3.3	Vyatta	25
3	Caracterização do sistema	27
3.1	Listagem de Requisitos	27
3.1.1	Requisitos Funcionais	27
3.1.2	Requisitos Não Funcionais	28
3.2	Escolhas tecnológicas	28
3.2.1	Interface Web	28
3.2.2	Servidor de Páginas	29
3.2.3	Servidor base de dados	29
3.2.4	Software <i>PBX</i>	29
3.2.5	Servidor de <i>VPN</i>	29
3.2.6	Solução de Alta-Disponibilidade	30
3.3	Arquitetura do sistema	30
3.3.1	Servidor de Fax	30
3.3.2	Servidor de <i>VPN</i>	33
4	Desenvolvimento do Sistema	35
4.1	Base de Dados	35
4.2	Servidor Web	37

4.2.1	Interface de administração do servidor Web	37
4.3	<i>jhEDI</i>	42
4.4	<i>jhAS400</i>	47
4.5	<i>Email-to-Fax</i>	50
4.6	<i>Fax-to-Email</i>	51
4.7	<i>Taxação</i>	52
4.8	Servidor <i>OpenVPN</i>	52
4.8.1	Interface de Administração do servidor <i>OpenVPN</i>	53
4.8.2	<i>Heartbeat</i> - Alta Disponibilidade	55
5	Testes e Resultados	57
5.1	Envio e Recepção de Fax via Operador <i>SIP</i>	57
5.2	Envio de Fax via <i>jhAS400</i>	59
5.3	Dados estatísticos do envio de Encomenda via <i>jhEDI</i>	60
5.4	Dados estatísticos do envio de Fax via <i>jhAS400</i> e via Email	60
5.5	Dados estatísticos da recepção de Faxes	61
5.6	Testes de conectividade cliente- <i>VPN</i> -sede	61
5.7	Testes de conectividade cliente- <i>VPN</i> -Escritório Remoto	62
5.8	Testes de <i>throughput</i> cliente- <i>VPN</i> -sede	62
5.9	Dados estatísticos da <i>VPN</i>	63
5.10	Discussão de Resultados	65
6	Conclusões	67
6.1	Síntese do Trabalho Desenvolvido	67
6.2	Desenvolvimento Futuro	68
A	Caracterização do sistema	69
A.1	Implementação	69
A.1.1	Asterisk	69
A.1.2	<i>Hylafax</i>	75
A.1.3	<i>IAXModem</i>	81
A.1.4	<i>Postfix</i>	81
A.1.5	<i>Nginx</i>	81
A.1.6	<i>jhAS400</i>	84
A.1.7	<i>OpenVPN</i>	86
A.1.8	<i>Heartbeat</i>	87
	Referências	88

Lista de Figuras

1.1	Arquitectura da rede <i>VoIP</i> a implementar	3
1.2	Estrutura da Rede <i>VPN</i> a implementar	4
2.1	Arquitectura de uma rede <i>SIP</i>	11
2.2	Fluxo de mensagens no estabelecimento de uma chamada <i>SIP</i> [1]	11
2.3	Cabeçalho do Pacote <i>RTP</i>	14
2.4	Fluxo de mensagens no estabelecimento de uma chamada <i>IAX</i> [2]	15
2.5	Centrais Telefónicas - Percentagem de Mercado	17
2.6	Distribuição de número de telefones por Sistema	18
2.7	Cenário <i>Lan-to-Lan</i>	20
2.8	Cenário <i>Roadwarrior</i>	20
2.9	Cabeçalho <i>IP</i> com <i>ESP</i>	22
2.10	Cabeçalho <i>IP</i> com <i>AH</i>	23
3.1	Modelo <i>MVC</i> do sistema	31
3.2	Arquitectura do sistema de Fax	31
3.3	Casos de Uso - Servidor de Fax	32
3.4	Arquitectura da plataforma de <i>VPN</i>	33
3.5	Casos de Uso - Servidor de <i>VPN</i>	34
4.1	Modelo da Base de Dados	35
4.2	Menu de configuração de Utilizadores	37
4.3	Menu de configuração de Utilizadores	37
4.4	Menu de configuração de Utilizadores - Adicionar Utilizador	38
4.5	Menu de configuração de Servidores <i>VoIP</i>	38
4.6	Menu de configuração de Servidores <i>VoIP</i> - Adicionar Servidor <i>SIP</i>	39
4.7	Menu de configuração de Servidores <i>VoIP</i> - Adicionar Servidor <i>IAX</i>	39
4.8	Menu de configuração de Modems	40
4.9	Menu de configuração de Modems - Adicionar Modem	40
4.10	Menu de Envio de FAX	41
4.11	Menu de Estatísticas	41
4.12	Mensagem de <i>email</i> gerada pelo Servidor de Encomendas	42
4.13	Mensagem de <i>email</i> gerada pelo Servidor de Encomendas	42
4.14	Mensagem de <i>email</i> gerada pelo Servidor de Encomendas	43
4.15	Fluxograma da Aplicação <i>jhEDI</i>	44
4.16	Diagrama de Sequência Temporal - Envio de Encomenda via Fax	45
4.17	Diagrama de Sequência Temporal - Envio de Encomenda via Fax e Email	45
4.18	Diagrama de Sequência Temporal - Envio de Encomenda via Email	46
4.19	Fluxograma da Aplicação <i>jhAS400</i>	47

4.20	Envio de Fax via ERP - Listagem de Impressoras	48
4.21	Envio de Fax via ERP - Escolha de Ficheiro	48
4.22	Envio de Fax via ERP - Alteração da Impressora	49
4.23	Envio de Fax via ERP - Confirmação do envio	49
4.24	<i>Email-to-Fax</i> - Arquitectura da solução	50
4.25	<i>Email-to-Fax</i> - Diagrama Temporal	50
4.26	<i>Fax-to-Email</i> - Arquitectura da solução	51
4.27	<i>Email-to-Fax</i> - Diagrama Temporal	51
4.28	Arquitectura da plataforma de <i>VPN</i> - Sincronismo	52
4.29	Criação de uma <i>VPN</i>	53
4.30	<i>Webmin</i> - Menu de configuração de gestão de certificados	54
4.31	<i>Webmin</i> - Menu de configuração de <i>VPN's</i>	54
4.32	<i>Webmin</i> - Estatísticas das <i>VPN's</i>	55
4.33	<i>Heartbeat</i> - Exemplo de Implementação	55
5.1	Envio de Fax - Estabelecimento da chamada <i>SIP</i>	57
5.2	Receção de Fax - Receção da chamada através de um canal <i>SIP</i>	58
5.3	Envio de Fax - Estabelecimento da chamada <i>SIP</i> canal <i>RTP</i>	58
5.4	Envio de Fax via <i>jhAS400</i>	59
5.5	Gráfico da utilização de largura da banda pelo servidor de <i>VPN</i> - Semana	63
5.6	Gráfico da utilização de largura da banda pelo servidor de <i>VPN</i> - Mês	63
5.7	Relatório do número de conexões activas do servidor de <i>VPN</i>	64
A.1	<i>Asterisk Menuselect 1</i>	70
A.2	<i>Asterisk Menuselect 2</i>	70
A.3	<i>Asterisk Menuselect 3</i>	71
A.4	<i>Asterisk Menuselect 4</i>	71
A.5	<i>Asterisk Menuselect 5</i>	72
A.6	<i>Asterisk Menuselect 6</i>	72
A.7	<i>Asterisk Menuselect 7</i>	73
A.8	<i>Asterisk Menuselect 8</i>	73

Lista de Tabelas

1.1	Dimensão do Projecto	5
2.1	Distribuição de Sistemas " <i>Open-Source</i> " PBX por tamanho da Empresa	17
5.1	Distribuição de Fax-Encomendas por Mês	60
5.2	Distribuição de Faxes por Mês	60
5.3	Distribuição de Faxes Recebidos por Mês	61
5.4	Distribuição de Fax/Aplicação por Mês	65

Lista de abreviaturas e siglas

3GPP	3rd Generation Partnership Project
ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
AES	Advanced Encryption Standard
AH	Authentication Header
ASCII	American Standard Code for Information Interchange
ASSOFT	Associação Portuguesa de Software
BGP	Border Gateway Protocol
BSD	Berkeley Software Distribution
CA	Certificate Authority
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
DOC	Microsoft Word file extension
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECM	Error Correcting Mode
ERP	Enterprise Resource Planning
ESP	Encapsulating Security Payload
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IAxv2	Inter Asterisk eXchange - versão 2
ICMP	Internet Control Message Protocol

IDE	Integrated Development Environment
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	Internet Protocol Security
ISP	Internet Service Provider
IVR	Interactive Voice Response
JDBC	Java Database Connectivity
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MGCP	Media Gateway Control Protocol
MVC	Model–View–Controller
NAS	Network Attached Storage
NAT	Network Address Translation
NFS	Network File System
NTP	Network Time Protocol
ODT	Open Document Word Processing file format
ODS	Open Document Spreadsheet file format
OSPF	Open Shortest Path First
PBX	Private Branch Exchange
PDF	Portable Document Format
PHP	Hypertext Preprocessor
PPTP	Point-to-Point Tunneling Protocol
PS	PostScript
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RTP	Real-Time Transport Protocol
RTCP	RTP Control Protocol
SCCP	Signalling Connection Control Part
SCTP	Stream Control Transmission Protocol

SDP	Session Description Protocol
SIP	Session Initiation Protocol
S/MIME	Secure/Multipurpose Internet Mail Extensions
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
TAP	Virtual Network Kernel Devices, Layer 2
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TIFF	Tagged Image File Format
TLS	Transport Layer Security
TUN	Virtual Network Kernel Devices, Layer 3
UDP	User Datagram Protocol
UM	Unified Messaging
UMTS	Universal Mobile Telecommunications System
VIP	Virtual IP
VLR	Visitor Location Register
VoIP	Voice over IP
VPN	Virtual Private Network
WWW	World Wide Web
XLS	Microsoft Excel file extension

Capítulo 1

Introdução

Este capítulo irá fornecer uma visão global do trabalho que se pretende realizar.

1.1 Motivação

O crescente movimento "*Open-Source*" tem produzido soluções de valor para diferentes áreas mas a sua implementação em ambiente empresarial ainda é reduzida. Algumas das razões referidas pelos decisores focam a ausência de documentação, a inexistência de uma entidade comercial que possam abordar e questionar, chegando alguns até a colocar em questão a segurança das aplicações. Apesar de nem todos os projectos apresentarem, à partida, uma documentação estruturada a maioria apoia-se no conceito de comunidade para produzir um conjunto de documentação útil, quer na forma de manual formal, quer recorrendo a fóruns de discussão.

É este constante escrutínio público que permite assegurar a contínua segurança, robustez e melhoria das aplicações.

Apesar de muitos focarem o carácter gratuito como a principal vantagem destas plataformas, essa abordagem acaba por ser redutora e errónea, uma vez que nem todas as aplicações são de uso gratuito especialmente em ambiente empresarial. As principais vantagens passam pelas sinergias, em termos de construção de soluções integradas e altamente personalizadas e adaptadas à lógica do negócio que se quer desenvolver. Além disso factores como desempenho, escalabilidade e resiliência conferem a estas plataformas uma vantagem competitiva no mercado empresarial.

Hoje, tal como foi referido, o movimento "*Open-Source*" produz soluções de alta qualidade para múltiplas áreas de actividade. Tomemos como exemplo, as distribuições de *Linux Debian* e *Ubuntu* que têm vindo a conquistar quota de mercado, não apenas em servidores mas também em computadores pessoais. Se adicionarmos a qualquer uma destas distribuições de *Linux* ferramentas de produtividade como *OpenOffice*, *Mozilla-Firefox* ou *Opera*, *Thunderbird* ou *Evolution* verificamos que dispomos de quase todas as ferramentas que constituem o típico escritório electrónico, ou seja, sistema operativo, editor de texto e folha de cálculo, navegador, cliente de *email*

sem incorrer em qualquer gasto.

Se olharmos para os serviços típicos que uma empresa necessita de disponibilizar aos seus funcionários verificamos mais uma vez que as soluções existem e muitas vezes são líderes na sua área:

- Servidor *Web* - *Apache*
- Servidor de DNS - *Bind9*
- Servidor de Autenticação - *OpenLDAP, Radius*
- Servidor de *Email* - *Postfix*
- Servidor de *FTP* - *ProFTPD*
- Servidores Aplicacionais - *Tomcat, Glassfish*
- Servidores de *Web Proxy* - *Squid*
- Serviços de Rede - *DHCPD*

E como veremos ao longo deste trabalho existem soluções para as áreas de Redes Privadas, telefonia *IP* e outros serviços de rede, como encaminhamento, servidor de tempo, servidores de partilhas, etc.

A fraca adesão a algumas destas plataformas é facilmente explicada pela utilização indevida de *software* proprietário, ou seja, pela pirataria desse mesmo *software*. O grau de impunidade deste ilícito em conjunção com uma mentalidade que não classifica o *software* como um bem e, por conseguinte, algo a que está associado um valor monetário leva à disseminação de cópias ilegais. Um recente estudo da *Associação Portuguesa de Software, ASSOFT*, revela que a taxa de pirataria [3] ronda os 54 por cento.

Um caso de cópias ilegais numa empresa, para além dos aspectos judiciais e financeiros associados a um acto ilícito, mancha o nome da empresa na sociedade, o que afecta de forma clara a sua reputação e mais valia no mercado; isto são já aspectos que prendem a atenção dos decisores.

1.2 Descrição do Problema

As principais metas deste projecto passam pela criação de uma estrutura de comunicações integradas que permita manter os actuais parâmetros de qualidade adicionando novas funções, escalabilidade e centralizando na empresa as capacidades de gestão, manutenção e desenvolvimento, reduzindo desta forma a dependência de terceiros, bem como, custos. A normalização da plataforma é um aspecto que também não devemos descurar.

A primeira questão a endereçar neste projecto será a das comunicações telefónicas, *Private Branch Exchange, PBX*, de uma empresa. O objectivo final é interligar 6 pontos centrais (escritórios/sede) e um número crescente de pontos periféricos (lojas). Só após a criação de uma infra-estrutura de *Voice over IP, VoIP* será possível avançar com a criação de um serviço de Fax centralizado.

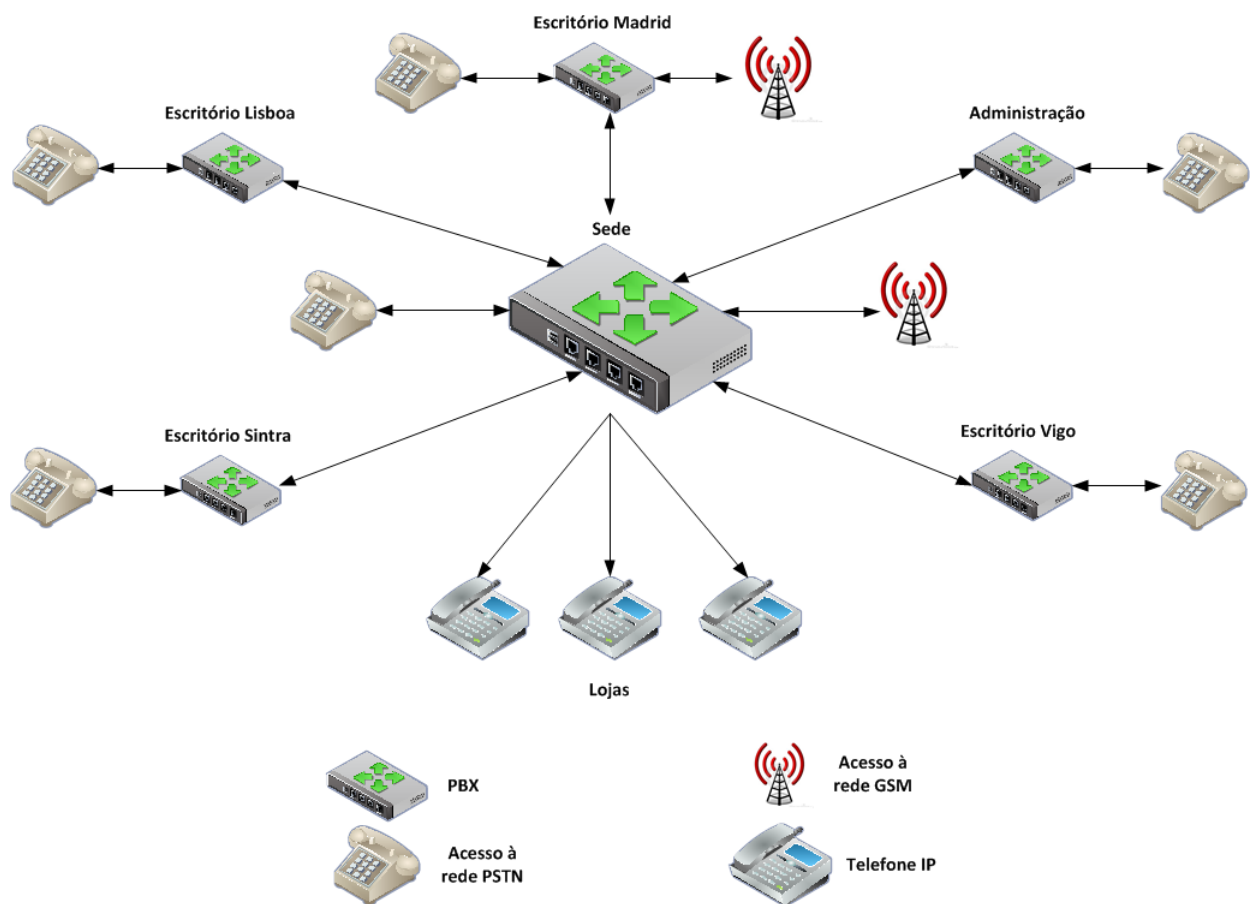


Figura 1.1: Arquitectura da rede VoIP a implementar

Na actual arquitectura da empresa existem centrais *Nortel Meridian* e *Alcatel OmniPCX*. Por serem plataformas fechadas ambas apresentam as seguintes desvantagens:

- Interligação com centrais de outros fabricantes é reduzida por ser difícil e dispendiosa
- Capacidade de crescimento requer sempre a compra de *hardware* específico e dispendioso encarecendo o *TCO*, *Total Cost of Ownership*
- Adição de novas funções requer quase sempre a compra de *hardware* específico e custo de licenciamento
- O custo dos terminais é alto e está-se sempre limitados a número reduzido de opções
- Os contratos de manutenção apresentam valores elevados

A segunda questão visa a criação de uma Rede Privada Virtual, *VPN*, assegurando um acesso fiável e controlado à informação disponível nos servidores centrais.

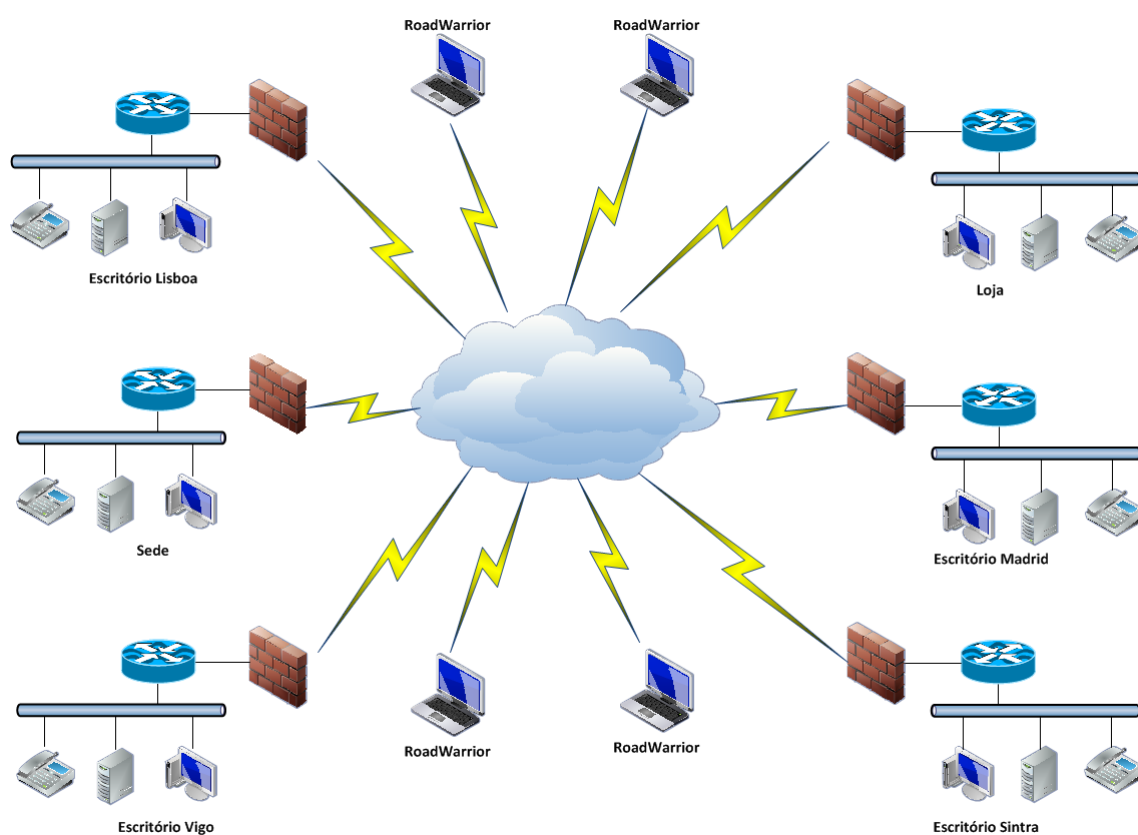


Figura 1.2: Estrutura da Rede *VPN* a implementar

Ao analisarmos estas duas questões verificamos que se encontram interligadas não podendo ser dissociadas uma da outra, o que leva, desde já, a um ponto vital neste projecto, que é a convergência das plataformas.

Na actual arquitectura este serviço encontra-se delegado no operador de telecomunicações, o que leva a que:

- A adição de qualquer novo ponto à rede requer uma acção por parte do operador
- A gestão da rede (publicação de rotas, implementação de políticas de qualidade de serviço) não é acessível ao cliente
- O cliente não pode optar por usar o acesso à rede de outro operador (cabo, ADSL, 3G) que não aquele que fornece o serviço
- Questões como expansão da rede ficam sempre condicionadas aos *timings* do operador

De forma resumida a Tabela 1.1 ilustra a dimensão do projecto de resolução das questões implementadas.

Tabela 1.1: Dimensão do Projecto

Projecto	Local	Nº de Pontos
PBX	Sede	100
	Administração	70
	Escritório Lisboa	30
	Escritório Sintra	20
	Escritório Vigo	30
	Escritório Madrid	30
	Lojas	25
VPN	Escritórios	4
	Lojas	120
	<i>RoadWarrior</i>	50

1.3 Estratégia de Resolução

A sequência de passos a implementar, no que se refere à plataforma de FAX, seria a seguinte:

- Realizar de testes com as diferentes soluções de Fax
- Escolher a plataforma de Fax
- Implementar os serviços *fax-to-email* e *email-to-fax*
- Criar as aplicações necessárias para interligar os sistemas existentes na empresa com a nova solução de FAX
- Efectuar testes à plataforma e corrigir eventuais falhas
- Efectuar o "*Roll-Out*" da solução adoptada

Este trabalho a realizar pressupõe um outro que não será abordado neste documento mas cuja realização foi fundamental para a criação da plataforma de FAX, de uma forma sucinta são apresentadas as etapas

- Escolher a plataforma de *VoIP* que se melhor se adapte a este projecto
- Colocar a nova central entre a central antiga e os acessos a rede *Public Switched Telephone Network, PSTN*
- Criar uma estrutura (plano de numeração) que permita que as 2 centrais coabitem e se interliguem
- Efectuar a troca faseada dos terminais de forma a diagnosticar possíveis falhas na implementação
- Corrigir essas mesmas falhas
- Terminar a troca dos terminais e desactivar da central antiga
- Efectuar o "*Roll-Out*" das centrais *VoIP* nos diferentes locais

Quanto à Rede Virtual Privada, a sequência será a seguinte:

- Efectuar testes com as plataformas analisadas
- Escolher a plataforma de *VPN* que se melhor se adapte a este projecto
- Efectuar testes (conectividade, latência, *throughput*) quer nos cenários *RoadWarrior* quer *Lan-to-Lan*, utilizando diferentes *ISP's*, quer de acessos de banda larga *ADSL* ou *EU-RODOCSIS - Cabo*, quer acessos de banda larga móvel *UMTS/GPRS*, vulgarmente conhecidos como *3G*

- Analisar os resultados obtidos e corrigir as falhas que se constatem
- Efectuar a definição completa de todos os elementos da estrutura à implementar, como por exemplo, características dos servidores (processamento, memória), equipamentos terminais (modem's, placa de acesso 3G)
- Definir as políticas de redundância (uso de múltiplos *Internet Service Provider, ISP's* e existência de pelo menos 2 servidores centrais)
- Efectuar o "*Roll-Out*" faseado da solução adoptada

Capítulo 2

Estado da Arte

Neste capítulo apresentam-se os diferentes protocolos e *software* estudados para a realização deste trabalho.

2.1 VoIP

2.1.1 SIP

A última versão do protocolo *Session Initiation Protocol, SIP* [4], foi desenvolvido no seio do *Internet Engineering Task Force, IETF*, e actualmente é aceite pela *3GPP* como protocolo de sinalização.

O protocolo *SIP* foi desenhado de forma a apresentar independência em relação à camada de transporte, operando sobre *Transmission Control Protocol - TCP* [5], *User Datagram Protocol (UDP) - UDP* [6] e *Stream Control Transmission Protocol - SCTP* [7], e a exibir, um comportamento do tipo *peer-to-peer*. Sendo um protocolo baseado na troca de mensagens de texto podemos encontrar similaridades com outros protocolos, como por exemplo, o *Hypertext Transfer Protocol, HTTP* [8] e o *Simple Mail Transfer Protocol, SMTP* [9].

Esta característica permite a inspecção directa dos pacotes e poderá ser entendida como uma vulnerabilidade em termos de segurança. Para corrigir esta situação, está prevista adição de protecção e confidencialidade às mensagens *SIP*, recorrendo, por exemplo, a:

- *Internet Protocol Security, IPSec*, que fornece a capacidade de comunicação segura entre 2 pontos com a implementação de protocolos segurança, quer no modo *Transport* onde a protecção ocorre para camadas superiores ao *IP* quer em Modo Túnel em que a protecção é completa
- *Secure/Multipurpose Internet Mail Extensions, S/MIME*, faz segurança de conteúdo, encriptando as mensagens *SIP*.

- *Transport Layer Security, TLS* que proporciona uma camada segura de transporte envolvendo *TCP*, encriptando as mensagens trocadas e assim diminuindo o risco associado à inspecção dos pacotes

Sendo um protocolo de sinalização, o *SIP*, não transporta qualquer informação de áudio ou vídeo, relegando esse papel para o *Real-time Transport Protocol, RTP* [10]. As informações relativas aos *codec's*, portas e protocolos envolvidos são negociados com recurso ao *Session Description Protocol, SDP* [11]. O *Real Time Control Protocol, RTCP* [12], é outro protocolo que também se encontra envolvido numa chamada *SIP*, sendo que o seu papel é controlar os diferentes canais *RTP* activos.

Os quatro elementos básicos de uma arquitectura *SIP* e as suas funções são apresentados de seguida:

- *SIP User Agents (UA)* - É a entidade *SIP* que interage com o utilizador, possuindo para tal, a capacidade de enviar e receber pedidos, podendo desta forma assumir o papel de cliente, enviando pedidos e recebendo respostas ou, como servidor, enviando respostas e recebendo pedidos.
- *SIP Proxy Servers* - A sua acção é também de cliente e servidor, recebendo os pedidos e reencaminhando-os para servidores mais próximos do destino. Existem dois tipos de servidores *Proxy*, o *Stateful Proxy Server* e o *Stateless Proxy Server*. O *Stateful Proxy Server* mantém o estado das transacções e permite "dividir" a chamada para múltiplos servidores na tentativa de localizar o utilizador, criando desta forma uma árvore de busca. A contabilização do gasto do cliente e a utilização protocolo *TCP* são outras das características relevantes.

As *Proxy's* do tipo *Stateless Proxy Server* não armazenam o estado da transacção apenas reencaminham os pedidos e as respostas, possuindo por tal maior velocidade, mas são incapazes de calcular gastos do utilizador.

- *SIP Redirect Server* - É um tipo de servidor *SIP*, que responde ao pedido do *UA* fornecendo o nome e a localização do utilizador. Este tipo de servidor não reencaminha os pedidos.
- *SIP Registrar Server*- A sua função é armazenar os dados sobre utilizadores, fornecendo um serviço de localização, de certa forma, é algo equivalente ao *Visitor Location Register, Visitor Location Register, VLR*, da rede *Global System for Mobile Communications, GSM* [13].

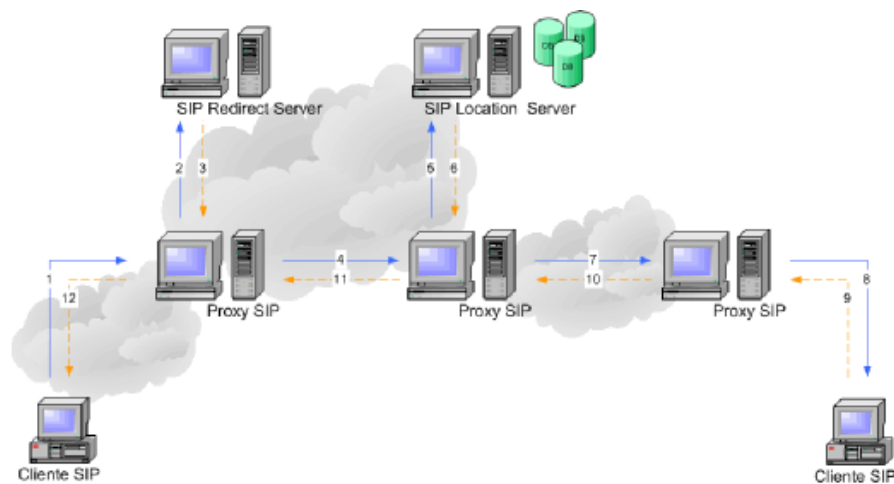
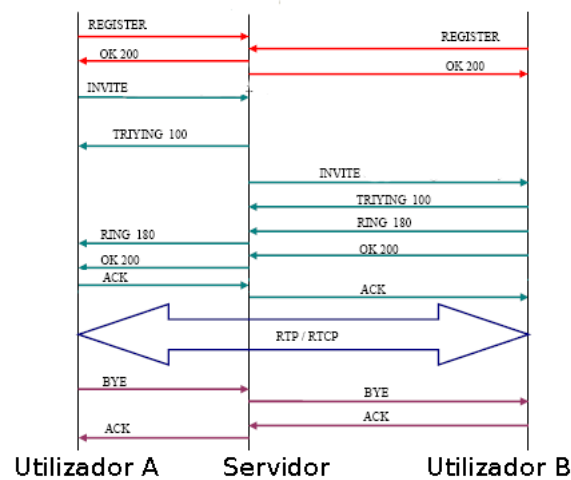


Figura 2.1: Arquitectura de uma rede SIP

Tal como foi referido anteriormente o *SIP* baseia-se na troca de mensagens de texto entre os intervenientes, as mensagens trocadas no estabelecimento de uma chamadas são apresentadas na seguinte imagem.

Figura 2.2: Fluxo de mensagens no estabelecimento de uma chamada *SIP* [1]

Como se pode observar, o primeiro passo é o registo do utilizador. Os utilizadores têm que se registar para que outros utilizadores os possam contactar. O terminal envia um pedido *REGISTER* em que os campos *from* e *to* correspondem ao utilizador registado. Neste caso, o *Proxy Server*, que desempenha o papel de *Registrar* verifica se o utilizador pode ou não autenticar-se e envia uma mensagem *OK* caso não exista nenhum problema.

As trocas que se seguem visam estabelecer uma sessão. Essa sessão consiste no envio de um pedido de *INVITE* do utilizador A para a *Proxy*, que de imediato, envia uma mensagem *TRYING 100* de forma a parar a difusão e redirecciona o pedido de *INVITE* para o utilizador B.

O utilizador B envia uma mensagem *Ringling 180* quando o telefone começa a tocar que também é redireccionado pela *Proxy* para o utilizador A. No final é trocada uma mensagem *OK 200* que corresponde ao aceitar da chamada por parte do utilizador B.

Neste momento a chamada encontra-se estabelecida e o protocolo de transporte *Real-Time Transport Protocol, RTP*, inicia a negociação de alguns parâmetros como portos, codec's, etc.

A última troca de mensagens marca o fim da sessão e é executada recorrendo a transmissão de uma mensagem *BYE* pelo utilizador A à qual o utilizador B responde com uma mensagem *OK 200* para confirmar que a mensagem foi recebida correctamente.

2.1.2 RTP

O *Real-Time Transport Protocol* [12] foi definido inicialmente no *RFC 1889* e revisto em 2003 no *RFC 3550*. O *RTP* assegura o transporte ponto-a-ponto de dados multimédia de tempo-real, como áudio e vídeo, em redes IP *unicast* e *multicast*, podendo ser utilizado para transportar diversos formatos de áudio e vídeo.

Os fragmentos de áudio e vídeo gerados do lado do servidor são encapsulados em pacotes *RTP*, sendo de seguida encapsulados sobre *UDP*. Normalmente, o *RTP* é transportado sobre *UDP*, por questões de desempenho e tolerância do vídeo a perdas. O protocolo define a utilização de números de sequência, marcas temporais e tipo de dados a transportar em cada pacote, informação utilizada pelas aplicações multimédia para detecção de perdas e cálculo do *jitter*. Desta forma, qualquer aplicação que implemente o protocolo *RTP*, poderá reutilizar estes mecanismos em detrimento de outros protocolos proprietários. Esta opção permite uma maior interoperabilidade com outras aplicações multimédia que utilizem o mesmo protocolo.

O *RTP* não implementa reserva de recursos, e não garante *QoS* para serviços multimédia de tempo-real. Devido ao encapsulamento dos pacotes *RTP* em *UDP*, não é igualmente garantida a entrega dos pacotes nos extremos. No entanto, foi definido um protocolo auxiliar, o *Real Time Control Protocol*, que pode ser utilizado em conjunto com o *RTP*. Os pacotes *RTCP* são trocados periodicamente entre todos os participantes durante a sessão *unicast* ou *multicast*, num porto diferente do *RTP*. Estes pacotes não transportam qualquer informação multimédia, sendo utilizados para transmitir relatórios com estatísticas do estado da rede. Estas estatísticas podem incluir perdas de pacotes, número de pacotes enviados, *jitter*, etc, sendo muito úteis às aplicações multimédia. Como exemplo, um servidor ao receber informação sobre elevados valores de perdas, poderá modificar a sua taxa de transmissão.

O pacote *RTP* é formado por um cabeçalho e um *payload*. O cabeçalho tem no mínimo 12 bytes e contém vários campos, entre os quais: número de sequência, marca temporal (informação de tempo, usado para a sincronização dos *streams*), tipo de *payload* (para identificar o tipo de codificação do conteúdo), *marker bit* (para detectar o final de um conjunto de pacotes relacionados), *source identifiers* (sincronização e contribuição) e versão.

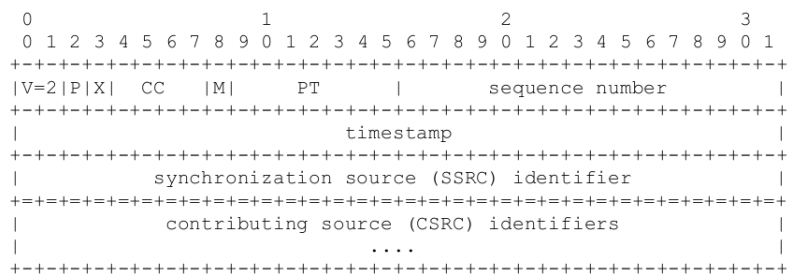


Figura 2.3: Cabeçalho do Pacote *RTP*

2.1.3 IAX

O protocolo *Inter Asterisk eXchange versão 2, IAX2* [14], foi desenvolvido pela empresa *Digium* originadora do projecto *Asterisk* com o objectivo de criar uma opção ao *SIP* no que se refere a interligação de centrais telefónicas.

Atendendo a que o *IAX2* se assume com uma alternativa ao *SIP*, a sua equipa de desenvolvimento tentou resolver alguns dos problemas de implementação que este apresenta, nomeadamente a questão associada as *NAT's* e à segurança.

Em oposição ao *SIP* o *IAX2*:

- apenas usa um porto, o 4569 em *UDP* para sinalização e transporte da voz; como tal não necessita de recorrer a protocolos como o *RTP*
- ao utilizar apenas um porto o problema relacionado com o uso de *firewall's* e de *Network Address Translation, NAT*, é atenuado
- as mensagens são em binário
- a segurança das mensagens pode ser assegurada nativamente recorrendo à cifra *Advanced Encryption Standard, AES*
- a autenticação recorre ao método desafio-resposta e este processo pode ser cifrado frustrando tentativas de escuta
- existe um conjunto de mensagens que permitem a monitorização da rede, da chamada e de parâmetros associados a ela que podem ser trocadas a qualquer momento
- a utilização de um *buffer*, o *jitterbuffer*, que melhora de forma considerável a qualidade do áudio sobre ligações não ideais, como por exemplo, *ADSL* ou acessos de banda reduzida e latência variável.

Existem algumas semelhanças entre os dois protocolos nomeadamente na arquitectura da rede, existindo um *IAX Registrar Server* em que os *UA's* têm que se registar.

A figura 2.4 ilustra o fluxo de mensagens no estabelecimento de uma chamada IAX, tal como descrito no artigo de Mark Spencer e Frank W. Miller [15].

Como se pode observar, existem diferenças com o ponto ilustrado na figura 2.2 sendo que a maior é a capacidade do protocolo IAX transportar o conteúdo multimédia da chamada.

O primeiro passo do estabelecimento da ligação é o envio de uma mensagem *New* do utilizador A para o utilizador B; este responde com uma mensagem *Accept* recebendo de seguida uma mensagem *ACK* que confirma a recepção da mensagem de aceitação. Para concluir o estabelecimento da chamada o utilizador B envia uma mensagem *Answer* e o utilizador A responde com mais uma mensagem de *ACK*.

O fluxo do conteúdo multimédia é assegurado por *M* e *F frames*. Cada fluxo é constituído na sua maioria por *IAX Mini Frames (M frames)* que contêm um cabeçalho simples de 4 bytes que visa assegurar eficiência quanto à largura de banda. Periodicamente são enviadas *Full Frames (F frames)* que incluem informação quanto a sincronização. Estas mensagens são enviadas pelo mesmo canal do protocolo de sinalização (ao contrário do que acontece com o *SIP*), o que evita questões associadas ao uso de *NAT's*.

O término da chamada requer apenas a troca de duas mensagens, uma de *HANGUP* e outra de *ACK*.

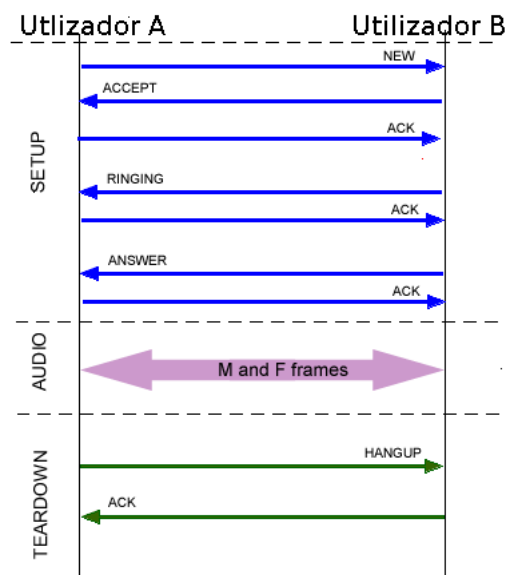


Figura 2.4: Fluxo de mensagens no estabelecimento de uma chamada IAX [2]

2.1.4 Asterisk

O *Asterisk* foi criado em 1999, por Mark Spencer, sendo um dos primeiros servidores *Private Branch Exchange*, *PBX*, open-source, tendo sido desenvolvido para sistemas *Linux*, hoje em dia já pode ser executado em diversos sistemas operativos, *BSD*, *MacOS*, *Solaris* e até *Microsoft Windows*.

O *Asterisk* possui diversas funcionalidades presentes nos sistemas tradicionais de *PBX*, tais como, *voice-mail*, chamadas em conferência, música de espera. Suporta vários protocolos *VoIP*, tais como, *SIP* [4], *Media Gateway Control Protocol*, *MGCP* [16], *H.323* [17], *IAX* [14] e pode operar com a grande maioria dos telefones *SIP*, funcionando como *Registrar Server* e *gateway* entre telefones IP e telefones de linha analógica tradicionais.

As configurações típicas de um servidor *Asterisk* são apresentadas de seguida:

- ***Asterisk como "switch" - PBX***

Asterisk pode ser configurado para fazer transferência de chamadas, administração de rotas e conexão de chamadas com o mundo exterior através de conexões IP, analógicas e digitais.

- ***Asterisk como gateway***

Pode funcionar como uma ponte entre as comunicações analógicas e o mundo da telefonia *IP* uma vez que a arquitectura do *Asterisk* permite converter chamadas através uma enorme variedade de protocolos de comunicação e *codecs*.

- ***Asterisk como servidor de conteúdos***

Permite *Interactive Voice Response*, *IVR*, atendimento automático de chamadas, salas de conferência, *voice-mail*, *Unified Messaging*, *UM*.

- ***IVR - Interactive Voice Response***

É uma tecnologia que permite a um computador detectar voz e sinais telefónicos no decorrer de uma chamada telefónica. Um sistema *IVR* pode ser capaz de responder ou interagir com o interlocutor através de áudio pré-gravado ou gerado dinamicamente permitindo, por exemplo, instruir o interlocutor das acções a tomar, e estar dotado de reconhecimento de voz para possibilitar o reconhecimento de mensagens vocais (tipicamente simples, embora não necessariamente), como "Sim" ou "Não", nomes, etc. Um *IVR* serve, portanto, qualquer função cujo interface possa ser decomposto em séries de opções de menu e, uma vez concebidos, de uma forma geral, são facilmente escaláveis para suportar grandes volumes de chamadas.

- ***UM - Unified Messaging***

Integra diferentes formas de comunicação (*e-mail*, *SMS*, *Fax*, voz, vídeo, etc.) numa única central de informação, acessível a partir de uma variedade de aparelhos.

Segundo um estudo da *Eastern Management Group* [18], apesar da plataforma *Asterisk* ser relativamente recente, quando comparado com outras soluções, ele já apresenta uma percentagem de mercado muito interessante, como podemos comprovar no seguinte gráfico.

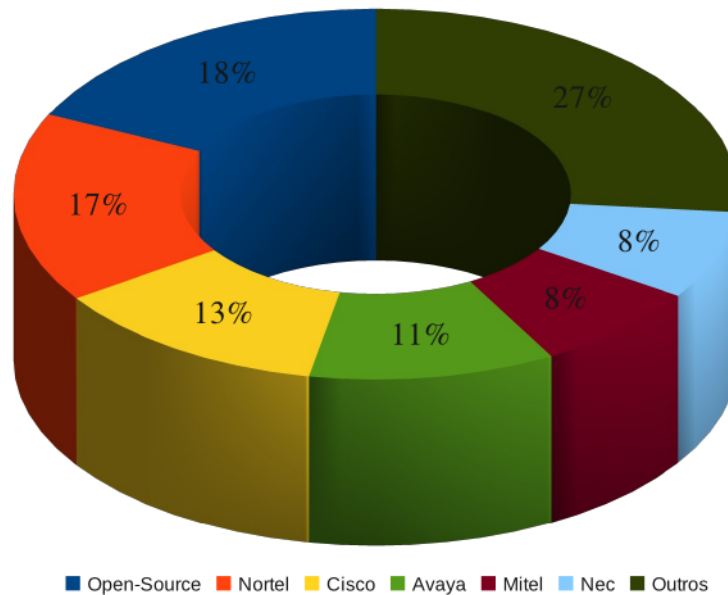


Figura 2.5: Centrais Telefônicas - Percentagem de Mercado

Outro dado interessante é a relação entre a dimensão da empresa e o número de sistemas instalados, tal como podemos constatar na tabela 2.1 são as empresas de média e pequena dimensão que mais apostam neste tipo de infraestrutura.

Tabela 2.1: Distribuição de Sistemas "Open-Source" PBX por tamanho da Empresa

Número de Filiais	Percentagem
1	32
2-10	55
11-50	8
51-100	2
101-500	2
> 500	1

O seguinte gráfico demonstra que 75 por cento dos sistemas são de pequena dimensão, ou seja, menos de 50 terminais por central.

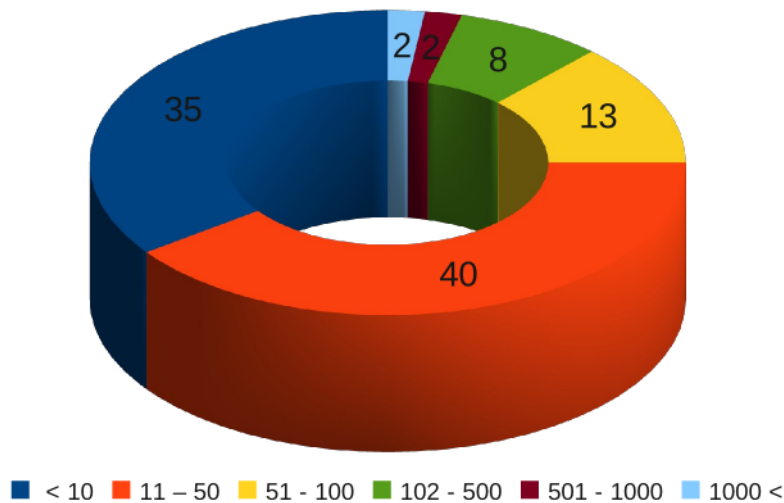


Figura 2.6: Distribuição de número de telefones por Sistema

2.1.5 Fax via VoIP

Para integramos o serviço de Fax com o *Asterisk* existem diversas soluções, as duas mais populares são:

- *SpanDSP* [19]
- *Hylafax* [20]

O *SpanDSP* é uma biblioteca que permite ao *Asterisk* receber e enviar fax podendo ser integrada directamente sobre a plataforma. Suporta quer o tradicional envio de fax recorrendo ao canal de áudio quer a norma *T.38* [21] de fax sobre *IP*, podendo agir como interface de conversão entre a normas de Fax via *PSTN* e via *T.38*.

Esta biblioteca implementa modems do tipo V.21, V.27, V.29 e V.17, ou seja, permite envio/recepção de fax a taxas desde 2400bps até 14400bps, suportando ainda *Error Correcting Mode*, *ECM*.

A equipa de desenvolvimento do *Hylafax* define o projecto como *Sistemas de Telecomunicações para sistemas UNIX*. Ao contrário do *SpanDSP* este projecto precisa de bibliotecas ou controladores de terceiros para aceder à *PSTN*.

A interligação ao *Asterisk* é obtida pela utilização de outro projecto *open-source*, o *iaxmodem* [22] que cria os modems que serão utilizados.

Funções como *fax-to-email*, *email-to-fax* são de fácil implementação existindo ainda uma variedade de clientes multi-plataforma [23] capazes de procederem ao envio e recepção de fax a partir dos computadores na nossa rede local. Existe ainda um amplo lote de conectores para diferentes linguagem de programação [24] e *software* de fax via *web* onde se destaca o *Avantfax* [25] pelo seu conjunto de funções.

2.2 VPN - Redes Privadas Virtuais

O termo *VPN* é o acrónimo de *Virtual Private Networks* que consiste numa rede criada sobre a já existente rede de comunicação. O seu carácter privado advém de toda a informação ser transmitida de uma forma segura quer usando encriptação ou por recurso a encapsulamento do tráfego por outro protocolo criando desta forma um túnel entre os dois extremos da comunicação.

Tal como descrito em [26] a segurança numa *VPN* foca 3 pontos essenciais que têm que ser assegurados:

- **Privacidade/Confidencialidade**

A informação transferida apenas pode se encontrar disponível as partes autorizadas

- **Integridade**

A informação transferida não pode ser alterada entre o emissor e o receptor

- **Disponibilidade**

A informação tem que estar acessível quando se verificar a necessidade de acesso

Os 2 cenários típicos de utilização são o *Lan-to-Lan* e *Roadwarrior*. O primeiro visa interligar duas redes, como por exemplo a rede da sede da empresa com uma rede de uma filial; já o segundo cenário aplica-se a mobilidade, uma vez que permite que um membro da organização aceder à informação ou serviços que se encontram alojados na sede tornando o terminal usado num ponto da rede da empresa.

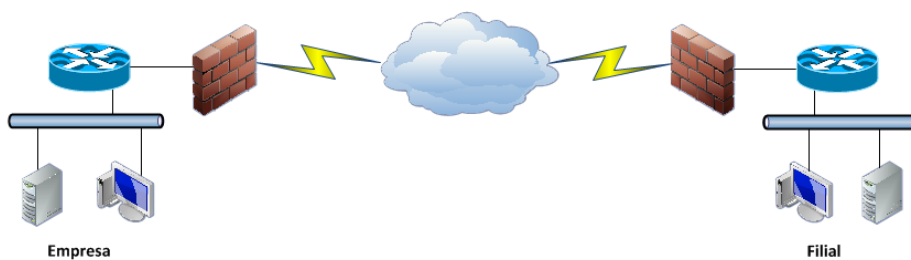


Figura 2.7: Cenário *Lan-to-Lan*

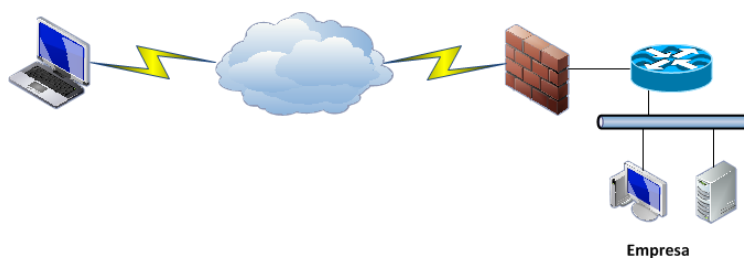


Figura 2.8: Cenário *Roadwarrior*

2.2.1 OpenVPN

O projecto *OpenVPN* data de 2001 sendo que a versão 2, que introduziu a possibilidade de múltiplas ligações por múltiplos clientes, foi revelada em 2004. A versão estável do projecto é a 2.09 e data de 2006 estando em desenvolvimento a versão 2.1.

A *OpenVPN* utiliza a biblioteca *OpenSSL* para o processo de encriptação, quer do canal de informação quer do canal de controle, desta forma o administrador pode escolher entre qualquer uma das cifras disponíveis no *OpenSSL* [27]. Opcionalmente, podemos ainda usar *Hash-based Message Authentication Code, HMAC*, que é um método criptográfico que permite calcular o *Message Authentication Code, MAC*, assegurando simultaneamente a autenticidade da mensagem e a sua integridade.

A autenticação dos clientes pode ser realizado quer por senha, por um binómio *utilizador/senha* ou pelo recurso a certificados. Neste último caso podemos recorrer ao *OpenSSL* para gerar os referidos certificados.

Este projecto foi desenhado de forma, a apresentar independência em relação à camada de transporte, conseguindo correr sobre *TCP, UDP* tendo a *Internet Assigned Numbers Authority, IANA*, atribuído-lhe o porto 1194. Outras característica interessantes são, por exemplo:

- A capacidade de estabelecer uma ligação do tipo *layer 3* recorrendo a um túnel *IP, TUN*, ou *layer 2* utilizando para tal uma interface *Ethernet TAP*
- A possibilidade de utilizar compressão no canal de dados, utilizando para tal a biblioteca *LZO* [28]
- A capacidade de informar o cliente de que *IP's*, rotas, servidores de *DNS* deve utilizar
- O uso de *proxys* encontra-se previsto o que permite evitar *firewalls* e problemas associados a *NAT's*

2.2.2 IPsec

O *Internet Protocol Security*, *IPsec* [29] é uma extensão do protocolo *IP* que visa aumentar o grau de segurança das comunicações na *Internet* recorrendo a autenticação e encriptação de cada pacote *IP* de fluxo de dados. O *IPsec* prevê um conjunto de rotinas para o estabelecimento de túneis seguros utilizando para tal, a autenticação mútua dos agentes envolvidos e a troca de chaves criptográficas no início da sessão que serão utilizadas para cifrar todo o fluxo de dados entre os intervenientes.

Este protocolo opera sob a camada de rede (ou camada 3) do modelo *OSI*. Outros protocolos de segurança da *Internet* como *Secure Sockets Layer*, *SSL* e *TLS* operam desde a camada de transporte (camada 4) até a camada de aplicação (camada 7). Isto torna o *IPsec* mais flexível, já que pode ser usado para proteger os protocolos *TCP* e *UDP*, não obrigando as aplicações a ter em conta o uso de canais encriptados.

Os modos de operação previstos são dois:

- **Modo Transporte** - Neste modo o cabeçalho de autenticação, *AH*, está localizado após o cabeçalho *IP* e antes do protocolo de camada superior ou de outros cabeçalhos adicionados pelo *IPSec*, como se pode verificar na figura 2.10. Uma desvantagem do modo transporte é a possibilidade de modificação dos endereços *IP* de origem e destino em caso de interceptação do pacote, uma vez que neste modo os endereços em questão ainda encontram-se abertos para modificações.
- **Modo Túnel** - Neste modo, todo o pacote *IP* é autenticado e criptografado, encontrando-se apenas visível o cabeçalho *IP* externo (com o último endereço de origem e destino), estando todo o conteúdo interno cifrado, tal como, a figura 2.9 ilustra.

Para garantir a segurança dos fluxos dos pacotes e mudanças de chaves existem dois métodos:

- **Encapsulating Security Payload (ESP)** - que oferece autenticação, confidencialidade dos dados e integridade da mensagem.

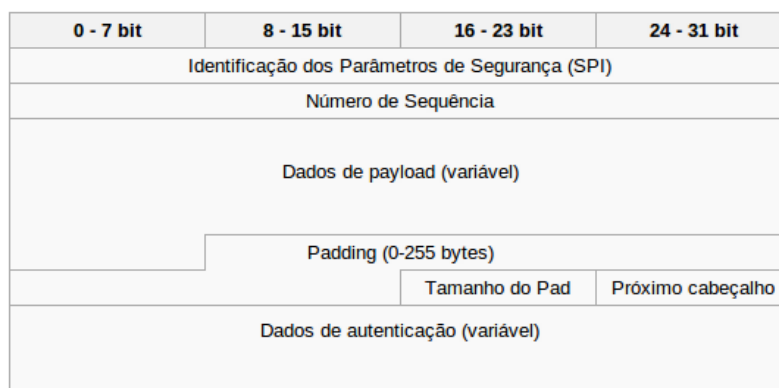


Figura 2.9: Cabeçalho *IP* com *ESP*

- **Cabeçalho de autenticação (AH)** - que oferece a autenticação e integridade dos dados, mas não a confidencialidade da mesma.

0 - 7 bit	8 - 15 bit	16 - 23 bit	24 - 31 bit
Próximo cabeçalho	Tamanho da mensagem	RESERVADO	
Identificação dos Parâmetros de Segurança (SPI)			
Número de Sequência			
Dados de autenticação (variável)			

Figura 2.10: Cabeçalho *IP* com *AH*

Quanto a funções criptográficas específicas, o *IPsec* pode utilizar múltiplas, com destaque para:

- *HMAC-SHA1* - para garantir integridade e autenticidade
- *TripleDES-CBC* - para garantir confidencialidade
- *AES-CBC* - para garantir confidencialidade

2.2.3 Distribuições de *Firewall/Router*

A utilização de uma das três distribuições apresentadas de seguida visa suprir a necessidade da instalação de *Router* e/ou concentrador de túneis na nossa arquitectura de Rede Virtual Privada.

Todos os projectos referidos apresentam as seguintes características:

- Implementa funções de *Firewall* e *NAT*
- Implementa Balanceamento de Carga e Redundância das ligação à *Internet*
- Suporta *OpenVPN*, *IPsec*, *PPTP*
- Suporta *Vlan's*
- Suporta algoritmos de *routing* com o *Ripv2*
- Inclui serviços como *DHCP*, *DNS*, *NTP*, *Web Proxy*, monitorização de tráfego

2.2.3.1 PfSense

O projecto *Pfsense* [30] iniciou-se em 2004 tendo na sua génese um outro de seu nome *MOn0wall*. Este projecto ao suportar tanto o *OpenVPN* como *IPsec*, é um excelente candidato a desempenhar o papel de *Router* e/ou concentrador de túneis. Como principais características diferenciadoras podemos referir:

- É baseado em *Berkeley Software Distribution*, *BSD*
- Suporta *QoS* via *Layer 7 Traffic Shaping*
- Suporta a adição de novas funções disponibilizadas pela comunidade através de um repositório central
- Pode *correr* sobre *hardware* de baixos recursos

2.2.3.2 Endian

A semelhança do projecto *PfSense* a *Endian Firewall* [31] este descende também de um outro projecto, neste caso o *IPCOP*. Este projecto apresenta soluções denominadas por *Community* cuja utilização é gratuita bem como outras de cariz comercial; iremos apenas focar a versão de uso livre.

Como características diferenciadoras mais relevantes podemos referir:

- É baseado em *Linux From Scratch*
- Suporta *Sip Proxy*
- Suporta filtragem de tráfego de email aplicando Anti-vírus e regras *Anti-spam*

- Suporta *Intrusion Detection System, IDS*, o que permite análise de possíveis ataques exteriores à nossa rede

2.2.3.3 Vyatta

O projecto *Vyatta*, tal como, a *Endian* apresenta soluções de uso livre e outras de carácter comercial.

A equipa de desenvolvimento deste projecto pretende ser uma opção face a soluções como a série *1800* e *ASA 5500* da Cisco realçando o custo e flexibilidade inerentes a um projecto "*Open-Source*".

Este projecto apresenta ainda soluções para ambientes virtuais suportados por *Xen* e *VMWare* o que é factor que os distingue dos restantes.

Como características relevantes podemos referir:

- É baseado em *Debian*
- Implementa Balanceamento de Carga e Redundância das ligação à *Internet*
- Suporta *Sip Proxy*
- Suporta algoritmos de *routing* externo como o *BGPv4*
- Permite criar um *cluster* de forma a oferecer alta disponibilidade.

Capítulo 3

Caracterização do sistema

Este capítulo começará por fazer uma exposição dos requisitos (funcionais e não funcionais) que o sistema a desenvolver deverá satisfazer. De seguida, serão apresentadas as tecnologias escolhidas a serem utilizadas para o desenvolvimento do sistema.

3.1 Listagem de Requisitos

3.1.1 Requisitos Funcionais

Os requisitos funcionais são aqueles que descrevem o comportamento do sistema, ou seja, funções ou serviços que o sistema deve ser capaz de realizar. Apresentam-se de seguida os considerados para o sistema de Fax:

- Garantir a interoperabilidade do novo sistema com as plataformas já existentes:
 - Serviço de Tratamento e Envio de Encomendas, *EDI*, que apresenta como principal característica uma Base de Dados *Oracle 8*
 - Serviço de Correio Electrónico
 - Centrais Telefónicas *VoIP*, baseadas em *Asterisk*
- Garantir compatibilidade do sistema desenvolvido com plataformas *Linux* e *Microsoft Windows*
- Arquivo de dados da base dados
 - Do ponto de vista do administrador deve ser possível consultar todos os trabalhos submetidos e recebidos por utilizador, bem como, as configurações globais do sistema;
 - Do ponto de vista do utilizador deve ser possível a consulta de todos os trabalhos que lhe estejam associados, bem como, dos dados do seu perfil;
 - Devem ser armazenados na base de dados as contas, perfis e históricos de todos os eventos que o sistema despolete.

Os requisitos funcionais considerados para o sistema de *VPN* são:

- Garantir a alta disponibilidade do sistema
- Permitir a gestão/manutenção dos ambiente *Lan-to-Lan* e *Roadwarrior* numa única interface
- Garantir compatibilidade do sistema desenvolvido com plataformas *Linux* e *Microsoft Windows*

3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais são requisitos que impõem restrições ou atributos de qualidade ao sistema como segurança, precisão e desempenho. Listam-se de seguida os considerados para o sistema de Fax:

- Minimizar os custos com as comunicações e com outros gastos acessórios
- Centralizar o envio e recepção de *Fax*, eliminando os tradicionais aparelhos físicos
- Aceder ao sistema utilizando autenticação

Os requisitos não funcionais considerados para o sistema de *VPN* são:

- Minimizar os custos com as comunicações
- Garantir a independência na gestão da rede virtual privada dos *ISP's*
- Aceder ao sistema utilizando autenticação
- Alargar o acesso remoto aos sistemas centrais ao maior número possível de utilizadores

3.2 Escolhas tecnológicas

3.2.1 Interface Web

A interface foi desenvolvida com recurso a várias ferramentas de software *open-source*. A linguagem escolhida para o seu desenvolvimento foi o *Hypertext Preprocessor, PHP* [32], pela sua extensa biblioteca de funções assim como pela capacidade de programação orientada a objectos. Além do uso de *PHP* para desenvolver a interface visual da página Web, foi utilizado *javascript*, especialmente as bibliotecas *jquery* [33] e *jqueryUI* para menus e outras animações tais como tabelas e formulários.

3.2.2 Servidor de Páginas

O software eleito para Servidor de Páginas foi o *Nginx* [34]. Após comparação com software como o *Apache* [35] e *lighttpd*, [36] este revela uma utilização menor de recursos de sistema. As capacidades deste *software*, que mais uma vez é uma aplicação *open-source*, têm vindo a convencer cada vez utilizadores encontrando-se em franco crescimento como se pode comprovar pelo estudo da *Netcraft* [37] que o coloca no terceiro lugar da sua categoria.

3.2.3 Servidor base de dados

Para servidor de base de dados foi feita uma ponderação entre o *PostgreSQL* [38] e o *MySQL* [39], pois são dos servidores de base de dados mais utilizados devido à sua robustez, fiabilidade e desempenho, e por serem *open source* e possuírem um grande suporte da comunidade ao nível de documentação, manuais e fóruns de discussão. Foi escolhido o *MySQL* uma vez que nunca tinha trabalhado com este tipo de sistema e considere que a sua utilização seria uma mais-valia em termos de conhecimento; por outro lado permite que os dados armazenados possam ser usados por outras aplicações e permite também efectuar análise de dados directamente da base de dados.

3.2.4 Software PBX

Neste capítulo a escolha recai sobre o *Asterisk* devido à elevada qualidade de documentação, de módulos, de interligação à base de dados, do suporte que possui a diferentes protocolos como por exemplo, *SIP*, *IAX*, *Signalling Connection Control Part - SCCP*, bem como, pela sua posição dominante no mercado.

3.2.5 Servidor de VPN

Para servidor de *VPN* a solução escolhida foi a *OpenVPN*. Este *software* é *open-source* e tem vindo a ser adoptado como solução na criação de redes virtuais privadas por diversas soluções, quer de uso livre quer comerciais. Entre as plataformas que adoptaram a *OpenVPN* encontramos:

- PfSense
- untangle
- Endian
- Vyatta
- ClearOS [40]
- RouterOS [41]

Dos projectos referidos a maioria apresenta versões de uso limitado ou pessoal e versões de uso comercial. Por exemplo, *OpenVPN* desenvolveu uma versão comercial que foca as questões da administração da plataforma e da *Cloud*, o que revela também a crescente quota de mercado

que tem vindo a adquirir. O seu carácter multi-plataforma, existindo versões para os sistemas *Linux*, *BSD*, *MacOS*, *Microsoft* e até mesmo para *Android* e *Iphone*, aliado a utilização protocolos como *X.509* [42], *SSL/TLS* e *AES* [43] e sua capacidade de interligação a sistemas de autenticação centralizada, *Lightweight Directory Access Protocol*, *LDAP* [44], tornam-se mais valias claras para o desenho desta solução.

3.2.6 Solução de Alta-Disponibilidade

De forma a minimizar tempos de paragem e assegurar redundância, a escolha para implementar um *cluster* recaiu sobre o *Heartbeat* [45], um projecto da *High-Availability Linux (Linux-HA)*. Este *software* permite monitorizar o estado dos servidores e dos processos que correm nos mesmo e em caso de falha substituir o nó em falha por outro.

O *Heartbeat* permite a monitorização de diverso *softwar*, como por exemplo, *Apache*, *MySQL*, *PostgreSQL*, *Oracle*, além do próprio nó, ou seja se o servidor responde ao não a pedidos *Internet Control Message Protocol*, *ICMP*.

3.3 Arquitectura do sistema

3.3.1 Servidor de Fax

Tendo em conta as funcionalidades e os requisitos funcionais e não funcionais conjugados com as vantagens das tecnologias e técnicas escolhidas, apresenta-se de seguida o resumo da arquitectura do serviço de Fax. O sistema a implementar será constituído pelos seguintes elementos base:

- Servidor Web com acesso a base dados relacional
- Servidor *Asterisk*
- Servidor *Hylafax*
- Aplicação para o processamento da encomendas geradas pelo sistema de *EDI*, o *jhEDI*
- Aplicação para o processamento de faxes gerados pelo *Enterprise Resource Planning*, *ERP*, o *jhAS400*

O sistema desenvolvido, obedece ao modelo *MVC*, *Model-View-Controller* (ver figura 3.1). Este modelo de arquitectura divide as aplicações interactivamente em três componentes: modelos, vistas e controlos.

Os componentes de modelos encapsulam o núcleo de acesso aos dados e as funcionalidades, e são independentes da apresentação da informação e dos comportamentos na introdução de dados. Os componentes de vistas assumem a responsabilidade de fornecer aos utilizadores a informação, cabendo aos componentes de controlos, a manipulação dos pedidos e a introdução de dados.

A interface com o utilizador fica a cargo das vistas e dos controladores. Desta forma, modificações na apresentação de informação não afectam o modo como os dados são manipulados, tal como, uma alteração na lógica não altera o modo como os dados são apresentados. De uma forma automática é mantido um mecanismo de consistência entre as interfaces e os modelos.

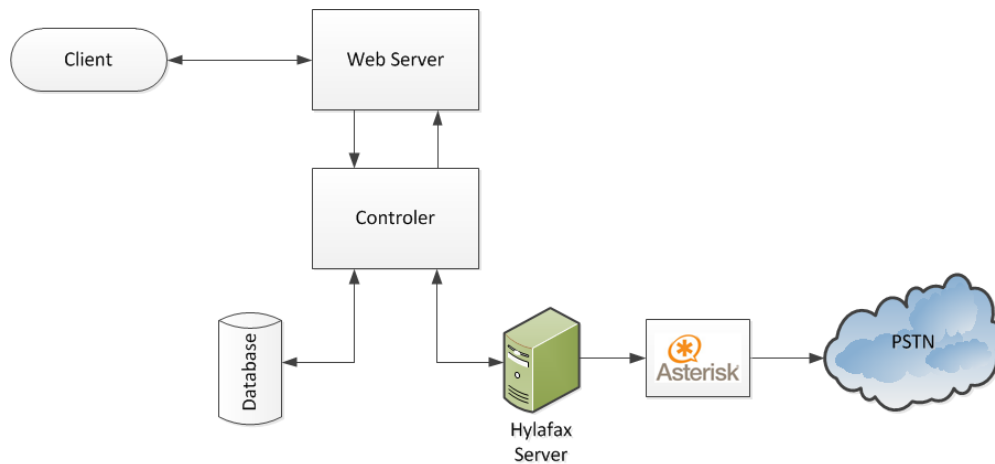


Figura 3.1: Modelo *MVC* do sistema

Em termos de implementação física a figura 3.2 ilustra os diferentes componentes e as suas interligações

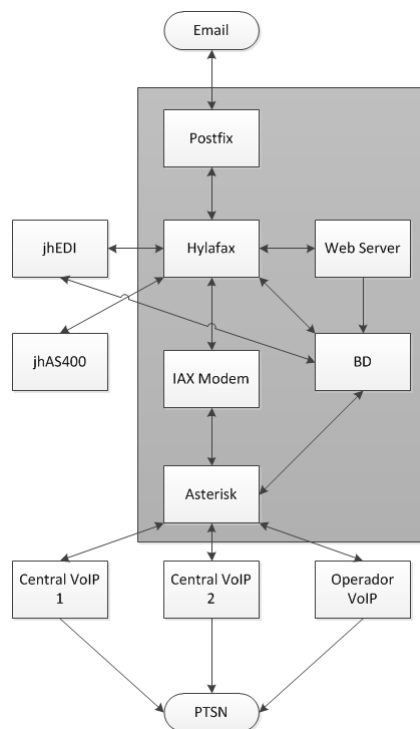


Figura 3.2: Arquitectura do sistema de Fax

O núcleo do sistema (zona cinzenta da figura) é constituído por 5 componentes:

- *Web Server, Nginx*
- Base de Dados, *MySQL*
- Servidor de Fax, *Hylafax*
- *IAXModem*
- *Software de PBX, Asterisk*

Para interagir com a rede telefónica tradicional existem duas ligações *IP* recorrendo ao protocolo *IAX* entre servidor *Asterisk* e duas centrais telefónicas *VoIP*, ambas suportadas pelo mesmo software. A existência destas duas ligações assegura redundância ao serviço centralizado de envio e recepção de faxes. Existe ainda a possibilidade de estabelecer ligações *SIP* ou *IAX* a operadores *VoIP* externos, este facto confere ao sistema desenvolvido resiliência, uma vez que adquirimos independência em relação ao operadores tradicionais.

A interligação aos sistemas já existentes na empresa exigiu o desenvolvimento de novo *middleware*, duas aplicações que implementam a mediação entre as plataformas, ambas desenvolvidas tendo por base a linguagem de programação *Java* e recorrendo ao *Integrated Development Environment - IDE, NetBeans*.

Por fim, existe ainda a possibilidade do sistema interagir com o servidor de *Email*, essa função não requereu o desenvolvimento de uma aplicação específica bastando a criação de um *script bash* para lidar com a recepção das mensagens e seu posterior envio.

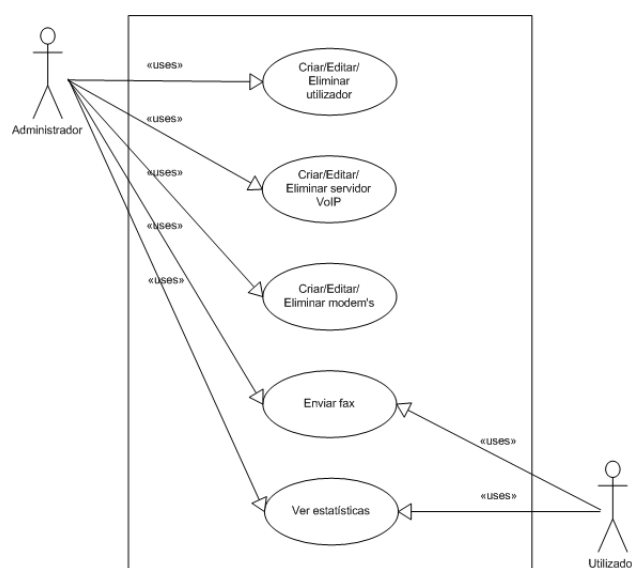


Figura 3.3: Casos de Uso - Servidor de Fax

As interacções dos administradores e utilizadores com a plataforma são ilustradas da na figura 3.3

3.3.2 Servidor de VPN

Atendendo as funcionalidades e aos requisitos funcionais e não funcionais e as vantagens das tecnologias e técnicas escolhidas, apresenta-se de seguida o resumo da arquitectura do serviço de VPN. O sistema a implementar será constituído pelos seguintes elementos base:

- Firewall
- Servidor de *Network File System, NFS*
- Servidor VPN, nó 1
- Servidor VPN, nó 2
- Aplicação de gestão e configuração da plataforma, *Webmin*
- Serviço de monitorização e HA, *Heartbeat*

O *cluster* de servidores que asseguram o serviço de VPN encontram-se configurados em *failover*, ou seja, o nó 1 encontra-se em produção e nó 2 em prevenção. Estes estados são assegurados por um mecanismo de controlo e monitorização desempenhado pelo *Heartbeat*.

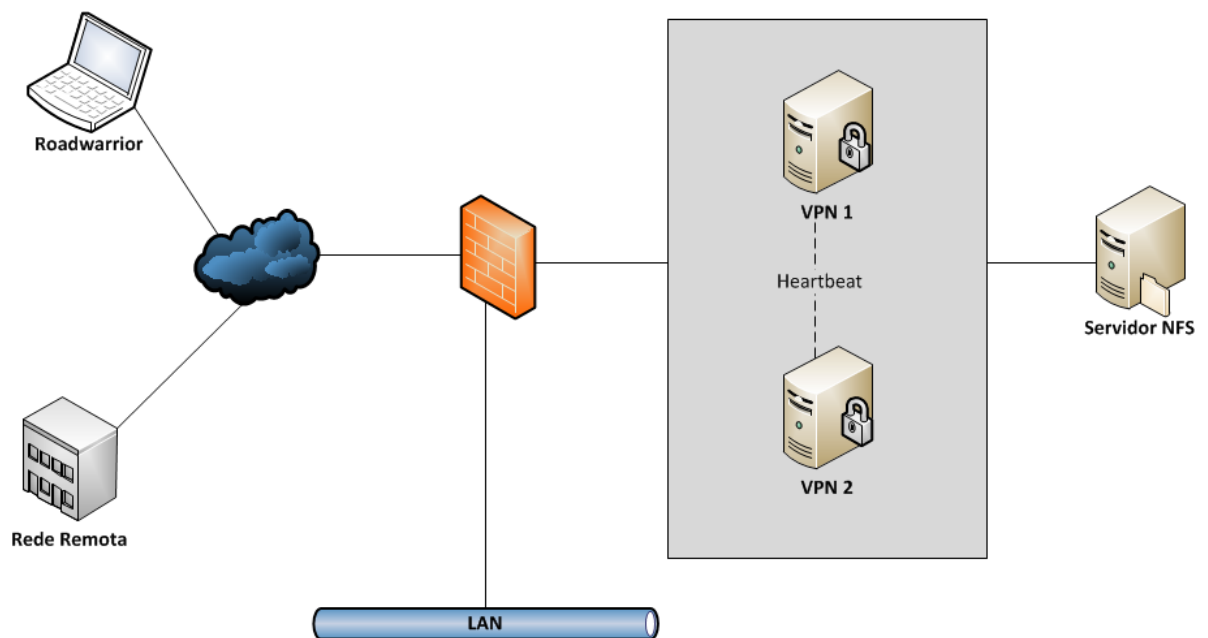


Figura 3.4: Arquitectura da plataforma de VPN

As interações dos administradores com a plataforma de gestão do servidor de *VPN* são ilustradas da na figura 3.5.

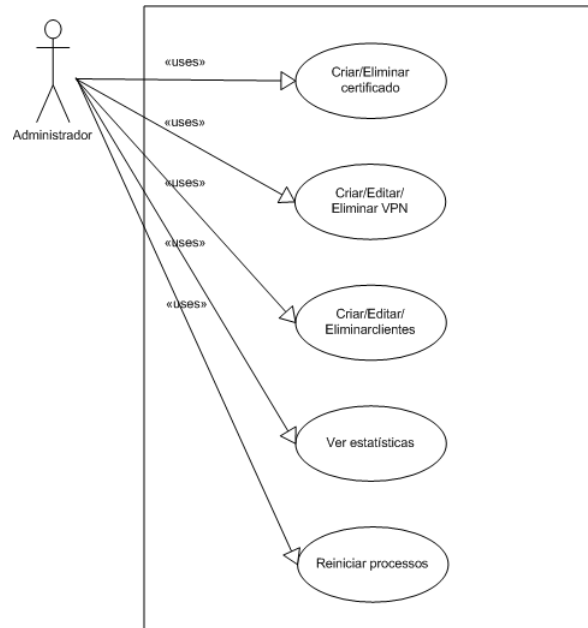


Figura 3.5: Casos de Uso - Servidor de *VPN*

Capítulo 4

Desenvolvimento do Sistema

Neste capítulo é apresentada informação acerca da abordagem e desenvolvimento dos vários componentes do sistema.

4.1 Base de Dados

O esquema da base de dados utilizada neste trabalho é apresentado na figura 4.1

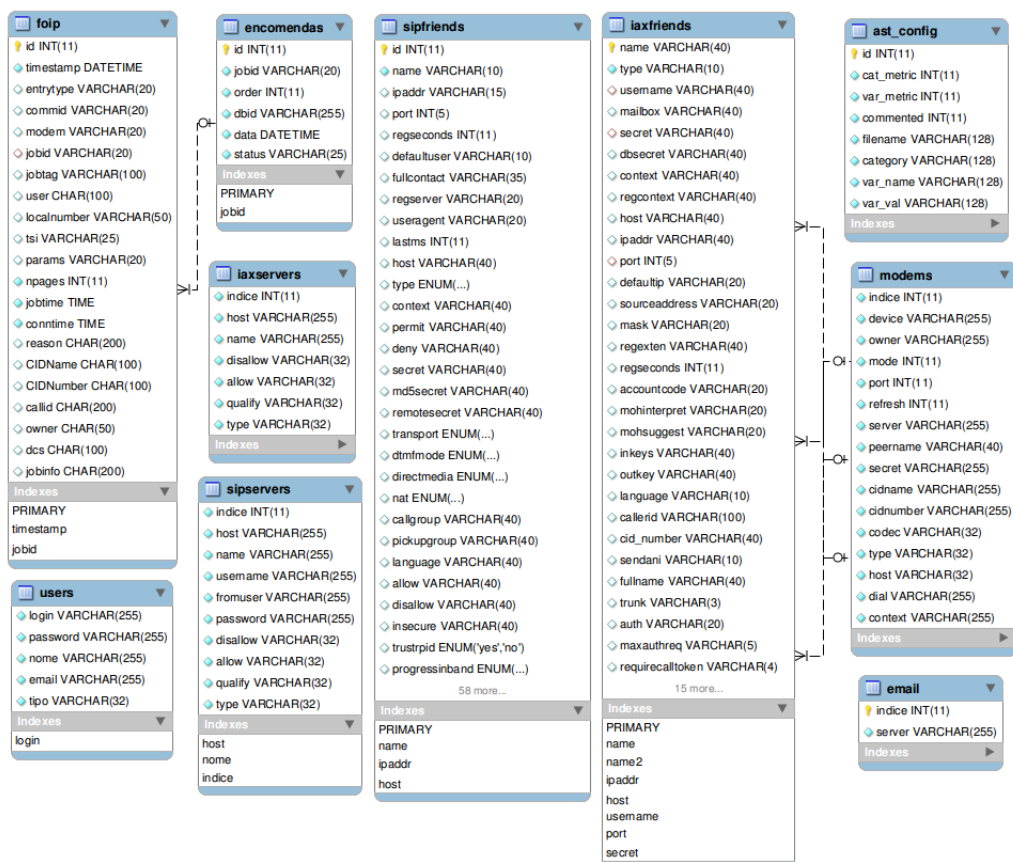


Figura 4.1: Modelo da Base de Dados

As tabelas usadas pela interface *web* são:

- email - contem o endereço *IP* do servidor de correio da empresa
- iaxservers - contem o as configurações dos servidores *IAX* que o sistema pode utilizar
- modems - contem as configurações dos diversos *modems* que a plataforma pode utilizar
- sipservers - contem o as configurações dos servidores *SIP* à disposição da plataforma
- users - contem as informações referentes aos utilizadores

A tabelas apresentadas de seguida são utilizadas pelo *Asterisk* no modo de funcionamento *Asterisk RealTime* em que as configurações encontram-se na base de dados e não na habitual estrutura de ficheiros:

- ast_config - configurações genéricas do *Asterisk*
- iaxfriends - configurações relacionadas com clientes e servidores *IAX*
- sipfriends - configurações relacionadas com clientes e servidores *SIP*

A aplicação *jhEDI* utiliza a tabela encomendas e tabela foip é utilizada pelo sistemas de taxação.

4.2 Servidor Web

4.2.1 Interface de administração do servidor Web

A figura 4.2 ilustra o menu de autenticação e página de boas vindas à plataforma de FAX.



Figura 4.2: Menu de configuração de Utilizadores

Após a autenticação o administrador pode consultar o quadro de utilizadores, figura 4.3, e adicionar/eliminar/editar essas entrada como exemplificado na figura 4.4.

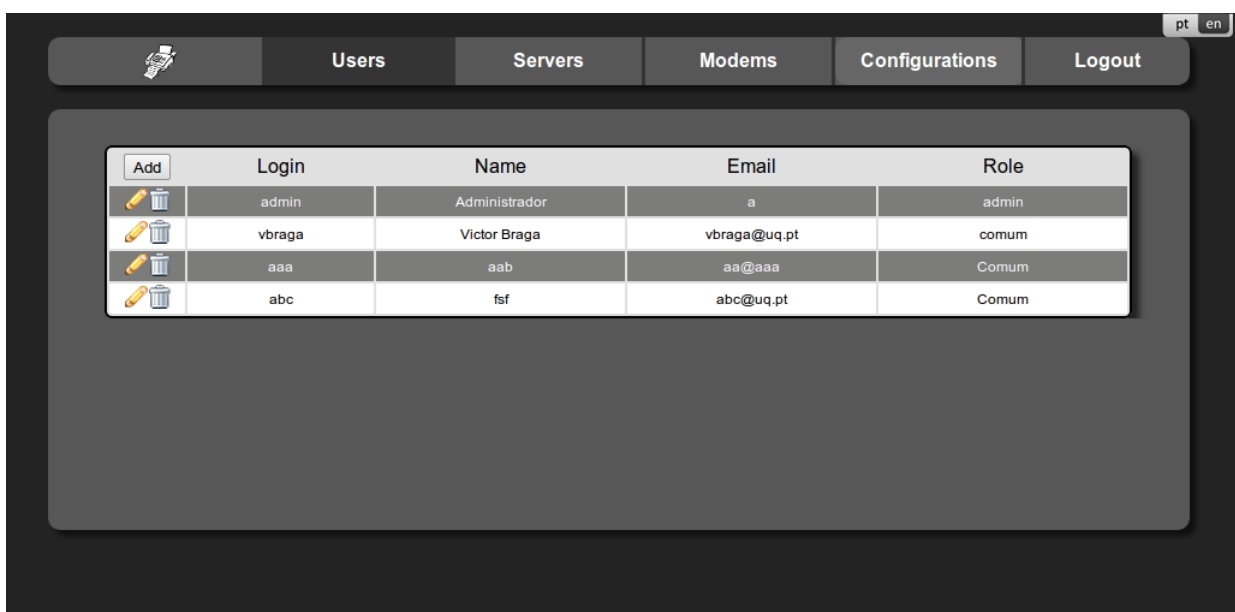


Figura 4.3: Menu de configuração de Utilizadores

Como se pode observar ao adicionar um utilizador a interface valida se o *login* se encontra disponível além de verificar se senha introduzida está correta.

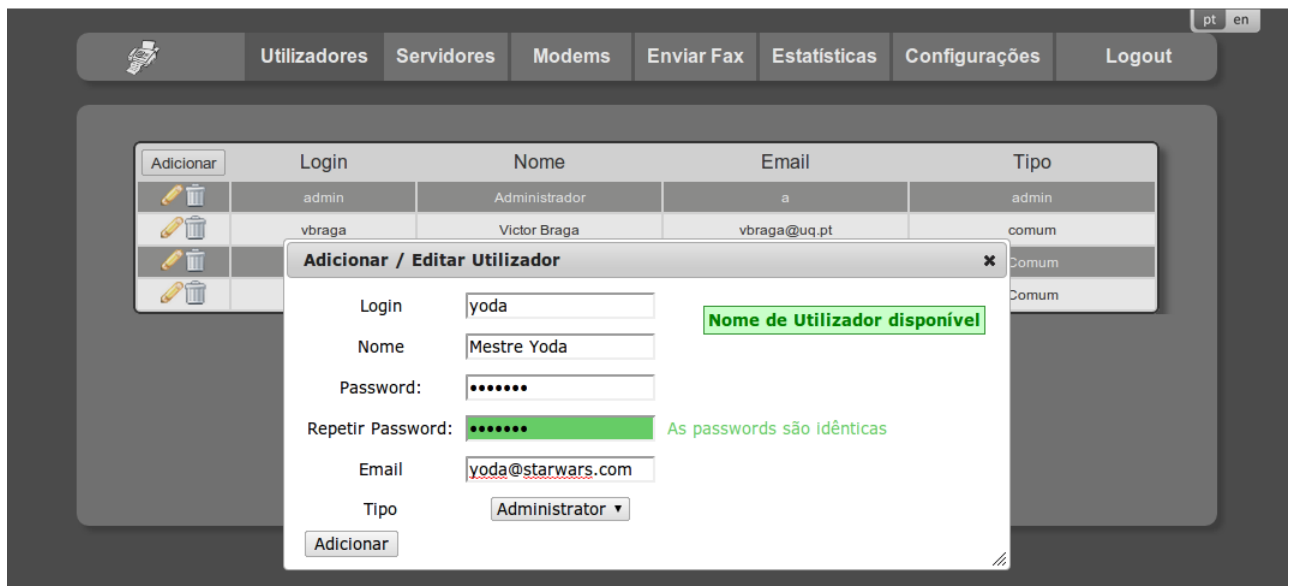


Figura 4.4: Menu de configuração de Utilizadores - Adicionar Utilizador

Na secção *Servidores* o administrador visualiza uma listagem dos servidores *SIP* e *IAX* já configurados, situação ilustrada pela figura 4.5. O administrador dispõe da capacidade de adicionar/eliminar/editar qualquer servidor.

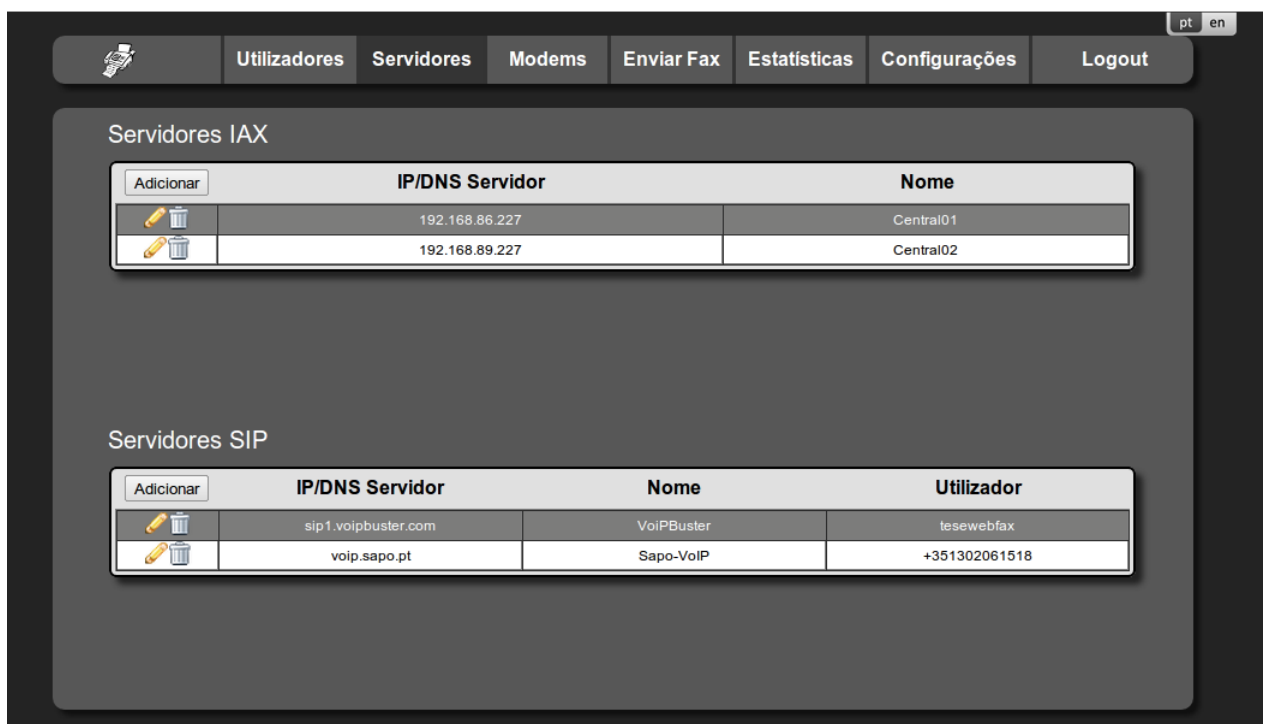


Figura 4.5: Menu de configuração de Servidores VoIP

As figuras 4.6 e 4.7 demonstram, respectivamente, a adição de um servidor *SIP* e *IAX* à configuração da plataforma.

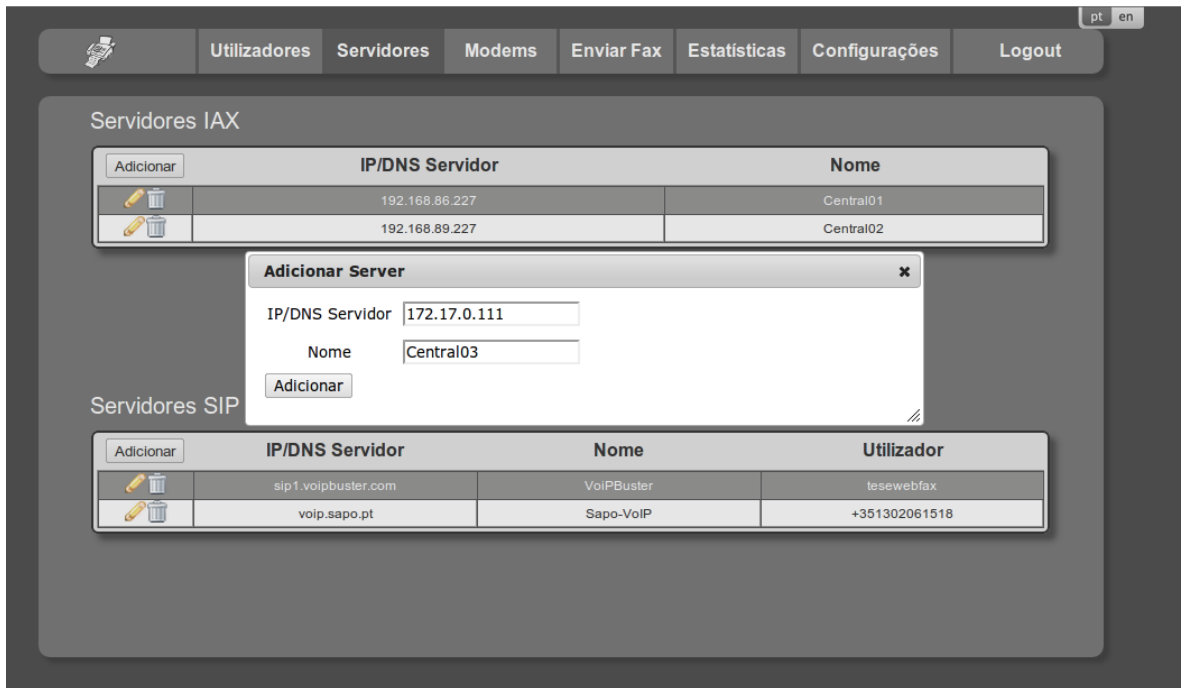


Figura 4.6: Menu de configuração de Servidores *VoIP*- Adicionar Servidor *SIP*

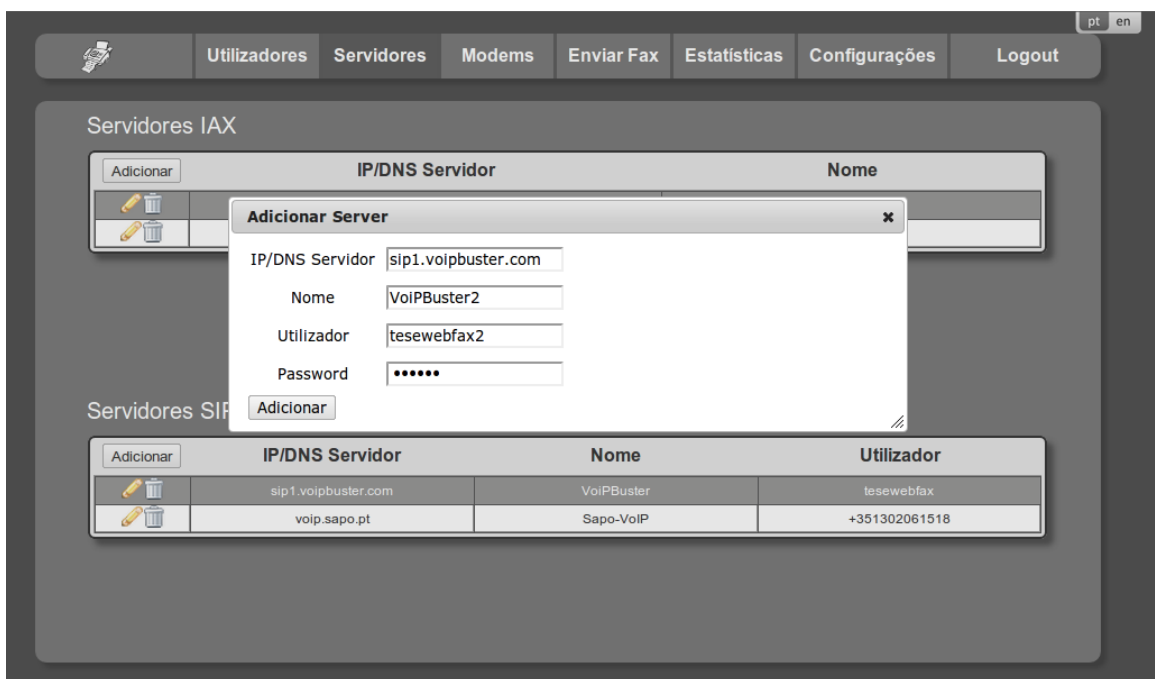


Figura 4.7: Menu de configuração de Servidores *VoIP*- Adicionar Servidor *IAX*

As figuras 4.8 e 4.9 ilustram, respectivamente a listagem dos *modems* configurados e a adição de mais um *modem* à configuração da plataforma.

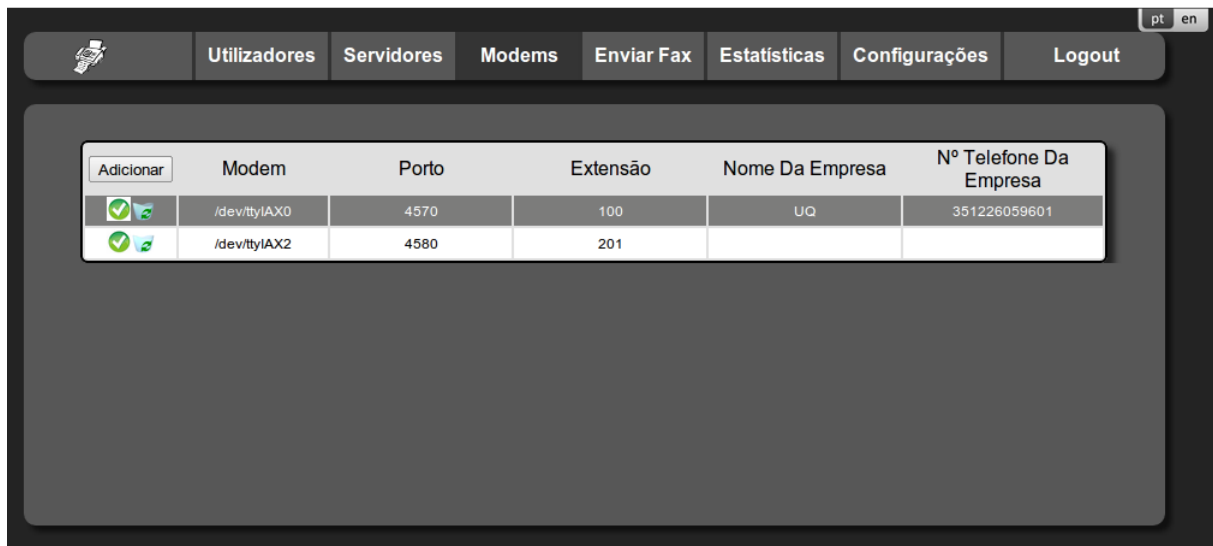


Figura 4.8: Menu de configuração de Modems

Nesta secção o administrador além de adicionar novas entradas pode editar e apagar entradas existentes.

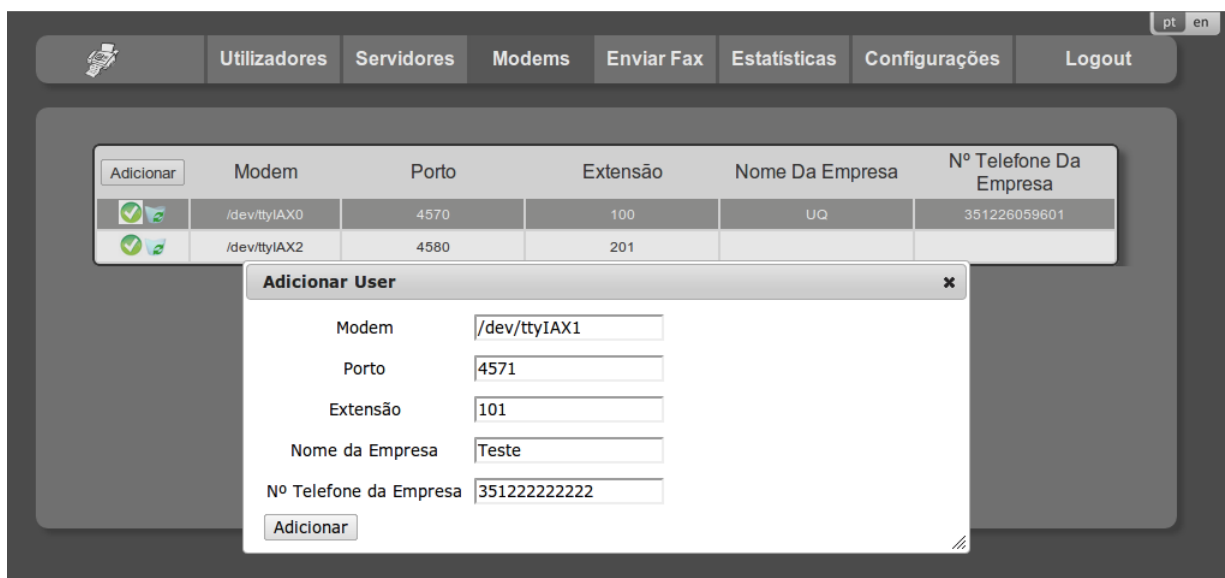


Figura 4.9: Menu de configuração de Modems - Adicionar Modem

A figura 4.10 ilustra a secção que permite o envio de um FAX, como se pode observar é possível adicionar múltiplos ficheiros ao mesmo trabalho, sendo as únicas limitações o formato dos mesmos, ou seja, encontra-se limitado ao envio de ficheiros *PDF*, *PS* e *TIFF* e o tamanho individual máximo de 20 *MBytes*.

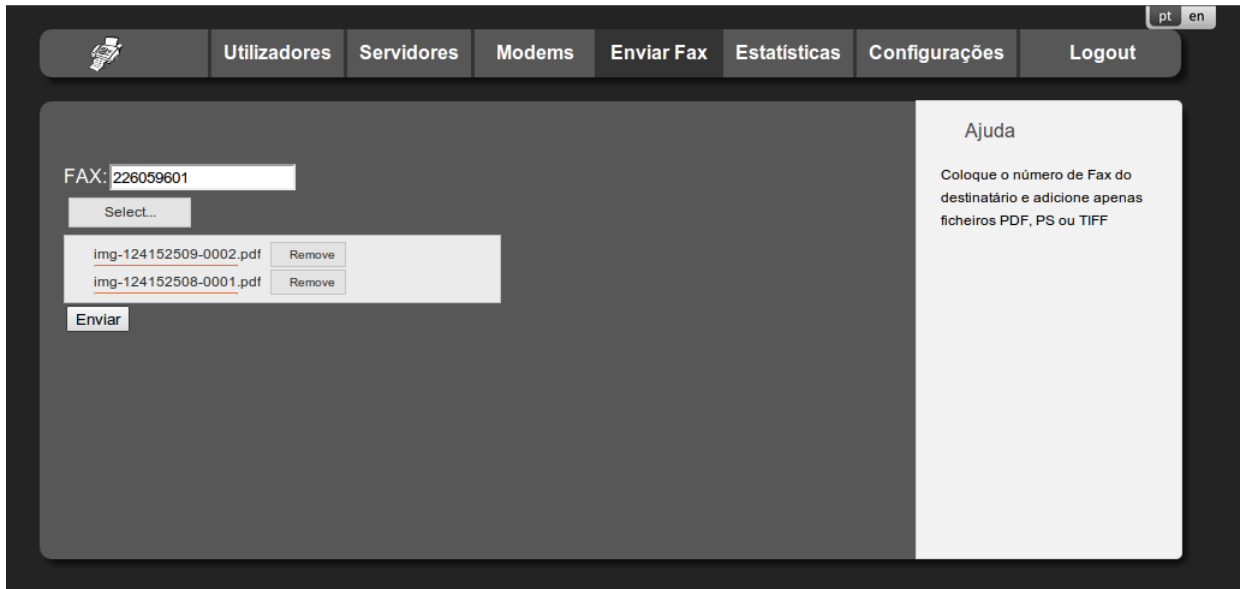


Figura 4.10: Menu de Envio de FAX

A figura 4.11 demonstra uma consulta a base de dados sobre os envios de FAX.

Data	ID Trabalho	Nº De Destino	Nº De Páginas
2011-08-01 00:45:16	209132	296302119	1
2011-08-01 00:46:03	209133	296302119	1
2011-08-01 02:12:51	209148	0034983303265	1
2011-08-01 02:13:00	209151	261940409	1
2011-08-01 02:13:02	209153	219540409	1
2011-08-01 02:13:39	209154	0034913292021	1
2011-08-01 02:13:43	209155	272321091	1
2011-08-01 02:14:31	209157	229993370	1
2011-08-01 02:14:32	209156	219540409	1
2011-08-01 02:14:33	209152	0034983303265	1
2011-08-01 02:15:18	209158	229993370	1
2011-08-01 02:15:22	209164	223395119	1
2011-08-01 02:16:05	209163	229993370	1
2011-08-01 02:17:05	209165	261331663	1
2011-08-01 02:19:34	209150	296642265	1
2011-08-01 02:21:05	209159	219747735	1

Figura 4.11: Menu de Estatísticas

4.3 *jhEDI*

Para o desenvolvimento da aplicação, *jhEDI*, optou-se pela linguagem de programação *Java*, de forma a alcançar um dos requisitos funcionais da interoperabilidade entre sistemas operativos. Para assegurar a ligação da aplicação com as duas bases de dados intervenientes nesta solução foi necessário utilizar conectores *Java Database Connectivity*, *JDBC*, nomeadamente:

- MySQL Connector/J [46]
- Oracle8i JDBC Drivers [47]

A interacção com o servidor de Fax, *HylaFAX*, e com o servidor de correio electrónico foram implementadas, respectivamente, recorrendo à *API GNU-HylaFAX* [48] e *JavaMAIL* [49].

A criação do ficheiro *PostScript* que serve de base ao Fax resulta da acção de uma classe *Java* desenvolvida pelo autor e que teve por base a informação constante no livro *PostScript Language Reference Manual* [50].

A estrutura do *email* que o servidor de encomendas gera foi adaptado de forma a incluir os campos correspondentes ao identificador da encomenda, número de fax e/ou endereço de email, passando o assunto a apresentar o seguinte formato:

#	Data	De	Assunto [Conversa]
7	12:15	foip@uq.pt	Enc=9027133 ID=e2336307.fax Fax=[fax=219748122]
6	12:15	foip@uq.pt	Enc=9027132 ID=e2336306.fax Fax=[fax=214398319]
5	12:15	foip@uq.pt	Enc=9027131 ID=e2336305.fax Fax=help.desk@progelcone.com
4	12:15	foip@uq.pt	Enc=9012011 ID=e2336303.fax Fax=[fax=218924282]
3	12:15	foip@uq.pt	Enc=9003703 ID=e2336302.fax Fax=[fax=0034923228607]
2	12:15	foip@uq.pt	Enc=9003702 ID=e2336301.fax Fax=[fax=0034923233623]
1	12:15	foip@uq.pt	Enc=9003701 ID=e2336300.fax Fax=pedidoscrm@casbegamail.com

Figura 4.12: Mensagem de *email* gerada pelo Servidor de Encomendas

Os campos correspondentes ao número de fax, endereço de email do fornecedor e número da encomenda encontram-se delimitados, respectivamente com as cores, vermelho, verde e preto. No caso da encomenda que apresenta o delimitador de cor vermelha o seu processamento ocorre como exemplificado na figura 4.16, o tratamento da encomenda com o delimitador de cor verde é apresentado no caso de uso da figura 4.18.

#	Data	De	Assunto [Conversa]
5	Quarta, 19 Agosto 20	VOIP Fax	Enc=9002376 ID=e1707601.fax Fax=[fax=0034968334628];pedidos@kernelexport.es
4	12:17	fax@uqfoip.uq.pt	facsimile job 239181 to 219748122 completed
3	12:16	fax@uqfoip.uq.pt	facsimile job 239179 to 218924282 completed

Figura 4.13: Mensagem de *email* gerada pelo Servidor de Encomendas

Nesta imagem podemos observar que o campo delimitado a preto apresenta quer o número de fax, quer o endereço de email do fornecedor. Este tipo de encomenda é processado segundo o caso de uso da figura 4.17.

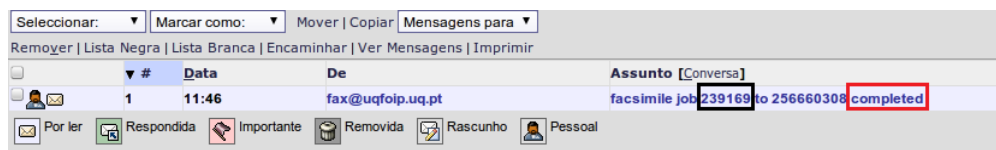


Figura 4.14: Mensagem de *email* gerada pelo Servidor de Encomendas

Nesta figura, o campo delimitado a preto corresponde ao identificador do trabalho de fax gerado pelo *HylaFAX* e o campo a vermelho o estado do mesmo trabalho. Este campo pode apresentar três variáveis:

- *completed* - corresponde a um envio de fax com sucesso
- *requeue* - a tentativa de envio do fax falhou; o ficheiro voltou a ser colocado na fila de saída para posterior tentativa
- *failed* - o trabalho de fax excedeu o número máximo de tentativas (12) e como tal a sua entrega é considerada falhada

O seguinte fluxograma apresenta as decisões e processos que a aplicação *jhEDI* toma no seu funcionamento.

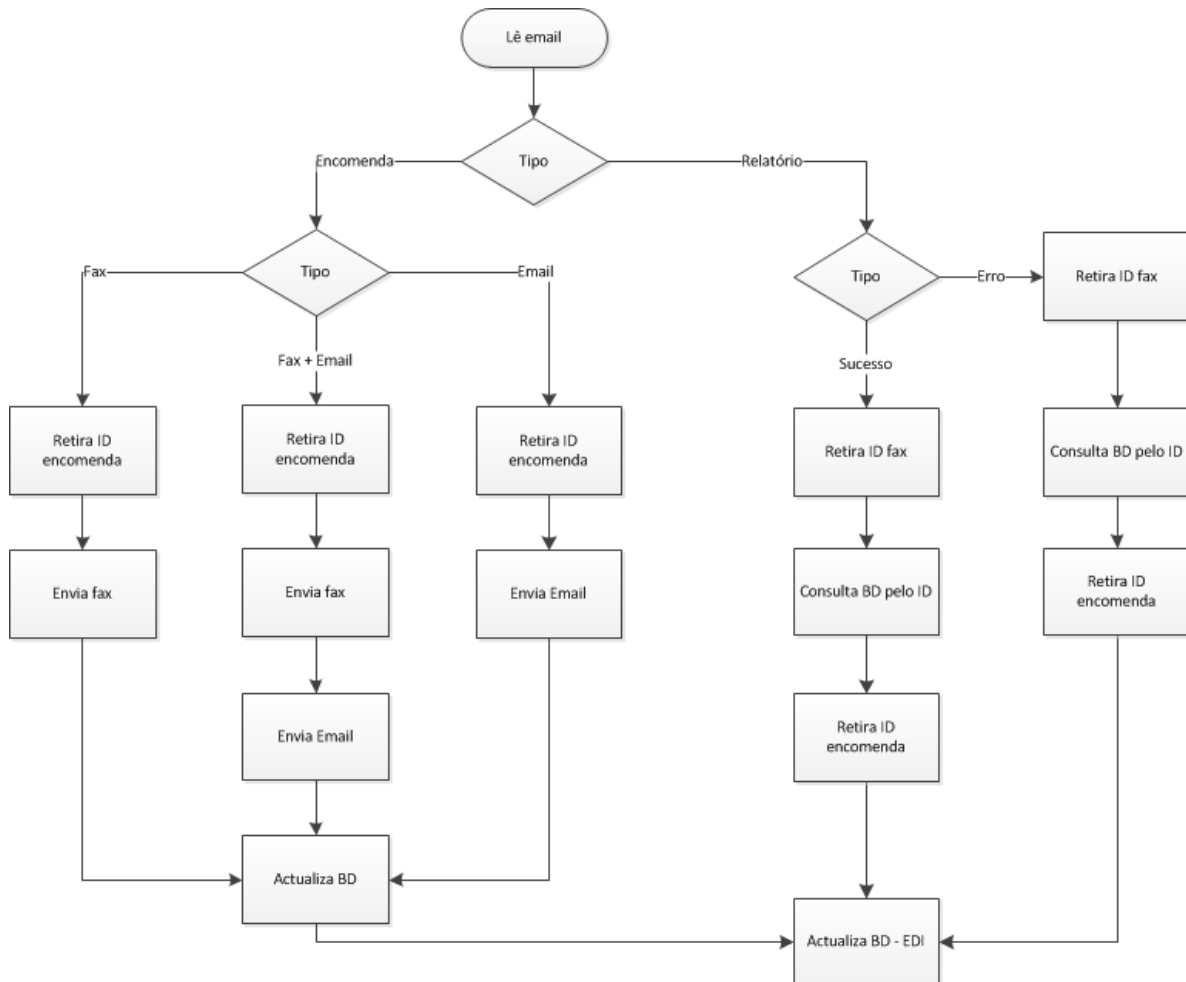


Figura 4.15: Fluxograma da Aplicação *jhEDI*

A figura 4.16 ilustra todos os passos que a aplicação *jhEDI* segue para o envio de uma encomenda via Fax. Como se pode observar existem dois ciclos, o primeiro lê o email transforma-o num fax e envia-o, guardando os dados relevantes na base de dados da aplicação e na base de dados do servidor *EDI*. No segundo ciclo a aplicação lê o email do relatório de entrega e cruza os dados para actualizar o estado da encomenda nas diferentes base de dados.

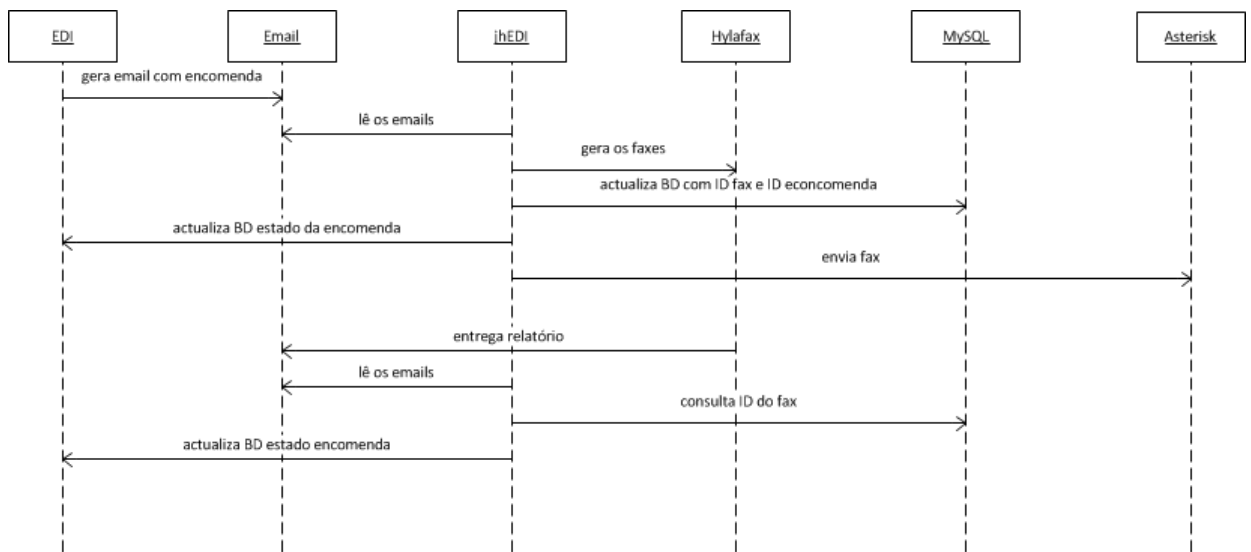


Figura 4.16: Diagrama de Sequência Temporal - Envio de Encomenda via Fax

O caso ilustrado pela figura 4.17 comparado como da figura 4.16 apresenta mais um passo que é o envio de um email logo após o envio do fax, de resto mantêm a mesma estrutura de dois ciclos, sendo que a confirmação do estado da encomenda obtido pelo relatório de entrega do fax.

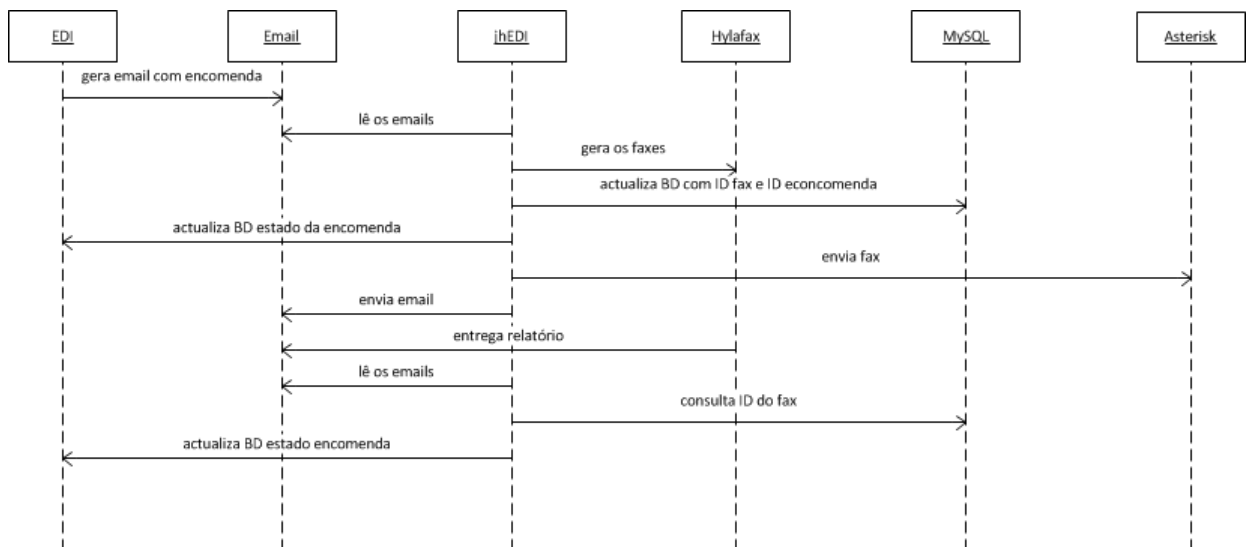


Figura 4.17: Diagrama de Sequência Temporal - Envio de Encomenda via Fax e Email

Neste último caso, ilustrado pela figura 4.18 existe apenas um ciclo visto que a confirmação da entrega da encomenda é realizada automaticamente após o envio do email por parte da aplicação *jhEDI*. Verifica-se ainda que não existe interações com o servidor de Fax nem com a base de dados *MySQL*.

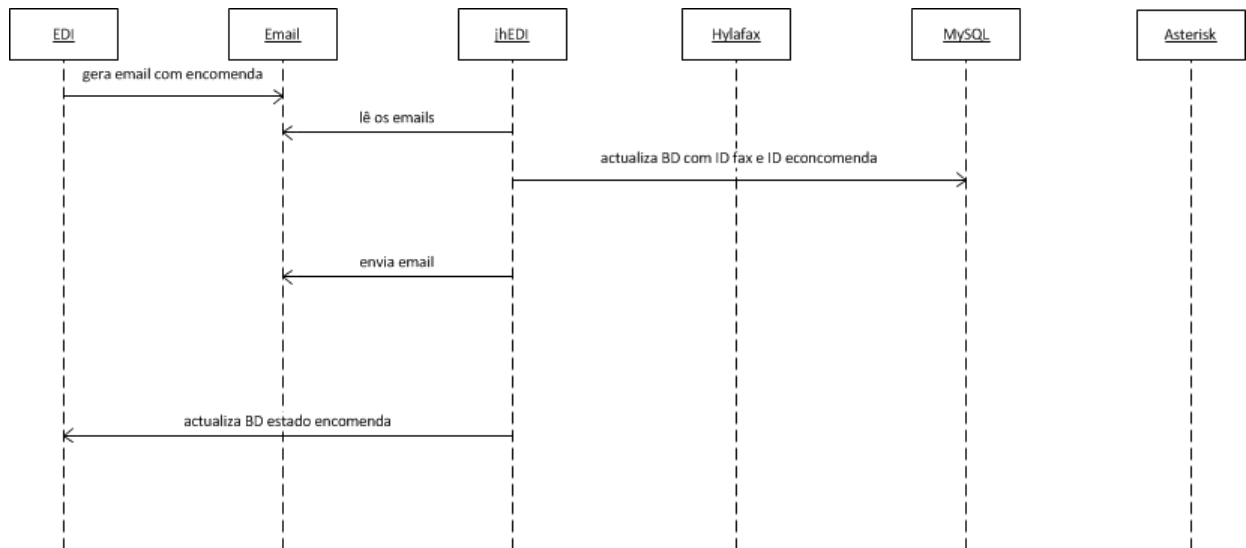


Figura 4.18: Diagrama de Sequência Temporal - Envio de Encomenda via Email

4.4 *jhAS400*

A aplicação *jhAS400* também foi desenvolvida em *Java*, utilizando tal como no *JhEDI*, a API *GNU-HylaFAX*. Para interligar o *ERP* ao nosso servidor de fax foi necessário criar uma impressora virtual; esta operação foi assegurada por uma aplicação *open-source*, *lp5250d* [51].

O fluxograma da figura 4.19 apresenta as várias etapas no envio de um fax via o *ERP* da empresa.

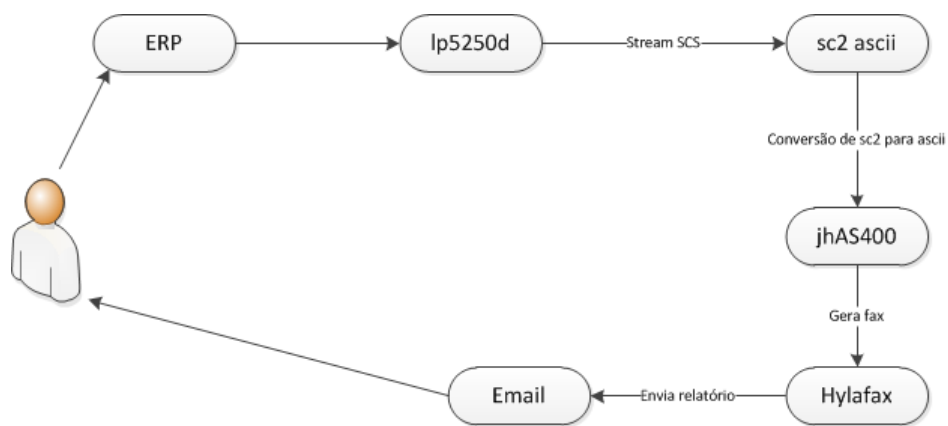


Figura 4.19: Fluxograma da Aplicação *jhAS400*

Como podemos observar, o utilizador após enviar o Fax recebe uma notificação do estado do processo. Para que esta funcionalidade fosse possível, foi necessário alterar a estrutura dos ficheiros gerados pelo *ERP* para que estes incluíssem o endereço de correio electrónico do utilizador. Foi, também, necessário garantir que o todos os números de telefone constantes na Base de Dados do *ERP*, associados a entidades externas, fossem do género *00351222222222*, de forma a assegurar uniformidade e que qualquer utilizador independentemente da sua localização geográfica pudesse beneficiar deste serviço.

A figura 4.20 apresenta a listagem de impressoras disponíveis no sistema. Dentro do delimitador de cor amarela podemos observar a nossa impressora virtual, cujo nome é *faxlogisti*; como podemos constatar existem parâmetros extras em relação as outras impressoras 'físicas', característica que a distingue.

As figuras 4.21 e 4.22 demonstram os procedimentos necessários para enviar um fax. Basicamente é necessário escolher o trabalho/ficheiro a enviar e modificar a impressora para *faxlogisti*, 'libertando' de seguida o ficheiro, tal como se pode ver na figura 4.23.

```

Trabalhar com Todas as Impressoras

Indique as opções, prima Enter.
 1=Iniciar   2=Alterar   3=Reter   4=Terminar   5=Trabalhar com   6=Libertar
 7=Ver mensagens   8=Trabalhar com fila de output

Opç  Disp      Est  Sep   Tipo Papel  Ficheiro  Utilizador  Dados Utlz
-  ALLWORK    END
-  DFX8000    END
-  FAXLOGISTI STR  *FILE *ALL
-  FAXTESTE   END
-  HP2600     END
-  IERAX2     END
-  IMPRESORA  END
-  INTEGRFIN  END
-  INTFACFOR  END
-  INTMOVCAIX END
-  INTRECIN   END

Mais...

Parâmetros para opções 1, 2, 3, 4, 6 ou comando
==> wrksplf
F3=Sair  F11=Vista 2  F12=Cancelar  F17=Início  F18=Fim  F24=Mais teclas

```

Figura 4.20: Envio de Fax via ERP - Listagem de Impressoras

```

Trabalhar com Todos os Ficheiros em Spool

Escreva as opções, prima Enter.
 1=Enviar   2=Alterar   3=Reter   4=Eliminar   5=Visualizar   6=Libertar
 7=Mensagens 8=Atributos   9=Trabalhar com estado de impressão

Opç  Ficheiro  Utilizador  Disp ou  Dados Utlz  Est  Total  Pág
Fila  Cópys
-  QPJOBLOG  GSANTOS    PRT03    QPADEV0013  RDY  4      1
-  QPJOBLOG  GSANTOS    QEZJOBLOG QPADEV006R  HLD  2      1
-  QPJOBLOG  GSANTOS    QEZJOBLOG QPADEV0069  HLD  4      1
-  QPJOBLOG  GSANTOS    QEZJOBLOG QPADEV007C  HLD  1      1

Fim

Parâmetros para as opções 1, 2, 3 ou comando
==>
F3=Sair  F10=Vista 4  F11=Vista 2  F12=Canc.  F22=Impressoras
F24=+ teclas

```

Figura 4.21: Envio de Fax via ERP - Escolha de Ficheiro

```

Alterar Atribs Fich em Spool (CHGSPLFA)

Indique opções, prima Enter.

Ficheiro em Spool . . . . . > QPJOBLOG      Nome, *SELECT
Nome trab. . . . . > QPADEV0013  Nome, *
  Utilizador . . . . . >   GSANTOS   Nome
  Número . . . . . >   976117   000000-999999
Número do ficheiro em Spool . . . > 1          1-999999, *ONLY, *LAST, *ANY
Nome do sistema do trabalho . . . > UQP7       Nome, *ONLY, *CURRENT, *ANY
Criação do ficheiro em spool:
  Data de criação . . . . . > 10122011   Data, *ONLY, *LAST
  Hora de criação . . . . . > 121939     Hora, *ONLY, *LAST
Impressora . . . . . > faxlogisti Nome, *SAME, *OUTQ
Sequência de impressão . . . . . > *SAME      *SAME, *NEXT
Tipo de papel . . . . . > *STD       Tipo de papel, *SAME, *STD
Cópias . . . . . > 1          1-255, *SAME
Reiniciar impressão . . . . . > *STRPAGE   Número, *SAME, *STRPAGE...

Fim

F3=Sair  F4=Parâmetros  F5=Actualizar  F10=Parâmetros adicionais
F12=Cancelar  F13=Como utilizar este ecrã  F24=Mais teclas

```

Figura 4.22: Envio de Fax via ERP - Alteração da Impressora

```

Trabalhar com Todos os Ficheiros em Spool

Escreva as opções, prima Enter.
1=Enviar  2=Alterar  3=Reter  4=Eliminar  5=Visualizar  6=Libertar
7=Mensagens  8=Atributos  9=Trabalhar com estado de impressão

Opc  Ficheiro  Utilizador  Disp ou  Dados Utlz  Est  Total  Pág
     Ficheiro  Utilizador  Fila      Dados Utlz  Est  Págs  Act  Cóps
-----
6  QPJOBLOG  GSANTOS  FAXLOGISTI  QPADEV0013  *CHG  4      1
   QPJOBLOG  GSANTOS  QEZJOBLOG  QPADEV006R  HLD    2      1
   QPJOBLOG  GSANTOS  QEZJOBLOG  QPADEV0069  HLD    4      1
   QPJOBLOG  GSANTOS  QEZJOBLOG  QPADEV007C  HLD    1      1

Fim

Parâmetros para as opções 1, 2, 3 ou comando
==>
F3=Sair  F10=Vista 4  F11=Vista 2  F12= Canc.  F22=Impressoras
F24=+ teclas

```

Figura 4.23: Envio de Fax via ERP - Confirmação do envio

4.5 Email-to-Fax

A funcionalidade *Email-to-Fax* foi obtida conjugando o servidor de correio electrónico disponibilizado pelo *Postfix* e por um *script bash*, *mail2fax.sh* (Anexo A.1.1). Este *script* retira o endereço do utilizador que enviou o *email* e número de telefone do recetor do fax; para tal, o destinatário do email deverá ser algo do género *222222222@fax.empresa.pt*.

Para que esta operação seja bem sucedida é necessário adicionar no servidor de correio electrónico da empresa uma regra de transporte para que todos os *emails* destinados ao domínio *fax.empresa.pt* sejam entregues no servidor de fax.

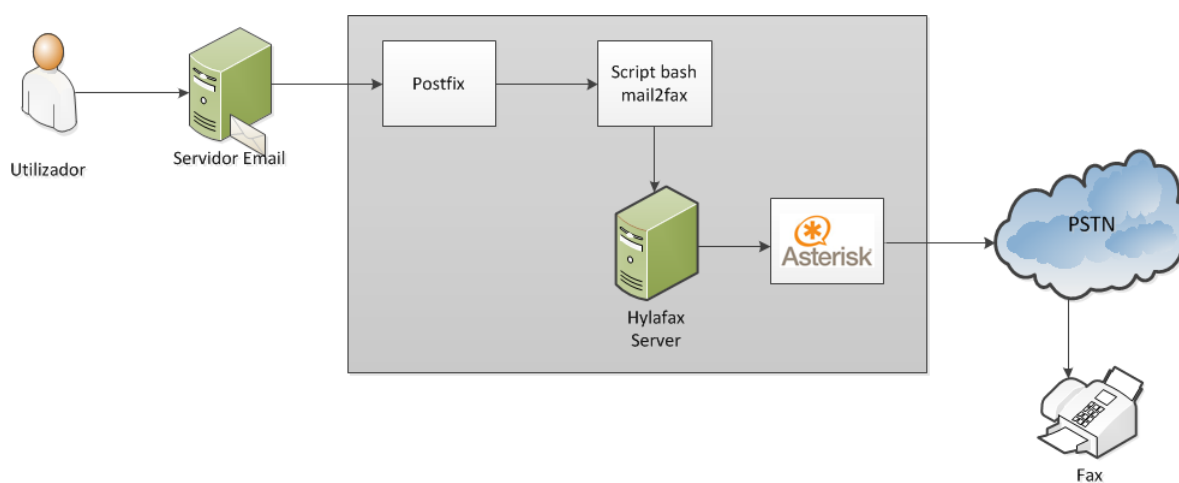


Figura 4.24: *Email-to-Fax* - Arquitectura da solução

Esta função encontra-se limitada ao envio de mensagens de correio electrónico cujos anexos sejam do tipo *PDF*, *PS* e *TIFF*, ou que a mensagem seja simples texto *ASCII*.

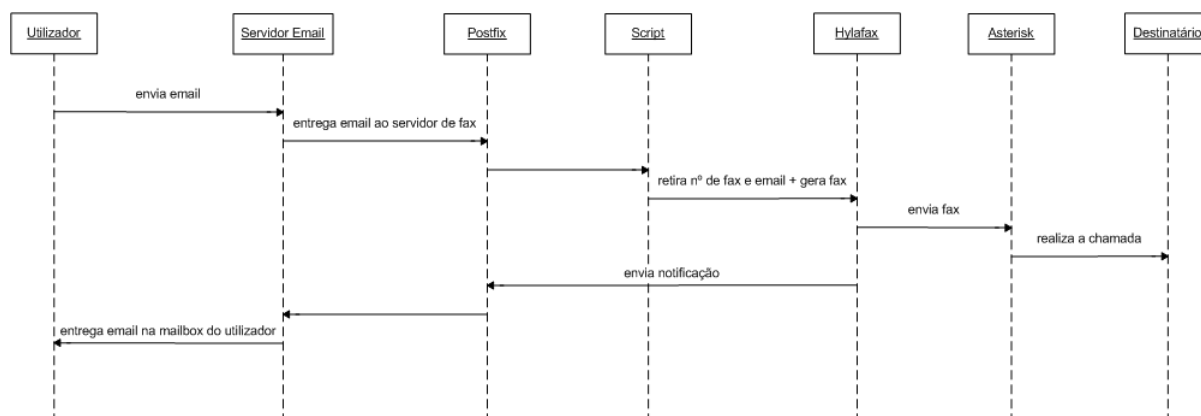


Figura 4.25: *Email-to-Fax* - Diagrama Temporal

A figura 4.25 ilustra as diferentes etapas no envio de Fax via correio electrónico.

4.6 Fax-to-Email

A semelhança da funcionalidade anterior, foi necessário conjugar um servidor *Postfix* e um *script bash*, *FaxDispatch*, que o servidor *HylaFAX* consulta aquando da receção de trabalho 4.26. Nesse ficheiro existem associações entre os *modem's* e os endereços de email do(s) utilizador(es) a quem deve ser entregue o Fax. Na mensagem de correio a enviar aos destinatários, segue, no corpo, os dados do emissor original em anexo um ficheiro *PDF* com a imagem do fax recebido. Para tudo funcionar como descrito, foi necessário configurar o *Postfix* para que o seu *relayhost* fosse o servidor de correio electrónico da empresa.

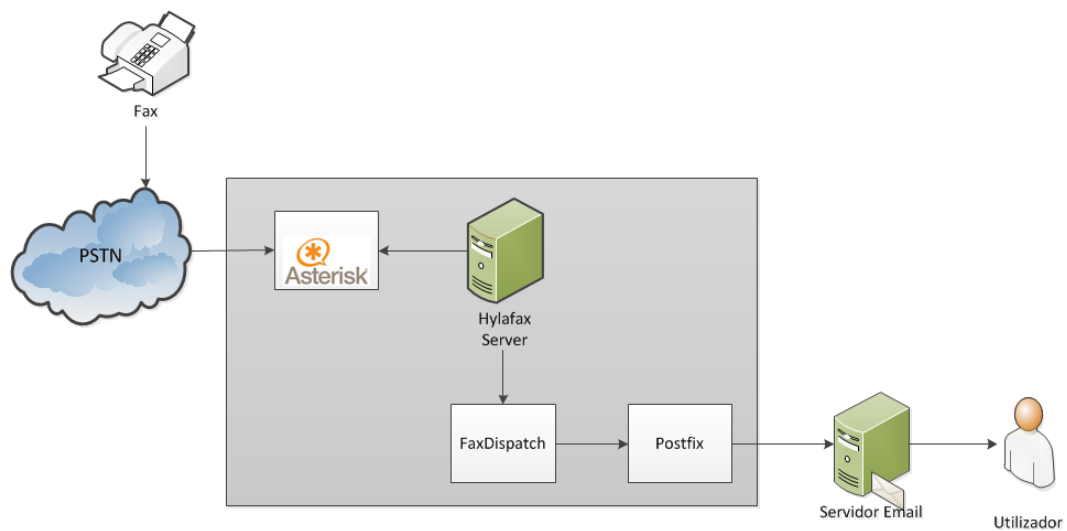


Figura 4.26: *Fax-to-Email* - Arquitectura da solução

A figura 4.27 ilustra as diferentes etapas no envio de Fax via correio electrónico.

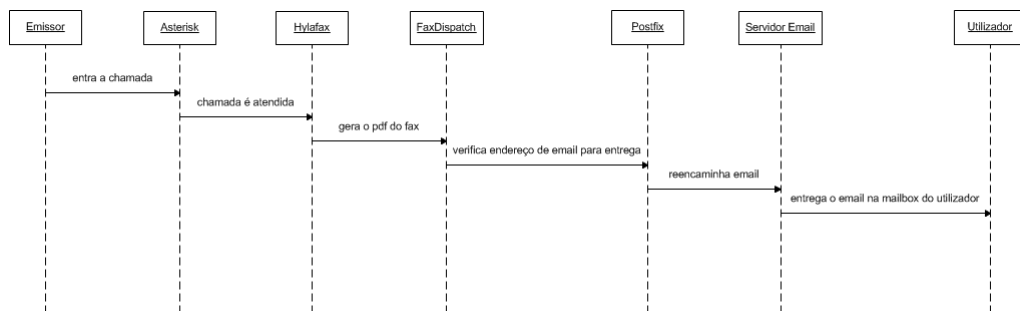


Figura 4.27: *Email-to-Fax* - Diagrama Temporal

4.7 Taxação

De forma a registar todos os movimentos criados pelo *HylaFAX* foi criado um *script bash* que procede à leitura do ficheiro de registos da aplicação e actualiza uma tabela na base de dados; desta forma, além dos dados referentes ao faxes enviados e recebidos é possível obter estatísticas sobre o número de tentativas e dos tempos médios gastos e o número de páginas por trabalho .

4.8 Servidor *OpenVPN*

Atendendo aos requisitos funcionais impostos para este serviço, nomeadamente a alta disponibilidade, foi necessário implementar dois servidores a correrem o *software OpenVPN*. Este fato levou ao desenho da solução ilustrado na figura 4.28.

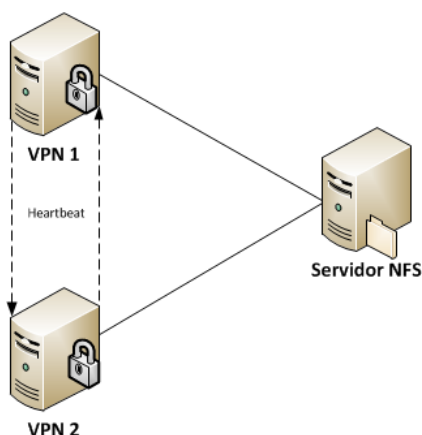


Figura 4.28: Arquitectura da plataforma de *VPN* - Sincronismo

Para garantir a sincronismo de contas e configurações do servidor de *VPN*, o autor decidiu-se aproveitar uma das características do *software OpenVPN*, o fato de toda a estrutura se basear em ficheiros de “texto”. Desta forma se ambos os servidores partilharem a mesma estrutura de ficheiros fica assegurado o sincronismo de contas e configurações, tornando o objetivo da alta disponibilidade menos complexo de se alcançar.

Para implementar esse sincronismo, a escolha passou pela criação de uma área partilhada entre esses servidores, disponibilizada via *NFS*, por uma *Network-Attached Storage, NAS*, existente na infraestrutura da empresa; assim, qualquer alteração realizada por um dos nós ficará instantaneamente disponível para o outro.

4.8.1 Interface de Administração do servidor *OpenVPN*

A Interface de Administração do servidor *OpenVPN* fica a cargo do *software Webmin* [52] em conjunção com um módulo [53]. Desta forma o administrador da plataforma pode desempenhar todas as seguintes tarefas:

- Criar/Eliminar o certificado *CA*
- Criar/Eliminar o certificado da *VPN*
- Criar/Eliminar o certificado dos clientes
- Criar/Editar/Eliminar as configurações associadas a cada *VPN*
- Obter estatísticas em tempo real do tráfego realizado e dos clientes conectados

A sequência de passos necessários para criar uma *VPN* funcional com clientes associados é exemplificada na figura 4.29 e consistem em:

- Criar certificado *Certificate Authority, CA* (são certificados “auto-assinados” criados via *OpenSSL*)
- Criar certificado para o servidor *VPN*
- Criar certificado para o cliente
- Associar cliente à *VPN*

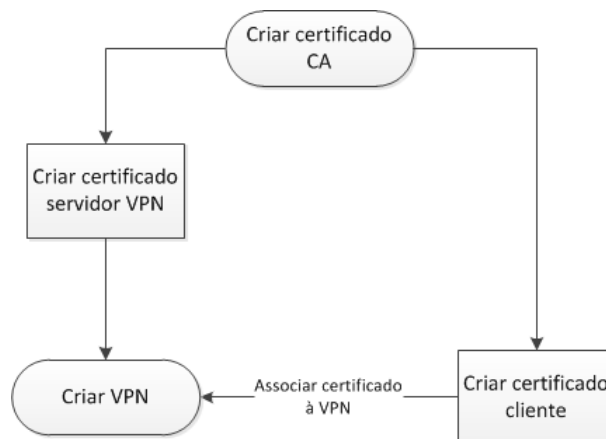


Figura 4.29: Criação de uma *VPN*

A figura 4.30 é um excerto da interface de criação e revogação dos certificados usados pelo servidor de *VPN*. É neste menu que podemos criar os *Certificate Authority, CA*, que serviram de base para gerar os certificados associados às instâncias de *VPN* e aos utilizadores associados a elas.

[Webmin Index](#)
[Help..](#)
[Module Config](#)

OpenVPN Administration

OpenVPN version 2.0_rc16, OpenSSL version 0.9.7e

[OpenVPN Administration](#)

Certification Authority List

Name	Notes	Info	Keys list	Remove
uq-lan2lan		CA Info	Keys list	Remove
uqcallcenters		CA Info	Keys list	Remove
uqconsult-vpn		CA Info	Keys list	Remove
uqgestor		CA Info	Keys list	Remove
uqibs		CA Info	Keys list	Remove
uqout		CA Info	Keys list	Remove
vivacity		CA Info	Keys list	Remove

New Certification Authority

Name of Certification Authority	changeme
Complete path to openssl.cnf	/etc/openvpn/openvpn-ssl.cnf
Keys directory	/etc/openvpn/keys
Key size (bit)	2048
Expiration time of Certification Authority key (days)	3650
State	US
Province	NY
City	New York
Organization	My Org
Email	me@my.org

Figura 4.30: Webmin - Menu de configuração de gestão de certificados

Na figura 4.31 podemos observar a listagem das VPN's existentes e aceder às suas opções de configuração.

[Webmin Index](#)
[Help..](#)
[Module Config](#)

OpenVPN Administration

OpenVPN version 2.0_rc16, OpenSSL version 0.9.7e

[OpenVPN Administration](#)

VPN server list:

Name	management	CA	proto	port	local	Logs	Client List	Status	Remove	Actions
UQCALLCENTER-SUPERVISAO		uq-lan2lan	udp		ALL	Log	Client List	Disable		stop
UQCALLCENTERS		uqcallcenters	udp		ALL	Log	Client List	Disable		stop
UQCC		uqconsult-vpn	udp		ALL	Log	Client List	Disable		stop
UQLAN2LAN		uq-lan2lan	udp		ALL	Log	Client List	Disable		stop
UQLAN2LAN2		uq-lan2lan	udp		ALL	Log	Client List	Disable		stop
UQLAN2LAN3		uq-lan2lan	udp		ALL	Log	Client List	Disable		stop
UQLAN2LAN4		uq-lan2lan	udp		ALL	Log	Client List	Disable		stop
VIVACITY		vivacity	udp		ALL	Log	Client List	Disable		stop
uqgestor		uqgestor	udp		ALL	Log	Client List	Disable		stop
uqibs		uqibs	udp		ALL	Log	Client List	Disable		stop
uqout		uqout	udp		ALL	Log	Client List	Disable		stop
vpnug		uqconsult-vpn	udp		ALL	Log	Client List	Disable		stop
vpnug-tcp		uqconsult-vpn	tcp		ALL	Log	Client List	Disable		stop

ca (Certification Authority): uq-lan2lan

Creation of new VPN Server: select the Certification Authority and click New VPN server

VPN server list with simmetric key

VPN List is empty

Creation of new VPN Server with symmetrical key

Figura 4.31: Webmin - Menu de configuração de VPN's

A figura 4.32 é um excerto da página de estatísticas disponibilizada pela Interface de Administração. Os campos principais são *Name* e *Virtual IP*, que representam respectivamente, a identificação do cliente conectado ao servidor e seu *IP* na rede da empresa. Existe ainda informação sobre o tráfego gerado, do seu *IP* público e do momento em que a ligação foi iniciada.

Active connection to VPN server						
Name	Virtual IP	Bytes received	Bytes sent	Real address	Connected since	remove key
VPN server: vpnuq						
armseixal	10.10.10.22	2511983	8325441	89.180.48.100:33091	Sat Jan 21 08:18:35 2012	Stop and Disable
cristinareis	10.10.10.46	34866027	246902204	93.102.32.26:55978	Thu Jan 19 17:13:42 2012	Stop and Disable
gabriel_santos	10.10.10.170	106417533	3250615765	93.108.190.202:65507	Sat Jan 21 17:41:13 2012	Stop and Disable
helenapereira	10.10.10.98	10237470	16309514	89.153.20.130:1840	Sat Jan 21 18:07:32 2012	Stop and Disable
luismeiros	10.10.10.238	382246	658343	94.132.79.170:51276	Sat Jan 21 11:22:58 2012	Stop and Disable
mariajarre	10.10.10.166	40888863	33822723	89.181.185.135:52452	Sat Jan 21 13:57:29 2012	Stop and Disable
martapaiva	10.10.10.162	7997880	18277630	78.130.10.218:1070	Sat Jan 21 13:23:17 2012	Stop and Disable
mjviana	10.10.10.42	77925	84853	188.82.183.42:60535	Sat Jan 21 19:25:38 2012	Stop and Disable
nilopes	10.10.10.34	8098951	11047514	82.154.91.28:59867	Fri Jan 20 19:03:37 2012	Stop and Disable
nunoalcobia	10.10.10.130	74968	170710	78.130.32.167:1290	Sat Jan 21 22:16:40 2012	Stop and Disable
pbcosta	10.10.10.158	1207196	34447878	62.169.108.200:47070	Sat Jan 21 21:28:32 2012	Stop and Disable
ruivelar	10.10.10.126	7018968	11683583	85.247.97.229:57318	Sat Jan 21 06:25:46 2012	Stop and Disable
ruiramalho	10.10.10.50	147221	19786	41.72.2.117:64338	Sat Jan 21 22:18:01 2012	Stop and Disable
sobibliotag	10.10.10.78	1694352	2834899	93.102.51.137:32768	Sat Jan 21 11:07:59 2012	Stop and Disable
soniajesus	10.10.10.6	137225889	1425310218	93.102.137.238:1035	Fri Jan 20 22:02:57 2012	Stop and Disable
lvbraga	10.10.10.82	46484	85730	87.196.168.180:34048	Sat Jan 21 22:18:15 2012	Stop and Disable

Figura 4.32: Webmin - Estatísticas das VPN's

4.8.2 Heartbeat - Alta Disponibilidade

O serviço *Heartbeat*, tal como foi referido na secção 3.2.6, utiliza pedidos *ICMP* para avaliar o estado de cada nó, por esta razão é necessário, para o caso de dois nós utilizar 3 *IP*'s; um para cada nó e terceiro será o *Virtual IP*, *VIP*.

O *VIP* será o *IP* 172.16.0.2, pelo qual o sistema deverá ser acedidos por todos os outros activos de rede, sendo que os outros dois *IP*'s servem apenas para manutenção e/ou para a implementação do mecanismo de monitorização do *Heartbeat*.

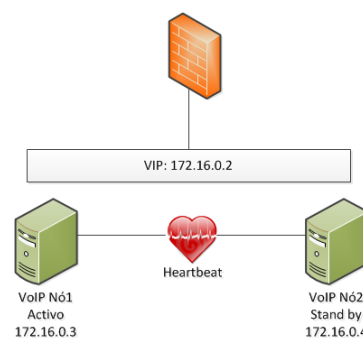


Figura 4.33: Heartbeat - Exemplo de Implementação

A figura 4.33 ilustra a implementação utilizada neste trabalho.

Além dos nós usarem pedidos *ICMP* entre si, devem também, trocar pedidos com um terceiro equipamento de forma a eliminarem problemas de conectividade com a rede da empresa.

Apesar da implementação deste mecanismo a plataforma não assegura cem por cento de disponibilidade, uma vez, que ao ser detetada uma falha no nó de produção a comutação do *VIP* demora alguns segundos e esta operação obriga a que todos os clientes voltem a autenticar-se no servidor de *VPN*, que neste caso seria o nó 2.

Capítulo 5

Testes e Resultados

Neste capítulo são apresentados os testes realizados bem como os resultados obtidos.

5.1 Envio e Recepção de Fax via Operador SIP

Para realizar testes ao servidor de Fax foram criado os seguintes cenários de teste:

- Envio de um Fax tendo por base um documento com texto e imagens no formato *PostScript*
- Recepção de um Fax tendo por base um documento de texto

Para validar o funcionamento da plataforma todo o tráfego do servidor de Fax foi recolhido com a ferramenta *tcpdump*.

A imagem 5.1 ilustra o estabelecimento da chamada SIP entre o servidor e o operador VoIP.

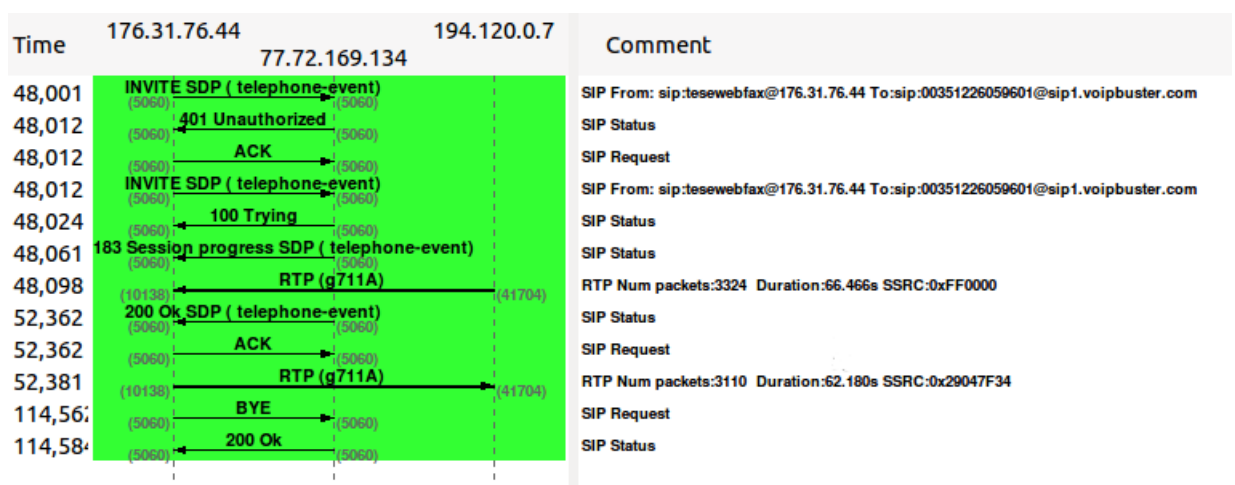


Figura 5.1: Envio de Fax - Estabelecimento da chamada SIP

A figura 5.1 demonstra a recepção de um Fax enviado pelo número 223097000 através um canal *SIP* fornecido pelo serviço *Sapo VoIP*.

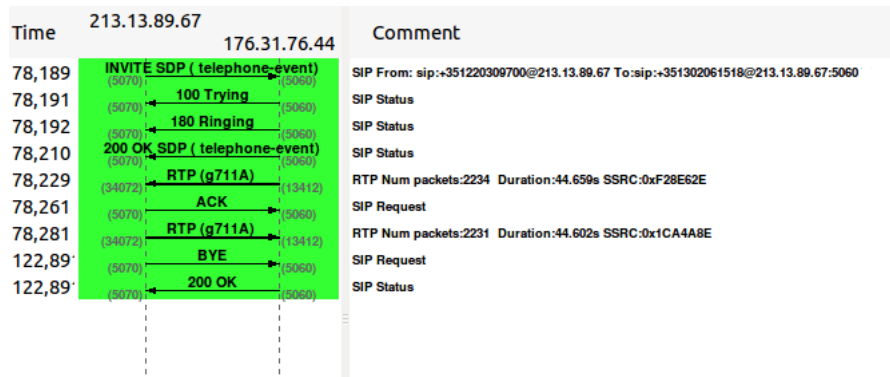


Figura 5.2: Recepção de Fax - Recepção da chamada através de um canal *SIP*

Como podemos observar na figura a chamada entrou pelo canal *SIP* e após a negociação dos parâmetros da chamada (*200 OK SDP*) foi estabelecido uma ligação *RTP* utilizando o *codec g.711A*.

Na figura 5.3 pode-se observar o fluxo de áudio no canal *RTP* da chamada em análise.

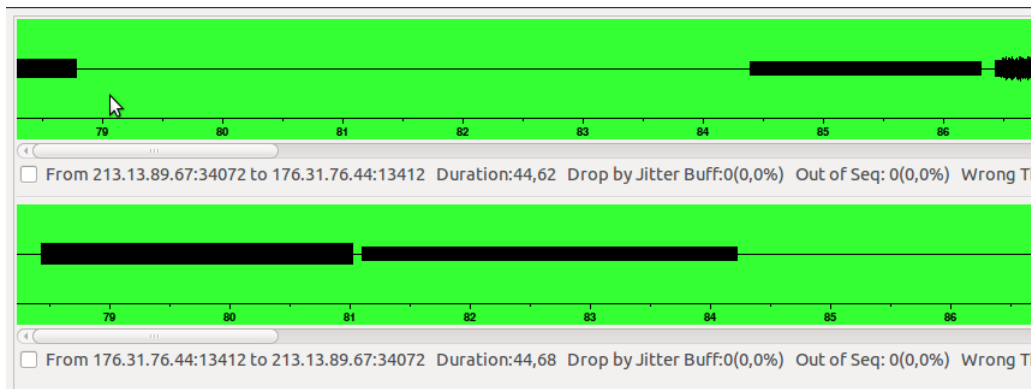


Figura 5.3: Envio de Fax - Estabelecimento da chamada *SIP* canal *RTP*

5.2 Envio de Fax via *jhAS400*

Time	AS400	JhAS400	HylaFAX	Comment
6,755		Telnet Data ...		TELNET: Telnet Data ...
6,755		Telnet Data ...		TELNET: Telnet Data ...
6,755		telnet > 38166 [PSH]		TCP: telnet > 38166 [PSH, ACK] Seq=32 Ack=13 Win=32768 Len=0 TSV=1316934000 TSER=4294954052
6,766		Telnet Data ...		TELNET: Telnet Data ...
6,766		Telnet Data ...		TELNET: Telnet Data ...
6,766		telnet > 38166 [PSH]		TCP: telnet > 38166 [PSH, ACK] Seq=819 Ack=25 Win=32768 Len=0 TSV=1316934000 TSER=4294954053
6,766		Telnet Data ...		TELNET: Telnet Data ...
6,766		Telnet Data ...		TELNET: Telnet Data ...
6,767		telnet > 38166 [PSH]		TCP: telnet > 38166 [PSH, ACK] Seq=1605 Ack=37 Win=32768 Len=0 TSV=1316934000 TSER=4294954053
6,767		Telnet Data ...		TELNET: Telnet Data ...
6,767		Telnet Data ...		TELNET: Telnet Data ...
6,767		telnet > 38166 [PSH]		TCP: telnet > 38166 [PSH, ACK] Seq=2391 Ack=49 Win=32768 Len=0 TSV=1316934000 TSER=4294954054
6,767		Telnet Data ...		TELNET: Telnet Data ...
6,767		Telnet Data ...		TELNET: Telnet Data ...
6,768		telnet > 38166 [PSH]		TCP: telnet > 38166 [PSH, ACK] Seq=3177 Ack=61 Win=32768 Len=0 TSV=1316934000 TSER=4294954054
6,768		Telnet Data ...		TELNET: Telnet Data ...
6,768		Telnet Data ...		TELNET: Telnet Data ...
6,768		telnet > 38166 [PSH]		TCP: telnet > 38166 [PSH, ACK] Seq=3432 Ack=73 Win=32768 Len=0 TSV=1316934000 TSER=4294954054
6,768		Telnet Data ...		TELNET: Telnet Data ...
6,768		Telnet Data ...		TELNET: Telnet Data ...
6,769		telnet > 38166 [PSH]		TCP: telnet > 38166 [PSH, ACK] Seq=3451 Ack=85 Win=32768 Len=0 TSV=1316934000 TSER=4294954054
6,929		60496 > hylafax [SYN]		TCP: 60496 > hylafax [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=4294954070 TSER=0 WS=7
6,929		hylafax > 60496 [SYN]		TCP: hylafax > 60496 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=1352783947 TSER=4294954070
6,929		60496 > hylafax [ACK]		TCP: 60496 > hylafax [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=4294954070 TSER=1352783947
6,931		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=1 Ack=1 Win=5888 Len=85 TSV=1352783949 TSER=4294954070
6,931		60496 > hylafax [ACK]		TCP: 60496 > hylafax [ACK] Seq=1 Ack=86 Win=5888 Len=0 TSV=4294954070 TSER=1352783949
6,932		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=86 Ack=1 Win=5888 Len=55 TSV=1352783950 TSER=4294954070
6,932		60496 > hylafax [ACK]		TCP: 60496 > hylafax [ACK] Seq=1 Ack=141 Win=5888 Len=0 TSV=4294954070 TSER=1352783950
6,932		60496 > hylafax [PSH]		TCP: 60496 > hylafax [PSH, ACK] Seq=1 Ack=141 Win=5888 Len=11 TSV=4294954070 TSER=1352783950
6,932		hylafax > 60496 [ACK]		TCP: hylafax > 60496 [ACK] Seq=141 Ack=12 Win=5888 Len=0 TSV=1352783950 TSER=4294954070
6,932		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=141 Ack=12 Win=5888 Len=33 TSV=1352783950 TSER=4294954070
6,932		60496 > hylafax [PSH]		TCP: 60496 > hylafax [PSH, ACK] Seq=12 Ack=174 Win=5888 Len=11 TSV=4294954071 TSER=1352783950
6,932		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=174 Ack=23 Win=5888 Len=26 TSV=1352783955 TSER=4294954071
6,937		60496 > hylafax [PSH]		TCP: 60496 > hylafax [PSH, ACK] Seq=23 Ack=200 Win=5888 Len=6 TSV=4294954071 TSER=1352783955
6,937		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=200 Ack=29 Win=5888 Len=30 TSV=1352783956 TSER=4294954071
6,937		60496 > hylafax [PSH]		TCP: 60496 > hylafax [PSH, ACK] Seq=29 Ack=230 Win=5888 Len=13 TSV=4294954071 TSER=1352783956
6,937		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=230 Ack=42 Win=5888 Len=31 TSV=1352783956 TSER=4294954071
6,941		60496 > hylafax [PSH]		TCP: 60496 > hylafax [PSH, ACK] Seq=42 Ack=261 Win=5888 Len=29 TSV=4294954071 TSER=1352783956
6,941		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=261 Ack=71 Win=5888 Len=30 TSV=1352783959 TSER=4294954071
6,941		60496 > hylafax [PSH]		TCP: 60496 > hylafax [PSH, ACK] Seq=71 Ack=291 Win=5888 Len=6 TSV=4294954071 TSER=1352783959
6,942		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=291 Ack=77 Win=5888 Len=60 TSV=1352783960 TSER=4294954071
6,977		60496 > hylafax [ACK]		TCP: 60496 > hylafax [ACK] Seq=77 Ack=351 Win=5888 Len=0 TSV=4294954075 TSER=1352783960
6,977		hylafax > 60496 [PSH]		TCP: hylafax > 60496 [PSH, ACK] Seq=351 Ack=77 Win=5888 Len=50 TSV=1352783995 TSER=4294954075
6,977		60496 > hylafax [ACK]		TCP: 60496 > hylafax [ACK] Seq=77 Ack=401 Win=5888 Len=0 TSV=4294954075 TSER=1352783995
6,980		60496 > hylafax [PSH]		TCP: 60496 > hylafax [PSH, ACK] Seq=77 Ack=401 Win=5888 Len=6 TSV=4294954075 TSER=1352783995

Figura 5.4: Envio de Fax via *jhAS400*

A figura 5.4 demonstra as duas etapas no envio via *jhAS400*. Numa primeira fase assisti-se a uma “conversação” entre o *AS400* e o servidor onde corre a aplicação *jhAS400*, o conteúdo deste fluxo TCP é um stream em EBCDIC (Anexo A.1.6). Na segunda fase intervêm o servidor

HylaFax e o *jhAS400*, neste caso podemos observar a conjunto de mensagens trocadas em *ASCII* muito similar ao que encontramos numa sessão de *FTP* (Anexo A.1.6).

5.3 Dados estatísticos do envio de Encomenda via *jhEDI*

Os dados apresentados na tabela 5.1 referem-se aos últimos 4 meses em que a aplicação *jhEDI* esteve em produção.

Tabela 5.1: Distribuição de Fax-Encomendas por Mês

Mês / Fax Encomendas	Enviadas	Sucesso	Sucesso (%)	Falhas	Falhas (%)
Agosto	5679	5560	97,9	119	2,1
Setembro	5466	5240	95,9	226	4,1
Outubro	5393	5210	96,6	183	3,4
Novembro	5160	4970	96,3	190	3,7
Dezembro	4712	4639	98,4	73	1,8
Total	26410	25619	97,0	791	3,0

Como podemos observar as percentagens de envio com sucesso, por mês, são sempre superiores a 95 %, sendo a percentagem total de 97%.

5.4 Dados estatísticos do envio de Fax via *jhAS400* e via Email

Os dados apresentados na tabela 5.2 referem-se aos últimos 4 meses em que a aplicação *jhAS400* e o serviço *Email-to-Fax* estiveram em produção.

Tabela 5.2: Distribuição de Faxes por Mês

Mês / Faxes	<i>jhAS400</i>	Email
Agosto	468	132
Setembro	356	102
Outubro	265	98
Novembro	349	92
Dezembro	212	69
Total	1650	493

5.5 Dados estatísticos da recepção de Faxes

Os dados apresentados na tabela 5.3 referem-se aos últimos 4 meses em que o serviço *Fax-to-Email* esteve em produção.

Tabela 5.3: Distribuição de Faxes Recebidos por Mês

Mês	Faxes recebidos
Agosto	135
Setembro	125
Outubro	117
Novembro	95
Dezembro	80
Total	552

5.6 Testes de conectividade cliente-VPN-sede

Este primeiro bloco demonstra o túnel VPN e o *IP* atribuído ao cliente

```
vbraga@vbraga-vaio:/$ ifconfig tun0
tun0      Link encap: N o Especificado  Endere o de HW
          00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet end.: 10.10.10.82  P-a-P:10.10.10.81  Masc:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  M trica:1
          pacotes RX:16465  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:13391  erros:0  descartados:0  excesso:0  portadora:0
          colis es:0  txqueuelen:100
          RX bytes:14385474 (14.3 MB) TX bytes:958194 (958.1 KB)
```

Recorrendo ao comando *tracpath* podemos validar a conectividade entre o cliente e a sede e visualizar os *saltos* que ocorrem

```
vbraga@vbraga-vaio:/$ tracpath 192.168.86.1
1:  10.10.10.82                0.110ms pmtu 1500
1:  10.10.10.1                  24.993ms
1:  10.10.10.1                  24.191ms
2:  192.168.86.1                26.445ms reached
Resume: pmtu 1500 hops 2 back 254
```

5.7 Testes de conectividade cliente-VPN-Escritório Remoto

Neste bloco testamos o acesso à um escritório remoto também interligado à sede via *VPN*

```
vbraga@vbraga-vaio:/$ traceroute 172.28.1.227
 1:  10.10.10.82                0.124ms pmtu 1500
 1:  10.10.10.1                 25.823ms
 1:  10.10.10.1                 25.511ms
 2:  no reply
 3:  172.28.1.227               81.581ms reached
Resume: pmtu 1500 hops 3 back 62
```

5.8 Testes de *throughput* cliente-VPN-sede

Para realizar os testes de *throughput* utilizou-se a aplicação *NetPerf*.

Neste primeiro teste foi utilizado um acesso de 100 *Mbit/s*:

```
braga@vbraga-vaio:~$ netperf -f k -p 12345 -H 192.168.86.92
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.86.92 (192.168.86.92)
port 0 AF_INET : demo
Recv  Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time  Throughput
bytes bytes bytes secs.  10^3 bits/sec

87380 16384 16384 10.09 99308.65
```

No segundo teste foi utilizado um acesso *ADSL* cujo velocidade de *download* rondava os 14 *Mbit/s*:

```
vbraga@vbraga-vaio:~$ netperf -f k -p 12345 -H 192.168.86.92
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.86.92 (192.168.86.92)
port 0 AF_INET : demo
Recv  Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time  Throughput
bytes bytes bytes secs.  10^3 bits/sec

87380 16384 16384 11.17 3419.19
```

No terceiro teste o acesso utilizado foi acesso *UMTS*:

```
braga@vbraga-vaio:~$ netperf -f k -p 12345 -H 192.168.86.92
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.86.92 (192.168.86.92)
port 0 AF_INET : demo
Recv  Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time  Throughput
bytes bytes bytes secs.  10^3 bits/sec

87380 16384 16384 10.12 13904.82
```

5.9 Dados estatísticos da VPN

As imagens 5.5, 5.6 ilustram os valores de tráfego que a plataforma de VPN gera.

A imagem 5.5 ilustra o tráfego gerado ao longo de uma semana:

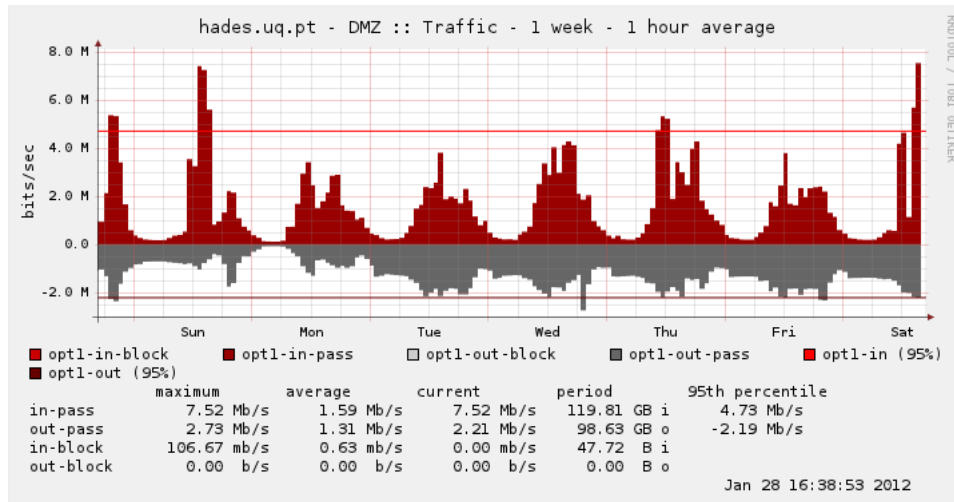


Figura 5.5: Gráfico da utilização de largura da banda pelo servidor de VPN - Semana

A imagem 5.6 ilustra o tráfego gerado ao longo de um mês:

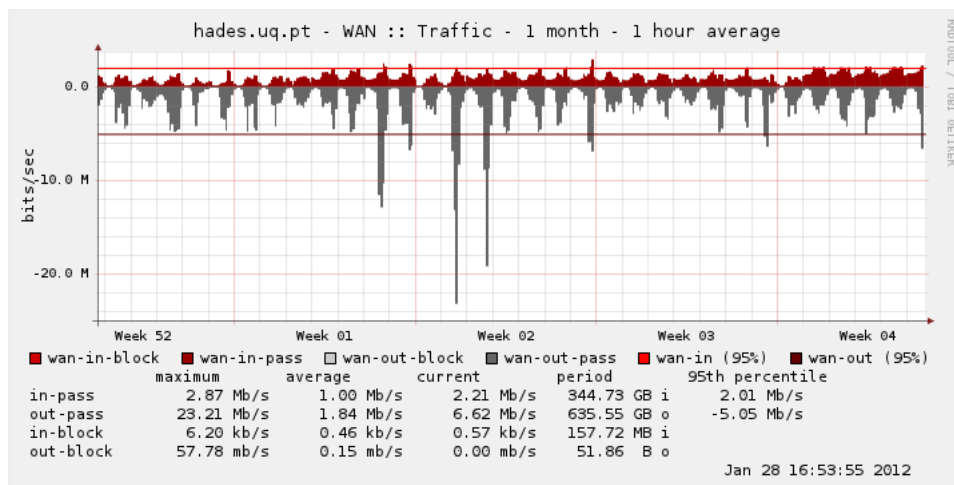


Figura 5.6: Gráfico da utilização de largura da banda pelo servidor de VPN - Mês

Como se pode observar o perfil de tráfego não se altera significativamente ao longo dos dias de semana, tendo que a plataforma responder a médias de tráfego inferiores à 10 MBits.

Em termos de conexões a figura 5.7 apresenta os dados obtidos na *Firewall*. Este dado é importante já que demonstra a capacidade da plataforma de lidar com múltiplas conexões de *VPN*, com recurso a cifra *AES*, sem perda de desempenho.

Diagnostics: Show States

States Reset states

Current state count: 710 Filter expression: Filter

Proto	Source -> Router -> Destination	State
udp	172.16.0.2:11940 <- 195.23.61.186:11940 <- 88.2.224.227:54387	MULTIPLE: MULTIPLE
udp	88.2.224.227:54387 -> 172.16.0.2:11940	MULTIPLE: MULTIPLE
udp	172.16.0.2:11948 <- 195.23.61.186:11948 <- 88.24.243.68:46762	MULTIPLE: MULTIPLE
udp	88.24.243.68:46762 -> 172.16.0.2:11948	MULTIPLE: MULTIPLE
udp	172.16.0.2:11940 <- 195.23.61.186:11940 <- 212.170.203.121:53702	MULTIPLE: MULTIPLE
udp	212.170.203.121:53702 -> 172.16.0.2:11940	MULTIPLE: MULTIPLE
udp	172.16.0.2:11948 <- 195.23.61.186:11948 <- 88.26.181.43:10000	MULTIPLE: MULTIPLE
udp	88.26.181.43:10000 -> 172.16.0.2:11948	MULTIPLE: MULTIPLE
udp	172.16.0.2:11942 <- 195.23.61.186:11942 <- 194.38.156.27:50772	MULTIPLE: MULTIPLE
udp	194.38.156.27:50772 -> 172.16.0.2:11942	MULTIPLE: MULTIPLE
udp	172.16.0.2:11940 <- 195.23.61.186:11940 <- 87.196.125.64:50772	MULTIPLE: MULTIPLE
udp	87.196.125.64:50772 -> 172.16.0.2:11940	MULTIPLE: MULTIPLE
udp	172.16.0.2:11940 <- 195.23.61.186:11940 <- 89.180.58.24:34090	MULTIPLE: MULTIPLE
udp	89.180.58.24:34090 -> 172.16.0.2:11940	MULTIPLE: MULTIPLE
udp	172.16.0.2:11942 <- 195.23.61.186:11942 <- 194.38.156.31:44698	MULTIPLE: MULTIPLE
udp	194.38.156.31:44698 -> 172.16.0.2:11942	MULTIPLE: MULTIPLE
udp	172.16.0.2:11940 <- 195.23.61.186:11940 <- 80.37.77.118:49083	MULTIPLE: MULTIPLE

Figura 5.7: Relatório do número de conexões activas do servidor de *VPN*

5.10 Discussão de Resultados

Antes de abordar a Discussão de Resultados, propriamente dita, o autor considera importante referir que ambas as plataformas se encontram em produção e que os dados obtidos, com os testes apresentados previamente, reflectem a realidade das duas plataformas. Uma última nota sobre os testes, estes são representativos das principais funcionalidades das plataformas e/ou representam requisitos exigidos aquando da formulação da caracterização do sistema.

Relativamente à plataforma de Fax, ao analisarmos os dados das tabelas 5.1 e 5.2 obtemos a seguinte distribuição por canal:

Tabela 5.4: Distribuição de Fax/Aplicação por Mês

Mês / Faxes	<i>jhEDI</i>	<i>jhEDI</i> %	<i>jhAS400</i>	<i>jhAS400</i> %	Email	Email %
Agosto	5560	90	468	8	132	2
Setembro	5240	92	356	6	102	2
Outubro	5210	93	265	5	98	3
Novembro	4970	92	349	6	92	3
Dezembro	4639	94	212	4	69	2
Total	25619	92	1650	6	493	2

Como se pode observar nas tabelas 5.4 e 5.3, num período de quatro meses, foram enviados quase 28 mil faxes e foram recebidos mais de 550. A dimensão da amostra permite conferir aos resultados obtidos um elevado grau de confiança.

Se atendermos aos dados constantes na tabela 5.1 verifica-se que a taxa de sucesso menor ronda os 96 % e a maior 98,5 %, sendo que a total é de 97 %. As falhas, apesar de representarem apenas 3%, podem ser explicadas devido a avarias nos equipamentos recetores, más condições de transmissão e por algum problema na gestão do próprio servidor da velocidade de envio do Fax.

Focando apenas os dados da tabela 5.2 podemos constatar que o serviço de Fax via *ERP* e Fax via *Email* apresentam uma adesão interessante por parte dos utilizadores, reunindo um pouco mais de 2 mil faxes durante o período de testes.

Relativamente à plataforma de redes virtuais privadas, analisando os testes de conectividade e de *throughput* que foram realizados comprova-se que um cliente e/ou um escritório remoto podem aceder à rede da sede com valores de latência baixos e com valores de largura de banda próximos daqueles que seriam o máximo, em termos de largura banda, do seu acesso à *Internet*.

Optou-se por focar em parâmetros como a latência e *throughput* na realização dos testes pelas seguintes razões:

- a latência (e a sua variação) influi directamente com a qualidade das comunicações *VoIP*
- elevadas latências (> 150 ms) conferem às comunicações *VoIP* uma baixa qualidade além de proporcionarem uma experiência de acesso aos sistemas centrais (plataformas *Web*) de fraca qualidade
- uma baixa capacidade de *throughput* prejudica a sensação que o utilizador tem de velocidade ao aceder aos sistemas centrais

Pode-se ainda afirmar que o servidor de *VPN* é capaz de lidar com cerca de duas centenas de conexões simultâneas sem que exista degradação de desempenho, uma vez que os testes apresentados neste capítulo foram efectuados em condições normais de funcionamento da plataforma.

Capítulo 6

Conclusões

Este último capítulo apresenta uma síntese do trabalho apresentado no documento, referindo os resultados obtidos e as conclusões alcançadas. São também apresentadas as perspectivas de desenvolvimento futuro.

6.1 Síntese do Trabalho Desenvolvido

O trabalho desenvolvido levou à criação duas plataformas, baseadas em tecnologias *open-source*, de forma a disponibilizar uma solução de carácter empresarial para os serviços de Fax e redes virtuais privadas, *VPN*.

No que respeita à plataforma de Fax o trabalho dividiu-se da seguinte forma:

- A fase inicial do trabalho centrou-se no estudo da diferentes tecnologias existentes no mercado que se perfilassem como opções válidas para o desenvolvimento das plataformas, incidindo sobre as suas potencialidades, limitações e possíveis sinergias entre elas.
- A segunda fase visou a criação da plataforma de Fax. Foi desenvolvida uma interface funcional a partir de aplicações e linguagens de programação livres, na qual é possível configurar todas as componentes envolvidas sem que sejam precisos conhecimentos especiais do modo de funcionamento e detalhes das diversas aplicações envolvidas.
- A fase seguinte visou a integração do servidor de Fax com os restantes serviços existentes na empresa. Esta fase exigiu o desenvolvimento de aplicações específicas de modo a responder às necessidades do ambiente empresarial.

No que respeita à plataforma de redes virtuais privadas o trabalho dividiu-se da seguinte forma:

- A fase inicial do trabalho centrou-se no estudo da diferentes tecnologias existentes no mercado que se perfilassem como opções válidas para o desenvolvimento das plataformas, incidindo sobre as suas potencialidades, limitações e possíveis sinergias entre elas.

- A segunda fase visou a criação de uma plataforma de testes integrando as soluções adotadas
- A terceira fase passou pela realização dos testes apresentados no Capítulo 5 e por pequenas alterações de afinação de operação.

Por último concretizou-se a validação das duas plataformas apresentadas através dos testes efetuados e cujos resultados são apresentados no capítulo 5, podendo-se afirmar que os objectivos principais desta dissertação foram atingidos.

6.2 Desenvolvimento Futuro

Em termos da plataforma de Fax, apesar de a solução apresentada cumprir com os requisitos propostos, ela pode ser enriquecida pela inclusão das seguintes funcionalidades:

- Permitir o envio por correio electrónico de documentos *DOC* (*MS Word file extension*), *XLS* (*MS Excel file extension*), *ODT* (*OpenDocument Word Processing file format*), *ODS* (*OpenDocument Spreadsheet file format*);
- Notificação de receção de fax por SMS (Short Message Service);
- Permitir o uso de servidores *T.38*
- Adicionar funcionalidade de “backup” e restauro de toda a configuração da plataforma

Em termos da plataforma de redes virtuais privadas, esta pode ser enriquecida pela inclusão das seguintes funcionalidades:

- Interligação a um sistema de gestão de identidades centralizada, neste caso, *LDAP*
- Criação de uma ferramenta que a partir do certificado e da gama de endereços atribuída, gere o ficheiro de configuração para a plataforma *PfSense*
- Criação de mecanismos de controlo de banda por cada ligação

Anexo A

Caracterização do sistema

A.1 Implementação

A.1.1 Asterisk

Optou-se por utilizar a versão mais recente deste software o seu processo de instalação é descrito em seguida.

O primeiro conjunto de comandos visa resolver as dependências necessárias para a compilação do software.

```
sudo apt-get install build-essential libtool libncurses5-dev libssl-dev \
libssl-dev libxml2-dev libmysqlclient-dev
```

Este segundo conjunto de comandos descarrega a última versão disponível do *Asterisk* e executa a rotina de configuração do mesmo.

```
sudo su
cd /usr/src/
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8.8.0-rc4.tar.gz
tar -xvzf asterisk-1.8.8.0-rc4.tar.gz
cd asterisk-1.8.8.0-rc4
./configure
make menuselect
```

De seguida são apresentadas um conjunto de imagens do menu de configuração do *Asterisk* e as opções que seleccionamos:

```
*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[ ] app_mysql
[ ] app_saycountpl
[] cdr_mysql
XXX chan_mobile
[ ] chan_ooh323
[ ] format_mp3
[ ] res_config_mysql

MySQL CDR Backend
Depends on: mysqlclient(E)

Support Level: deprecated, Replaced by: cdr_adaptive_odbc
```

Figura A.1: *Asterisk Menuselect 1*

```
*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[*] chan_agent
XXX chan_alsa
[*] chan_bridge
XXX chan_console
XXX chan_dahdi
XXX chan_gtalk
XXX chan_h323
[*] chan_iax2
XXX chan_jingle
[*] chan_local
[ ] chan_mgcp
XXX chan_misdn
[ ] chan_multicast_rtp
XXX chan_nbs
[ ] chan_oss
[*] chan_phone
[*] chan_sip
[ ] chan_skinny
[] chan_unistim
XXX chan_usbradio
XXX chan_vpb

UNISTIM Protocol (USTM)

Support Level: extended
```

Figura A.2: *Asterisk Menuselect 2*

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[*] codec_a_mu
[*] codec_adpcm
[*] codec_alaw
XXX codec_dahdi
[ ] codec_g722
[ ] codec_g726
[ ] codec_gsm
[ ] codec_ilbc
[ ] codec_lpc10
XXX codec_resample
XXX codec_speex
[*] codec_ulaw

ITU G.726-32kbps G726 Transcoder

Support Level: core

```

Figura A.3: Asterisk Menuselect 3

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[ ] format_g719
[ ] format_g723
[ ] format_g726
[ ] format_g729
[ ] format_gsm
[ ] format_h263
[ ] format_h264
[ ] format_ilbc
[ ] format_jpeg
XXX format_ogg_vorbis
[*] format_pcm
[ ] format_siren14
[ ] format_siren7
[*] format_sln
[ ] format_sln16
[*] format_vox
[ ] format_wav
[ ] format_wav_gsm

Raw Signed Linear 16KHz Audio support (SLN16)

Support Level: core

```

Figura A.4: Asterisk Menuselect 4

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[ ] DONT_OPTIMIZE
[ ] DEBUG_THREADS
[ ] STATIC_BUILD
[*] LOADABLE_MODULES
[ ] DEBUG_FD_LEAKS
[ ] REBUILD_PARSERS
[ ] LOW_MEMORY
XXX BETTER_BACKTRACES
XXX USE_HOARD_ALLOCATOR
[ ] LOTS_OF_SPANS
[ ] RADIO_RELAX
[*] G711_NEW_ALGORITHM
[*] G711_REDUCED_BRANCHING
[*] TEST_CODING_TABLES
[*] TEST_TANDEM_TRANSCODING
[ ] MALLOC_DEBUG
[ ] BUSYDETECT_TONEONLY
[ ] BUSYDETECT_COMPARE_TONE_AND_SILENCE
[ ] BUSYDETECT_DEBUG
[ ] IAX_OLD_FIND
[ ] INTEGER_CALLERID
... More ...

Disable Optimizations by the Compiler

Support Level: core

```

Figura A.5: Asterisk Menuselect 5

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[ ] FILE_STORAGE
XXX ODBC_STORAGE
XXX IMAP_STORAGE

Storage of Voicemail using filesystem

Conflicts with: ODBC_STORAGE(M), IMAP_STORAGE(M)
Support Level: core

```

Figura A.6: Asterisk Menuselect 6

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[ ] CORE-SOUNDS-EN-WAV
[*] CORE-SOUNDS-EN-ULAW
[M] CORE-SOUNDS-EN-ALAW
[ ] CORE-SOUNDS-EN-GSM
[ ] CORE-SOUNDS-EN-G729
[ ] CORE-SOUNDS-EN-G722
[ ] CORE-SOUNDS-EN-SLN16
[ ] CORE-SOUNDS-EN-SIREN7
[ ] CORE-SOUNDS-EN-SIREN14
[ ] CORE-SOUNDS-ES-WAV
[ ] CORE-SOUNDS-ES-ULAW
[ ] CORE-SOUNDS-ES-ALAW
[ ] CORE-SOUNDS-ES-GSM
[ ] CORE-SOUNDS-ES-G729
[ ] CORE-SOUNDS-ES-G722
[ ] CORE-SOUNDS-ES-SLN16
[ ] CORE-SOUNDS-ES-SIREN7
[ ] CORE-SOUNDS-ES-SIREN14
[ ] CORE-SOUNDS-FR-WAV
[ ] CORE-SOUNDS-FR-ULAW
[ ] CORE-SOUNDS-FR-ALAW
... More ...

English, a-Law format

Support Level: core

```

Figura A.7: Asterisk Menuselect 7

```

*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

[] MOH-OPSOUND-WAV
[ ] MOH-OPSOUND-ULAW
[ ] MOH-OPSOUND-ALAW
[ ] MOH-OPSOUND-GSM
[ ] MOH-OPSOUND-G729
[ ] MOH-OPSOUND-G722
[ ] MOH-OPSOUND-SLN16
[ ] MOH-OPSOUND-SIREN7
[ ] MOH-OPSOUND-SIREN14

opsound.org Music On Hold Files, WAV format

Support Level: core

```

Figura A.8: Asterisk Menuselect 8

Após seleccionar-se as opções que se necessita para esta versão de *Asterisk* deve-se salvar a configuração e de seguida iniciar o processo de compilação e posterior instalação recorrendo a este conjunto de comandos:

```
make
make install
make samples
```

Para garantir que o *Asterisk* é correctamente iniciado com o arranque do servidor e que o mesmo passa a ser um serviço tem-se que executar os seguintes passos:

Copiar o ficheiro que irá inicializar e para o serviço:

```
cd /usr/src/asterisk-1.8.8.0-rc4/contrib/init.d/
cp rc.debian.asterisk /etc/init.d/asterisk
```

Editar esse mesmo ficheiro de forma a apresentar estes 3 parâmetros:

```
DAEMON=/usr/sbin/asterisk
ASTVARRUNDIR=/var/run/asterisk
ASTETCDIR=/etc/asterisk
```

Por fim adicionar este serviço ao arranque do servidor:

```
update-rc.d asterisk defaults
```

Para iniciar o serviço basta executar o seguinte comando:

```
/etc/init.d/asterisk start
```

Para finalizar o processo de instalação basta confirmar se a consola do *Asterisk* se encontra operacional:

```
root@fax01:~# asterisk -vr
Asterisk 1.8.8.0-rc4, Copyright (C) 1999 - 2011 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.8.0-rc4 currently running on fax01 (pid = 441)
Verbosity is at least 7
fax01*CLI>
```


Para que o *Asterisk* leia as configurações da Base de Dados e não da estrutura habitual de ficheiros temos que adicionar ao ficheiro */etc/asterisk/res_config_mysql.conf* o seguinte conteúdo:

```
[ asterisk ]
dbhost = 127.0.0.1
dbname = NOME_DA_BASE_DE_DADOS
dbuser = UTILIZADOR
dbpass = PASSWORD_DO_UTILIZADOR
dbport = 3306
dbsock = /var/run/mysql/mysql.sock
requirements=warn ;
```

e ao ficheiro */etc/asterisk/extconfig.conf* as seguintes linhas:

```
[ settings ]
sipusers => mysql,asterisk,users ; SIP user
sippeers => mysql,asterisk,users ; SIP peers
extensions => mysql,asterisk,extensions ; SIP extensions
voicemail => mysql,asterisk,voicemails ; SIP voicemailboxes
queues => mysql,asterisk,queues ; SIP queue
queue_members => mysql,asterisk,queue_members ; SIP queue members
sip.conf => mysql,asterisk,ast_config
```

A.1.2 Hylafax

Optou-se por utilizar a versão mais recente deste software o seu processo de instalação é descrito em seguida.

O primeiro conjunto de comandos visa resolver as dependências necessárias para a compilação do software:

```
sudo apt-get install gawk ghostscript gsfonts libcups2 libcupsimage2 \
libgs8 libnetpbm10 libpaper-utils libpaper1 libtiff-tools libtiff4 \
netpbm psfontmgr libtiff-dev
```

Este segundo conjunto de comandos descarrega a última versão disponível do *HylaFAX* e executa a rotina de configuração do mesmo:

```
sudo su
cd /usr/src
wget http://sunet.dl.sourceforge.net/project/hylafax/hylafax/hylafax-5.5.0.tar.gz
tar -xvzf hylafax-5.5.0.tar.gz
cd hylafax-5.5.0
./configure
```

HylaFAX configuration parameters (part 1 of 2) are:

```
[ 1] Directory for applications:      /usr/local/bin
[ 2] Directory for lib data files:    /usr/local/lib/fax
[ 3] Directory for lib executables:   /usr/local/sbin
[ 4] Directory for system apps:      /usr/local/sbin
[ 5] Directory for manual pages:     /usr/local/man
[ 6] Directory for spooling:         /var/spool/hylafax
[ 7] Directory for uucp lock files:   /var/lock
[ 8] Uucp lock file scheme:         ascii
[ 9] PostScript imager package:      gs
[10] PostScript imager program:     /usr/bin/gs
```

```

[11] Manual page installation scheme:   bsd-source-cat
[12] Default page size:                 North American Letter
[13] Default vertical res (lpi):        98

Are these ok [yes]? no
Directory to install applications [/usr/local/bin]?
Directory to install library data files [/usr/local/lib/fax]?
Directory to install library executables [/usr/local/sbin]?
Directory to install system apps [/usr/local/sbin]?
Directory to install manual pages [/usr/local/man]?
Directory to setup server spooling area [/var/spool/hylafax]?
Directory for uucp lock files [/var/lock]?
UUCP lock file scheme [ascii]?
PostScript imager package [gs]?
PostScript imager program [/usr/bin/gs]?
Manual page installation scheme [bsd-source-cat]?
Default page size [North American Letter]? A4
Default vertical res (lpi) [98]?
Location of getty program [/sbin/getty]?

HylaFAX configuration parameters (part 1 of 2) are:

[ 1] Directory for applications:         /usr/local/bin
[ 2] Directory for lib data files:       /usr/local/lib/fax
[ 3] Directory for lib executables:      /usr/local/sbin
[ 4] Directory for system apps:         /usr/local/sbin
[ 5] Directory for manual pages:        /usr/local/man
[ 6] Directory for spooling:            /var/spool/hylafax
[ 7] Directory for uucp lock files:      /var/lock
[ 8] Uucp lock file scheme:             ascii
[ 9] PostScript imager package:         gs
[10] PostScript imager program:         /usr/bin/gs
[11] Manual page installation scheme:   bsd-source-cat
[12] Default page size:                 ISO A4
[13] Default vertical res (lpi):        98

Are these ok [yes]?

make
make install

```

Para configurar o *Hylafax* basta executar o comando *faxsetup* num terminal e responder as questões tal como apresentado de seguida:

```

faxsetup

Update /var/spool/hylafax/status/any.info.

HylaFAX configuration parameters are:

[1] Init script starts faxq:           yes
[2] Init script starts hfaxd           yes
[3] Start paging protocol:             no

Are these ok [yes]?

```

```

Modem support functions written to /var/spool/hylafax/etc/setup.modem.
Configuration parameters written to /var/spool/hylafax/etc/setup.cache.

No scheduler config file exists , creating one from scratch.
Country code [1]? 351
Area code [000]?
Long distance dialing prefix [1]?
International dialing prefix [011]?
Dial string rules file (relative to /var/spool/hylafax) ["etc/dialrules"]?
Tracing during normal server operation [1]?
Default tracing during send and receive sessions [0xFFFF]?
Continuation cover page (relative to /var/spool/hylafax) []?
Timeout when converting PostScript documents (secs) [180]?
Maximum number of concurrent jobs to a destination [1]?
Define a group of modems []?
Time of day restrictions for outbound jobs ["Any"]?
Timeout before purging a stale UUCP lock file (secs) [30]?
Max number of pages to permit in an outbound job [0xffffffff]?
Syslog facility name for ServerTracing messages [daemon]?

The non-default scheduler parameters are:

CountryCode:          351

Are these ok [yes]?

Creating new configuration file /var/spool/hylafax/etc/config...

Restarting HylaFAX server processes.
Should I restart the HylaFAX server processes [yes]?

/etc/init.d/hylafax start
HylaFAX: faxq hfaxd (without SNPP support).

You do not appear to have any modems configured for use.  Modems are
configured for use with HylaFAX with the faxaddmodem(8C) command.
Do you want to run faxaddmodem to configure a modem [yes]? no

Done verifying system setup.

```

Deve-se ainda criar o seguinte ficheiro *hylafax* em */etc/default*:

```

# default settings for hylafax server

# try to check for the correct USE_FAXGETTY value
if grep -E '^[^#]*:respawn:/usr/sbin/fax(getty|modem).*$' /etc/inittab >/dev/null 2>&1
  || grep -E '^exec /usr/sbin/fax(getty|modem)' /etc/event.d/* >/dev/null 2>&1
then
    USE_FAXGETTY=init
else
    USE_FAXGETTY=yes
fi

# Arguments for faxgetty command. At least -D should be specified
# as explained in debian bug #462459.
FAXGETTYARGS="-D"

# In a send only installation you may want to use faxgetty

```

```

# or faxmodem on lines already configured with faxaddmodem.
# faxgetty is the new default, faxmodem the old one.
# For a better understanding of the difference between faxmodem
# and faxgetty, see http://www.hylafax.org/archive/1999-09/msg00043.php
#
# In a send/receive installation you must use faxgetty.
#
# To use faxmodem change the following variable value to "no".
#
# You may also prefer to run faxmodem/faxgetty via inittab instead of
# running it here from a script. In this case set its value as
# "init".
USE_FAXGETTY=yes

#
# Uncomment this line once hylafax has been fully configured and/or
# you want to enable the server.
#
RUN_HYLAFAX=1

#
# If you need to bind hylafax to one address only, just uncomment
# this variable and change the value using your preferred IP.
#
# BINDTO="-l 127.0.0.1"

#
# The following variables are automatically assigned by running
# the faxsetup utility. If you leave them commented out then
# faxsetup values are used. You may specify here the values to
# use.
#
# OLDPROT="-o 4557"
# NEWPROT="-i 4559"
# SNPP="-s 444"

#
# hfaxd may be run from inetd while faxq should still be executed
# by the init script. If you run hfax via inetd or xinetd then
# uncomment the following line
# HFXD_FROM_INET=true

```

Para configurar *modem's* deve-se criar em */var/spool/hylafax/etc/* um ficheiro, por exemplo, com o seguinte nome *config.ttyIAX0* e o seu conteúdo deverá ser deste tipo:

```

# ----- Start config.ttyIAX Example -----
CountryCode:          351
AreaCode:
FAXNumber:            351226059601 # replace this with a real phone
number
LongDistancePrefix:
InternationalPrefix:
DialStringRules:     etc/dialrules
ServerTracing:       0xFFFF
SessionTracing:      0xFFFF
RecvFileMode:        0600
LogFileMode:         0600

```

```

DeviceMode:          0600
RingsBeforeAnswer:   1
SpeakerVolume:       off
GettyArgs:           "-h %l dx_%s"
LocalIdentifier:     ""
TagLineFont:         etc/lutRS18.pcf
TagLineFormat:       "De %%l%%c|Pagina %%P de %%T"
MaxRecvPages:       600
#
#
# Modem-related stuff: should reflect modem command interface
# and hardware connection/cabling (e.g. flow control).
#
ModemType:           Class1          # use this to supply a hint

#
# The modem is taken off-hook during initialization, and then
# placed back on-hook when done to prevent glare.
#
ModemResetCmds:      "ATH1\nAT+VCID=1"      # enables CallID display
ModemReadyCmds:      ATH0

Class1AdaptRecvCmd:  AT+FAR=1
Class1TMConnectDelay: 400              # counteract quick CONNECT response
Class1RMQueryCmd:    "!24,48,72,96"      # V.17 fast-train recv doesn't work
well

CallIDPattern:       "NMBR="
CallIDPattern:       "NAME="
CallIDPattern:       "ANID="
CallIDPattern:       "NDID="
# Uncomment these if you really want them, but you probably don't.
#CallIDPattern:      "DATE="
#CallIDPattern:      "TIME="
#20060302 JDG - Fix for broken libtiff JBIG support in tiff2pdf
Class1JBIGSupport:   no

```

O conteúdo do *script mail2fax.sh* é o seguinte:

```

#!/bin/sh
#
# simple mail-to-fax utility using both faxmail and sendfax.
#
# - Lee Howard
#
RANDOMFAX=/tmp/mail2fax.$$
mkdir $RANDOMFAX
cat >> $RANDOMFAX/_message_
JOBID=$(grep -e "subject:" -i $RANDOMFAX/_message_ | sed q | sed 's/^[^:]*: */g' )
TOLINE=$(grep -e "to:" -i $RANDOMFAX/_message_ | sed q)
FROMLINE=$(grep -e "from:" -i $RANDOMFAX/_message_ | sed q)
if [ "$TOLINE" | grep '<.*>' != "" ]; then
    TONUMBER=$(echo $TOLINE | sed -e 's/.*<([^\@]*)@.*>.*//1/' )
else
    TONUMBER=$(echo $TOLINE | sed -e 's/^[Tt]o://g' -e 's/[ ]*(.*)@.*//1/' )
fi
if [ "$FROMLINE" | grep '<.*>' != "" ]; then

```

```

FROMPATH='echo $FROMLINE | sed -e 's/.*<(\.*\).*>.*\1/'
else
FROMPATH='echo $FROMLINE | sed -e 's/^[Ff]rom://g' -e 's/[ ]*\([^\ ]*\).*\1/'
fi

cat $RANDOMFAX/_message_ | faxmail -v -T $FROMPATH | sendfax -vv -n -R -f "$FROMPATH" -i
"$JOBID" -d $TONUMBER

rm -rf $RANDOMFAX
exit 0

```

Para que exista um registo permanente de todos os movimentos que o servidor de FAX gera deve-se criar o ficheiro `/var/spool/hylafax/etc/FaxAccounting` e atribuir permissões de execução:

```

#!/bin/sh

$USERNAME = ""
$PASS = ""
$DB = ""

DATETIME="date +%Y-%m-%d\ %H:%M:%S"; shift
ENTRYTYPE="$1"; shift
COMMID="$1"; shift
MODEM="$1"; shift
JOBID="$1"; shift
JOBTAG="$1"; shift
USERID="$1"; shift
LOCALNUMBER="$1"; shift
TSI="$1"; shift
PARAMS="$1"; shift
NPAGES="$1"; shift
JOBTIME="$1"; shift
CONNTIME="$1"; shift
REASON="$1"; shift
CIDNAME="$1"; shift
CIDNUMBER="$1"; shift
CALLID="$1"; shift
OWNER="$1"; shift
DCS="$1"; shift
JOBINFO="$1"; shift

echo "INSERT INTO foip SET timestamp = \"$DATETIME\", entrytype = \"$ENTRYTYPE\", commid = \"$COMMID\", modem = \"$MODEM\", jobid = \"$JOBID\", jobtag = \"$JOBTAG\", user = \"$USERID\", localnumber = \"$LOCALNUMBER\", tsi = \"$TSI\", params = \"$PARAMS\", npages = \"$NPAGES\", jobtime = \"$JOBTIME\", conntime = \"$CONNTIME\", reason = \"$REASON\", CIDName = \"$CIDNAME\", CIDNumber = \"$CIDNUMBER\", callid = \"$CALLID\", owner = \"$OWNER\", dcs = \"$DCS\", jobinfo = \"$JOBINFO\"" | mysql -u $USERNAME -p$PASS $DB

exit 0

```

A.1.3 IAXModem

Para instalar o *IAXModem* basta executar o seguinte comando num terminal.

```
apt-get install iaxmodem
```

Para criar *modems* é necessário criar um ficheiro por modem em */etc/iaxmodem/*, o seguinte ficheiro ilustra um desses casos

```
device      /dev/ttyIAX0
owner       uucp:uucp
mode        660
port        4570
refresh     60
server      127.0.0.1
peername    100
secret      foip
cidname     UQ CONSULT
cidnumber   351220005110
codec       slinear
```

A.1.4 Postfix

Para instalar o *Postfix* basta executar o seguinte comando num terminal.

```
apt-get install postfix
```

A.1.5 Nginx

Para instalar o *Nginx* basta executar o seguinte comando num terminal.

```
apt-get install nginx php5-cli php5-cgi spawn-fcgi psmisc
```

De seguida é preciso criar um ficheiro na directoria */etc/nginx/sites-available/* como o nome *webfax* com o seguinte conteúdo:

```
server {
    server_name localhost;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
    root /var/www;

    location / {
        index index.html index.htm;
    }

    location ~ \.php$ {
        include /etc/nginx/fastcgi_params;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME /var/www/$fastcgi_script_name;
    }
}
```

bem como ficheiro */usr/bin/php-fastcgi* que deverá apresentar este conteúdo

```
#!/bin/bash

FASTCGI_USER=www-data
FASTCGI_GROUP=www-data
ADDRESS=127.0.0.1
PORT=9000
PIDFILE=/var/run/php-fastcgi/php-fastcgi.pid
CHILDREN=6
PHP5=/usr/bin/php5-cgi

/usr/bin/spawn-fcgi -a $ADDRESS -p $PORT -P $PIDFILE -C $CHILDREN -u $FASTCGI_USER -g
$FASTCGI_GROUP -f $PHP5
```

O passo final é criar um ficheiro *php-fastcgi* em */etc/init.d/* com este conteúdo:

```
#!/bin/bash

PHP_SCRIPT=/usr/bin/php-fastcgi
FASTCGI_USER=www-data
FASTCGI_GROUP=www-data
PID_DIR=/var/run/php-fastcgi
PID_FILE=/var/run/php-fastcgi/php-fastcgi.pid
RET_VAL=0

case "$1" in
  start)
    if [[ ! -d $PID_DIR ]]
    then
      mkdir $PID_DIR
      chown $FASTCGI_USER:$FASTCGI_GROUP $PID_DIR
      chmod 0770 $PID_DIR
    fi
    if [[ -r $PID_FILE ]]
    then
      echo "php-fastcgi already running with PID `cat $PID_FILE`"
      RET_VAL=1
    else
      $PHP_SCRIPT
      RET_VAL=$?
    fi
    ;;
  stop)
    if [[ -r $PID_FILE ]]
    then
      kill `cat $PID_FILE`
      rm $PID_FILE
      RET_VAL=$?
    else
      echo "Could not find PID file $PID_FILE"
      RET_VAL=1
    fi
    ;;
  restart)
    if [[ -r $PID_FILE ]]
    then
      kill `cat $PID_FILE`
      rm $PID_FILE
      RET_VAL=$?
    fi
  *)
    echo "Usage: $0 {start|stop|restart}"
    RET_VAL=1
  esac
```



```
else
    echo "Could not find PID file $PID_FILE"
fi
$PHP_SCRIPT
RET_VAL=$?
;;
status )
    if [[ -r $PID_FILE ]]
    then
        echo "php-fastcgi running with PID 'cat $PID_FILE'"
        RET_VAL=$?
    else
        echo "Could not find PID file $PID_FILE, php-fastcgi does not appear to be
            running"
    fi
;;
*)
    echo "Usage: php-fastcgi { start|stop|restart|status }"
    RET_VAL=1
;;
esac
exit $RET_VAL
```

Para assegurar o arranque do processo com o iniciar do servidor temos que executar os seguintes comandos:

```
chmod +x /etc/init.d/php-fastcgi
update-rc.d php-fastcgi defaults
/etc/init.d/php-fastcgi start
/etc/init.d/nginx start
```

A.1.6 *jhAS400*

O seguinte bloco ilustra o *stream* em *EBCDIC* gerado pelo *ERP, AS400*:

```
.....  
.....+K  
.....  
.....  
.....  
.....  
.....+ J . a ..... + H .   .4D..4 { .IBERUSA CENTRAL DE COMPRAS PARA RESTAURACAO ACE...4 { .PRA  
.A BOM SUCESSO, N..105 A 159- 9...4 { .4150-146 PORTO.....4 { .ENCOMENDA N  
..4 { ^01.4 { / - .4 { .033485...4 { .TELEFONE: 00351 22 6089700.4 { .DATA.4 { ^2012/01/17...4 { .  
FAX       : 00351 22 6064009.....4 { .DATA INICIO ACTIVIDADE:01/01/04...4 { .NR.  
CONTRIBUINTE.4 { .506808904...4 { .FORNECEDOR  CONTA.4 { .2211101.4 { .PRIMOR CHARCUTARIA -  
PRIMA SA...4 { .NUMERO.4 { .005673.4 { .AV.SANTIAGO DE GAVIAO 1142 (AP.37) ...4 { .NR.  
CONTRIBUINTE.4 { .501276211.4 { .GAVIAO...4 { .VIA EXPEDICAO.4 { .001.4 { .V/CARRO  
.4 { .4784-960.4 { +V.N.FAMALICAO...4 { .FORMA DE PAGAMENTO...4 { .COND. PAGAMENTO  
.4 { .60.4 { .45 DIAS RESUMO MENSAL...4 { .NUMERO DE FAX.4 { .226059601...4 { +** LOCAL DE  
ENTREGA:...4 { .DATA DE ENTREGA:25 DE JANEIRO.4 { +GPT/PLATAFORMA CENTRAL....  
.  
.....  
.....  
.....4 { + VILA AM.LIA - CABANAS (PALMELA)...4 { +2950-805 QUINTA DO ANJO...4 { +PORTUGAL  
(+351) 212135747...4 { .QUEIRAM FORNECER-NOS OS SEGUINTES ARTIGOS, A ENTREGAR EM  
:4 { 2012/01/25...4 { +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
  C O D I G O | D E S C R I C A | A O          | QT.PEDIDA |UMI.4 { PR.UNITARIO |  
UPI VALOR LIQ.4 { !.EURO PALLETS|      PESO  
|.4 { +-----+-----+-----+-----+-----+-----+-----+-----+-----+  
$1.4 { .1.4 { .1.4 { .1.4 { .125497.4 { .1.4 { .15601240005231.4 { .1.4 { .      54,000.4 { <.4 { =CX  
.4 { ?1.4 {      40,20.4 { <.4 { (CX.4 { |1.4 { &      683,99.4 { $1.4 { *      1,200.4 { .1.4 { |  
      269,352.4 { .1...4 { .1          |.4 { .CH..  
.  
.  
.....  
.....  
.....=====...4 { .1.4 { <=.4 { +  
      1,000000.4 { $EUR...4 { .CUMPRIMENTOS...4 { .....4 { .  
LEGENDA:.4 { .LEGENDA:.4 { ( Victor Braga - UQ - "Ibersol"...4 { .CX.4 { .CAIXA.....  
.  
.....  
.....  
.....  
.....
```

O seguinte *log* apresenta as mensagens trocadas entre a aplicação *jhAS400* e o servidor de *Hylafax*:

```
130 Warning, no inverse address mapping for client host name "catsupervisor.uq.pt".  
220 webfax server (HylaFAX (tm) Version 5.5.0) ready.  
user ....  
331 Password required for foip.  
pass ....  
230 User foip logged in.  
noop  
200 NOOP command successful.  
tzon LOCAL
```

```
200 Using time values in GMT.
port 192,168,86,222,179,120
200 PORT command successful.
stot
150 FILE: /tmp/doc241301.ps (Opening new data connection).
226 Transfer complete (FILE: /tmp/doc241301.ps).
jnew
200 New job created: jobid: 241264 groupid: 241264.
job
200 Current job: jobid: 241264 groupid: 241264.
jparm FROMUSER vbraga@uq.pt
213 FROMUSER set to "vbraga@uq.pt".
jparm NOTIFYADDR vbraga@uq.pt
213 NOTIFYADDR set to "vbraga@uq.pt".
jparm LASTTIME 000300
213 LASTTIME set to 000300.
jparm MAXDIALS 12
213 MAXDIALS set to 12.
jparm MAXTRIES 3
213 MAXTRIES set to 3.
jparm SCHEDPRI 85
213 SCHEDPRI set to 85.
jparm DIALSTRING 226059601
213 DIALSTRING set to "226059601".
jparm VRES 196
213 VRES set to 196.
jparm NOTIFY done
213 NOTIFY set to "done".
jparm CHOPTHRESHOLD 3
213 CHOPTHRESHOLD set to 3.
jparm DOCUMENT /tmp/doc241301.ps
200 Added document /tmp/doc241301.ps as docq/doc241301.ps.241264.
jsubm 241264
200 Job 241264 submitted.
quit
221 Goodbye.
```

A.1.7 OpenVPN

Para instalar a aplicação *OpenVPN* basta executar o seguinte comando num terminal:

```
apt-get install openvpn openssl lzop
```

A aplicação *Webmin* pode ser instalada seguindo os seguintes passos:

```
cd /usr/src
wget http://downloads.sourceforge.net/project/webadmin/webmin/1.580/webmin_1.580_all.deb
dpkg -i webmin_1.580_all.deb
apt-get install -f
```

O seguinte bloco apresenta o conteúdo do */etc/openvpn/vpn-testes.conf* (ficheiro de configuração de uma instância do servidor de VPN):

```
port 1194
proto udp
dev tun
ca keys/vpn-testes/ca.crt
cert keys/vpn-testes/server_uq.crt
key keys/vpn-testes/server_uq.key
dh keys/vpn-testes/dh2048.pem
server 10.0.1.0 255.255.255.0
cr1-verify keys/vpn-testes/cr1.pem
cipher DES-CBC
user nobody
group nogroup
status servers/vpn-testes/logs/openvpn-status.log
log-append servers/vpn-testes/logs/openvpn.log
verb 2
mute 10
management 127.0.0.1 6666
keepalive 10 120
client-config-dir /etc/openvpn/servers/vpn-testes/ccd
comp-lzo
persist-key
persist-tun
ccd-exclusive
push "route 172.16.10.100 255.255.255.255"
```

A.1.8 *Heartbeat*

Para instalar o *Heartbeat* basta executar o seguinte comando num terminal.

```
apt-get install heartbeat
```

Deve-se editar o ficheiro */etc/heartbeat/ha.cf* para que apresente o seguinte aspecto:

```
autojoin none
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility daemon
warntime 10
keepalive 2
deadtime 15
initdead 30
udpport 694
bcast eth0
node IP_NO_1
node IP_NO_2
ping IP_GATEWAY_OU_OUTRO_SERVIDOR
debug 1
auto_failback off
```

bem como o ficheiro */etc/heartbeat/authkeys*

```
autojoin none
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility daemon
warntime 10
keepalive 2
deadtime 15
initdead 30
udpport 694
bcast eth0
node IP_NO_1
node IP_NO_2
ping IP_GATEWAY_OU_OUTRO_SERVIDOR
debug 1
auto_failback off
```

e ainda o ficheiro */etc/heartbeat/haresources*

```
uqvpn01 IPaddr::172.16.0.2
```

Estes ficheiros devem se encontrar em todos os nós do *cluster*.

Referências

- [1] Sip call example, Julho 2011. Disponível em http://www.en.voipforo.com/SIP/SIP_example.php, último acesso em 3 de Julho de 2011.
- [2] Iax call example, Abril 2011. Disponível em <http://www.en.voipforo.com/IAX/IAX-example-messages.php>, último acesso em 22 de Abril de 2011.
- [3] 7º estudo anual sobre pirataria de software, efectuado pela bsa e pela idc, Junho 2011. Disponível em <http://www.assoftware.pt/default.asp?flag=1&idmenu=2&idsubmenu=18>, último acesso em 22 de Junho de 2011.
- [4] J. Rosenberg. RFC3261: Session initiation protocol.
- [5] Information Sciences Institute University of Southern California. RFC793: Transmission control protocol. <http://www.ietf.org/rfc/rfc793.txt>.
- [6] J. Postel. RFC768: User datagram protocol.
- [7] Q. Xie R. Stewart. RFC2960: Stream control transmission protocol.
- [8] UC Irvine R. Fielding. RFC2616: Hypertext transfer protocol.
- [9] J. Postel. RFC821: User datagram protocol.
- [10] Columbia University H. Schulzrinne. RFC1889: A transport protocol for real-time applications.
- [11] V. Jacobson M. Handley. RFC2327: Session description protocol.
- [12] Columbia University H. Schulzrinne. RFC3550: A transport protocol for real-time applications.
- [13] Arquitectura da rede GSM, Abril 2011. Disponível em <http://www.pt.com/page/tutorials/gsm-tutorial>, último acesso em 22 de Abril de 2011.
- [14] Truphone E. Guy, Ed. RFC5457: Iana consideraocol description, tions for iax: Inter-asterisk exchange version 2.
- [15] Iax protocol description, Abril 2011. Disponível em <http://www.seteurocom.ru/materials/rus/iax.pdf>, último acesso em 22 de Abril de 2011.
- [16] RSL COM M. Arango. RFC2705: Media gateway control protocol.
- [17] Microsoft Corporation O. Levin. RFC3762: Telephone number mapping (enum) service registration for h.323.

- [18] Penetração de centrais pbx open-source, Abril 2011. Disponível em <http://www.nojitter.com/showArticle.jhtml?printable=true&articleID=212903167>, último acesso em 22 de Abril de 2011.
- [19] Spandsp, Junho 2011. Disponível em <http://www.soft-switch.org/spandsp-modules.html>, último acesso em 20 de Junho de 2011.
- [20] Hylafax, Junho 2011. Disponível em <http://www.hylafax.org/>, último acesso em 20 de Junho de 2011.
- [21] T.38, Junho 2011. Disponível em http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-T.38-200704-I!!PDF-E&type=items, último acesso em 20 de Junho de 2011.
- [22] Iaxmodem, Junho 2011. Disponível em <http://iaxmodem.sourceforge.net/>, último acesso em 20 de Junho de 2011.
- [23] Clientes - hylafax, Junho 2011. Disponível em http://www.hylafax.org/content/Desktop_Client_Software, último acesso em 20 de Junho de 2011.
- [24] Conectores - hylafax, Junho 2011. Disponível em http://www.hylafax.org/content/HylaFAX_Connectors, último acesso em 20 de Junho de 2011.
- [25] Avantfax, Junho 2011. Disponível em <http://www.avantfax.com/>, último acesso em 20 de Junho de 2011.
- [26] Markus Feilner. *OpenVPN: Building and Integrating Virtual Private Networks*. PacktPub, 2006.
- [27] Openssl, Abril 2011. Disponível em <http://www.openssl.org/>, último acesso em 22 de Abril de 2011.
- [28] Lzo, Abril 2011. Disponível em <http://www.oberhumer.com/opensource/lzo/>, último acesso em 22 de Abril de 2011.
- [29] R. Atkinson S. Kent, BBN Corp. RFC2401: Security architecture for the internet protocol.
- [30] Pfsense, Junho 2011. Disponível em <http://www.pfsense.com/>, último acesso em 20 de Junho de 2011.
- [31] Endian firewall, Junho 2011. Disponível em <http://www.endian.com/en/community/>, último acesso em 20 de Junho de 2011.
- [32] Php, Dezembro 2011. Disponível em <http://www.php.net/>, último acesso em 2 de Dezembro de 2011.
- [33] jquery, Dezembro 2011. Disponível em <http://jquery.com/>, último acesso em 2 de Dezembro de 2011.
- [34] Nginx, Dezembro 2011. Disponível em <http://nginx.org/>, último acesso em 2 de Dezembro de 2011.
- [35] Apache, Dezembro 2011. Disponível em <http://httpd.apache.org/>, último acesso em 2 de Dezembro de 2011.

- [36] lighttpd, Dezembro 2011. Disponível em <http://www.lighttpd.net/>, último acesso em 2 de Dezembro de 2011.
- [37] Estudo utilização de servidores de páginas, Dezembro 2011. Disponível em <http://news.netcraft.com/archives/2011/11/07/november-2011-web-server-survey.html>, último acesso em 2 de Dezembro de 2011.
- [38] Postgresql, Dezembro 2011. Disponível em <http://www.postgresql.org/>, último acesso em 2 de Dezembro de 2011.
- [39] Mysql, Dezembro 2011. Disponível em <http://www.mysql.com/products/community/>, último acesso em 2 de Dezembro de 2011.
- [40] Clearos, Janeiro 2012. Disponível em <http://www.clearfoundation.com/>, último acesso em 9 de Janeiro de 2012.
- [41] Routeros, Janeiro 2012. Disponível em <http://www.mikrotik.com/software.html#>, último acesso em 9 de Janeiro de 2012.
- [42] W. Ford R. Housley. RFC2459: internet x.509 public key infrastructure certificate and crl profile.
- [43] Skygate Technology P. Chown. RFC3268: Advanced encryption standard (aes) ciphersuites for transport layer security (tls).
- [44] Critical Angle Inc. M. Wahl. RFC2401: Lightweight directory access protocol.
- [45] Hearbeat, Dezembro 2011. Disponível em <http://linux-ha.org/wiki/Heartbeat>, último acesso em 2 de Dezembro de 2011.
- [46] Mysql connector/j, Janeiro 2012. Disponível em <http://www.mysql.com/downloads/connector/j/>, último acesso em 9 de Janeiro de 2012.
- [47] Oracle jdbc, Janeiro 2012. Disponível em <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc817-100207.html>, último acesso em 9 de Janeiro de 2012.
- [48] Gnu-hylafax, Janeiro 2012. Disponível em <http://gnu-hylafax.sourceforge.net>, último acesso em 9 de Janeiro de 2012.
- [49] Javamail, Janeiro 2012. Disponível em <http://www.oracle.com/technetwork/java/javamail/index.html>, último acesso em 9 de Janeiro de 2012.
- [50] Postscript language reference manual, Janeiro 2012. Disponível em www.adobe.com/products/postscript/pdfs/PLRM.pdf, último acesso em 9 de Janeiro de 2012.
- [51] lp5250d, Janeiro 2012. Disponível em <http://linux.die.net/man/1/lp5250d>, último acesso em 9 de Janeiro de 2012.
- [52] Webmin, Janeiro 2012. Disponível em <http://webmin.com/>, último acesso em 9 de Janeiro de 2012.
- [53] Modulo de administração openvpn, Janeiro 2012. Disponível em <http://www.openit.it/downloads/OpenVPNadmin/openvpn-2.5.wbm.gz>, último acesso em 9 de Janeiro de 2012.