

Faculdade de Engenharia da Universidade do Porto



**FEUP**

# **Redes e Serviços VoIP/3G e Web 2.0**

**Décio de Jesus Ferreira**

Relatório de Projecto realizado no Âmbito do  
Mestrado Integrado em Engenharia Informática e Computação  
Orientador: Gil Manuel Magalhães de Andrade Gonçalves (Mestre)

Julho de 2008



# **Redes e Serviços VoIP/3G e Web 2.0**

**Décio de Jesus Ferreira**

Relatório de Projecto realizado no Âmbito do  
Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Maria Henriqueta Nóvoa (Professora Doutora)

---

Arguente: Maria Manuela Cunha (Professora Doutora)

Vogal: Nuno Silva (Mestre)

18 de Julho de 2008



# Resumo

O presente documento constitui o relatório de estágio realizado pelo aluno Décio de Jesus Ferreira na PT Inovação, no âmbito do Mestrado Integrado de Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

As redes de terceira geração (3G) têm o objectivo de fundir dois dos mais bem sucedidos paradigmas na comunicação: as redes móveis e da Internet. O IP (Internet Protocol) Multimedia Subsystem (IMS) é o elemento-chave para a arquitetura 3G que torna possível fornecer acesso móvel omnipresente a todos os serviços que a Internet oferece. Imagine aceder à sua página Web favorita, ler o seu e-mail, assistir a um filme, ou fazer parte de uma videoconferência onde quer que esteja, precisando apenas de um aparelho 3G. Esta é a visão IMS.

Hoje em dia a Internet tornou-se mais importante do que nunca, deixou de se centrar nas ferramentas de software, e passou a focar-se nos dados. Os serviços da Web não são os servidores – embora eles sejam prestados através de uma maciça colecção de servidores de Internet – nem os browsers – embora sejam experimentados pelo utilizador dentro de um. Sem dados, as ferramentas são inúteis. Este é o conceito da Web 2.0.

Pretendeu-se com este projecto, numa primeira fase, desenvolver um Portal de Administração Web para uma plataforma de comunicações de nova geração (SIP Application Server), em cenários de redes fixas, redes móveis e de convergência fixo-móvel (VoIP/3G), tirando igualmente partindo de conceitos da Web 2.0.

A aplicação Web a desenvolver corre sobre serviços/aplicações que são compatíveis com o standard IP Multimedia Subsystem (IMS), materializados pela arquitectura de produtos shipnet® da PT Inovação, que representa a nova geração de redes e serviços para as redes de comunicação convergentes.

Numa segunda fase do projecto, foi realizada uma análise entre as tecnologias Jain SLEE e SIP Servlets. Estas duas tecnologias permitem desenvolver serviços para redes VoIP/IMS e consequentemente para o shipnet® Service Delivery Framework (SDF), que disponibiliza um layer de Service Enablers (tais como localização, presença, messaging, etc.) e um conjunto de plataformas (SIP Application Server, IM-SSF, etc.), potenciando assim uma *framework* para um rápido desenvolvimento de serviços/aplicações inovadores.



# Abstract

This is the report of probation made by the student Décio de Jesus Ferreira in PT Inovação, under Master in Informatics and Computing Engineering, Faculdade de Engenharia da Universidade do Porto.

Third Generation (3G) networks aim to merge two of the most successful paradigms in communications: cellular networks and the Internet. The IP (Internet Protocol) Multimedia Subsystem (IMS) is the key element in the 3G architecture that makes it possible to provide ubiquitous cellular access to all the services that the Internet provides. Picture yourself accessing your favorite web pages, reading your email, watching a movie, or taking part in a videoconference wherever you are by simply pulling a 3G hand-held device out of your pocket. This is the IMS vision.

Today, the Internet has become more important than ever, has left to focus on software tools, and started to focus on the data. The Web services are not the servers - although they are provided through a massive collection of Internet servers - non the browsers - although they are experienced by users within one. Without data, the tools are useless. This is the concept of Web 2.0.

Initially, this project aimed to develop a Web Administration portal for a platform of communications of new generation (SIP Application Server), in scenarios of fixed networks, mobile networks and fixed-mobile convergence (VoIP/3G), with concepts of Web 2.0.

The Web application runs under services/applications that are compatible with the standard IP Multimedia Subsystem (IMS), materialised by the architecture of products shipnet® of PT Inovação, which represents the new generation of networks and services for the converging communications networks.

On a second phase of the project, an analysis was performed between technologies Jain Slee and SIP Servlets. These two technologies allow to develop services for networks VoIP/IMS and the shipnet® Service Delivery Framework (SDF), which provides a layer of Service Enablers (such as location, presence, messaging) and a number of platforms (SIP Application Server, IM-SSF, etc.), boosting a framework for rapid development of innovative services/applications.



# Agradecimentos

Durante o desenvolvimento deste projecto, foram fundamentais os contributos de natureza diversa que não podem deixar de ser realçados.

Por essa razão, desejo expressar os meus sinceros agradecimentos a todas as pessoas da PT Inovação que contribuíram para o sucesso do trabalho aí realizado. São elas: Ana Luísa Lito, Filipa Jesus Ferreira, João Baltazar, José Alberto Leal, José Miguel Freitas, Marco Filipe Monteiro, Nuno Silva e Pedro Elói Braz.

Agradecer aos professores da Faculdade de Engenharia da Universidade do Porto António Augusto Sousa, Raul Moreira Vidal, e Gil Manuel Gonçalves, pela contribuição para a realização deste estágio.

Agradecer aos meus pais, irmão e namorada, pelo estímulo e apoio incondicional desde a primeira hora.

O Autor



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto/Enquadramento . . . . .	1
1.2	Projecto . . . . .	2
1.3	Planeamento . . . . .	3
1.3.1	Processo de Desenvolvimento de Sistemas da PT Inovação . . . . .	4
1.4	Motivação e Objectivos . . . . .	5
1.5	Estrutura da Dissertação . . . . .	6
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>7</b>
2.1	shipnet . . . . .	7
2.1.1	ip-Jib . . . . .	8
2.1.2	ip-Centrex . . . . .	10
2.2	Web 2.0 . . . . .	12
2.3	Ruby on Rails . . . . .	13
2.3.1	Arquitectura . . . . .	15
2.3.2	Algumas características diferenciadoras . . . . .	16
2.3.3	Comparação em relação a outras <i>frameworks</i> . . . . .	19
2.4	Web Services . . . . .	21
2.5	IP Multimedia Subsystem (IMS) . . . . .	21
2.6	Jain SLEE . . . . .	23
2.7	SIP Servlets . . . . .	25
2.8	Resumo . . . . .	25
<b>3</b>	<b>Redes e Serviços VoIP/3G e Web 2.0</b>	<b>27</b>
3.1	Conceitos . . . . .	27
3.2	Camada de Interface . . . . .	29
3.2.1	Melhoramento do Portal de Administração . . . . .	34
3.2.2	Modelo de Dados . . . . .	37
3.3	Camada de Serviços . . . . .	37
3.4	Resumo e Conclusões . . . . .	38
<b>4</b>	<b>Implementação</b>	<b>41</b>
4.1	Camada de Interface . . . . .	41
4.1.1	Melhoramento do Portal de Administração . . . . .	45
4.2	Camada de Serviços . . . . .	48
4.2.1	Implementação . . . . .	49
4.3	Resumo . . . . .	49

## CONTEÚDO

<b>5</b>	<b>Avaliação dos Resultados</b>	<b>51</b>
5.1	Análise Teórica . . . . .	51
5.2	Análise Prática . . . . .	52
5.3	Conclusões . . . . .	52
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>55</b>
6.1	Satisfação dos Objectivos . . . . .	55
6.2	Trabalho Futuro . . . . .	56
	<b>Referências</b>	<b>58</b>
<b>A</b>	<b>Requisitos</b>	<b>59</b>
A.1	Interface . . . . .	59
A.1.1	Apresentação . . . . .	59
A.1.2	Atributos . . . . .	62
A.1.3	Entidades . . . . .	62
A.1.4	Funcionalidades . . . . .	66
A.1.5	Grupos Tipo . . . . .	75
A.1.6	Provisão . . . . .	76
A.1.7	Restrições . . . . .	76
<b>B</b>	<b>Plano de Trabalho Final</b>	<b>79</b>
<b>C</b>	<b>Wireframe da segunda versão do Portal</b>	<b>81</b>
<b>D</b>	<b>Modelos de Dados</b>	<b>85</b>

# Lista de Figuras

1.1	Plano de Trabalho . . . . .	3
1.2	Diagrama do Processo de Desenvolvimento de Sistemas . . . . .	4
2.1	shipnet® SDF (Service Delivery Framework) [ipc] . . . . .	8
2.2	Arquitectura da plataforma ip-Jib [ipj] . . . . .	9
2.3	Arquitectura funcional do serviço ip-Centrex . . . . .	10
2.4	Página de configuração de serviços . . . . .	11
2.5	Exemplo de criação de regra de reencaminhamento . . . . .	12
2.6	Arquitectura do Ruby on Rails [A.P08] . . . . .	16
2.7	Arquitectura Web Service [Wik08b] . . . . .	21
2.8	Arquitectura IMS [Wik08a] . . . . .	22
2.9	Os quatro elementos básicos do JSLEE [Hai04] . . . . .	24
3.1	Print Screen do Portal de Self-Care . . . . .	29
3.2	Casos de Uso . . . . .	31
3.3	Casos de Uso - Administração . . . . .	32
3.4	Casos de Uso . . . . .	35
3.5	Casos de Uso - Administração . . . . .	36
3.6	Casos de Uso - Administração . . . . .	36
3.7	Click-to-Call - Diagrama de Sequência . . . . .	38
4.1	Portal de Administração . . . . .	41
4.2	Áreas e Extensões . . . . .	42
4.3	Menus Áreas e Extensões . . . . .	42
4.4	Numeração . . . . .	43
4.5	Funcionalidades . . . . .	44
4.6	Portal de Administração - Segunda Versão . . . . .	45
4.7	Adicionar Área . . . . .	47
4.8	Arquitectura Física do Sistema SIP . . . . .	48
B.1	Plano de Trabalho Final . . . . .	79
C.1	Definições . . . . .	81
C.2	Utilizadores . . . . .	82
C.3	Funcionalidades . . . . .	82
C.4	Terminais . . . . .	83
C.5	Numeração . . . . .	83
C.6	Perfis . . . . .	84

## LISTA DE FIGURAS

C.7	Relatórios . . . . .	84
D.1	Primeira Versão Modelo de Dados - Parte 1/2 . . . . .	86
D.2	Primeira Versão Modelo de Dados - Parte 2/2 . . . . .	87
D.3	Segunda Versão Modelo de Dados - Parte 1/2 . . . . .	88
D.4	Segunda Versão Modelo de Dados - Parte 2/2 . . . . .	89

# Lista de Tabelas

5.1	Comparação Jain Slee e SIP Servlets . . . . .	54
-----	---	----

## LISTA DE TABELAS

# Abreviaturas e Símbolos

3G	third generation
3GPP	3rd Generation Partnership Project
AJAX	Asynchronous Javascript And XML
AOP	Aspect Oriented Programming
API	Application Programming Interface
AS	Application Server
CMS	Content Management Systems
CS	Circuit Switched
DRY	Don't Repeat Yourself
DSL	Digital Subscriber Line
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HSS	Home Subscriber Servers
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IDE	Integrated Development Environment
IMS	IP Multimedia Subsystem
IP	Internet Protocol
J2EE	Java 2 Platform, Enterprise Edition
JDBC	Java Database Connectivity
JEE	Java Platform, Enterprise Edition
JSLEE	Java Service Logic Execution Environment
MVC	Model-View-Controller
OMA	Open Mobile Alliance
PBX	Private Branch eXchange
PDS	Processo de Desenvolvimento de Sistemas
PHP	Hypertext Preprocessor
PSTN	Public Switch Telephony Network
RA	Resource Adaptor
RPG	Redes de Próxima Geração
SBB	Service Building Blocks
SDF	Service Delivery Framework
shipnet	Service Handling on IP Networks
SIP	Session Initiation Protocol
SMS	Short Message Service

## ABREVIATURAS E SÍMBOLOS

SOAP	Simple Object Access Protocol
VoIP	Voice over Internet Protocol
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
XML	eXtensible Markup Language
W3C	World Wide Web Consortium
WLAN	Wireless Local Area Network
WSDL	Web Services Description Language

# Capítulo 1

## Introdução

### 1.1 Contexto/Enquadramento

Nos últimos anos tem-se assistido a um grande crescimento na utilização dos serviços disponibilizados pela Internet/redes IP (Skype, YouTube, SAPO Messenger, GoogleMaps, Mashups, etc) e pelas redes móveis (Pré-Pago, Virtual Private Networks, Ring Back Tones, etc.). O passo seguinte é combinar a flexibilidade das redes IP (e os serviços da Web 2.0) e os aspectos de mobilidade (3G) numa arquitectura de rede de próxima geração, oferecendo assim o melhor dos dois "mundos". O projecto 3GPP definiu a arquitectura "IP Multimedia Subsystem" (IMS) que permite a convergência de vários serviços (dados, voz, vídeo, etc.) sobre uma infra-estrutura de rede IP, utilizando o protocolo SIP (standard do IETF).

Contrastando com o domínio CS (Circuit Switched), o IMS permite o estabelecimento de qualquer tipo de sessão multimédia, possibilitando a modificação das sessões estabelecidas dinamicamente ("on the fly"), por exemplo, adicionar uma componente de vídeo a uma sessão de voz existente, adicionar/retirar de uma forma dinâmica, utilizadores a uma sessão, etc. Este tipo de capacidades abre um novo leque de potencialidades para o desenvolvimento de serviços inovadores.

Os "Servidores de Aplicações SIP" (SIP AS – SIP Application Servers) são plataformas que utilizam protocolos SIP (Session Initiation Protocol) para estabelecer sessões de comunicação tempo-real em redes IP, envolvendo qualquer combinação de média (voz, vídeo, dados, etc.), disponibilizando um ambiente para a criação e execução de aplicações em redes de próxima geração (VoIP/3G).

Virtual Call Centers, Voice Portals, Location-based services, Content Charging, Mobile Content Delivery, Multimedia Messaging e Mobile Gaming são alguns exem-

plos de aplicações inovadoras que utilizam estas plataformas como ambiente de runtime.

O suporte deste conjunto alargado de aplicações, de forma agnóstica ao tipo de rede de acesso, tem sido apresentado como um requisito por parte dos operadores de comunicações de forma a evoluir de uma forma consistente o leque de aplicações que disponibilizam aos seus clientes.

Os "Servidores de Aplicações SIP" endereçam igualmente o desafio crescente de convergência de serviços, permitindo que diferentes redes, terminais e aplicações trabalhem conjuntamente.

Em suma, os Servidores SIP transportam o modelo distribuído e aberto de desenvolvimento de aplicações da Internet para o mundo das telecomunicações, permitindo rapidamente desenvolver e instalar diferentes aplicações em redes convergentes.

Este estágio elaborado na PT Inovação, SA (uma empresa moderna, cuja credibilidade assenta em 50 anos de experiência tecnológica acumulada em telecomunicações), surgiu da necessidade de desenvolvimento de um portal de Administração para o projecto shipnet® , inserido no contexto acima descrito.

## 1.2 Projecto

Pretendeu-se com este projecto, numa primeira fase, desenvolver um Portal de Administração Web para uma plataforma de comunicações de nova geração (SIP Application Server), em cenários de redes fixas, redes móveis e de convergência fixo-móvel (VoIP/3G), tirando igualmente partido de conceitos da Web 2.0.

A aplicação Web a desenvolver corre sobre serviços/aplicações que são compatíveis com o standard IP Multimedia Subsystem (IMS), materializados pela arquitectura de produtos shipnet® da PT Inovação, que representa a nova geração de redes e serviços para as redes de comunicação convergentes.

Numa segunda fase do projecto, foi realizada uma análise entre as tecnologias Jain SLEE e SIP Servlets. Estas duas tecnologias permitem desenvolver serviços para redes VoIP/IMS e conseqüentemente para o shipnet® Service Delivery Framework (SDF), que disponibiliza um layer de Service Enablers (tais como localização, presença, messaging, etc.) e um conjunto de plataformas (SIP Application Server, IM-SSF, etc.), potenciando assim uma *framework* para um rápido desenvolvimento de serviços/aplicações inovadores.

No âmbito deste projecto, foram abordadas diversas tecnologias/linguagens de programação tais como Ruby on Rails, Java/J2EE, XML/Web Services, XHTML, CSS, JavaScript, Ruby. O projecto teve uma forte componente de Engenharia de

## Introdução

Software, complementado com noções avançadas de redes IP/SIP. Durante o desenvolvimento do projecto fui integrado numa equipa multidisciplinar do Programa shipnet®, abrangendo os vários leques de competências.

### 1.3 Planeamento

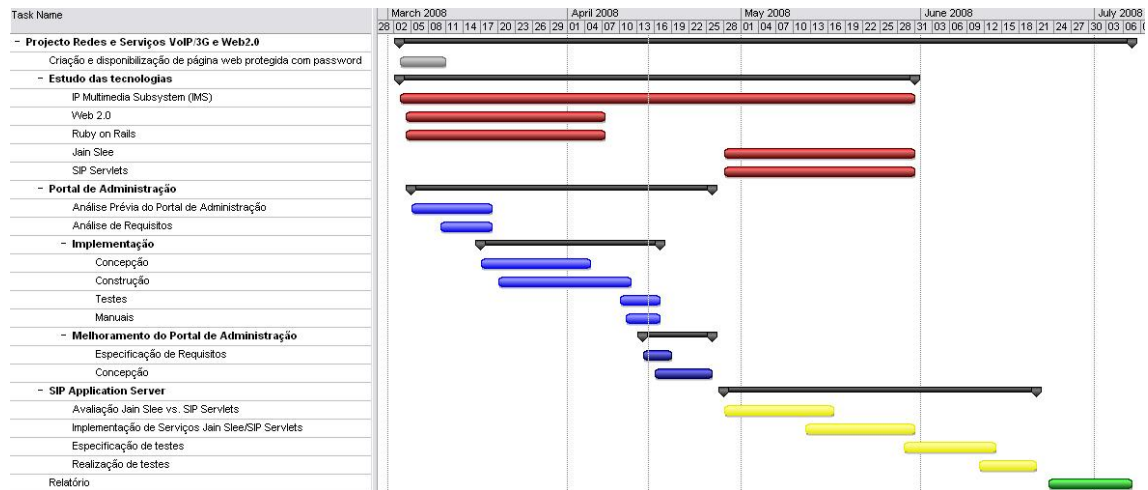


Figura 1.1: Plano de Trabalho

Na figura 1.1 é descrito o plano de trabalho proposto para o estágio. O projecto foi iniciado com a criação e disponibilização de uma página Web, com o objectivo de acompanhar a evolução do trabalho desenvolvido. O estudo das tecnologias foi outra tarefa iniciada cedo, tendo-se esta estendido ao longo do desenrolar do projecto.

A tarefa associada ao "Portal de Administração", teve por objectivo o desenvolvimento de uma Aplicação Web em Ruby on Rails, que seguiu o Processo de Desenvolvimento de Sistemas (PDS) da PT Inovação. Esta tarefa centrou-se na implementação do sistema, que cobriu as etapas de Concepção, Construção, Testes e elaboração dos Manuais. A etapa de Especificação de Requisitos, já se encontrava realizada, tendo o cliente PT Prime entregue o Documento de Especificação de Requisitos (Anexo A). Uma vez que o desenvolvimento desta aplicação foi apenas parte de um projecto, também a etapa de instalação não fez parte do plano de trabalho.

Terminada a implementação da aplicação Web, fez parte do plano de trabalho, o desenvolvimento de uma segunda versão da mesma aplicação, que envolveu a passagem do projecto para uma nova equipa de trabalho. Esta decisão foi motivada, em primeiro lugar pelo facto de na versão inicial não terem sido considerados todos os requisitos, e em segundo, pela necessidade de passar o desenvolvimento para uma nova equipa de trabalho. Dado isto, fez parte do meu projecto acompanhar esta "passagem de testemunho".

## Introdução

Na última fase do projecto, entitulado "SIP Application Server" foi elaborado um estudo comparativo entre as tecnologias Jain Slee e SIP Servlets, esta fase dividiu-se em quatro partes, uma primeira de avaliação das duas tecnologias, seguida de uma implementação de serviços, especificação de testes, e finalmente a realização desses mesmos testes retirando daí as conclusões.

Ao longo do estágio, o plano de trabalho sofreu algumas alterações (Anexo B), tendo resultado numa segunda versão. Esta alteração verificou-se na etapa de melhoria do Portal de Administração, que foi extendido até ao final do projecto.

### 1.3.1 Processo de Desenvolvimento de Sistemas da PT Inovação

O Processo de Desenvolvimento de Sistemas (PDS) da PT Inovação é um processo iterativo, que tem como objectivo desenvolver sistemas, podendo este ser feito a partir de um conjunto de requisitos apresentados por um cliente externo ou identificados internamente na própria PT Inovação.

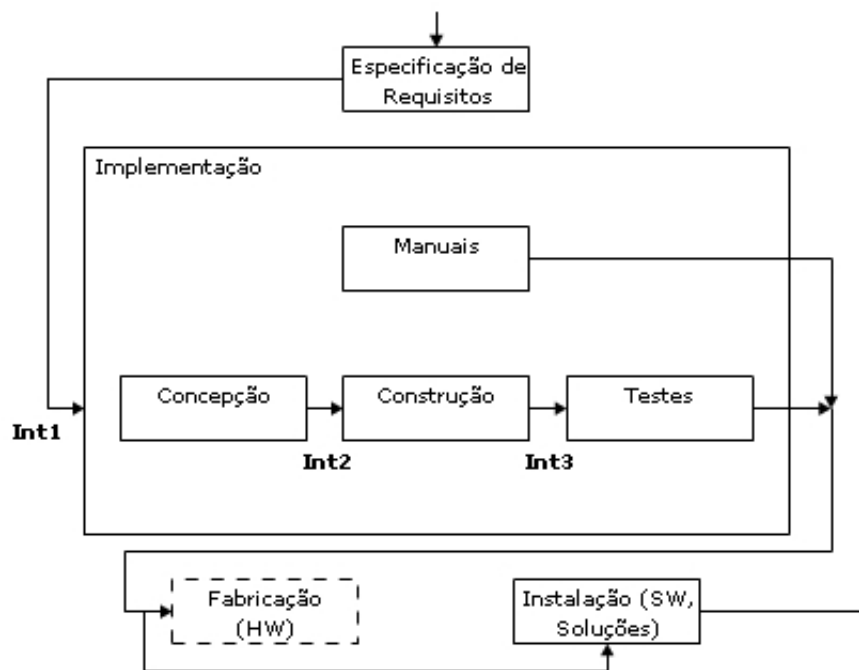


Figura 1.2: Diagrama do Processo de Desenvolvimento de Sistemas

Tal como se pode ver na Figura 1.2 da página 4, no PDS existem vários procedimentos, estando este divididos em três fases.

A Especificação de Requisitos corresponde à primeira fase, que resulta no documento de Definição de Requisitos do sistema a ser desenvolvido.

A segunda fase do processo, Implementação, é a fase que cobre a Concepção, Construção e os Testes. A Concepção do Sistema compreende a Análise e o Desenho

do Sistema. O documento resultante desta etapa é o Relatório de Concepção, no qual constam informações sobre a Especificação de Interface, Interface de Utilizador, e o Modelo de Informação e Arquitectura. Segue-se a Construção do Sistema. O controlo da Qualidade da construção é realizado através da fase de testes, última etapa da fase de implementação. De início é feita a especificação dos testes que têm como objectivo definir o domínio dos testes e as características/funcionalidades a serem testadas, garantindo o cumprimento dos requisitos definidos e a qualidade do produto realizado. Após esta especificação são executados os testes e é elaborado o Relatório de Testes.

Em paralelo com estas fases de implementação é necessário elaborar o Manual de Utilização.

Por fim é feita a instalação do Sistemas desenvolvidos na PT Inovação.

### 1.4 Motivação e Objectivos

A motivação principal para a realização deste trabalho resulta da importância crescente que as redes móveis e a Internet têm vindo a apresentar. O conceito de Web 2.0 e respectivas tecnologias associadas, a arquitectura IMS (IP Multimedia Subsystem) ou tecnologias como as redes de terceira geração, são temáticas em que ainda existe um grande potencial de crescimento.

A Internet tem experimentado um forte crescimento ao longo dos últimos anos, tendo começado por uma pequena rede de alguns sites de investigação e transformando-se numa enorme rede mundial. A principal razão para este crescimento tem sido a capacidade de fornecer um conjunto de serviços extremamente úteis para milhões de utilizadores. Os exemplos mais conhecidos são a World Wide Web e o e-mail.

Mas hoje em dia a Internet tornou-se mais importante do que nunca, deixou de se centrar nas ferramentas de software, e passou a focar-se nos dados. Os serviços da Web não são os servidores – embora eles sejam prestados através de uma maciça colecção de servidores de Internet – nem os browsers – embora sejam experimentados pelo utilizador dentro de um. Sem dados, as ferramentas são inúteis. Este é o conceito da Web 2.0.

Por outro lado, actualmente, as redes móveis prestam serviços a mais de 3.3 mil milhões de utilizadores em todo o mundo - equivalente a metade da população mundial [Reu07]. Esses serviços incluem, naturalmente, chamadas telefónicas, mas não se limitam a estas. As redes telefónicas de hoje prestam serviços que vão desde a simples mensagem SMS até às mensagens multimédia que incluem vídeo, áudio e texto. Os utilizadores de telemóveis são capazes de navegar na internet e ler e-mails utilizando ligações de dados, ou mesmo aceder a serviços locais de notificação de utilizadores quando um amigo ou colega está perto.

## Introdução

As redes de terceira geração (3G) têm o objectivo de fundir dois dos mais bem sucedidos paradigmas na comunicação: as redes móveis e da Internet. O IP (Internet Protocol) Multimedia Subsystem (IMS) é o elemento-chave para a arquitetura 3G que torna possível fornecer acesso móvel omnipresente a todos os serviços que a Internet oferece. Imagine aceder à sua página Web favorita, ler o seu e-mail, assistir a um filme, ou fazer parte de uma videoconferência onde quer que esteja, precisando apenas de um aparelho 3G. Esta é a visão IMS.

O projecto shipnet, em que se insere este trabalho, pretende combinar a flexibilidade das redes IP (e os serviços da Web 2.0) e os aspectos de mobilidade (3G) numa arquitectura de rede de próxima geração, oferecendo assim o melhor dos dois "mundos".

### 1.5 Estrutura da Dissertação

Este trabalho encontra-se estruturado em 6 capítulos dos quais, o primeiro é composto por esta introdução.

No capítulo 2, é descrito o estado da arte e são apresentados trabalhos relacionados.

No capítulo 3, é feita uma apresentação detalhada do problema a resolver.

O capítulo 4, é dedicado à apresentação de detalhes de nível mais baixo relacionados com o enquadramento e implementação das soluções preconizadas no capítulo anterior.

No capítulo 5, são analisados os resultados do estudo das tecnologias associadas à camada de serviços.

No capítulo 6 é apresentado um resumo do trabalho realizado e apreciada a satisfação dos objectivos do trabalho, uma lista de contribuições principais do trabalho e as direcções para trabalho futuro.

## Capítulo 2

# Revisão Bibliográfica

Uma das mais importantes características da plataforma a desenvolver na primeira fase do trabalho é o facto de fazer parte de um projecto actualmente em desenvolvimento, a interface Web ip-Centrex, que permite, por exemplo, que as funções das centrais telefónicas passem a ser virtualizadas e integralmente geridas através de um interface Web amigável. Embora as duas plataformas funcionem de forma independente, algumas das tecnologias a utilizar serão impostas por esta interface.

A segunda parte do projecto está inserido num contexto de investigação. Aqui são apresentadas as tecnologias Jain Slee e SIP Servlets, em análise.

Neste capítulo são apresentadas breves descrições de infra-estruturas tecnológicas implementadas na PT Inovação que estão directamente relacionadas com o projecto: shipnet (ip-Jib e ip-Centrex). São também descritas várias tecnologias e conceitos considerados para o trabalho realizado.

### 2.1 shipnet

shipnet – *Service Handling on IP Networks* é a arquitectura de referência que enquadra a família de produtos da PT Inovação, para dar resposta aos desafios e necessidades das redes de próxima geração (RPG) num cenário de convergência fixo-móvel.

Baseada nas especificações 3GPP IP Multimedia Subsystem (IMS), ETSI TIS-PAN (Telecoms & Internet converged Services & Protocols for Advanced Networks) e Open Mobile Alliance (OMA), a arquitectura shipnet® é implementada através de uma família de produtos nos níveis de serviços, controlo e transporte.

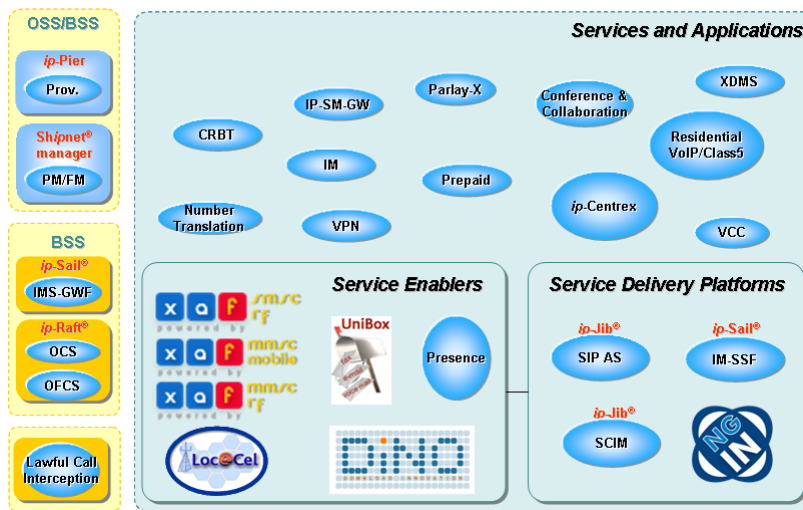


Figura 2.1: shipnet® SDF (Service Delivery Framework) [ipc]

Com os produtos shipnet a PT Inovação reforça a sua posição como um actor importante nas redes de próxima geração, garantindo a evolução das soluções de rede e serviços actualmente existentes [Ino07a].

O pacote de serviços Classe-5 consiste num conjunto de serviços em Jain SLEE preparados para correr na plataforma ip-Jib, que permite trazer a terminais vulgares POTS (Plain Old Telephone Service) funcionalidades típicas das redes de nova geração. Estes serviços estão conforme os standards do ETSI TS 181 002 4 e 3GPP TS 22.273 4 relativos aos requisitos necessários para dar suporte às funcionalidades de IP Multimedia Telephony sobre uma rede de próxima geração (RPG).

### 2.1.1 ip-Jib

A plataforma ip-Jib [Ino07b] é a solução convergente de SIP/IMS Application Server da família de produtos shipnet (Service Handling on IP Networks) da PT Inovação, baseado em tecnologia JAIN SLEE e JEE.

É uma plataforma carrier-grade escalável, baseada nos standards do 3GPP IMS, ETSI TISPAN e OMA, disponibilizando um conjunto de interfaces para integração com redes de nova geração IMS/TISPAN ou com sistemas legados, bem como um conjunto de interfaces para terceiros, para desenvolvimento de serviços personalizados.

O ip-Jib é composto pelas seguintes camadas lógicas: SLEE, Enablers e Resource Adapters, que disponibilizam ao criador de serviços um ambiente dinâmico e extensível para a criação e execução dos mesmos.

- Resource Adapter: Adaptador que converte os dados de um determinado protocolo, de um recurso externo ao JAIN SLEE para eventos prontos a serem processados pelos serviços JAIN SLEE.
- Service Enabler: Módulo utilizado na construção de serviços que permite ao programador obter uma API com um maior nível de abstracção e menor complexidade.
- Serviço: Lógica do negócio que executa na plataforma ip-Jib.

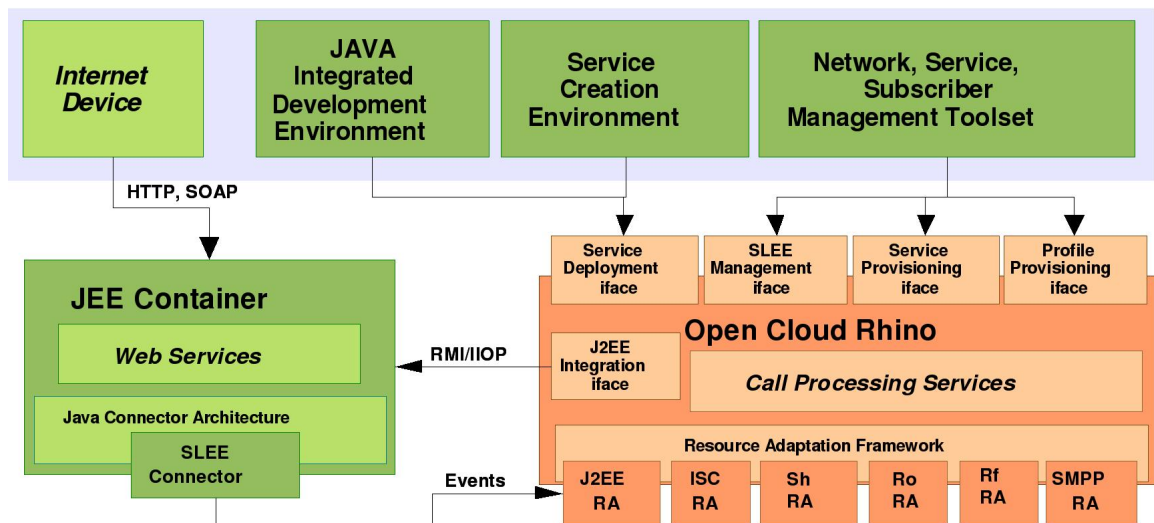


Figura 2.2: Arquitectura da plataforma ip-Jib [ipj]

Como podemos observar na Figura 2.2, esta plataforma é constituída por dois componentes principais: o Servidor JEE contendo a lógica aplicacional e de apresentação dos serviços disponibilizados, e o Servidor JSLEE, disponibilizando um ambiente de criação de serviços de tempo real assim como os interfaces necessários para uma infra-estrutura SIP/IMS.

Como servidor de JEE é usado o JBoss mas pode ser usada qualquer outra solução compatível com a especificação JEE e que cumpra os requisitos de desempenho exigidos pelas aplicações shipnet. No caso do servidor JSLEE é usado o servidor Rhino da empresa OpenCloud podendo-se usar uma outra solução compatível com a especificação JSLEE e que também cumpra os requisitos de desempenho exigidos pelas aplicações shipnet.

No interior do SIP AS temos os serviços. Estes serviços podem receber pedidos de utilizadores, iniciar pedidos para utilizadores e interagir com outros serviços no mesmo ou noutra SIP AS. Estes serviços podem suportar SIP, HTTP/SOAP entre outras, de forma alternada ou em simultâneo.



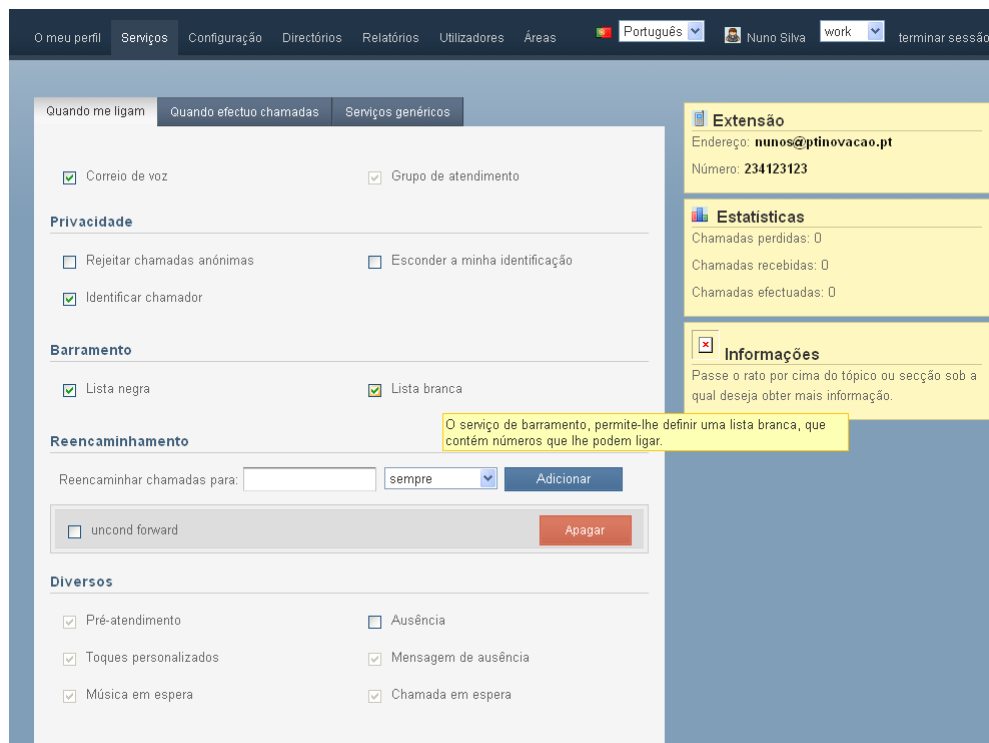


Figura 2.4: Página de configuração de serviços

Assim a construção de novos serviços baseados em enablers existentes torna-se mais simples.

Logo, como camadas na arquitectura teremos: a camada dos adaptadores, dos enablers e das aplicações. A interacção entre as diferentes camadas será realizada de várias formas consoante cada caso em concreto (i.e. consoante cada serviço).

Segundo a especificação, a comunicação entre a camada de adaptadores e de enablers/serviços é assíncrona (através do envio de eventos JSLEE – por exemplo o adaptador SIP quando recebe um pedido INVITE envia um evento JSLEE onde encapsula essa actividade externa). Na camada de serviços podemos efectuar comunicação síncrona ou assíncrona entre enablers/serviços dependendo dos casos.

Na camada de adaptadores de recursos (conhecidos por Resource Adaptors na nomenclatura JSLEE - têm como função fazer a ponte entre o contentor SLEE e os recursos externos) são necessários adaptadores para stacks SIP, adaptadores para servidores MRF, adaptadores para servidores de tarifação, entre outros.

Na camada de enablers são necessários SBBs (Service Building Blocks) que actuarão como childs (filhos ou slave) para os SBBs das aplicações (pai ou master). Isto é, estes SBBs serão comuns entre as várias aplicações permitindo que a sua utilização facilite a construção de serviços com maior complexidade usando para o efeito peças atómicas com funções bem definidas à partida. Um bom exemplo de

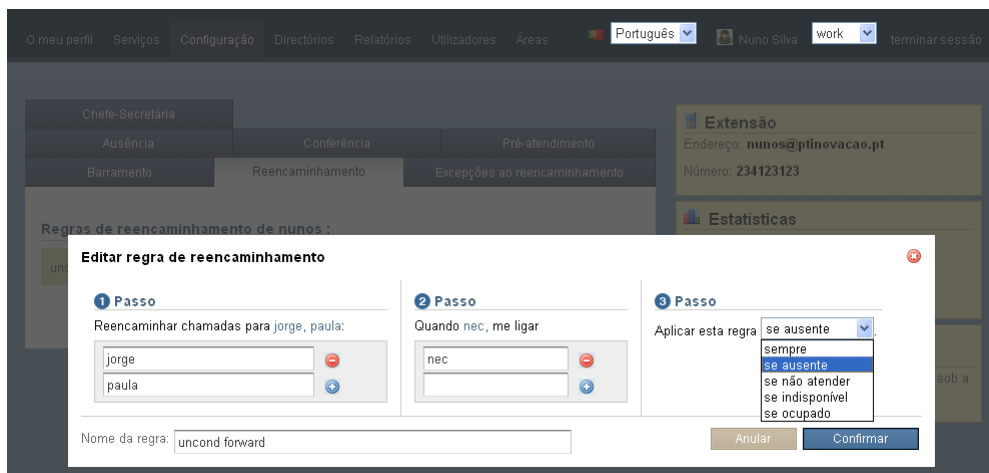


Figura 2.5: Exemplo de criação de regra de reencaminhamento

um SBB reutilizado por várias aplicações na construção dos serviços será o SBB de Proxy e/ou de B2BUA.

O SBB é o componente de código que tem a lógica do serviço. É neste componente que é programada a lógica a aplicar em cada evento recebido pelo mesmo. Por exemplo um SBB que defina que pretende receber eventos SIP que representem pedidos INVITE terá um método onde descreve a lógica a aplicar a esse INVITE.

Na camada de aplicações irão residir os serviços propriamente ditos que farão, ou não, uso directo dos enablers para cumprir a sua missão, ou então comunicam directamente com os adaptadores. A este nível podemos igualmente encontrar alguma comunicação assíncrona entre aplicações distintas através do envio de eventos SLEE. Para esta comunicação os SBBs das aplicações usam canais de comunicação partilhados entre eles (conhecidos por Activity Contexts no JSLEE).

Os vários serviços terão acesso à provisão dos vários utilizadores através de interface um JDBC disponibilizado pela plataforma.

## 2.2 Web 2.0

Embora o termo tenha uma conotação de uma nova versão da Web, ele não se refere à actualização nas suas especificações técnicas, mas a uma mudança na forma como ela é encarada por utilizadores e programadores.

O conceito de "Web 2.0" [O'R05] começou com uma conferência de brainstorming entre a O'Reilly e a MediaLive International. Dale Dougherty, pioneiro da web e vice-presidente da O'Reilly, notou que, ao contrário de haver "explodido", a Web estava mais importante do que nunca, apresentando interessantes aplicações novas e Websites aparecendo com surpreendente regularidade. E, o que é melhor, parecia

que as companhias que haviam sobrevivido ao colapso tinham algo em comum. O colapso "dot-com" marcou uma espécie de viragem que deu sentido a uma convocação do tipo "Web 2.0".

No ano e meio que se seguiu, o termo "Web 2.0" claramente se consagrou com mais de 9,5 milhões de citações no Google. Mas ainda existe um enorme desacordo sobre o que significa Web 2.0.

Na realidade, as competências centrais das companhias "Web 2.0" são:

- Serviços e não software empacotado, com "escalabilidade" de custo eficiente;
- Controlo sobre bases de dados únicas e difíceis de serem criadas e que ficam mais ricas quanto mais as pessoas as utilizarem;
- Confiança nos utilizadores como "co-programadores";
- Agregação de inteligência colectiva;
- Estimular a "cauda longa" através de auto-serviço para o cliente;
- Software para mais que um dispositivo;
- Interfaces de utilizador, modelos de desenvolvimento e modelos de negócios leves.

É com base nesta listagem de características que se avalia se uma aplicação está ou não de acordo com a definição de "Web 2.0". Quanto mais pontos englobar, mais fará jus ao nome. No entanto, a excelência numa área pode também contar mais do que alguns pequenos passos em todas as sete.

## 2.3 Ruby on Rails

Na generalidade, grande percentagem do tempo de desenvolvimento de software é gasto em componentes genéricos, que podem ser reutilizados nos vários componentes da aplicação. Ao resultado desta colecção de componentes é chamado de *framework* da aplicação. Ruby on Rails é uma *framework* que torna mais fácil o desenvolvimento, distribuição, manutenção de aplicações Web [DT06, Wil07].

Devido ao forte crescimento das aplicações Web, as *framework* Web foram desenvolvidas, de modo a fornecerem funcionalidades tais como: escalabilidade, performance, portabilidade e mecanismos de segurança. Do ponto de vista de desenvolvimento de uma aplicação, quando se usa uma *framework*, pretende-se tirar partido do maior número de funcionalidades implementadas, por forma a minimizar o tempo e esforço de desenvolvimento. Por forma a atingir este objectivo, os desenvolvimentos nas *framework* centram-se em fornecer soluções para os seguintes problemas:

- Apresentação
- Gestão de Sessões
- Validação
- Gestão de Eventos

Para muitas das *frameworks* Web actuais, é comum a criação de grandes ficheiros de configuração baseados em XML, ou junção e integração de vários componentes, provenientes de terceiros, exigindo que os programadores dediquem muito tempo com a configuração de funcionalidades.

David Heinemeir Hansson, ao criar Rails, desenvolveu uma *framework* que endereça os requisitos, não de todas as aplicações, mas das mais convencionais. Em particular, o Rails usa alguns princípios, ou ideias chave, de forma a libertar os programadores:

- **Convenção ao invés de configuração:** O Rails possibilita a criação de aplicações Web sem a necessidade de escrita de uma única linha de código XML para configuração. Ao eliminar tanto quanto possível os arquivos de configuração através da adoção de uma série de convenções, permite integrar as diferentes partes da *framework* sem praticamente nenhuma intervenção directa do programador. As convenções englobam todo tipo de aspectos de uma aplicação Web: o nome dos modelos e tabelas relacionais associadas, o nome das colunas das tabelas que constituem chaves primárias e estrangeiras, o nome dos arquivos e das classes que implementam cada camada da estrutura MVC (Model-View-Controller), o nome e identificador dos campos HTML entre outras.
- **Feedback instantâneo:** O Rails possibilita a alteração do sistema, e permite obter uma resposta imediata. Mudanças simples, em virtude da colaboração com o cliente, podem ser efectuadas, com pouco esforço por parte do programador, estando as alterações visuais disponíveis imediatamente.
- ***Framework* completa:** Ao invés de juntar várias extensões de múltiplos fornecedores, o Rails disponibiliza todos os componentes que são frequentemente precisos para as aplicações Web.

A escrita de uma aplicação Rails, deixando de lado configurações, consiste basicamente em três tarefas:

1. **Descrever e modelar o domínio da aplicação:** O domínio é o universo da aplicação, o que quer que este represente é necessário identificar tudo o que lhe pertence, que entidades existem, e como elas se relacionam com o universo e entre si.

2. Especificar a lógica da aplicação: Na lógica da aplicação são descritas as acções que o utilizador pode exercer sobre o sistema, bem como os procedimentos no caso do desencadeamento de eventos.
3. Escolher e desenhar os layouts: Neste ponto começa-se a pensar no comportamento na Web. Depois de se decidir as interações possíveis no sistema, decide-se a forma como estas serão apresentadas ao utilizador final.

Tudo o que é necessário para se desenvolver uma aplicação Rails, recai sobre estas categorias. Em alguns aspectos estas categorias estão correlacionadas, em particular, a escolha das acções que tem lugar no domínio da aplicação (categoria 2) e decidir quais os layouts que a aplicação terá (categoria 3). No entanto, as camadas de desenvolvimento são diferentes fases do desenvolvimento, assegurar que inter-operam sem problemas, é o trabalho da *framework*.

### 2.3.1 Arquitectura

A arquitectura de Rails é baseada em MVC (Model-View-Controller), um padrão de arquitetura de software. Com o aumento da complexidade das aplicações desenvolvidas torna-se fundamental a separação entre os dados (Model) e o layout (View). Desta forma, alterações feitas no layout não afectam a manipulação de dados, e estes poderão ser reorganizados sem alterar o layout.

O MVC resolve este problema através da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interacção com o utilizador, introduzindo um componente entre os dois: o Controller (controlador). MVC é usado em padrões de projetos de software, mas MVC abrange mais da arquitetura de uma aplicação do que é típico para um padrão de projecto.

O Model (dados) é onde se encontra a "lógica de negócio". Este define os "objectos do domínio" como o utilizador, produto, etc., as relações entre eles, e o processamento que ocorre.

O View (layout) é a forma como o sistema interage com os utilizadores. Em aplicações Web, isto traduz-se em HTML, Javascript, e provavelmente imagens e Flash.

O Controller (controlador) liga o Model e o View. Numa aplicação Web, o Controller recebe o pedido do utilizador, interage com o Model para processar e/ou receber dados, e disponibiliza-a através da View.

O View e o Model não devem, teoricamente, ter conhecimento sobre nada, um do outro. Já no caso do Controller, este sabe o suficiente sobre os dois para os interligar. Para o programador deve ser possível a partir de um Model, e aplicando um diferente View e Controller, produzir um diferente interface para o mesmo sistema (e.g. GUI, Web Service, linha de comandos).

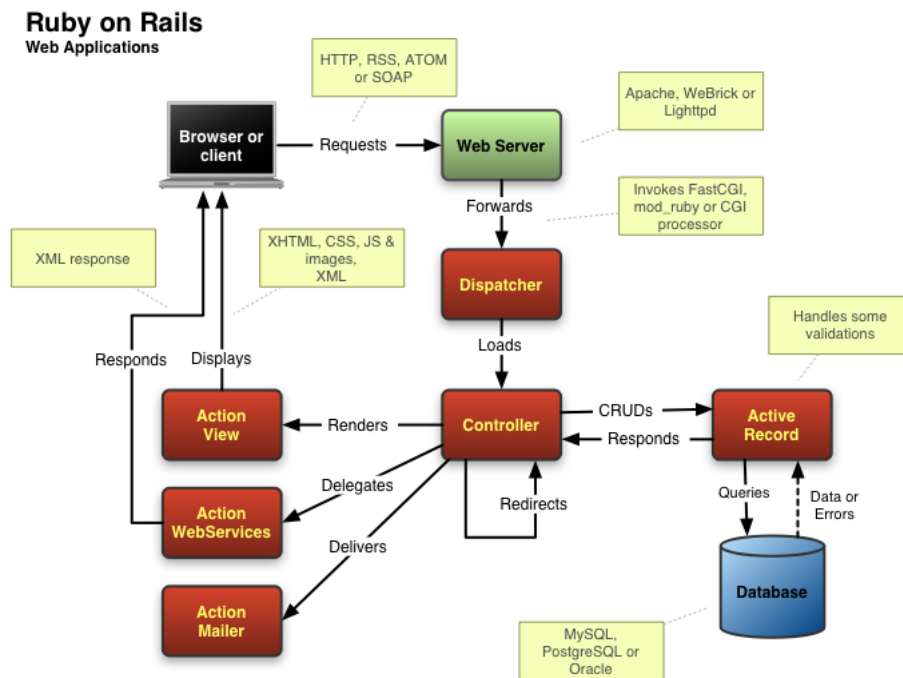


Figura 2.6: Arquitectura do Ruby on Rails [A.P08]

O rails não impõe uma separação completa dos conceitos (provavelmente nenhuma *framework* o faz), mas a sua mecânica é verdadeiramente MVC.

### 2.3.2 Algumas características diferenciadoras

Algumas das características diferenciadoras de Ruby on Rails são:

- Quantidade reduzida de código: Esta característica deve-se em grande parte a Ruby [Bla06], uma linguagem de programação dinâmica e open-source com foco na simplicidade e produtividade. Tem uma sintaxe elegante de leitura e escrita intuitiva.
- Poucas linhas de configurações: As aplicações escritas em Rails requerem poucas linhas de especificação para determinar a ligação à base de dados a usar, bem como para cada tipo de "routes" que a usar (tipicamente 3 a 4 linhas para toda a aplicação). Comparativamente às cerca de 20 linhas necessárias a uma aplicação Rails, uma aplicação Java de tamanho médio, requer centenas de linhas de configurações.
- DRY (Don't Repeat Yourself): O Rails dá ênfase à não repetição de código e obriga a manter a ortogonalidade à volta dos conceitos. Esta característica

permite reduzir as situações de simples aglomeração de código, que sendo respeitada permite uma redução dos custos de manutenção e aumento da produtividade, eliminando a redundância de código, que é muito comum nas aplicações Web desenvolvidas em Java: quando uma alteração é feita numa camada (por exemplo, adicionar uma coluna a uma tabela de base de dados) esta necessita de ser propagada por todas as outras camadas.

- Metodologias de desenvolvimento ágil: O Rails foca-se no "feedback" instantâneo, onde o programador efectua alterações, que estão imediatamente disponíveis para uso e testes. O desenvolvimento sobre Rails elimina o ciclo de "build", "deploy" e de reinício da aplicação.
- Uma *framework* completa: O Rails inclui todos os componentes necessários para desenvolver uma aplicação Web completa, tais como "drivers" de ligação a bases de dados (opensource e comerciais); servidor Web, completamente desenvolvido em Ruby, que permite iniciar o desenvolvimento rapidamente sem necessidade de configuração de um servidor como o Apache ou o Lighttpd; uma camada de mapeamento de dados a objectos (ORM); uma *framework* "Model-View-Controller"; testes unitários e funcionais; suporte nativo para Ajax e Stylesheets; suporte para múltiplos sistemas de templates; suporte para "deployments" automáticos em sistemas heterogéneos; suporte para envio e recepção de emails; web-services; etc.
- Não é imperativa a automatização de um IDE: É um facto que muita da produtividade em Java, advém do uso de ferramentas de automação de escrita e alteração de código, Java é defendido como sendo produtivo, desde que se use as ferramentas certas por forma a mitigar o impacto do volume de código e configurações. No entanto a curva de aprendizagem para estas ferramentas é elevada, e o processo de escolha destas, além de ser moroso, ainda se corre o risco do abandono das mesmas por parte dos fornecedores. Muita da pesquisa sobre anotações e "Aspect Oriented Programming" é feita de forma a tornar a linguagem Java produtiva, dada a sua adoção. Ao contrário, em Ruby on Rails, estas são dinâmicas e concisas, suportando o estilo de "Aspect Oriented Programming" de separação de conceitos (existem ferramentas disponíveis de AOP, para Rails, no entanto estas não são usadas unanimemente), minimização de configurações, e de IDE's desenhados para minimizar a necessidade de programação redundante. Ruby é extremamente flexível neste campo, permitindo uma alteração rápida de funcionalidades sem alterar o código da *framework*.
- Injecção de dependências (Dependency Injection or Inversion of Control): Existem bibliotecas em Ruby que implementam a injecção de dependências, mas

a natureza dinâmica do Ruby, aliada ao seu esquema de tipagem, fazem desta um extra, e não uma necessidade.

- Código livre: Tanto Ruby como o Rails estão disponíveis sob licenças de código aberto, que permitem o seu uso, alteração e redistribuição.
- Abstração de sistema de gestão de base de dados: Rails suporta uma grande panóplia de bases de dados, incluindo as mais usadas em ambientes de produção.
- Suporte para Web-services: Rails suporta os mais usados protocolos de Web-services, incluindo XML-RPC, SOAP e WSDL, existindo ainda uma grande flexibilidade ao nível da adição de API's externas, de forma a suportar futuros protocolos.
- Testes funcionais e unitários integrados: As funcionalidades de testes em Rails são simples, bem documentadas, concisas e integradas na *framework*. O Rails constrói as bases para os testes unitários e funcionais, e regista o rácio de testes por código automaticamente. O Rails inclui suporte para testes usando simulação de objectos padrão, e introdução automática de dados em bases de dados, permitindo assim testes automatizados. Como Rails não impõe a existência de um contexto para os objectos, os testes são efectuados ao nível do protocolo HTTP e dentro do browser.
- Produção vs. Testes vs. Desenvolvimento: O Rails tem, por defeito, três ambientes:
  - Production, onde a cache e as "performances enhancements" estão activas, onde é assumido que é código não está a ser desenvolvido, e a informação de desenvolvimento não é passada ao utilizador final.
  - Testing, onde uma base de dados volátil fica disponível com o propósito de correr as suites de testes.
  - Development, onde as "performance enhancements" estão desabilitadas, é usada uma base de dados para desenvolvimento, estando disponíveis ferramentas e feedback adicional, dando ênfase a metodologias de desenvolvimento ágil, expecting zero turnaround time development.
- Escalabilidade horizontal sem partilha de informação: Enquanto que o Rails beneficia da crescente potência dos servidores em que corre, foi desenhado com o conceito de "share nothing" em mente. Cada thread de Rails é auto-contido horizontalmente (i.e. não partilha nada com os outros threads de Rails), e verticalmente (i.e. não partilha nada com o servidor Web ou o servidor de base de dados). Quando se faz o deployment de uma aplicação de Rails, com

vista à escalabilidade, é comum o uso de servidores *lightweight* como é o caso do `lighttpd` e `FastCGI` fornecendo assim uma escalabilidade horizontal rápida e fazendo uso de poucos recursos.

### 2.3.3 Comparação em relação a outras *frameworks*

Existem dezenas de outras *frameworks* e muitas até há muito mais tempo que Rails. De seguida apresentam-se algumas, tendo sido escolhidas as mais usadas ou mais equiparáveis a Rails.

**JAVA (J2EE)** Apesar de Ruby on Rails ser uma *framework* nova e excitante para muitos, o core da arquitectura segue os padrões presentes em J2EE. A filosofia de desenvolvimento de aplicações Web é o que diferencia estas duas *frameworks*. Rails dá primazia a código explícito ao invés de ficheiros de configuração, e a natureza dinâmica do Ruby gera muito código em runtime. Grande parte da *framework* Ruby on Rails foi criada como um projecto único e o desenvolvimento de uma aplicação beneficia de um conjunto de componentes homogéneos. Em contraste, uma stack típica de J2EE tende a ser construída de componentes diferentes (um conjunto de bons componentes entre um grande leque) que são geralmente desenvolvidos independentemente, em que são extensivamente usados ficheiros XML para os ligar e tratar da configuração. Seria, no entanto, um erro dispensar-se completamente J2EE em favor de Rails. J2EE é um standard muito bem estabelecido com inúmeras implementações sólidas. É uma tecnologia com provas dadas e robustez comprovada. No entanto trabalhar com uma *framework* em Java implica conhecer bem muitas das tecnologias envolvidas (Struts, Hibernate, Spring, Axis, etc.) sendo o seu número muito elevado e por vezes bastante complexo, tornando a escolha das tecnologias adequadas a um projecto uma tarefa difícil. Para se atingir um nível de produtividade semelhante ao de outras *frameworks* exige-se assim muito mais prática e habituação. Além de ser necessárias muito mais configurações numa aplicação em J2EE, existe o overhead do processo de compilação e muitas vezes os erros cometidos no código não são detectados na compilação mas apenas no runtime e conseqüentemente será difícil reagir rapidamente a mudanças, característica fundamental numa interface Web. Apesar de tudo os programadores Java têm ao dispor um vasto leque de escolhas. Além de haver um grande número de *frameworks* J2EE, é relativamente fácil fazer integração com uma infinidade de aplicações Java já desenvolvidas para os mais variados propósitos.

**Zope + Plone** Zope (Z Object Publishing Environment) é um servidor de aplicações Web desenvolvido totalmente em Python e é um poderoso ambiente de programação Web, que permite desenvolver aplicações remotamente. As aplicações

de uso do Zope são as mais diversas, porém tem sido utilizado em larga escala por diversas corporações como CMS (Content Management Systems). As aplicações criadas no Zope são chamadas de "Produtos", sendo Plone um dos mais usados em aplicações Web. Plone é uma *framework* de gestão de conteúdo profissional. O Plone é sem dúvida, um dos melhores CMS Open Source do mercado, sendo utilizado pela NASA, IDGnow, AOL entre outros. A combinação Zope/Plone constitui uma *framework* muito sólida e permite construção de aplicações Web com um esforço mínimo. No entanto foca-se no desenvolvimento de aplicações típicas de gestão por um CMS, não sendo fácil a implementação de aplicações de outro carácter.

**Django** Django é uma *framework* Open Source desenvolvida em Python, que segue os padrões do modelo MVC de um modo não restritivo. O objectivo principal é facilitar a criação de Websites complexos e com muitos acessos a uma base de dados de forma rápida e simples. Foca-se na reutilização e ligação entre os diversos componentes bem como no desenvolvimento ágil e o princípio DRY (Don't Repeat Yourself). A linguagem Python é usada de modo transversal, da mesma forma que Ruby é usado na *framework* Ruby on Rails, sendo uma linguagem simples mas menos específica. Com uma interface de administração gerada automaticamente e com autenticação integrada, os projectos em Django estão mais virados para CMS sendo menos flexível do que Rails.

**TurboGears** TurboGears é mais uma *framework* de desenvolvimento Web em Python, em tudo semelhante a Django. Tem uma grande comunidade e conta já com um livro inteiramente dedicado ao desenvolvimento nesta *framework* e com algumas aplicações a correr no mundo real. É uma *framework* mais flexível do que Django, com melhor suporte para AJAX, não estando tão especializada em CMS, mas por essa razão também é mais complexa.

**Frameworks em PHP** PHP é uma linguagem muito usada para criação de aplicações Web, tendo sido criada especificamente para esse efeito. Como tal, não seria de estranhar que surgissem *frameworks* de desenvolvimento nesta linguagem. Na verdade existem dezenas de *frameworks* em PHP, que facilitam em muito o desenvolvimento de aplicações com esta linguagem. No entanto não resolvem o principal problema, razão que leva a escolher *frameworks* noutras linguagens – o código PHP tende a ficar muito confuso com o evoluir da aplicação. Além disso são pobres em ferramentas adicionais como consola, migrações e mecanismo de testes. Não deixam de ser uma boa solução para casos em que o servidor não permite usar outras linguagens menos comuns, uma vez que PHP foi desde a sua invenção uma das linguagens mais usadas para aplicações Web e por isso está muito bem disseminada.

## 2.4 Web Services

Web Service, segundo a definição da W3C, é descrito como "um sistema concebido para apoiar software interoperável de interação numa rede" [W3C02]. Web services são frequentemente apenas APIs Web que podem ser acedidos através de uma rede, tais como a Internet, e executados num sistema remoto que hospeda os serviços requeridos.

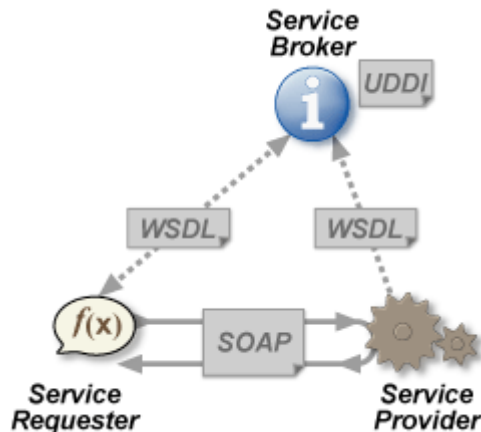


Figura 2.7: Arquitectura Web Service [Wik08b]

A definição de Web Service da W3C engloba muitos sistemas diferentes, mas em uso comum, o termo refere-se aos clientes e servidores que comunicam através de mensagens XML que seguem o standard SOAP. Nesses sistemas, existem frequentemente descrições legíveis por máquina das operações oferecidas pelo serviço, escritas nos Web Services Description Language (WSDL). Este último não é uma exigência de um endpoint SOAP, mas é uma condição prévia para geração de código do lado do cliente automatizados em muitas frameworks Java e .NET SOAP.

## 2.5 IP Multimedia Subsystem (IMS)

O IP Multimedia Subsystem (IMS) [GC06, JB04, Cha07] define uma arquitetura genérica para a oferta de Voice over IP (VoIP) e serviços multimédia.

O IMS foi especificado pela primeira vez pela Third Generation Partnership Project (3GPP/3GPP2) estando agora associada à ETSI/TISPAN que normaliza as redes convergentes que usam o IMS e a sua arquitectura core. A norma a partir da release 6 da especificação 3GPP definiu o IMS como sendo independente do tipo de acesso, suportando assim qualquer tipo de acesso como por exemplo: GSM, WCDMA, CDMA2000, DSL, WiMAX, GPRS e WLAN.

## Revisão Bibliográfica

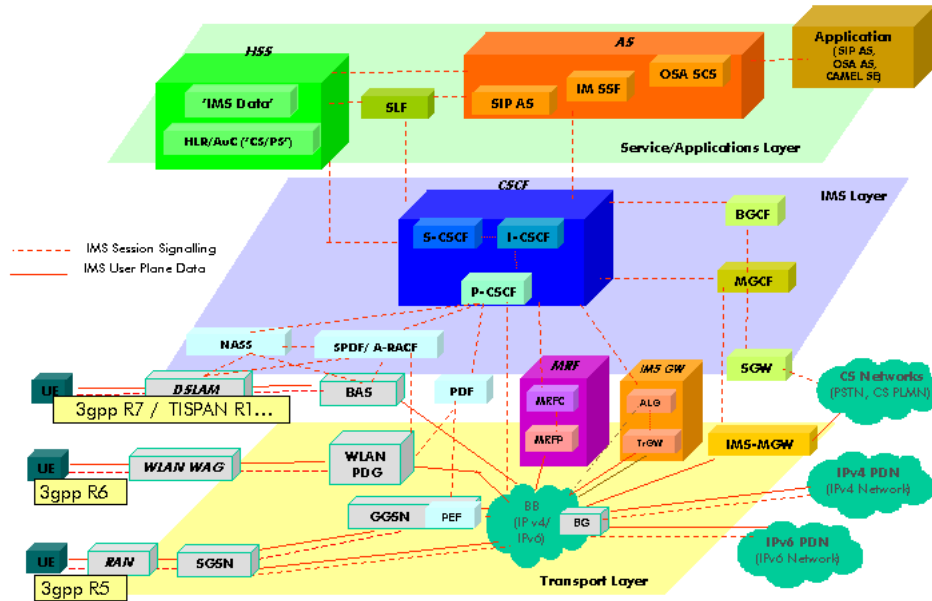


Figura 2.8: Arquitectura IMS [Wik08a]

Um aspecto importante para se compreender a arquitectura IMS é o facto de o 3GPP não normalizar nós, mas sim funções. Isto significa que a arquitectura IMS é um conjunto de funções ligadas por interfaces. Quem implementa é livre de juntar mais que uma função num só nó, assim como dividir uma função por vários nós. Como podemos ver na figura 2.8 este conjunto de funções lógicas pode ser dividido em 3 camadas: a camada de transporte, a de controlo de sessão e a de aplicação. Também podemos observar os seguintes nós:

- Uma ou mais bases de dados de utilizador, ditas HSSs (Home Subscriber Servers) e SLFs (Subscriber Location Functions);
- Um ou mais servidores SIP, os CSCFs (Call/Session Control Functions);
- Um ou mais ASs (Application Servers);
- Um ou mais MRFs (Media Resource Functions), cada um dividido em MRFC (Media Resource Function Controllers) e MRFP (Media Resource Function Processor);
- Um ou mais BGCFs (Breakout Gateway Control Functions);

- Uma ou mais PSTN (Public Switch Telephony Network) gateways, cada uma dividida num SGW (Signaling Gateway), num MGCF (Media Gateway Controller Function) e num MGW (Media Gateway).

## 2.6 Jain SLEE

Lançada pela SUN em 1998, a iniciativa JAIN estende a plataforma JAVA ao mundo das Comunicações.

O JAIN é um conjunto de APIs (Application Programming Interfaces) de JAVA que permitem desenvolver serviços de forma simples, económica e rápida, independentemente da tecnologia da rede utilizada.

O JSLEE é uma norma JAIN orientada para ambientes de execução de aplicações da área de Comunicações, sendo a plataforma de execução de serviços (ou SLEE – Service Logic Execution Environment), e disponibiliza um ambiente para o provisionamento e execução dos serviços.

Permite ao criador de serviços uma abstracção em relação às infra-estruturas e aos protocolos de sinalizações utilizados, ou seja, os serviços são totalmente independentes das mudanças da rede e dos protocolos utilizados.

O JSLEE [SM04] possui um modelo constituído por componentes otimizados para eventos e é projectado para as exigências de processamento de eventos (quase) em tempo real. Este modelo permite estruturar a lógica de aplicações em conjuntos reutilizáveis de componentes orientados para objectos e a composição desses componentes em serviços mais sofisticados e de mais alto nível.

Os objectivos do JSLEE são os seguintes:

- Portabilidade de serviços (write once, run anywhere) – oferece uma independência total em relação às interfaces do sistema, permitindo assim reutilizar os serviços para todos os tipos de redes;
- Independência da rede (on any network) – ao possibilitar o desenvolvimento de serviços independentemente das redes, possibilita a fusão de redes habitualmente incompatíveis;
- A segurança de acesso (by anyone) – a segurança oferecida pelo JSLEE autoriza acessos à rede interna por pessoas exteriores;
- Flexibilidade e disponibilidade do Java para criar novas aplicações;
- Ambiente apropriado para correr serviços de Telecomunicações;
- "High throughput";
- Latência baixa – orientado a eventos (comunicações assíncronas);

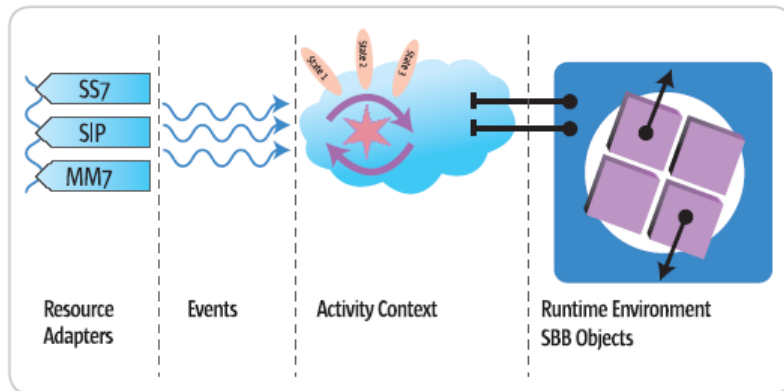


Figura 2.9: Os quatro elementos básicos do JSLEE [Hai04]

- Orientado ao modelo de estados;
- Fácil integração de adaptadores baseados em JAIN;
- Arquitectura flexível e distribuída;
- Escalabilidade Horizontal e Vertical.

O JSLEE é constituído por quatro elementos básicos:

- Resource Adapters (RA): são entidades externas ao SLEE (SIP Proxies, Bases de Dados, etc.) que são potenciais geradores de eventos a consumir por entidades do SLEE. Um RA adapta um recurso para que possa ser usado pelos serviços do JSLEE. Ao receber uma mensagem de uma identidade externa, o RA submete um evento ao SLEE. O SLEE irá rotar o evento pelos SBB interessados em o receber. Por outro lado, um SBB pode invocar um método no RA;
- Eventos: um evento representa uma ocorrência sujeita a processamento aplicativo. Os eventos podem ser originados por diversas fontes como os RAs, pelo próprio contentor SLEE, ou por outro componente SBB. Os serviços SLEE são activados por eventos (por exemplo o evento resultante do envio da mensagem “SIP INVITE” a pedir o estabelecimento duma sessão multimédia entre duas entidades);
- Activity Contexts: é uma entidade lógica do SLEE que recebe e faz o roteamento de eventos (através do Event Router) para os componentes do SBB. As actividades são abstracções de um fluxo de eventos relacionados e emitidos pelo recurso. O RA disponibiliza aos SBB interessados uma representação Java desta actividade (Objecto Actividade);

- Ambiente de “runtime”: hospeda os objectos SBB (Service Building Block). O SBB é um componente reutilizável de software que envia e recebe eventos e, com base no seu estado e nos eventos recebidos, executa uma determinada lógica computacional. Um SBB é um componente com estado e, como tal, pode lembrar os resultados de computações anteriores e usá-los em computações posteriores. Um SBB pode ser composto por outros SBBs para a construção de aplicações mais complexas com base em SBBs existentes. Um serviço é então um conjunto de SBBs que comunicam entre si.

## 2.7 SIP Servlets

Uma servlet é basicamente uma classe Java (aplicação) que permite a extensão das potencialidades de um servidor que contém aplicações acedidas no modelo pedido-resposta. No fundo estão para o servidor como as applets estão para o cliente.

A SIP Servlet API [Kri03] surgiu graças ao Java Specification *Request 116* com o objectivo de definir uma API de alto nível para servidores SIP, permitindo que aplicações SIP sejam instaladas e geridas segundo um modelo de servlet e, tendo por objectivo ser a chave para a standardização de uma plataforma que permita a distribuição de serviços SIP.

Devido às diferenças entre SIP e HTTP, quer na sua natureza quer na sua aplicabilidade, a API de SIP Servlet à parte das capacidades comuns de resposta a pedidos, oferece as seguintes capacidades:

- Request initiation;
- Request and Response reception;
- Request Proxying;
- Response sequencing.

As aplicações de SIP Servlet necessitam de registar o seu interesse nos eventos para os quais querem ser notificados. O método “Service” na interface “javax.servlet.Servlet” é responsável pela entrega desses eventos às aplicações.

## 2.8 Resumo

As tecnologias a considerar neste projecto foram várias. Desde a informação sobre o projecto a ser desenvolvido na PT Inovação (shipnet), passando pelo conceito IMS, Web 2.0, ou tecnologias como Ruby on Rails, podemos concluir que este estágio teve uma grande abordagem de diversas tecnologias.

## Revisão Bibliográfica

Este capítulo teve como principal objectivo apresentar as tecnologias envolvidas no projecto, de forma a introduzir os principais conceitos necessários para a compreensão dos restantes capítulos do documento.

## Capítulo 3

# Redes e Serviços VoIP/3G e Web 2.0

A PT Inovação, ciente dos desafios que se avizinham na área da convergência fixo-móvel, iniciou em 2005 o programa estratégico shipnet - Service Handling on IP Networks, com o objectivo primordial de consolidar a liderança como fornecedor de sistemas e plataformas para a criação, provisão e gestão de serviços e aplicações em redes *All-IP* convergentes.

O Laboratório shipnet IMS da PT Inovação disponibiliza um core IMS que integra as diversas plataformas e protótipos desenvolvidos. O shipnet IMS Lab opera com níveis elevados de interfuncionamento com as a redes ADSL, Wi-Fi ou redes móveis GPRS e UMTS.

Realizado na PT Inovação, e inserido neste contexto, o trabalho realizado fez parte do desenvolvimento do projecto shipnet.

Dividido em duas fases, e com uma duração aproximada de 20 semanas, o trabalho esteve orientado para as camadas de interface e serviços. De seguida são descritas com maior pormenor cada uma destas fases.

### 3.1 Conceitos

Nesta secção são listados os conceitos do sistema, utilizados ao longo deste relatório.

- **Área:** identificada por uma frase, que identifica uma empresa ou uma organização dentro de uma empresa. São Áreas todos os conjuntos de Utilizadores aos quais se aplicam as mesmas regras, restrições e se fornecem as mesmas regalias e os mesmos acessos a funcionalidades específicas. São também Áreas

as agregações por Sítio ou com base na organização de uma empresa (i.e. Departamentos).

- **Funcionalidade:** privilégio que pode ser delegado entre Níveis de Áreas que tem como finalidade a disponibilização de um serviço de valor acrescentado à entidade Utilizador - que usufrui do sistema. As Funcionalidades são, desta forma, atributos das Áreas e Utilizadores.
- **Grupo:** referido no documento de Definição de Requisitos como sinónimo de Área. Durante o desenvolver do projecto, a definição foi alterada, significando Grupo um subdomínio de Área, incluindo apenas Sítio e Departamento enquanto que Área engloba também a agregação de Empresa.
- **Numeração Interna/Extensão:** são números de cada Grupo Cliente (Nível 1), que podem ser atribuídos a Grupos e Utilizadores.
- **Numeração Externa:** são números através dos quais se pode chegar da Rede Externa ao Sistema a um Utilizador de uma qualquer Área. Estes podem ser atribuídos a Grupos e Utilizadores.
- **Perfil:** para facilitar o uso do Sistema por parte do Utilizador, é fornecida a possibilidade de cada utilizador configurar (por exemplo até 3) perfis diferentes de utilização das funcionalidades que detém. O Utilizador só poderá ter um perfil activo em cada momento e terá a hipótese de alternar entre perfis sempre que quiser.
- **Restrição:** limitação imposta pelos Grupos de Nível N-1 aos Grupos de Nível N, por eles criados. Estas têm como fim a restrição dos recursos consumidos por cada Grupo, Utilizador, assim como a restrição de acessos às funcionalidades por parte dos Grupos ou dos Utilizadores.
- **Terminal:** entidade que pode estar directamente associada a um ou mais Utilizadores ou a um ou mais Grupos. O Terminal estará sempre associado a uma ou mais extensões, podendo estar associado a zero ou mais Números Externos. O Terminal é o elemento VoIP do sistema, que só por si disponibiliza aos Utilizadores funcionalidades base.
- **Utilizador:** entidade que realmente usufrui dos serviços. A um Utilizador pode ser atribuído a gestão de um Grupo. Caso contrário, não poderá criar novos Grupos e só tem acesso à configuração de funcionalidades/características pessoais. A função de Delegação é então restrita à entidade Grupo. Um Utilizador é uma entidade que poderá gerir um ou mais Grupos, poderá pertencer a um ou mais Grupos e possuir uma ou mais extensões e Números Externos.

### 3.2 Camada de Interface

O ip-Centrex assenta na transferência da rede para IMS. As empresas deixam de ter de adquirir uma PBX (central telefónica), e a gestão dos contactos, chamadas e operações de voz é mantida numa interface web.

Com esta funcionalidade do IMS, pretende-se que o cliente possa gerir os serviços disponibilizados para os Recursos Humanos da sua empresa - portal de administração - assim como que seja possível para cada uma das pessoas que tem acesso ao sistema, activar e configurar os serviços que tem disponíveis - portal Self-Care.

A prossecução desta primeira fase do projecto de estágio, decorre de um requisito do sistema que define a necessidade de implementação do Portal de Administração para o projecto ip-Centrex, para que seja possível por parte do cliente, a administração dos serviços associados à sua empresa, monitorização de todas as operações de voz realizadas, entre outras.

Aquando do início do projecto, o portal de Self-Care já se encontrava em fase avançada de desenvolvimento. Uma vez que o portal de administração será integrado juntamente com a parte de Self-Care, terá que ser tida em consideração tanto o seu layout e design, assim como o modo de interação com o utilizador.



Figura 3.1: Print Screen do Portal de Self-Care

Na figura 3.1, são descritos alguns dos pontos mais importantes da interface. As opções, listadas no menu principal são:

**O meu perfil** Onde é apresentada informação do utilizador (login, nome, etc.), e das suas extensões (SIP-URI, número curto, *alias*, etc.).

**Serviços** Activação das funcionalidades disponibilizadas pelo sistema. São apresentadas em dois grupos: "Quando me ligam", e "Quando efectuo chamadas".

**Configuração** Tab em que são configuradas as diversas funcionalidades activadas na opção anterior.

**Directórios** Local onde são listados os contactos da empresa. Aqui é possível pesquisar os contactos de todos os utilizadores da empresa, sendo possível despoletar o serviço "Click to Call", em que é automaticamente estabelecida uma chamada entre o terminal do utilizador e o contacto seleccionado.

**Relatórios** Informação sobre chamadas perdidas, efectuadas, recebidas e reenca-minhadas.

Introduzido o Self-Care, os objectivos específicos desta fase de trabalho são os seguintes:

- Diferenciar o acesso à parte de Administração para dois tipos de utilizadores (sendo que o utilizador "normal" não tem acesso à parte de administração):
  - Administrador: Utilizador que tem acesso à gestão do grupo raiz do sistema, e que estende as funcionalidades do Gestor.
  - Gestor: Utilizador que tem acesso à gestão de uma entidade "Área", e que por isso controla os privilégios e recursos a ela associados ou atribuídos.
- Implementar um conjunto de funcionalidades:
  - Possibilidade de escolha da língua Portuguesa ou Inglesa.
  - Existência de "Grupos de Utilizadores" que estarão disposto em árvore, com características de Nome, Descrição e Tipo.
  - Possibilidade de Criação, Edição e Eliminação de Grupos.
  - Possibilidade de escolher as funcionalidades a que o Grupo terá acesso, considerando as restrições que o seu grupo pai lhe impõe.
  - Possibilidade de Criação, Edição e Eliminação de Utilizadores, organizados por Grupos.
  - Possibilidade de atribuição do papel de Gestor de grupos a Utilizadores.
  - Possibilidade de escolher as funcionalidades a que o Utilizador terá acesso, considerando as restrições que o grupo a que pertence lhe impõe.
- Elaborar Manual de Administração da plataforma

A análise dos requisitos foi então uma das tarefas realizadas inicialmente. O documento foi fornecido ao grupo de trabalho pela PT Prime.

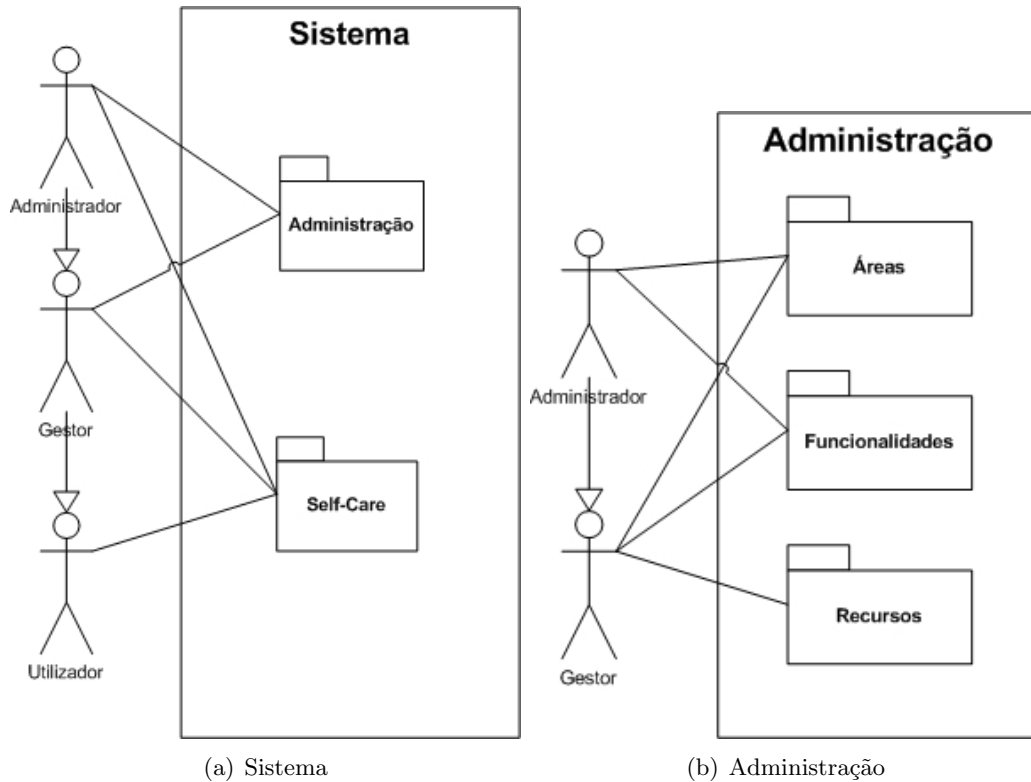


Figura 3.2: Casos de Uso

Na figura 3.2(a) é apresentado o sistema geral da aplicação Web ip-Centrex. Nesta fase do projecto, apenas a parte de Administração é abordada, já que a parte de Self-Care foi implementada por um outro colega do grupo de trabalho.

Como podemos observar no diagrama, existe um actor do sistema - o utilizador - que não vai interagir com a parte do Portal de Administração. Por este motivo, a parte de Administração vai apenas ter como actores o Gestor e o Administrador [Fow03].

Pelo facto de a aplicação Web ser apresentada ao cliente como um todo, o layout/design e o modo de interacção com o utilizador tiveram que ser tidos em conta para a implementação da parte de Administração - figura 3.2(b). Algumas funcionalidades do sistema são transversais a toda a aplicação. São disso exemplo a autenticação, o menu, a língua em que se apresenta a interface, entre outras.

Focando agora a parte em que esta fase do projecto se centrou, o portal de Administração vai estar dividido, no que diz respeito aos casos de uso, em três áreas principais: Áreas, Funcionalidades e Recursos.

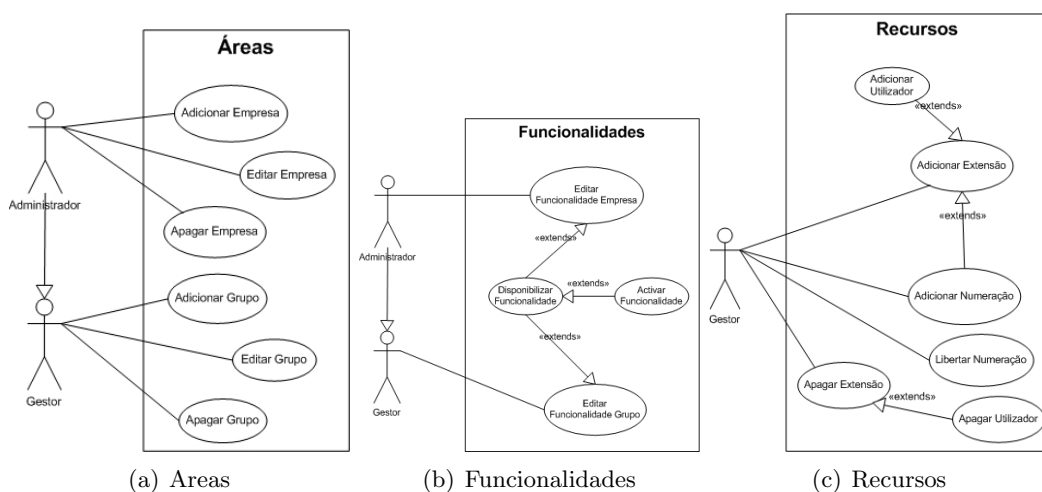


Figura 3.3: Casos de Uso - Administração

Tal como já foi referido, nesta aplicação, teremos dois tipos de actores: Gestor e Administrador.

O actor que representa o lado da empresa é intitulado de Gestor. Este actor terá a capacidade de configurar as funcionalidades (e.g. reencaminhamentos, barra-mentos) associadas à empresa a que pertence, disponibilizar ou retirar permissões a utilizadores, adicionar ou remover áreas e utilizadores. Resumindo, um Gestor é um actor com capacidade de gestão da área (i.e. empresa, sítio ou departamento) a que pertence.

O Administrador, é o actor do sistema que representa o lado do operador que gera a solução (e.g. PT, Optimus, Novis), e que por isso, tem a possibilidade de criação, edição e remoção de empresas, para além de ter acesso a todas as funcionalidades disponibilizadas ao gestor - Figura 3.3.

Antes de descrever os casos de uso das áreas (figura 3.3(a)) é preciso introduzir os termos usados para este esquema. São áreas (designadas como "Grupos" na especificação de requisito - a modificação de nomenclatura está associada a um pedido por parte do cliente durante a implementação, uma vez que no seu parecer, a nomenclatura inicial iria suscitar alguma confusão no utilizador) todos os conjuntos de Utilizadores aos quais se aplicam as mesmas regras, restrições e se fornecem as mesmas regalias e os mesmos acessos a funcionalidades específicas.

Existem três tipos de áreas: empresas e grupos (são considerados "grupos" as áreas do tipo sítio e departamento, mas para o sistema, estas são consideradas de igual forma).

A descrição dos casos de uso das áreas (Figura 3.3(a)), pode ser resumida em adicionar, editar e apagar grupos e empresas, sendo que o gestor apenas tem acesso

a "gerir" grupos que lhe estejam associados.

No que diz respeito às funcionalidades, os casos de uso estão descritos na figura 3.3(b). À semelhança dos casos de uso das áreas, o actor Administrador difere do Gestor pelo facto de ter permissões de gestão de empresas, enquanto que os gestores apenas têm acesso a gerir os grupos aos quais estão associados. As duas acções possíveis são activação e disponibilização das funcionalidades (e.g. reencaminhamentos, barramentos, permissões).

Por último, os casos de uso dos recursos são apresentados na figura 3.3(c). Neste caso, tanto o gestor como o Administrador têm acesso aos mesmos casos de uso (uma vez que o Administrador é uma extensão ao Gestor). Ambos os actores têm a possibilidade proceder às seguintes acções, nas áreas respectivas:

- adicionar extensão: criação de um número para um utilizador. Isto implica:
  - a existência de uma área onde a extensão será criada;
  - a associação de um utilizador da empresa, para a extensão, ou a possibilidade da sua criação (ver "adicionar utilizador");
  - existência de numeração externa associada à área (ver "adicionar numeração").
- adicionar utilizador: funcionalidade que estende a adição de extensão, não sendo possível aceder a esta acção de outra forma. Isto acontece, porque o sistema só permite a existência de utilizadores com extensões associadas.
- adicionar numeração: atribuição de numeração externa a uma área. Esta acção segue um esquema hierárquico, em que apenas pode ser adicionada a numeração disponível na área acima da actualmente seleccionada.
- apagar extensão: acção de remover uma extensão do sistema. No caso particular de o utilizador não ter mais extensões, também ele é removido, já que dentro do sistema essa situação não seria permitida.
- libertar numeração: acção em que a numeração é devolvida à área acima da actualmente seleccionada. Isto permite que a numeração possa estar reservada a uma área.
- apagar utilizador: acção de remover utilizador do sistema, pelo mesmo motivo pelo qual a criação de utilizador não é directamente acessível ao utilizador, também esta remoção não o é.

### 3.2.1 Melhoria do Portal de Administração

Terminada a fase de desenvolvimento do Portal de Administração, e consequente fecho da primeira versão do projecto, estarei responsável pelo acompanhamento da seguinte versão do portal.

Nesta fase, tal como foi dito anteriormente, os objectivos foram extendidos, tendo-se também considerado a seguinte lista de objectivos:

- Inventário dos terminais, com a descrição das suas características, e funcionalidades. Deve para além disso conter o número de terminais activos de cada modelo.
- Visualizar toda a informação de Detalhes de Chamadas para um dado utilizador.
- Visualizar toda a informação de Detalhes de Chamadas já definida, para toda a árvore com raiz no grupo que a visualiza.
- Provisão automática dos terminais e das configurações de utilizadores, por parte do cliente.

Mais uma vez, esta listagem é apenas um resumo dos objectivos pretendidos para o sistema de Administração, a versão completa dos requisitos pode ser encontrada em anexo.

Da análise dos requisitos foram produzidos novos diagramas de casos de uso.

A representação do sistema (figura 3.4(a)) é igual à que foi produzida anteriormente, dividido em duas áreas distintas: Administração e Self-Care. Mais uma vez, e tal como no caso anterior, apenas vamos abordar a parte do sistema que diz respeito à Administração (figura 3.4(b)). A grande diferença que se encontra nesta parte da representação, é uma adição de três áreas: Utilizadores, Perfis e Relatórios.

No caso das áreas (figura 3.5(a)), a alteração dos casos de uso foram apenas no aspecto de tornar o interface mais fácil e intuitiva para o utilizador. Para isso foram extendidos os casos de uso de adicionar empresa e grupo com:

- disponibilização de funcionalidades: configurar quais as funcionalidades disponíveis para a área a criar, assim como configurar a sua activação e valores por defeito.
- adição de numeração interna: adicionar gamas ou valores soltos de numeração interna para a área a criar. Esta acção será também posteriormente disponibilizada ao gestor da área para ele próprio poder configurar qual a numeração interna que quer para o grupo. Uma outra particularidade da numeração interna é que esta será única para cada empresa (e.g. apenas pode existir uma extensão "101", dentro da uma empresa e respectivos grupos filho).

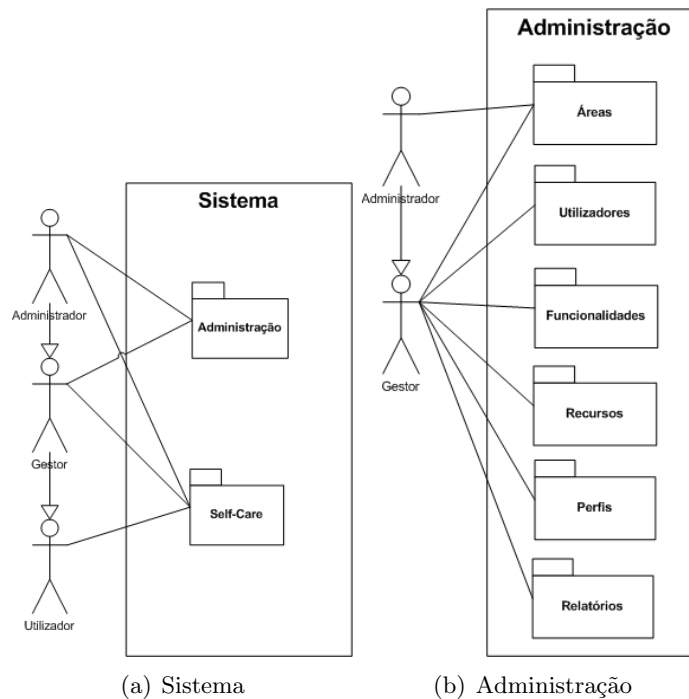


Figura 3.4: Casos de Uso

- adição de terminais: adicionar terminais a uma área, com as suas características (i.e. MAC, Marca, Modelo, IP, etc.). Esta necessidade tem por base facilitar o sistema de provisão e facilitar o controlo dos terminais.
- atribuição de gestor: adicionar um utilizador à área a ser criada. Esta acção tornou-se apenas possível nesta segunda versão do portal, através da alteração do modelo da base de dados com a ligação entre um utilizador e a empresa, sem ser para isso necessário existir uma extensão a liga-los.

Este "wizard" permite que a adição de uma área seja mais simples para o utilizador, uma vez que este consegue num passo apenas, configurar vários parâmetros da área a ser criada.

A secção de Utilizadores (figura 3.5(b)) foi adicionada nesta versão, para facilitar o controlo dos utilizadores. Para isso, contribuiu em muito a alteração do modelo da base de dados, com a ligação directa entre utilizador e empresa, não sendo obrigatória a existência de uma extensão para os associar.

Uma particularidade desta secção é que estará associada sempre a uma empresa (e não outra área) uma vez que os utilizadores estão ligados a empresas. Os casos de uso resumem-se à gestão de utilizadores (adicionar, editar, remover), atribuir numeração a um utilizador (tanto a numeração interna, como externa), gerir funcionalidades (configurar quais as funcionalidades a que o utilizador tem acesso), delegar

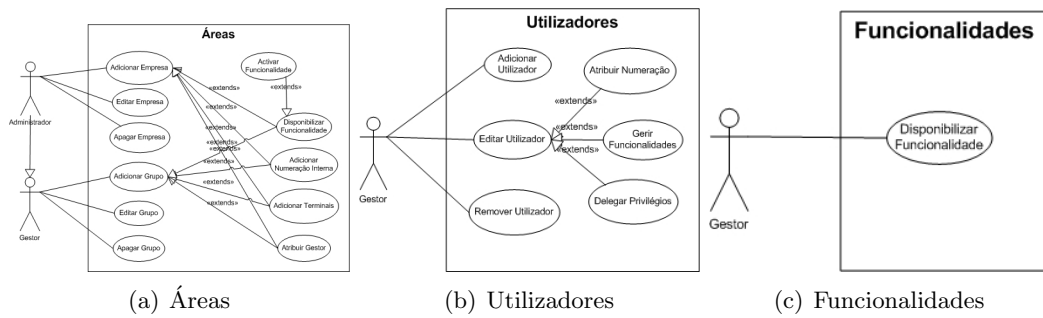


Figura 3.5: Casos de Uso - Administração

privilégios (definir quais os privilégios do utilizador sobre uma área (i.e. gestão de utilizadores, áreas ou privilégios)).

As Funcionalidades (figura 3.5(c)) foram simplificadas para um único caso de uso: o da sua disponibilização. Esta alteração deveu-se à existência de um novo conceito de perfil (ver mais à frente), em que as funcionalidades de cada área/utilizador são editadas.

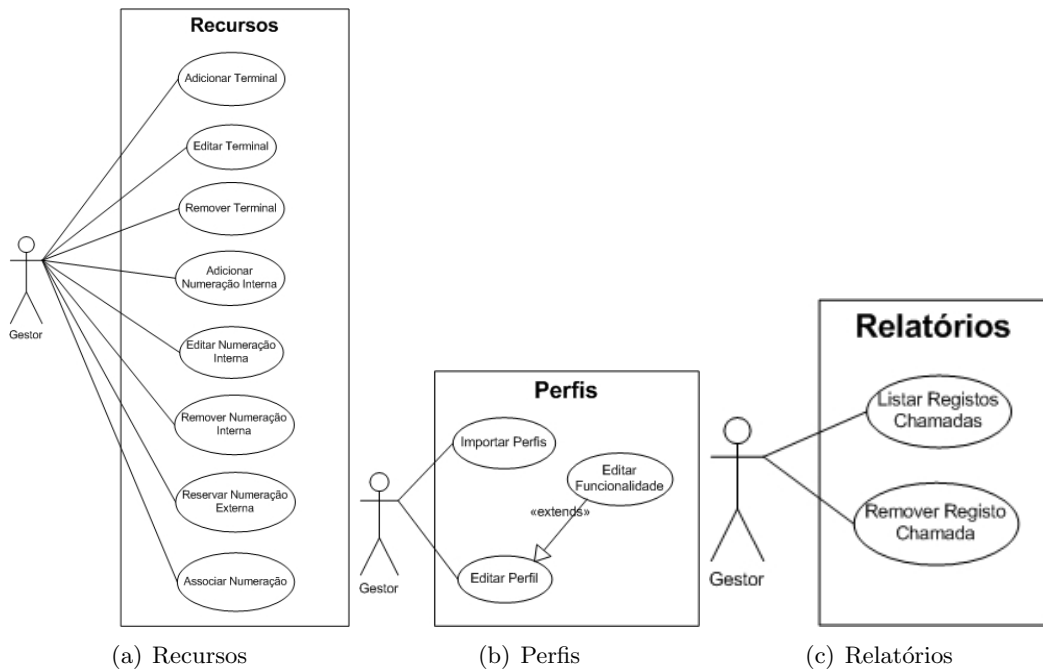


Figura 3.6: Casos de Uso - Administração

Aos casos de uso dos recursos foram adicionados os terminais (parte que não tinha sido implementada na versão anterior). Os casos de uso encontram-se representados na figura 3.6(a), e são descritos como:

- Gestão de terminais;

- Gestão de numeração interna;
- Reservar numeração externa;
- Associar numeração.

Na figura 3.6(b) são apresentados os casos de uso dos Perfis. Este foi um novo conceito introduzido nesta segunda versão do projecto, que pode ser descrito como um "template" de configuração das funcionalidades para uma dada área ou utilizador. No caso particular dos perfis associados às áreas, estes têm por objectivo funcionar como templates para os perfis dos utilizadores. No caso dos perfis dos utilizadores, estes funcionam como uma forma rápida de os utilizadores mudarem as funcionalidades que têm activas (assim como a configuração das mesmas).

Na secção dos relatórios (figura 3.6(c)) encontram-se os casos de uso de: listar registo de chamadas e remover registo de chamada. Nesta secção é possível visualizar o histórico de chamadas das numerações/utilizadores dentro de uma mesma área.

### 3.2.2 Modelo de Dados

O HSS (Home Subscriber Server) é a base de dados principal com a informação relacionada com o utilizador. Esta informação está relacionada com a identidade dos utilizadores, a sua localização e informação relativa à subscrição de serviços. Toda a informação relativa aos utilizadores ip-Centrex e aos serviços utilizados estão nesta base de dados de forma a ser possível aos membros do sistema configurar dados relativos aos serviços subscritos. A base de dados tanto é acedida pelo portal do utilizador como pelos serviços em si.

No Anexo D podemos ver o Diagrama Relacional da primeira versão da Base de Dados.

Para a segunda versão do Website, foi elaborada uma remodelação do Modelo de Dados, diminuindo a sua complexidade, e possibilitando maior flexibilidade de funcionalidades por parte do portal (e.g. ligação entre Utilizadores e Empresas, por forma a não ser necessária a criação de uma extensão para os "ligar"). No Anexo D está também representado o Diagrama Relacional da segunda versão da Base de Dados.

## 3.3 Camada de Serviços

Na última fase do estágio, e em paralelo com a tarefa de desenvolvimento da segunda versão do Portal de Administração, será elaborado um estudo comparativo entre as tecnologias SIP. A necessidade deste estudo deve-se principalmente ao facto de as tecnologias desta área estarem a evoluir de forma constante, sendo necessário

conhecer as alternativas existentes, e perceber as suas vantagens e/ou desvantagens em relação à escolha actual. Um outro motivo que leva à necessidade deste estudo, é o facto de durante o desenvolvimento do projecto shipnet (iniciado em 2005), os requisitos terem sido alterados por diversas vezes, o que faz com que alguns dos factores de decisão possam ter sido alterados no entretanto.

Resumindo, o objectivo deste trabalho consiste na elaboração de um estudo comparativo entre as tecnologias Jain Slee e SIP Servlets. Para esta análise será feito um estudo teórico das duas tecnologias, seguido da aplicação de um exemplo prático do serviço "Click to Call".

Este serviço é disponibilizado ao utilizador quando este acede ao seu Directório de Contactos no portal de Self-Care. Aí são-lhe apresentados links para fazer chamadas para os contactos listados. O sistema funciona através da chamada do serviço enviando uma mensagem SOAP/HTTP para o servidor (Controller), este faz em primeiro lugar uma chamada para a pessoa que pediu o serviço. Quando esta atende, o sistema faz uma outra chamada para o destinatário, estabelecendo assim a comunicação entre os dois.

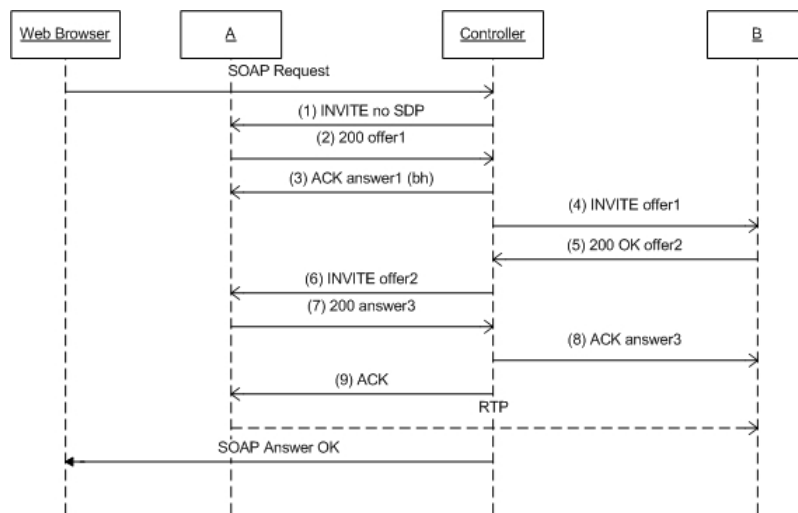


Figura 3.7: Click-to-Call - Diagrama de Sequência

O diagrama da figura 3.7 apresenta a sequência de sinalização dos eventos, com base na definição feita no documento de especificação [Ros04].

### 3.4 Resumo e Conclusões

Inserido no projecto shipnet, este estágio teve como base as camadas de interface e de serviços.

Tendo-se iniciado pela camada de interface, esta fase mostrou ter um elevado grau de dificuldade, dado o vasto número de requisitos associados e a quantidade de tecnologias envolvidas. Por este motivo os objectivos da implementação do Portal foram filtrados, apenas considerando-se os requisitos de maior prioridade. Já na segunda versão, em que se elaborou um "wireframe" do sistema, foram considerados todos os requisitos.

Para a camada de serviços foi necessário um estudo alargado de diversas tecnologias e conceitos, para além da aplicação prática de um exemplo (no caso, o serviço "Click to Call").

Os dois capítulos seguintes analisam a implementação do trabalho descrito neste capítulo, e apresentam os resultados da análise das tecnologias Jain SLEE e SIP Servlets.



## Capítulo 4

# Implementação

Este capítulo é dedicado à apresentação de detalhes de nível mais baixo relacionados com o enquadramento e implementação das soluções preconizadas no capítulo anterior.

### 4.1 Camada de Interface

Tal como já foi referido, nesta parte do trabalho que acentou sobre a camada de interface, foi necessário desenvolver um Portal de Administração para o ip-Centrex.



Figura 4.1: Portal de Administração

O Portal de Administração é obtido através do clique na opção do menu principal "Áreas" (figura 4.1). Toda a parte de Administração do Portal encontra-se nesta "tab". Dentro, é apresentado um submenu com as opções "Áreas e Extensões", "Numeração" e "Funcionalidades".

Dentro deste submenu são apresentadas as áreas e extensões, organizadas em árvore (figura 4.2(a)). Para termos acesso às acções disponíveis sobre estes dois tipos de entidades, precisamos de clicar no botão direito do rato sobre as mesmas. No caso da área é-nos apresentado um menu (figura 4.3(a)) com as seguintes opções:

## Implementação



Figura 4.2: Áreas e Extensões

- Detalhes: onde se pode ver os detalhes da área seleccionada (descrição/tipo, nome e domínio), e onde é possível alterar o campo nome.
- Inserir Área: acção para inserção de uma nova área dentro da área seleccionada.
- Apagar Área: acção de apagar uma área e respectivas áreas descendentes.
- Inserir Extensão: adicionar uma extensão na área seleccionada.

Para cada uma das acções é apresentada uma "modal window" (figura 4.2(b)) com a informação e/ou formulários associados.



Figura 4.3: Menus Áreas e Extensões

O outro menu existente diz respeito às extensões (figura 4.3(b)) e apresenta as seguintes opções:

- Detalhes: onde se pode ver os detalhes relativos à extensão seleccionada. Estes detalhes englobam a informação do Utilizador da Extensão, o SIP-URI associado à extensão, o seu "short number", e a informação sobre se esta extensão é ou não confidencial, sendo possível editar esta última informação.

## Implementação

- Funcionalidades: local onde são listadas as funcionalidades do utilizador, assim como a informação de quais as que estão disponíveis para uso, e dessas, quais as que estão activas. Neste menu é possível alterar o estado das funcionalidades entre activo e inactivo.
- Utilizador: apresenta a informação do utilizador (primeiro nome, último nome, "username" e email), e permite a geração de uma nova password para o utilizador.
- Apagar: apaga a extensão seleccionada. No caso de o utilizador associado à extensão onde se procedeu à acção, não ter mais nenhuma extensão associada, o utilizador é automaticamente apagado, uma vez que não faz sentido no sistema utilizadores sem extensões associadas.

Estas funcionalidades utilizadas no Portal são possíveis através da utilização de bibliotecas de *JavaScript* [Zak05], alteradas de forma a irem de encontro com as necessidades do portal.



Figura 4.4: Numeração

Na segunda opção - "Numeração" - é apresentada agora uma estrutura hierárquica das Áreas e Numerações. Tendo por base o mesmo princípio de clique com o botão direito sobre as áreas ou numerações, é possível realizar as seguintes acções:

- Adicionar Numeração: Em que são adicionados números à área seleccionada para realizar esta acção. Os números são "retirados" da área "pai", pelo que este tem que ter numeração livre para que a acção se realize. Esta acção é necessária para adicionar uma extensão à área, já que a extensão tem que ter por base um número para a criação da extensão.
- Libertar Numeração: Com esta acção é possível libertar um número. No caso de o número estar associado a uma extensão, a extensão é apagada, e o número fica livre para uma nova utilização. No caso de o número já estar livre, a numeração é libertada para a área acima da que se encontra.

## Implementação

Áreas e Extensões			Numeração			Funcionalidades		
Admin >> BES								
Serviços genéricos								
Nome			Disponível			Activo		
Chefe-Secretária			<input checked="" type="checkbox"/>			<input type="checkbox"/>		
Grupos de Atendimento			<input checked="" type="checkbox"/>			<input type="checkbox"/>		
Quando utilizadores recebem chamadas								
Nome			Disponível			Activo		
Não Incomodar			<input checked="" type="checkbox"/>			<input type="checkbox"/>		
Rejeitar chamadas anónimas			<input checked="" type="checkbox"/>			<input type="checkbox"/>		
Lista branca			<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>		
Lista negra			<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>		
Identificar chamador			<input type="checkbox"/>			<input type="checkbox"/>		
Identificar chamador sempre			<input type="checkbox"/>			<input type="checkbox"/>		
Esconder a minha identificação			<input checked="" type="checkbox"/>			<input type="checkbox"/>		
Quando utilizadores efectuam chamadas								
Nome			Disponível			Activo		
Esconder a minha identificação			<input checked="" type="checkbox"/>			<input type="checkbox"/>		
Ver informações do destino			<input type="checkbox"/>			<input type="checkbox"/>		
Lista negra			<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>		
Permitir chamadas externas			<input checked="" type="checkbox"/>			<input type="checkbox"/>		
Permitir chamadas para rede fixa			<input checked="" type="checkbox"/>			<input type="checkbox"/>		

Figura 4.5: Funcionalidades

Por último, na opção "Funcionalidades" (figura 4.5) são listadas as funcionalidades (e.g. reencaminhamentos, barramentos) do grupo escolhido no momento. A informação relativa a cada uma das funcionalidades é: se está ou não disponível (i.e. se os utilizadores deste grupo e grupos "filhos" têm acesso a esta funcionalidade) e se esta se encontra activa por defeito (i.e. a configuração das funcionalidades herdada pelos utilizadores quando estes são criados).

De destacar que:

- Todo o portal está disponível em Português e Inglês.
- A "tecnologia" Ajax é utilizada nas "modal windows" e apresentação de mensagens de erro/notificação, por forma a tornar as páginas mais interactivas com o utilizador.
- As linguagens utilizadas para este desenvolvimento foram:

Ruby Linguagem base da meta-framework "Ruby on Rails".

XHTML + CSS Para produção das páginas Web.

JavaScript Script que corre do lado do cliente para validação de formulários, interacção com a página, etc..

### 4.1.1 Melhoramento do Portal de Administração

Numa segunda versão do projecto, foi feita uma reavaliação da aplicação desenvolvida, e foram consideradas algumas melhorias. Com o objectivo de apresentar a nova abordagem da aplicação e conseqüente validação, foi elaborado um "wire-frame" do novo Website, figura 4.6.

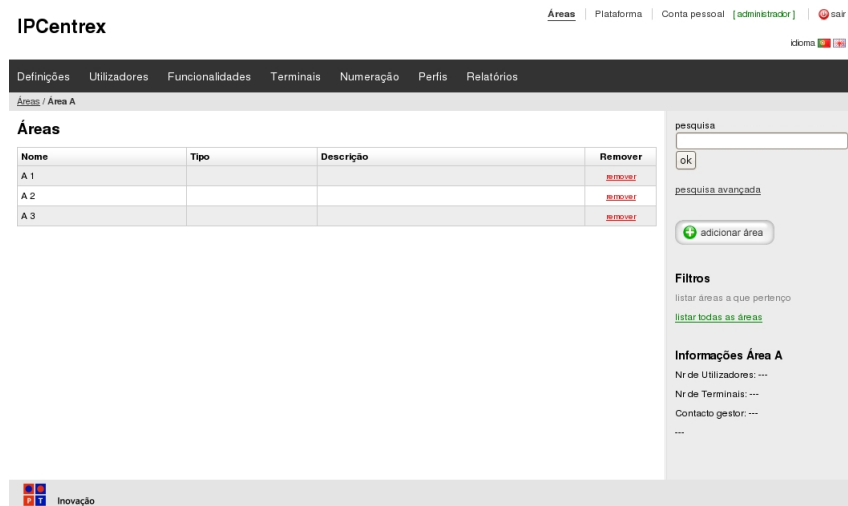


Figura 4.6: Portal de Administração - Segunda Versão

Detalhando a interface, esta encontra-se dividida em seis partes principais. Uma primeira área, que pode ser encontrada na parte superior da página e que se destina a um banner da aplicação, assim como a um menu geral da plataforma (opção "Áreas" para gestão de dados, "Plataforma" para administrar configurações do sistema do lado da operadora, e "Conta Pessoal" que tal como o nome indica é uma secção em que o utilizador configura as suas definições no sistema). Ainda nesta área é apresentada informação como o nome do utilizador, idioma e botão para fazer o *logout* do sistema.

Numa segunda área (abaixo da primeira), é apresentado o menu respectivo a cada uma das opções descritas acima (Áreas, Plataforma e Conta Pessoal). O caso que estamos a analisar é o da Áreas (i.e. Portal de Administração) pelo que o menu será respectivo a esta.

A terceira área, pouco mais abaixo que a anterior, destina-se a representar o local actual em que nos encontramos, sendo assim possível que quando seleccionamos alguma opção do menu, sabermos a que área essa acção diz respeito. Esta área permite também uma mais fácil navegação no sistema.

Na quarta e quinta áreas são apresentadas, respectivamente, a área principal da aplicação Web, e um menu que contém o formulário de pesquisa, e opções para

## Implementação

efectuar acções sobre as entidades seleccionadas. Uma última área, apresenta apenas um logo da empresa.

Em Anexo podem ser encontrados mais "print screens" da aplicação, que detalham melhor a aplicação e a sua interface.

Esta interface centra-se na entidade área sendo que os restantes componentes, listados no menu principal, são relativos à área seleccionada (por exemplo, no "tab" Utilizadores são listados os utilizadores de uma dada área). A informação da área onde nos encontramos é dada pelo "caminho" apresentado logo abaixo do menu principal. Pormenorizando o menu apresentado, temos:

- Definições: Onde são apresentados os detalhes da área seleccionada (descrição/tipo, nome, domínio).
- Utilizadores: Aqui são listados os utilizadores da área seleccionada, sendo possível adicionar, editar e remover utilizadores. Para além destas três acções principais é também possível atribuir numeração, gerir funcionalidades e delegar privilégios a um utilizador seleccionado na lista.
- Funcionalidades: Permite disponibilizar funcionalidades para a área em questão.
- Terminais: listagem dos terminais associados a uma área, com possibilidade de adicionar, editar e remover terminais, existindo ainda a possibilidade de associar extensões a um terminal seleccionado na lista.
- Numeração: Aqui é listada a numeração interna e externa, sendo possível adicionar e remover numeração interna, assim como reservar ou remover numeração externa. Para além disso é também disponibilizada a acção de associar numeração a um utilizador.
- Perfis: Aqui é possível editar os perfis associados a cada área (se se encontra activa ou não, e qual a sua configuração).
- Relatórios: Local onde são listadas as chamadas efectuadas através do sistema, assim como a informação essencial associada.

No caso particular de "Adicionar Área" é apresentado um "wizard" para a criação de uma área, em que se pode definir detalhes da área, funcionalidades, numeração interna, terminais, e gestor. Estas acções são "casos particulares" das existentes nas

## Implementação

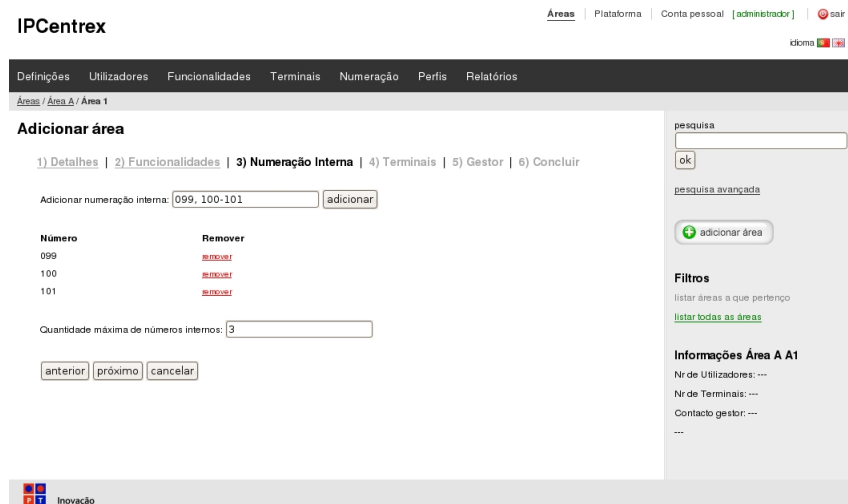


Figura 4.7: Adicionar Área

”tabs” respectivas, e têm por objectivo facilitar a interacção do utilizador com o portal.

Antes de clicar em alguma das opções apresentadas no menu, são listadas as áreas às quais temos permissões de gestão. O menu apresentado é relativo à área aqui seleccionada.

Na tab de Definições são apresentados os detalhes gerais da área seleccionada: descrição/tipo, nome, domínio.

Na tab Utilizadores é apresentada uma tabela com a informação sobre os utilizadores da área em questão, (ou da área e respectivas sub-áreas). É possível adicionar, editar e remover utilizadores, atribuir numeração ao utilizador (em que se associa uma numeração interna ou externa ao utilizador), gerir funcionalidades (em que se configura se um utilizador tem ou não acesso a determinadas funcionalidades), e delegar privilégios (em que se configura os privilégios de administração que o utilizador tem sobre uma dada área).

- Na opção ”Funcionalidades” é possível disponibilizar ou restringir o acesso a determinada funcionalidade para a área seleccionada.
- Na tab de ”Terminais” é possível visualizar os terminais associados à área, assim como efectuar as acções básicas de adição, edição e remoção de terminais. Para além disso é também possível associar numeração interna a um terminal (i.e. associar um número interno de um utilizador a um terminal).

## Implementação

- Na opção "Numeração" é listada a numeração interna e externa da área. É possível remover numeração (interna e externa), adicionar numeração interna, e reservar numeração externa. Aqui também é apresentada a informação sobre se algum utilizador está associado à numeração listada.
- Na tab "Perfis" são listados os perfis associados à área, sendo possível activar por defeito e editar cada uma das funcionalidades. Também é disponibilizada a acção de importar as definições dos perfis da área "pai".
- Na última das tabs, "Relatórios", são listadas as "actividades" (chamadas ou outras) do sistema para a área seleccionada, sendo possível exportar os relatórios ou remover alguma das entradas da listagem.

Uma característica que é transversal a todas as tabs (excepto para a tab de "Perfis"), é a existência de um formulário de pesquisa com possibilidade de detalhe avançado.

## 4.2 Camada de Serviços

Nesta fase do projecto procedeu-se à implementação do serviço "Click-to-Call" em Jain Slee. A arquitectura física do sistema está descrita na figura 4.8.

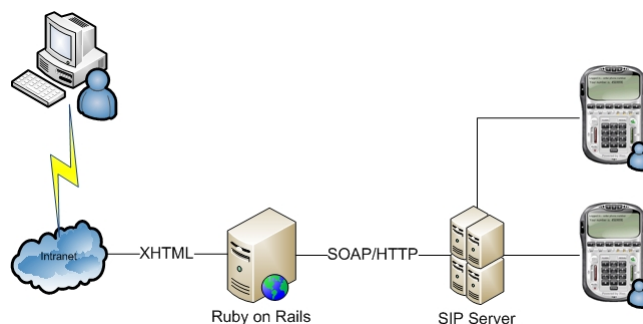


Figura 4.8: Arquitectura Física do Sistema SIP

O caso de uso do serviço é descrito como: um utilizador consulta o seu directório de contactos (aproveitando o portal de Self-Care em desenvolvimento). Ao clicar no link correspondente, é enviada uma mensagem SOAP/HTTP para o servidor Jain Slee que por sua vez efectua a chamada desejada entre os dois contactos.

Na mensagem SOAP/HTTP é enviada a informação sobre qual o contacto da pessoa que está a efectuar o pedido e qual o contacto do destinatário.

### 4.2.1 Implementação

Na implementação com Jain Slee, foram reutilizados dois Resource Adaptors (HTTP e SIP) incluídos na distribuição do Application Server (Open Cloud - Rhino). Foi então desenvolvido um SBB que recebe eventos HTTP Request referentes ao pedido inicial, proveniente do portal. Este evento origina a criação de uma instância SBB que será responsável pelo estabelecimento dos diálogos SIP. Numa primeira fase é validado o envelope que é enviado na mensagem SOAP. Caso esteja mal estruturada é logo enviada uma mensagem de erro SOAP para o portal. No caso de a informação do envelope estar correcta é usado o RA de SIP para a criação de dois diálogos para os respectivos utilizadores. O primeiro é realizado para o utilizador que efectuou o pedido, esperando uma resposta de atendimento, e só depois se realiza o segundo dialogo para o destinatário (ver a figura 3.7).

Depois de desenvolvida a versão Jain Slee do sistema, passou-se para a abordagem em SIP Servlets. Neste caso foi encontrado um exemplo da aplicação já desenvolvida, em versões para Tomcat e JBoss. Escolhendo a versão JBoss, procedemos à instalação do exemplo.

Graças à convergência entre o mundo SIP e o mundo J2EE, presente num container SIP Servlet, é espectável que a criação de aplicações mais poderosas seja facilitada, devido ao acesso transparente ("seamless") entre diferentes contexto (JEE, HTTP e SIP). Pelo que a abordagem para o despoltar deste serviço foi diferente. Usando uma chamada através de um POST/HTTP directamente para a Servlet responsável simultaneamente pela parte de SIP e HTTP - em vez da chamada através de um SOAP/HTTP.

Esta abordagem não será a mais correcta uma vez que o container SIP Servlet é responsável tanto pela parte HTTP como SIP. A forma mais correcta de o fazer, seria colocar ambas as aplicações, Web e SIP, numa mesma aplicação/servidor para que fosse possível aproveitar as vantagens do container. Como neste caso a parte Web está desenvolvida em "Ruby on Rails", a utilização de SIP Servlets perde parte das suas vantagens, já que não foi pensada com esse objectivo.

## 4.3 Resumo

Neste capítulo foi descrito o trabalho elaborado no decorrer do estágio. Implementando as soluções descritas no capítulo anterior, o projecto teve duas fases: Camada de Interface e Serviços.

Na Camada de Interface, através do uso de diversas linguagens e tecnologias, foi possível dar forma ao requisito de implementação do Portal de Administração, assim como um planeamento para a sua segunda versão com melhoramentos associados.

## Implementação

Na Camada de Serviços foi possível fazer o estudo previsto das tecnologias Jain SLEE e SIP Servlets, e a implementação do serviço "Click to Call" a ser chamado pelo Portal de Self-Care.

## Capítulo 5

# Avaliação dos Resultados

### 5.1 Análise Teórica

O Jain SLEE e as SIP Servlets são ambos standards definidos no Java Community Process direccionados para a criação de aplicações. Ambos os standards definem um modelo de contentor para aplicações, definem como é que as aplicações são desenvolvidas e instaladas no contentor e definem os serviços e características fornecidos pelos respectivos contentores.

O Jain SLEE define um ambiente aplicacional vocacionado para aplicações orientadas à rede. Define um modelo agnóstico ao protocolo, por componentes, orientado a objectos e com programação transaccional.

O Jain SLEE é adaptado para usar um determinado recurso graças a um Resource Adaptor, o que torna o SLEE extensível.

As SIP Servlets definem um modelo baseado em contentor que é uma extensão do bem conhecido modelo Servlet. Foi desenhado para simplificar a criação de aplicações SIP logo enaltecendo a adopção do protocolo SIP.

Podemos considerar um conjunto de características por forma a comparar e distinguir estes dois modelos para se perceber os benefícios de cada um e como podem ser usados. Vamos considerar os seguintes aspectos:

- Arquitectura da aplicação – A maneira como é estruturada uma aplicação na lógica de negócio.
- Estado da aplicação – A maneira como o estado da aplicação pode ser representado, acedido e protegido.
- Controlo de concorrência – A maneira como são controlados múltiplos threads em execução.

- Facilidades disponíveis às aplicações – Que facilidades são disponibilizadas às aplicações pelo contentor.
- Mecanismos de disponibilidade – Como é que as características e conceitos do modelo podem ser tratados para garantir disponibilidade contínua.
- Gestão – Quais os mecanismos disponibilizados pelo contentor para gerir aplicações e o próprio contentor.

Na tabela 5.1 é feita esta comparação.

## 5.2 Análise Prática

Da aplicação prática do serviço "Click to Call" pôde-se verificar que no caso do Jain SLEE, a configuração é bastante extensa e com uma elevada probabilidade de erro. São necessários diversos ficheiro XML com configurações para colocar uma funcionalidade base a funcionar. A curva de aprendizagem mostra-se por isso bastante acentuada.

No caso da aplicação prática em SIP Servlets, com base no exemplo fornecido pela Java, pôde-se concluir que uma consequência da co-habitação entre os containers HTTP e SIP é a possibilidade de criar facilmente aplicações convergentes, como é o caso do "Click to Call" num Website. A informação entre o mundo Web e SIP pode ser facilmente trocada pela partilha nos objectos de sessão.

Contudo, neste caso em particular a interface Web está a ser desenvolvida num contexto diferente, em Ruby on Rails. O que tornou esta "suposta" vantagem do sistema, num ponto de pouca relevância. Para além disso a interação entre a parte Web e o container SIP teve que ser feita por POST/HTTP, o que significa que teve que ser chamado o container Web para dispoletar uma acção na parte SIP.

## 5.3 Conclusões

No que diz respeito à parte prática da análise, podemos concluir que seriam necessários mais testes e tempo para analisar todas as vertentes das duas tecnologias.

Já no caso do estudo teórico a análise foi bastante mais "profunda", tendo-se constatado que, na verdade, Jain SLEE e SIP Servlets são tecnologias mais complementares que concorrentes - existem zonas de conforto naturais para cada tipo de aplicação pretendida.

Jain SLEE é consideravelmente mais baixo-nível (baseado em eventos), que SIP Servlets, e com uma camada de abstracção neutra em relação a SIP.

## Avaliação dos Resultados

Jain SLEE é uma tecnologia JavaSE, com a qual se pode escrever, por exemplo, um container de SIP Servlets. Tem um modelo definido para a composição de serviços e demarcação de transacções que envolvem diálogos SIP.

A tecnologia SIP Servlets permite a programadores Java com muita ou pouca experiência SIP desenvolverem facilmente e em pouco tempo aplicações convergentes de telecomunicações numa rede IMS reutilizando o seu leque de conhecimentos e minimizando o *time-to-market*.

No caso em particular do projecto shipnet, penso que o Jain SLEE é a melhor opção, pela sua flexibilidade, robustez e pelo facto de poder ser integrado com maior facilidade com outras plataformas.

## Avaliação dos Resultados

Tabela 5.1: Comparação Jain SLEE e SIP Servlets

SIP Servlets	JAIN SLEE
Arquitectura da aplicação	
Baseado nas HTTP Servlets; Unidade lógica da aplicação é o Servlet; Nenhum modelo standard para composição ou reutilização	Baseado em componentes, arquitectura orientada a objectos; Unidade lógica da aplicação é o SBB; Suporte para composição e reutilização
Estado da aplicação	
As Servlets são stateless; A partilha de um estado pode ser armazenada num objecto sessão separado com um nome String/Objecto; Este estado partilhado é visível a todas as Servlets com acesso à sessão	Os SBBs podem ser stateful (ou stateless); O estado do SBB é privado e uma propriedade do próprio SBB; A partilha de estado pode ser armazenada num Activity Context separado através de um interface seguro; O acesso aos estados partilhados pode ser especificado no momento do deploy
Controlo de concorrência	
Apenas gestão da aplicação, i.e. uso de monitores Java	Gestão do sistema, i.e. isolamento de transacções concorrentes
Suporte de protocolos	
SIP e HTTP	Agnóstico ao protocolo; Pode ser estendido para suportar protocolos adicionais e recursos externos; Modelo de eventos consistente em relação a protocolos/recursos
Facilidades disponíveis às aplicações	
Timer	Timer; Trace; Alarm; Statistics and Usage; Profiles
Mecanismos de disponibilidade	
Gestão do estado do contentor (objecto sessão) que pode ser replicado; Não existe contexto de transacção para o processamento das mensagens SIP; Operações de estado do container não são transaccionais; As facilidades não são transaccionais; Não existe modelo de falha definido	Gestão do estado do contentor (SBB, CMP, Activity Context, Facility state, etc) que pode ser replicado; Contexto de transacção para a entrega de eventos; As operações de gestão do estado do container são transaccionais; As facilidades são transaccionais; Modelo de falhas bem definido através de transacções
Gestão	
Não tem mecanismo de gestão definido	Interface de gestão baseado em JMX (Java for Management Extensions); Independente do protocolo de gestão; Interface para gestão das aplicações, incluindo ciclo de vida, perfis, traces, etc.; Interface para gestão do ciclo de vida do SLEE

## Capítulo 6

# Conclusões e Trabalho Futuro

### 6.1 Satisfação dos Objectivos

Dado por terminado o estágio, podemos concluir que os objectivos a que este estágio se propôs foram cumpridos na totalidade. A plataforma de administração corresponde aos requisitos e objectivos inicialmente propostos, assim como o wire-frame desenvolvido para uma segunda versão.

Na segunda fase do projecto, relativa à Camada de Serviços, foi possível implementar o serviço “Click to Call” a que me propus no início do estágio. O estudo teórico permitiu avaliar as duas tecnologias SIP (Jain SLEE e SIP Servlets), por forma a definir qual a mais apropriada para o projecto shipnet.

Este estágio contribuiu para o completar da solução ip-Centrex pertencente ao projecto shipnet, através de uma primeira versão do portal de administração, e do encaminhamento de uma segunda versão para os requisitos finais pretendidos. Uma segunda contribuição fez-se através do estudo das tecnologias de desenvolvimento de serviços - Jain SLEE e SIP Servlets - o que contribui para uma melhor percepção do estado da arte nesta área em constante evolução.

Este estágio permitiu-me aprofundar os meus conhecimentos em diversos níveis. No plano tecnológico adquiri conhecimentos a nível de IMS e redes de comunicação. Aprofundei o meu conhecimento a nível de arquitectura de sistemas, implementação, e desenvolvimento web.

A nível pessoal a minha integração com uma equipa de trabalho, com o desenrolar do dia-a-dia forneceu-me competências a nível de trabalho em equipa, noções de assertividade e desenvolveu a minha capacidade de resolução de problemas.

## 6.2 Trabalho Futuro

Relativamente ao futuro do trabalho desenvolvido, podemos dizer que para a Camada de Interface, prende-se com a implementação do projecto definido para a segunda versão do Portal de Administração.

Na Camada de Serviços, o trabalho futuro é bastante mais alargado, já que seria necessário testar de forma mais abrangente tanto a tecnologia Jain SLEE como SIP Servlets. Isto deve-se ao facto de o serviço de "Click to Call" implementado não abranger todas as vertentes que as duas tecnologias podem seguir.

# Referências

- [A.P08] Manoj A.P. Ruby on rails, Fevereiro 2008. <http://webscape-india.blogspot.com/2008/02/rubby-on-rails.html>.
- [Bla06] David Black. *Ruby for Rails: Ruby Techniques for Rails Developers*. Manning Publications, 2006.
- [Cha07] Shyam Chakraborty. *IMS Multimedia Telephony over Cellular Systems: VoIP Evolution in a Converged Telecommunication World*. Wiley, 2007.
- [DT06] David Hansson Dave Thomas. *Agile Web Development with Rails, 2nd Edition*. Pragmatic Bookshelf, 2006.
- [Fow03] Martin Fowler. *UML Distilled, Third Edition*. Addison-Wesley Professional, 2003.
- [GC06] Miguel-Angel García-Martín Gonzalo Camarillo. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, Second Edition*. Wiley, 2006.
- [Hai04] Sven Haiges. Jain/slee: Ejb for communications, Setembro 2004. <http://java.sys-con.com/read/46230.htm>.
- [Ino07a] PT Inovação. Service handling on ip networks, Novembro 2007. [http://www.ptinovacao.pt/produtos/SHipNet/F\\_SHipNET.pdf](http://www.ptinovacao.pt/produtos/SHipNet/F_SHipNET.pdf).
- [Ino07b] PT Inovação. Sip application server, Novembro 2007. [http://www.ptinovacao.pt/produtos/SHipNet/f\\_IP-Jib.pdf](http://www.ptinovacao.pt/produtos/SHipNet/f_IP-Jib.pdf).
- [Ino07c] PT Inovação. Voip convergent business solutions, Novembro 2007. [http://www.ptinovacao.pt/produtos/SHipNet/f\\_IP-Centrex.pdf](http://www.ptinovacao.pt/produtos/SHipNet/f_IP-Centrex.pdf).
- [ipc] *Descrição Técnica IP Centrex*.
- [ipj] *Descrição Técnica IP Jib*.
- [JB04] Sebastian Coope Jeffrey Bannister, Paul Mather. *Convergence Technologies for 3G Networks: IP, UMTS, EGPRS and ATM*. Wiley, 2004.
- [Kri03] Anders Kristensen. Sip servlet api, Fevereiro 2003. <http://jcp.org/en/jsr/detail?id=116>.
- [O'R05] Tim O'Reilly. What is web 2.0, Setembro 2005. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.

## REFERÊNCIAS

- [Reu07] Reuters. Global cellphone penetration reaches 50 pct, Novembro 2007. <http://investing.reuters.co.uk/news/articleinvesting.aspx?type=media&storyID=nL29172095>.
- [Ros04] J. Rosenberg. *Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)*, Abril 2004.
- [SM04] Inc. Sun Microsystems. Jain s1ee 1.0 specification, Março 2004. <http://jcp.org/en/jsr/detail?id=22>.
- [W3C02] W3C. Web services @ w3c, Janeiro 2002. <http://www.w3.org/2002/ws/>.
- [Wik08a] Wikipedia. Ip multimedia subsystem, Julho 2008. [http://en.wikipedia.org/wiki/IP\\_Multimedia\\_Subsystem](http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem).
- [Wik08b] Wikipedia. Web service, Julho 2008. [http://en.wikipedia.org/wiki/Web\\_service](http://en.wikipedia.org/wiki/Web_service).
- [Wil07] Justin Williams. *Rails Solutions: Ruby on Rails Made Easy*. friends of ED, 2007.
- [Zak05] Nicholas C. Zakas. *Professional JavaScript for Web Developers*. Wrox, 2005.

# Anexo A

## Requisitos

Elaborado pela PT Prime, este documento apresenta a especificação de requisitos do sistema ip-Centrex.

### A.1 Interface

#### A.1.1 Apresentação

Personalizado para PT Prime

1. Utilizar cores, logo e imagem típicos da PT Prime.

(107-109) Escolha de Língua a Apresentar

1. Deverá ser possível escolher uma língua predefinida, sendo que este a poderá alterar:
  - (a) Português (prioritário)
  - (b) Inglês

Inventário de Terminais

1. Inventário dos terminais, com a descrição das suas características, funcionalidades.
2. Para facilitar o sistema de provisão e facilitar o controlo de terminais.
3. Deve conter o número de terminais activos de cada modelo.

Funcionalidades Gerais da Interface

1. New, Delete, Undo, Save, Edit.
2. Deverão ser utilizadas **Check Boxes** para seleccionar funcionalidades (com algumas seleccionadas por defeito - características de Grupo por defeito).

## Requisitos

3. Deverão ser utilizadas **Listas** para gestão de utilizadores ou terminais.
4. Deverão ser **Combo Boxes** para atribuição de recursos, juntamente com funcionalidade **ADD** ou **Retrieve**.

Estrutura - Gestão de Sistema e Utilização de Sistema

1. A estrutura da interface deverá ser dividida em duas grandes áreas:
  - (a) (113-121, 126) Gestão do Sistema
    - i. Grupo Administrador PT (Nível 0)
      - A. Configuração de novos Clientes - sendo o único que pode criar os Grupos de Topo - e gestão ou eliminação destes.
      - B. **Criação, Gestão e Eliminação** de todos os Grupos .
      - C. Na primeira área, deverão ser introduzidos os campos que são o **Nome, Descrição e Características** do Grupo.
      - D. Numa área serão escolhidas as **funcionalidades** a que o Grupo terá acesso.
      - E. Noutra serão escolhidos os **recursos** que lhe serão atribuídos.
      - F. Finalmente, numa outra área será atribuído o Utilizador - **Gestor de Grupo**, que fará também parte do grupo e serão atribuídos a este grupo um **conjunto de Utilizadores**.
      - G. Acesso à Árvore de Grupos de Cliente (pode-se configurar um qualquer grupo a um qualquer nível de toda a árvore de grupos, gerada com raiz no Nível 0).
      - H. Somente o **Grupo de Nível 0** terá acesso a configurar todos os parâmetros de toda a **árvore** de Grupos.
      - I. Deverá ser apresentada a árvore de grupos existente.
      - J. Poderá ser seleccionado um qualquer grupo da árvore e **surgirão as áreas** de Criação, Gestão e Eliminação já **indicadas no requisito anterior**.
      - K. Criação de Utilizadores.
      - L. Área para gestão de Utilizadores, onde se pode **criar um utilizador** com os **recursos associados**, incluindo o(s) terminais.
      - M. Serão lhe atribuídas **funcionalidades** que este poderá ou não utilizar.
      - N. Será necessário o controlo do **número de funcionalidades realmente utilizadas por tipo, no total dos utilizadores do grupo**. Haverá a **listagem dos Utilizadores** deste grupo.
      - O. Detalhes de Chamadas.
      - P. Possibilitar visualizar toda a informação de **Detalhes de Chamadas** já definida, para toda a árvore com **raiz no grupo que a visualiza**.
      - Q. Possibilitar visualizar toda a informação de **Detalhes de Chamadas** já definida, apenas para o **utilizador em causa**.
      - R. Provisão dos Clientes - terminais, configurações e acessos.

## Requisitos

- S. Possibilitará a provisão automática dos terminais e das configurações de utilizadores.
  - T. Para tal, terá uma área com a Listagem das marcas e modelos dos terminais usados na solução. Terá a listagem dos MAC Address de todos os terminais dos Clientes, agrupadas por marca e modelo. Possibilitará colocar novas versões de FW por UPLOAD directamente no servidor de TFTP (ou outro), para serem carregados nos terminais.
  - ii. Grupo de Topo (Nível 1)
    - A. Configuração de **novos Grupos** e **gestão** ou **eliminação** destes.
    - B. Acesso a **Árvore de Grupos** criada para baixo do Grupo de Topo, ou seja **árvore do Cliente**, com raíz no nível 1.
    - C. Criação de Utilizadores.
    - D. Detalhes de Chamadas.
  - iii. Grupo Geral (Nível 1+N)
    - A. O apresentação da página de configuração de um Grupo geral Nível 1+N será exactamente a mesma que a de um **Grupo de Topo**, sendo que a **árvore** de Grupos a que terá acesso será só a que tem **raíz em si mesmo**.
- (b) (122-125, 127, 128) Utilização do Sistema
- i. Esta área da interface será constituída pela **interface Web de Configuração própria do Utilizador**.
  - ii. Escolha e Configuração de Funcionalidades
    - A. Nesta área o Utilizador poderá **escolher qual das funcionalidades que lhe foram atribuídas ele pretende utilizar**. O utilizador poderá depois **configurar cada uma destas funcionalidades**, com a informação que for necessária.
  - iii. Escolha e Configuração de Terminais Utilizados
    - A. O Utilizador poderá **introduzir novos terminais** (desde que estes estejam disponíveis na solução) que tenha adquirido nesta área. Poderá, ainda nesta área, **configurar as várias características dos terminais** a que tiver acesso.
  - iv. Gestão de Recursos
    - A. Área onde se possibilita a **atribuição dos recursos disponíveis para o utilizador aos terminais (extensões e DDI's)**.
  - v. Configuração de Perfis / Consulta de Grupos
    - A. Área onde se possibilita a **configuração de três perfis diferentes**.
    - B. Este perfil deverá ser **construído a partir das configurações do utilizador de momento**, podendo se configurar todos os parâmetros necessários para a criação de um perfil.
    - C. Cada utilizador poderá **consultar a listagem dos grupos** a que pertence e as **funcionalidades e recursos que lhe foram delegadas por cada um dos grupos**.

### A.1.2 Atributos

#### Extensões

1. São **números** da Numeração Interna de cada Grupo Cliente (**Nível 1**), que podem ser **atribuídos em Gamas** ou em **números Soltos a Grupos e Utilizadores**.

#### DDI's

1. São **números** através dos quais se pode chegar da **Rede Externa** ao Sistema a um Utilizador de um qualquer Grupo.
2. Estes podem ser **atribuídos em Gamas** ou em **números Soltos a Grupos e Utilizadores**.

#### Restrições

1. As Restrições são as **Limitações impostas pelos Grupos de Nível N-1** aos **Grupos de Nível N**, por eles criados. Estas têm como fim a **restrição dos recursos** consumidos por cada **Grupo, Utilizador**, assim como a restrição de acessos às funcionalidades por parte dos **Grupos** ou dos **Utilizadores**.

#### Funcionalidades

1. As Funcionalidades são os **privilégios** que **podem ser delegados entre Níveis de Grupos** que têm como fim a **disponibilização de um serviço de valor acrescentado à entidade Utilizador** - que usufrui do sistema.
2. As Funcionalidades são, desta forma, **atributos dos Grupos e Utilizadores**.

### A.1.3 Entidades

#### 1. Grupo

##### (a) Definição de Grupo

- i. Para a criação de todos os utilizadores, é **obrigatória a definição prévia de um Grupo**.
- ii. Um Grupo é **identificado por uma frase**, que em principio identifica uma empresa ou uma organização dentro de uma empresa.
- iii. São Grupos todos os **conjuntos de Utilizadores** aos quais se aplicam as **mesmas regras, restrições e se fornecem as mesmas regalias e os mesmos acessos a funcionalidades específicas**.
- iv. São também Grupos as agregações por **Site** ou com base na **organização de uma empresa**.

##### (b) Níveis de Grupo

- i. Os Níveis de Grupo são úteis para visualização da Hierarquia de Grupos.
- ii. O **Nível 0** é o mais alto na Hierarquia (**PT**).

## Requisitos

- iii. Grupos do **Nível N**, só poderão criar **Grupos N+1**.
  - iv. Grupos do **Nível N+1** estarão sempre **sujeitos às imposições** do Grupo de **Nível N** que o criou.
- (c) Dois Grupos definidos por Defeito
- i. Todos Grupos de topo (**Cliente/Empresa - Nível 1**) derivam do **Super Grupo PT - Administrador (Nível 0)**, onde **todos os privilégios estão disponíveis**.
  - ii. Quando criado o primeiro Grupo para uma empresa é criado e atribuído um Grupo que serve de padrão para todos os Grupos criados a partir do primeiro.
- (d) Criação de Grupos
- i. Se um Grupo tiver acesso ao **privilégio da criação de outros Grupos**, tem de o fazer dentro do **mesmo Nível de Grupo a que pertence**.
  - ii. Existirá também a **funcionalidade** adicional que, quando atribuída, **permite ao Grupo criar novos Grupos a um Nível inferior**.
  - iii. Qualquer Grupo criado deste modo fica relacionado com o Grupo de topo (**Nível Superior**), o que significa que:
    - A. **Só é válido dentro da organização onde foi criado**.
    - B. Podem existir Grupos com a **mesma designação em organizações diferentes**.
- (e) Diferenciação entre Grupos
- i. A separação em Grupos pretende **distinguir entre acessos a acções/funcionalidades** dos vários utilizadores ou grupos de utilizadores.
  - ii. Cada Grupo poderá **gerir os acessos dos Grupos criados por si**, com base na delegação de acessos a funcionalidades.
- (f) Delegação de Competências e Acessos
- i. A **delegação de acessos a funcionalidades** é baseada no Conceito de Grupo, sendo que **todos os elementos pertencentes ao mesmo grupo** terão acesso às **mesmas funcionalidades**.
  - ii. Os **privilégios de configuração dados a cada um dos Grupos** pode, **no máximo, ser idêntica** aos do Grupo que o cria, mas pode também ser um **subconjunto** dessas **funcionalidades**.
- (g) Atributos
- i. Grupo de Utilizadores
    - A. Um Grupo terá um **Utilizador que o Gere**, e que **pertence** também ao grupo, e os Utilizadores que o constituem.
  - ii. Nome, Descrição e Características de um Grupo
    - A. Um Grupo terá associada um **Nome**, uma **Descrição (Empresa/Direcção/ Site/ ...)** e características como **Morada, Função** na organização, **Responsabilidades**, etc.
  - iii. Restrições de um Grupo

## Requisitos

- A. Um Grupo terá que suportar um Conjunto de Restrições, que limitam os Recursos desse grupo.
- B. São restrições as que se seguem:
- C. Extensões;
- D. DDI's;
- E. de Chamadas em Paralelo;
- F. Gamas ou números soltos de Extensões e DDI's;
- G. Níveis de custo em chamadas e/ou minutos.
- H. Um Grupo que seja um **Site** terá ainda a **restrição extra do número de chamadas máximo em simultâneo que todos os Grupos/Utilizadores** desse Site podem efectuar, limitados **em função do acesso VoIP do Site**.

### iv. Privilégios de um Grupo

- A. Um Grupo terá disponível um Conjunto de Privilégios, como:
- B. Criar um Grupo ao Nível N+1.
- C. Uso de Call Forwarding.
- D. Existência de um Voice Mail.
- E. Possibilidade de Delegação de privilégios.
- F. Configuração de Música on Hold própria.
- G. Serão possíveis privilégios de um grupo, resumidamente, todas as funcionalidades disponíveis no sistema.

## 2. Terminal

### (a) Definição de Terminal

- i. Terminal é uma entidade que pode estar directamente **associada a um ou mais Utilizadores** ou **a um ou mais Grupos**.
- ii. Terminal estará **sempre associado a uma ou mais extensões**, podendo estar **associado a zero ou mais DDI's**.
- iii. Terminal é o **elemento VoIP do sistema**, que só por si **disponibiliza aos Utilizadores funcionalidades base**.

### (b) Atributos

- i. Utilizadores do Terminal
  - A. A cada terminal estarão **associados um ou mais Utilizadores ou Grupos**.
  - B. **Não faz sentido** ter um Terminal no sistema **sem que lhe seja atribuído um Utilizador ou um Grupo**.
- ii. Extensões e DDI's de um Terminal
  - A. A um **Utilizador** serão atribuídos **uma ou mais Extensões SIP e um ou mais DDI's**, para uso próprio.
  - B. Estes **DDI's** não deverão ser somente **numeração nómada (3XX XXX XXX)**, mas também **numeração fixa (2XX XXX XXX)**.

## 3. Utilizador

- (a) Definição de Utilizador
  - i. Utilizador é a **entidade que realmente usufrui dos serviços**.
  - ii. A um Utilizador **pode ser atribuída a gestão de um Grupo**, caso contrário, não poderá criar novos Grupos e só tem **acesso à configuração de funcionalidades/características pessoais**. A função de **Delegação é então restrita à entidade Grupo**.
  - iii. Um Utilizador é uma **entidade que poderá gerir um ou mais Grupos**, poderá **pertencer a um ou mais Grupos** e **possuir uma ou mais extensões e DDI's**.
- (b) Definição de Utilizador Gestor de Grupo
  - i. Gestor, é um Utilizador que tem acesso à **gestão de uma entidade Grupo** e todos os privilégios e recursos a ele associados ou atribuídos.
- (c) Concorrência de Configurações de Grupo
  - i. Quando se detecta que um Utilizador **pertence a mais do que um Grupo** deverão ser **aplicados a soma dos dois privilégios**.
- (d) Configuração de Funcionalidades Disponíveis
  - i. O Utilizador terá acesso a uma **interface** onde poderá **configurar as várias funcionalidades** que lhe forem **disponibilizadas pelos Grupos** a que pertence.
- (e) Atributos
  - i. Nome e Descrição de um Utilizador
    - A. Um Utilizador terá associado um **Nome** e uma **Descrição (Posição/ Responsabilidades/ Função/ etc.)**.
  - ii. Grupos a que pertence Utilizador
    - A. Um Utilizador poderá estar **associado a um ou a mais Grupos**, podendo ser **Gestor de um ou mais desses Grupos**.
  - iii. Extensões e DDI's de um Utilizador
    - A. A um Utilizador serão **atribuídos um número de Extensões e de DDI's**, para uso próprio.
    - B. Um Utilizador só poderá **executar chamadas exclusivamente por essa numeração atribuída**.
    - C. A um utilizador poderão ser atribuídos **um ou mais DDI's e uma ou mais Extensões**.
    - D. Deverá ser possível atribuir a um utilizador diferentes DDI's de entrada e de saída e em quantidades diferentes, ou seja, deverá haver a possibilidade de **definir de forma distinta quais os DDI's a utilizar nas chamadas de inbound e outbound**, por parte de um dado utilizador.
  - iv. Terminais de um Utilizador
    - A. A um Utilizador serão atribuídos **um ou mais terminais**, podendo ser o uso de cada um dos terminais **partilhado com outros Utilizadores**.

- B. Um Cliente **”regista-se” num terminal**, por **introdução de um código na primeira chamada** (por exemplo o AS terá que evocar o pedido de update de configuração do user respectivo ao código introduzido do terminal. Passado X tempo, será evocado novamente o pedido para limpar configuração do terminal). Isto implica que **existam terminais sem Utilizadores univocamente atribuídos**.
- C. **Logout automático numa só utilização**.
- v. Privilégios de Utilizador
  - A. Um Utilizador **herda os privilégios dos Grupos a que pertence**, podendo estes ser **diferenciados por Utilizador do grupo**, por **limitação imposta pelo Gestor de Grupo**.
- vi. Restrições de um Utilizador
  - A. Um Utilizador verá os seus **recursos como Extensões, DDI’s ou Minutos por Mês, restringidos conforme imposições** do(s) Utilizador(es) que gere(m) o(s) Grupo(s) a que pertence.
- vii. Perfis de um Utilizador
  - A. Para facilitar o Uso do Sistema por parte do Utilizador, é fornecida a possibilidade de cada **utilizador configurar (por exemplo até 3) perfis** diferentes de **utilização das funcionalidades que detém**.
  - B. O Utilizador **só poderá ter um perfil activo em cada momento** e terá a **hipótese de alternar entre perfis sempre que quiser**.
- viii. Utilizador VIP
  - A. A cada Utilizador poderá ser associada a **característica de VIP no acesso aos canais de voz, quando estes se encontram saturados**.
  - B. Neste caso, deverá aguardar X segundos (ex.: 10 segundos) a partir do momento em que deverá ser interrompida a chamada a decorrer há mais tempo (com a excepção de ser uma chamada efectuada por outro Utilizador VIP) sendo esta substituída pelo pedido do Utilizador VIP.

#### A.1.4 Funcionalidades

##### 1. Auto-Atend

- (a) Custom Automatic Greeting
  - i. **Mensagens personalizadas para Greeting inicial configuráveis por Grupo**.
- (b) Custom Automatic Menus
  - i. Introdução do comandos via DTMF’s para interacção com os IVR front-end.
  - ii. **Os Menus IVR poderão ser configurados por Grupo**.

2. Call Back

- (a) Permitir que quando uma **chamada não é atendida**, por o terminal estar ocupado com outra chamada, o utilizador possa **realizar uma chamada para esse número não atendido**, através da **introdução de um código**.

3. Call Forwarding

- (a) Call Forwarding Incondicional
  - i. **Reencaminhamento automático sempre** que se executa uma chamada para esse **terminal/extensão**, para um determinado número a definir.
- (b) Call Forwarding no caso de indisponibilidade
  - i. **Reencaminhamento** no caso de o terminal/extensão **não se encontrar disponível na rede** - reencaminhar **para um determinado número** a definir.
- (c) Call Forwarding se Ocupado
  - i. **Reencaminhamento se ocupado** (definir número para o qual vai reencaminhar).
- (d) Call Forwarding Temporal
  - i. **Reencaminhamento temporal** (definir periodos de tempo: por exemplo, todos os dias das 17h às 9h a chamada é reencaminhada para o VoiceMail).
- (e) Call Forwarding por timeouts
  - i. **Reencaminhamento com timeout** (passados X segundos reencaminha para outra extensão).
- (f) Call Forwarding – Numeração
  - i. É necessário garantir que quando **A telefona para B** e **B reencaminha para C**, a chamada é **facturada a B**.
  - ii. Ou seja, o **A number** passará a ser o elemento **B** na segunda chamada.

4. Call Parking

- (a) Call Park
  - i. Permitir que uma chamada seja colocada em **espera numa sala virtual** e que **qualquer utilizador** a possa retirar dessa sala, através do estabelecimento de uma chamada para uma dada extensão.
- (b) Call Pick Up
  - i. Permitir que uma **chamada seja retirada de uma sala virtual de espera**, através da **marcação de uma extensão**.
  - ii. Esta funcionalidade **permite que alguém atenda uma chamada que era direccionada a outra pessoa**, através da **marcação de uma extensão**.

## Requisitos

- iii. Esta funcionalidade deverá **funcionar por grupo**, sendo possível **escolher quais os utilizadores que poderão atender chamadas destinadas a outros utilizadores**.

### 5. Call Queue

- (a) Incoming Call Queue - Pré-Atendimento
  - i. Permitir a **existência de uma sala virtual onde se coloquem as chamadas que não podem ser atendidas de momento**, sendo estas listadas com informação do originante.
  - ii. Lista de espera.
  - iii. A **extensão com pré-atendimento terá uma fila de espera associada**. Se a extensão estiver ocupada ou com chamadas em espera, o chamador deverá ouvir a mensagem que foi configurada pelo utilizador. Esta **mensagem deverá ser configurável via web**, incluindo o **upload da própria mensagem**.

### (b) Outgoing Call Queue

- i. Permitir a existência de uma **sala virtual onde se coloquem as chamadas que não podem ser executadas de momento**, sendo estas realizadas assim que existir disponibilidade por hierarquia de utilizador, **por verificação se é VIP e por antiguidade na lista de espera**.
- ii. Lista de Espera.

### 6. Call Recording

- (a) Permitir a **gravação de chamadas**, sendo um dos utilizadores o **responsável por pedir esta gravação** e os outros envolvidos na chamada **avisados através de uma mensagem de audio automática**. Possibilidade de **configurar a gravação de uma percentagem das chamadas** ou da **gravação de todas as chamadas** de um grupo.

### 7. Call Trace

- (a) Permitir **marcar uma chamada nos registos de chamadas** para que lhe seja **dedicada mais atenção e facilitar a monitoria**.

### 8. Call Waiting

- (a) Suportar a activação, desactivação, e a funcionalidade no próprio sistema, em geral suportada nos terminais. Depois, enviar a indicação (semelhante à existente na PSTN) de um **toque com uma cadência própria quando existe uma chamada em espera**.

### 9. Chefe-Secretária

- (a) Esta funcionalidade consiste na **combinação de outras funcionalidades base**, que poderão ser configuradas manualmente pelo utilizador.
- (b) No entanto, **comercialmente** faz sentido disponibilizar esta configuração como uma funcionalidade extra e fácil de usar.

## Requisitos

- (c) A um **chefe** poderão ser associadas **uma ou mais secretárias**.
- (d) Deverá ser possível garantir que todas as chamadas são reencaminhadas para a secretária, **excepto as VIP's**.
- (e) Será necessário **garantir que as secretárias podem efectuar chamadas ou transferir chamadas para os chefes**.
- (f) Finalmente, **caso seja mais do que uma secretaria**, deverá ser possível seleccionar se quando é efectuada uma chamada para o chefe, **tocará sequencialmente ou ao mesmo tempo os terminais das secretárias**.
- (g) **Existirá uma lista de números, os quais quando forem a origem da chamada esta não é reencaminhada**. Neste caso nesta lista poderemos acrescentar a **“secretária”**, os **outros “chefes”**, lista de número pessoais que o **“chefe”** gostaria que não passassem pela **“secretária”**.

### 10. Detalhe de Tráfego

- (a) Estimativa de Consumo - Definição de Plano de Tarifário
  - i. Para que um Cliente possa estimar o seu consumo, deverá ser possível **associar custos às suas chamadas com 2 períodos horários de taxaço, e 3 períodos semanais** (fim-de-semana, feriados e dias úteis).
- (b) Estimativa de Consumo por Utilizador
  - i. Possibilidade de permitir **determinar o consumo por utilizador**.
  - ii. Possibilidade ainda de **separar o consumo em empresarial e privado**, sendo que determinadas chamadas de um utilizador sejam consideradas com consumo privado e não da empresa.
  - iii. Esta separação deverá ser possível **com base em listas de contactos típicos da empresa ou em horários a definir pela empresa**.
- (c) Estimativa de Consumo em relatórios com detalhes das chamadas
  - i. Permitir a inclusão de informação sobre a **estimativa de consumo** nos relatórios com os detalhes das chamadas.
- (d) Custo Estimado total entre duas datas
  - i. Permitir obter informação do **custo estimado total em chamadas entre duas datas**.
- (e) Registo de Eventos para taxaço por Evento
  - i. São **relatórios/logs do histórico de eventos das actividades realizadas pelos utilizadores**, nas alterações de configuraçoes ou de utilizaço de dados serviços, **permitindo a posterior taxaço com base em eventos**.
- (f) Lista com Detalhes de Chamadas – Filtros
  - i. Deverá ser possível efectuar **filtros dos Relatorios por extensáo**, por **destino**, com filtros diferentes para **destinos** (locais, nacionais, internacionais, móveis) e por **intervalos de tempo**. Deverá ser possível efectuar **filtros por qualquer uma das colunas de informaçáo dos detalhes de chamadas**.

## Requisitos

- (g) Definição de Colunas - Informação a disponibilizar (v2)
  - i. Deverá conter as seguintes colunas:
    - A. N° de Sequência;
    - B. User;
    - C. Origem;
    - D. Destino;
    - E. Duração (hr:min:seg);
    - F. Detecção de Erro Ocorrido;
    - G. Custo estimado por chamada, iniciada ou terminada numa extensão GlobalPhone.
  - ii. Deverá também incluir a indicação de:
    - A. Ser uma chamada transferida;
    - B. Ser um reencaminhamento (e qual o tipo);
    - C. Ser uma conferência.
  - iii. Ter a indicação da chamada ser terminada, ou no caso de não ser, ter a **indicação do motivo** (não atendida (time-out), ocupada ou número inexistente).
- (h) Detalhes de chamadas - Definição de Colunas a Visualizar
  - i. Permitir **escolher qual a informação** a conter/visualizar num **relatório de detalhes de chamadas** ou na **listagem apresentada na Interface** em tempo real.
- (i) Gravação de Relatórios
  - i. Permitir **gravar os detalhes de chamadas** com as Filtragens e com as Colunas desejadas.
- (j) Download de Relatórios
  - i. Permitir executar o **download de listagens** de detalhes de chamadas gravados ou a ser listados de momento, através do download de um **ficheiro Excel**.
- (k) Análise de detalhes de chamadas - Em Gráficos (v2)
  - i. Ser um **gráfico é uma aposta em ser User Friendly**. Permitir observar ao longo do tempo (**dia, semana, mês, ano**) a evolução do número de chamadas efectuadas ou recebidas ou do custo estimado das chamadas. Esta análise poderá ser por Grupo ou Utilizador. Permite a avaliação dos recursos face a necessidades e algum auto-controlo.
- (l) Monitorização de chamadas a Decorrer (v2)
  - i. Permitir **observar em tempo real um gráfico ou uma tabela com as chamadas a decorrerem** (utilizadores, destinos, totais, durações, etc...).

## 11. Dial Plan

- (a) Dialplan específico por Utilizador e Grupo

## Requisitos

- i. Possibilidade de **configurar diferentes regras de dialplan por Grupo ou Utilizador.**
  - (b) Dial por nome ou extensão
    - i. Possibilidade de introduzir **nome de utilizador ou extensão para iniciar chamada.**
  - (c) Numeração de Extensões - Dígitos
    - i. Permitir a existência de **extensões de 1 a n dígitos**, quando for **necessário marcar 0 para sair.**
    - ii. No caso de não ser necessário marcação do zero, será **apenas possível a existência de extensões que não sejam parte do PNN**, tal como indicado no requisito 164.
12. Direct Line or Hot Line
- (a) Permitir que se possa **configurar que uma extensão**, quando activa, irá **automaticamente estabelecer uma chamada para um destinatário previamente indicado.**
13. Directório de Utilizadores
- (a) Cada Utilizador poderá **ver a lista de todos os Utilizadores que pertencem ao seu Grupo e Grupos dele dependentes.**
  - (b) A possibilidade de **ver toda a árvore de Utilizadores até ao topo da organização** a que pertence é uma funcionalidade que **pode ou não ser concedida.**
14. Directório Pessoal
- (a) Directório de Contactos Pessoal
    - i. Cada Utilizador deverá ter à sua disposição um **directório de contactos pessoais** onde poderá **colocar os contactos** que pretender.
  - (b) Click to Call
    - i. Possibilidade de **clicar num dos contactos** do directório de contactos disponível na interface para que se **estabeleça automaticamente uma chamada entre o terminal do utilizador (hard ou soft) e o destino seleccionado.**
  - (c) Directório de Contactos Pessoal - On-Click
    - i. Cada Utilizador deverá ter a possibilidade de **efectuar chamadas On-Click**, directamente a **partir dos directórios de contactos disponíveis.**
15. Last Number Redial
- (a) Permitir que pela **introdução de um código** se possa **estabelecer automaticamente uma chamada para o último número marcado.**

16. Lista VIP Pessoal

- (a) Cada Utilizador deverá ter a possibilidade de **configurar uma Lista VIP**.
- (b) Todos os Contactos incluídos nesta Lista VIP poderão **efectuar directamente chamadas para o utilizador, não sofrendo reencaminhamentos para secretárias ou operadoras, ou mesmo no caso do utilizador se encontrar no estado DND (*do not disturb*)**.

17. Media Server

- (a) Disponibilização de Audio on Hold
  - i. Quando um Utilizador ou um chamador de fora do Sistema forem **colocados em Espera (*on hold*)**, será lhes enviado pelo Media Server algo.
- (b) Upload de Música
  - i. Para os Grupos, Utilizadores que tenham **disponíveis esta funcionalidade**, é possível carregar músicas via interface web.
  - ii. No upload para o sistema central, **todas as músicas têm de ser classificadas**.
  - iii. Só pode haver uma **música standard**.
  - iv. A nomeação de uma música standard, **quando já existe outra com essa classificação**, só pode ser **efectuada depois de uma dupla confirmação pelo Administrador (Nível 0)**.
  - v. Uma música anteriormente carregada pode ser **eliminada**, com **excepção da standard** que tem de permanecer sempre no sistema.
- (c) Classificações de Músicas
  - i. As músicas têm quatro classificações:
    - A. A **música standard (Nível 0)** que é usada na ausência de outra configuração;
    - B. **Pertencentes ao Grupo de Topo (Nível 1)** à organização. Todos os Utilizadores têm acesso a estas músicas;
    - C. Pertencentes a um Grupo de **Nível N** que apenas é **válido e está disponível para esse Grupo** e para os que se lhe derivarem (Nível N+1);
    - D. As **músicas dos Utilizadores** que só podem ser usadas por este.
- (d) Música Personalizada por Receptor
  - i. A **música a aplicar** poderá ser **diferente por Grupo ou por Utilizador**.
  - ii. O sistema deve permitir aos **Utilizadores e Grupos** que tenham esta **funcionalidade activa a possibilidade de escolher músicas para situações de *on hold***.
- (e) Musica Personalizada por Chamador

## Requisitos

- i. Permitir ao utilizador **escolher qual a música a ouvir enquanto espera por ser atendido**, via menu **IVR**, através de **DTMF's**.
18. Provisão automática dos clientes
  - (a) Assim que um qualquer nível de utilizador **altera algum parâmetro na configuração do terminal, via a Interface, isto repercute-se no ficheiro gerado no servidor de TFTP**, que será acedido regularmente pelos terminais.
  - (b) Deverá ser efectuado através de um **Servidor TFTP (inicialmente)**.
  - (c) O AS irá comunicar com uma entidade que fará de interface única para o TFTP, tendo esta toda a lógica de comunicação com o Servidor TFTP (ou outro), **adaptando ao Terminal as configurações indicadas pelo AS**.
  - (d) Esta entidade será ainda **responsável por adaptar o pedido de alteração de configuração de um terminal** (EX: terminais Dummy), sendo que deverá dar a indicação ao terminal que terá que obter junto do TFTP uma nova configuração.
  - (e) O **AS deverá comunicar as alterações efectuadas a nível de interface de utilizador** que impliquem alterações a nível da configuração dos terminais a esta entidade.
19. Remote Call Access - WEB Callback - Virtual Calling Card
  - (a) Permitir aceder via Internet ou VPN ao sistema com o intuito de permitir realizar chamadas através do PBX.
  - (b) A ideia é, **a pedido, ser o próprio PBX a executar a chamada entre dois números**.
  - (c) Tem-se um serviço que reduzirá o custo das chamadas quando um utilizador de um dado cliente se deslocar para fora de PT, trazendo tráfego para a Rede.
  - (d) Esta funcionalidade deverá ser **protegida por certificação e por password**.
  - (e) Para impedir abusos, será feito um **track das chamadas efectuadas** com base na extensão do utilizador que pede para efectuar estas chamadas.
20. Seize Call - Interrupção de Chamada
  - (a) Permitir que um Utilizador possa **retirar uma chamada de outro Utilizador**, terminando a chamada com esse outro Utilizador e **apoderando-se da chamada**.
21. User Profiles
  - (a) Multiplicidade de Perfis de Utilizador/Configuração

## Requisitos

- i. Para evitar constantes alterações a configuração de utilizador e para **tornar o funcionamento mais User Friendly**, possibilitar a configuração de diferentes **perfis**, por exemplo perfil **Noite** VS perfil **Dia** ou **Trabalho** VS **Férias**.
    - ii. Cada utilizador poderá ter, **por exemplo até 3 perfis**.
    - iii. O Utilizador **só poderá ter um perfil activo em cada momento** e terá a hipótese de alternar entre perfis sempre que quiser.
  - (b) Perfis Genéricos aplicáveis a vários Utilizadores
    - i. Deverão existir **Perfis genéricos a aplicar pelo Gestor de Grupo a todos os Utilizadores desse grupo**.
    - ii. Os **Perfis configurados pelo próprio Utilizador serão prioritários** aos definidos pelo Gestor do Grupo.
  - (c) Perfis Genéricos - Night Service
    - i. **Reencaminhamento Automático para número a definir, fora do horário definido como laboral no serviço** para um dado Utilizador.
22. Voice Mail
  - (a) Voicemail de Grupo ou Utilizador – Externo
    - i. A cada **extensão** deve ser possível associar **uma caixa de voicemail externa** - recorrendo, por exemplo, ao **voicemail da PTC**.
    - ii. Implementação fácil:
      - A. Suficiente o encaminhamento para a actual plataforma de voicemail.
    - iii. O voicemail vai reencaminhar a chamada para o número “+0xxxxxxxxx” sendo que os x correspondem ao DDI da extensão.
    - iv. No reencaminhamento para o voicemail deve ser mantido o número de origem, de forma a que no voicemail possa ficar registado o número de telefone da pessoa que ligou.
  - (b) Voicemail de Grupo ou Utilizador – Interno
    - i. A cada extensão deve ser possível associar uma caixa de voicemail interna, isto é, integrar com o Exchange de forma a que nativamente suporte **voicemail com notificações por email**.
23. Wake Up Call
  - (a) Chamadas de Lembrança
    - i. Permitir ao utilizador **agendar chamadas automáticas que o avisem de algum compromisso, como acordar a uma dada hora**.
  - (b) Mensagens Gravadas para Memo's
    - i. Permitir ao utilizador proceder à gravação de **audio no sistema de forma a que lhe seja transmitido essa gravação aquando da chamada automática se realizar**.

### A.1.5 Grupos Tipo

#### 1. Hunting Groups - Round Robin

- (a) **Ligando para um determinado número toca nos vários telefones que estiverem definidos nesse grupo, um de cada vez, até um deles atender (*round-robin*).**
- (b) Deverá ser possível **configurar qual a ordem dos terminais a tocar.**
- (c) **Só se pode atender a chamada no terminal que estiver a tocar no momento.**
- (d) Deverá ser configurável, ao **critério do administrados de cada Grupo**, se nesse grupo **prevalencem as regras de utilizador ou as regras do Hunting Group.**

#### 2. Call Forking - Chamadas em Paralelo

- (a) Permitir **configurar grupos onde extensões associadas a utilizadores do grupo toquem todos ao mesmo tempo e que a chamada seja estabelecida com o terminal que a atender primeiro.**
- (b) Permitir ao IPCentrex que, quando recebe o pedido de uma chamada para um dado Utilizador, a **encaminha para os vários terminais configurados pelo utilizador** para fazerem parte do **Forking Group.**
- (c) Ao atender um dos terminais do grupo, **todos os outros deixam de tocar.**
- (d) O **número máximo** de terminais a colocar num grupo poderá ser **limitado pelo Serviço e também pelo Gestor de Grupo do Utilizador.**

#### 3. Intercom Groups

- (a) Permitir configurar grupos onde extensões associadas a utilizadores do grupo possam comunicar por intercomunicador - do género **walkie-talkie, pressionando um botão.**
- (b) Tem que ser suportado também no **terminal.**
- (c) É uma funcionalidade **interessante para Secretária-Chefe ou grupos de individuos da mesma equipa**, que não se encontrem no mesmo espaço.

#### 4. Hunting Groups

- (a) Round Robin Início Sequencial
  - i. Deverá ser possível activar a possibilidade de cada vez que uma chamada é efectuada para este grupo, o **primeiro terminal a tocar seja o seguinte na lista** ao que tocou na chamada anterior.
- (b) Round Robin em função de Utilização

## Requisitos

- i. Deverá ser possível activar a possibilidade de cada vez que uma chamada é efectuada para este grupo, o **primeiro terminal a tocar seja a extensão menos utilizada** até ao momento.
- (c) Round Robin Aleatório
  - i. Deverá ser possível activar a possibilidade de cada vez que uma chamada é efectuada para este grupo, o **primeiro terminal a tocar seja escolhido aleatoriamente** da lista de extensões do grupo.

### A.1.6 Provisão

1. Envio de Numeração final para Gateway
  - (a) **Não haver necessidade** de alterar nada na **gateway manualmente**. Ou seja, a gateway ser um elemento passivo na arquitectura do IP Centrex. Para tal, é necessário, por exemplo, que o IP Centrex garanta que é enviada para a Gateway pedidos com a Numeração correcta de saída para a PSTN.
2. Terminais
  - (a) Inventário de Terminais
    - i. **Inventário dos terminais**, com a descrição das suas características, funcionalidades.
    - ii. Para facilitar o sistema de provisão e facilitar o controlo de terminais. Deve conter o **número de terminais activos** de cada modelo.
    - iii. Os terminais a incluir neste inventário serão somente os **testados e aceites no funcionamento com o IP Centrex**.
    - iv. Existe ficheiro de configuração de base para cada terminal.
  - (b) Provisão de Novos Terminais
    - i. Sempre que se pretender vender um novo terminal, a interface e o sistema estarem preparados para **suportar a fácil criação da provisão de outro terminal, via TFTP**.
  - (c) Provisão automática dos Parâmetros de Cliente nos Terminais
    - i. Assim que um qualquer nível de utilizador altera algum parâmetro na **configuração do terminal**, via a Interface, isto repercute-se no **ficheiro gerado no servidor de TFTP**, que será acedido regularmente pelos terminais.

### A.1.7 Restrições

1. Restrição do número máximo de chamadas simultaneas
  - (a) Deverá ser possível definir o **número máximo de chamadas simultaneas por Grupo**. Esta definição é imposta pelo **Grupo de Nível N** aos grupos dele **dependentes de Nível N+1**.

## Requisitos

- (b) Um Grupo que seja um **Site terá a restrição do número de chamadas máximo em simultâneo que todos os Grupos/Utilizadores desse Site podem efectuar**, limitados em função do acesso VoIP do Site.
2. Restrição do número máximo de chamadas simultaneas Contratadas no Acesso
    - (a) Deverá ser possível definir o **número máximo de chamadas simultaneas por acesso VoIP do Site**, independentemente da LB disponível para o Cliente, sendo limitado com base no número de chamadas de voz simultaneas permitidas contratualmente para um dado acesso.
  3. Controlo da LB utilizada pelo Cliente nas suas chamadas para garantir QoS
    - (a) Deverá ser possível **controlar o número de chamadas e o respectivo codec utilizado**, sendo somada a LB ocupada.
    - (b) Só se deverá permitir fazer uma chamada quando ainda **existe LB de banda suficiente** no acesso que permita a utilização do Codec a ser utilizado nessa chamada. Esta análise deverá ser efectuada logo no momento do INVITE.
    - (c) O IP Centrex deverá alterar o conteúdo da mensagem, **seleccionando os Codecs cuja LB ainda é permitida**. Caso não haja nenhum disponível, então é barrada a chamada.
  4. Restrição de Tipo de Chamada
    - (a) Possibilidade de **restringir o tipo de chamadas** de cada **extensão**, por exemplo:
      - i. Apenas chamadas internas e para números de emergência.
      - ii. Apenas chamadas para a rede fixa nacional.
      - iii. Apenas chamadas para a rede móvel e a rede fixa nacional.
      - iv. Todas as chamadas.
  5. OutBound - Call Bloking
    - (a) Permitir identificar certos números de telefone ou **prefixos de zona para onde não serão aceites chamadas**, automaticamente.
    - (b) Permite **poupar tempo aos trabalhadores e dinheiro**.
    - (c) Permite impossibilitar chamadas para zonas onde se sabe **não se poder fazer negócio**.
  6. InBound - Call Bloking - BlackLists
    - (a) Permitir identificar Utilizadores ou Grupos que **não possam receber chamadas do exterior** ou de outros Grupos ou Utilizadores.
    - (b) Permite garantir BlackLists, para restringir acesso a **Spammers**.
  7. Extensões de Atendimento

## Requisitos

- (a) Permitir identificar **Utilizadores ou Grupos** que só possam **receber chamadas**. Não as podem efectuar.

### 8. Extensões de Execução de Chamadas

- (a) Permitir identificar **Utilizadores ou Grupos** que só possam **efectuar chamadas**. Não as podem receber.

### 9. Numeração

- (a) Restrição do número de DDI's

- i. A cada Grupo estão atribuídos um **conjunto de DDI's** (números discretos ou intervalos de numeração) que podem ser usados exclusivamente por esse Grupo.
- ii. Os DDI's disponíveis num determinado Grupo podem ser reservados para serem distribuídos em **Grupos de Nível igual (N)** ou **Inferior (N+1)**. Este processo é recursivo.

- (b) Restrição do número de Extensões

- i. A cada Grupo estão atribuídos um conjunto de **números de extensão** (números discretos ou intervalos de numeração) que podem ser **usados em mais do que Grupo Cliente (Nível 1)**.
- ii. Os números disponíveis num determinado Grupo de Nível N podem ser reservados para serem distribuídos em outros **Grupos dependentes deste e de Nível N+1**. Este processo é recursivo.

- (c) Garantir a Restrição de Numeração

- i. O IP Centrex deverá garantir que um dado **Grupo só utiliza a numeração (extensão ou DDI) que lhe foi exclusivamente atribuída**.
- ii. Caso este procure efectuar chamadas com **numeração fora da que lhe foi atribuída**, as chamadas serão **rejeitadas**.

- (d) Gestão da Restrição da Numeração

- i. Cada Utilizador **Gestor de um Grupo** tem disponível uma **interface de gestão de numeração disponível**, onde está visível a **numeração de DDI's e Extensões** de facto utilizada e a não utilizada nesse Grupo (Nível N) e em todos os que dependam dele (Nível N+1).
- ii. É possível nessa interface **recuperar numeração não utilizada e associa-la a qualquer outro Grupo**.
- iii. A numeração recuperada fica obviamente **indisponível** para ser usada onde **previamente tinha sido alocada**.

# Anexo B

## Plano de Trabalho Final

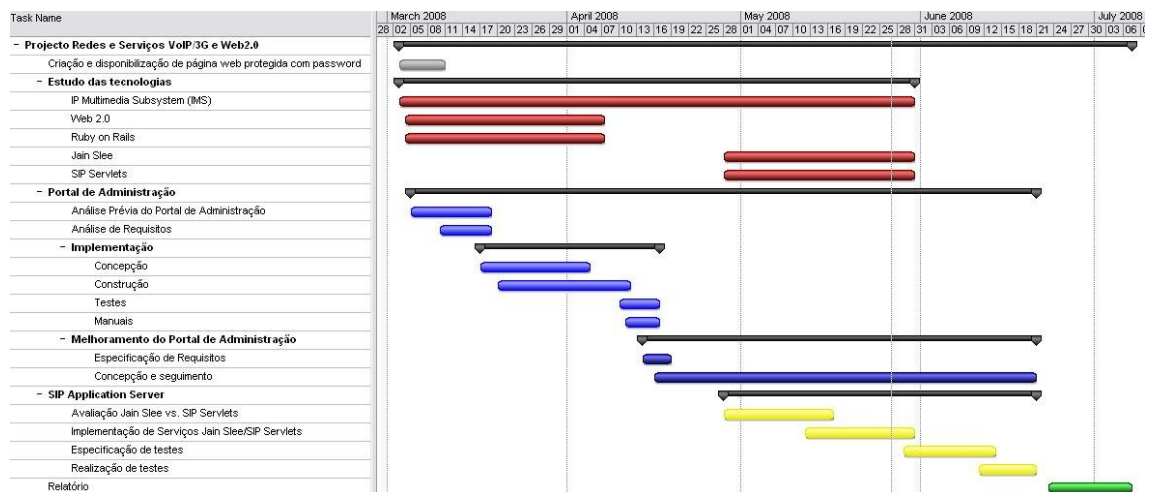


Figura B.1: Plano de Trabalho Final

## Plano de Trabalho Final

## Anexo C

# Wireframe da segunda versão do Portal

The image shows a wireframe of the IPCentrex portal. The top navigation bar includes the logo 'IPCentrex' and links for 'Áreas', 'Plataforma', 'Conta pessoal', and 'administrador'. A user profile 'administrador' and a 'sair' button are also visible. A language selector 'idioma' is present in the top right. Below the navigation bar, a breadcrumb trail reads 'Áreas / Área A / Área 1'. The main content area is titled 'Formulário de edição' and contains a 'Detalhes' section with the following fields:

<b>Grupo Pai:</b>	
PT	Descrição <input type="text" value="Empresa"/>
--Mentes	Nome <input type="text"/>
--Comercial	Domínio <input type="text"/>

On the right side of the form, there is a search section with a 'pesquisa' input field, an 'ok' button, and a 'pesquisa avançada' link. Below this is a '+ adicionar área' button. Further down, a 'Filtros' section contains a 'listar áreas a que pertença' link and a 'listar todas as áreas' link. At the bottom, an 'Informações Área A A1' section displays statistics: 'Nr de Utilizadores: ---', 'Nr de Terminais: ---', and 'Contacto gestor: ---'.

Figura C.1: Definições

# Wireframe da segunda versão do Portal

IPCentrex

Áreas | Plataforma | Conta pessoal [administrador] | sair

idioma

Definições Utilizadores Funcionalidades Terminais Numeração Perfis Relatórios

Áreas / Área Δ / Área 1

### Utilizadores

E-mail	Nome	Extensão	Acções
ze@mv.pt	Zé Manel	100	<a href="#">editar</a> <a href="#">remover</a>
joao@mv.pt	João Miguel	100	<a href="#">editar</a> <a href="#">remover</a>
pedro@mv.pt	Pedro Bia	100	<a href="#">editar</a> <a href="#">remover</a>
ana@mv.pt	Ana Maria	100	<a href="#">editar</a> <a href="#">remover</a>
rita@mv.pt	Rita Sofia	100	<a href="#">editar</a> <a href="#">remover</a>
rui@mv.pt	Rui To	100	<a href="#">editar</a> <a href="#">remover</a>

pesquisa

ok

pesquisa avançada

[+ adicionar utilizador](#)

#### Filtros

[Listar por sub áreas](#)

#### Acções

- Atribuir numeração
- Gerir funcionalidades
- Delegar privilégios

Inovação

Figura C.2: Utilizadores

IPCentrex

Áreas | Plataforma | Conta pessoal [administrador] | sair

idioma

Definições Utilizadores Funcionalidades Terminais Numeração Perfis Relatórios

Áreas / Área Δ / Área 1

### Funcionalidades

Descrição	Disponível
Voicemail	<input type="checkbox"/>
Voicemail	<input type="checkbox"/>

#### Filtros

[Esconder as que não têm permissões](#)

Inovação

Figura C.3: Funcionalidades

# Wireframe da segunda versão do Portal

IPCentrex

Áreas | Plataforma | Conta pessoal [administrador] | sair

idioma

Definições Utilizadores Funcionalidades **Terminais** Numeração Perfis Relatórios

Áreas / Área A / Área 1

### Terminais

MAC	Marca	Modelo	IP	Extensões Associadas	Acções
AA::FF:FF:FF:FF	Linksys	Voip	10.0.0.1	100, 102	<a href="#">editar</a> <a href="#">remover</a>
AA::FF:FF:FF:FF	Linksys	Voip	10.0.0.1	100, 102	<a href="#">editar</a> <a href="#">remover</a>
AA::FF:FF:FF:FF	Linksys	Voip	10.0.0.1	100, 102	<a href="#">editar</a> <a href="#">remover</a>
AA::FF:FF:FF:FF	Linksys	Voip	10.0.0.1	100, 102	<a href="#">editar</a> <a href="#">remover</a>

pesquisa

ok

pesquisa avançada

+ adicionar terminal

**Acções**

Associar extensões

Inovação

Figura C.4: Terminais

IPCentrex

Áreas | Plataforma | Conta pessoal [administrador] | sair

idioma

Definições Utilizadores Funcionalidades Terminais **Numeração** Perfis Relatórios

Áreas / Área A / Área 1

### Numeração

Nr	Utilizador	Acções
099	Manuel@xpto.pt	<a href="#">remover</a>
100	Pedro@xpto.pt	<a href="#">remover</a>
101	Zé@xpto.pt	<a href="#">remover</a>
102	Ana@xpto.pt	<a href="#">remover</a>

pesquisa

ok

pesquisa avançada

+ adicionar numeração interna

**Listar**

Numeração

[Numeração externa](#)

**Filtros**

[Listar por área a que pertença](#)

[Listar por sub área](#)

[Listar por disponíveis](#)

[Listar por reservados](#)

[Listar por atribuídos](#)

Inovação

Figura C.5: Numeração

# Wireframe da segunda versão do Portal

IPCentrex

Áreas | Plataforma | Conta pessoal [administrador] | sair

kioma

Definições Utilizadores Funcionalidades Terminais Numeração Perfis Relatórios

Áreas / Área A / Área 1

### Perfis

Funcionalidades	Trabalho	Casa	Férias
Reencaminhamento de chamadas	<input type="checkbox"/> <a href="#">detalhes</a>	<input type="checkbox"/> <a href="#">detalhes</a>	<input type="checkbox"/> <a href="#">detalhes</a>
Voice mail	<input type="checkbox"/> <a href="#">detalhes</a>	<input type="checkbox"/> <a href="#">detalhes</a>	<input type="checkbox"/> <a href="#">detalhes</a>
Sieze Call	<input type="checkbox"/> <a href="#">detalhes</a>	<input type="checkbox"/> <a href="#">detalhes</a>	<input type="checkbox"/> <a href="#">detalhes</a>

importar perfis de grupo

Inovação

Figura C.6: Perfis

IPCentrex

Áreas | Plataforma | Conta pessoal [administrador] | sair

kioma

Definições Utilizadores Funcionalidades Terminais Numeração Perfis Relatórios

Áreas / Área A / Área 1

### Relatórios

Nr	Data	Hora	Nr Destino	Duração chamadas	Valor com IVA	Remover
1	06-03-2008	19:38	234123456	0:05:00	1,010	<a href="#">remover</a>
2	07-03-2008	19:38	234123456	0:06:38	1,020	<a href="#">remover</a>
3	08-03-2008	19:38	234123456	0:07:11	0,150	<a href="#">remover</a>
4	09-03-2008	19:38	234123456	0:06:31	1,036	<a href="#">remover</a>
5	09-03-2008	19:38	234123456	0:04:20	0,159	<a href="#">remover</a>
6	10-03-2008	19:38	234123456	0:06:31	1,036	<a href="#">remover</a>

pesquisa

ok

pesquisa avançada

exportar

Filtros

[Listar por área](#)

[Listar por utilizador](#)

Inovação

Figura C.7: Relatórios

Anexo D

## Modelos de Datos



# Modelos de Dados

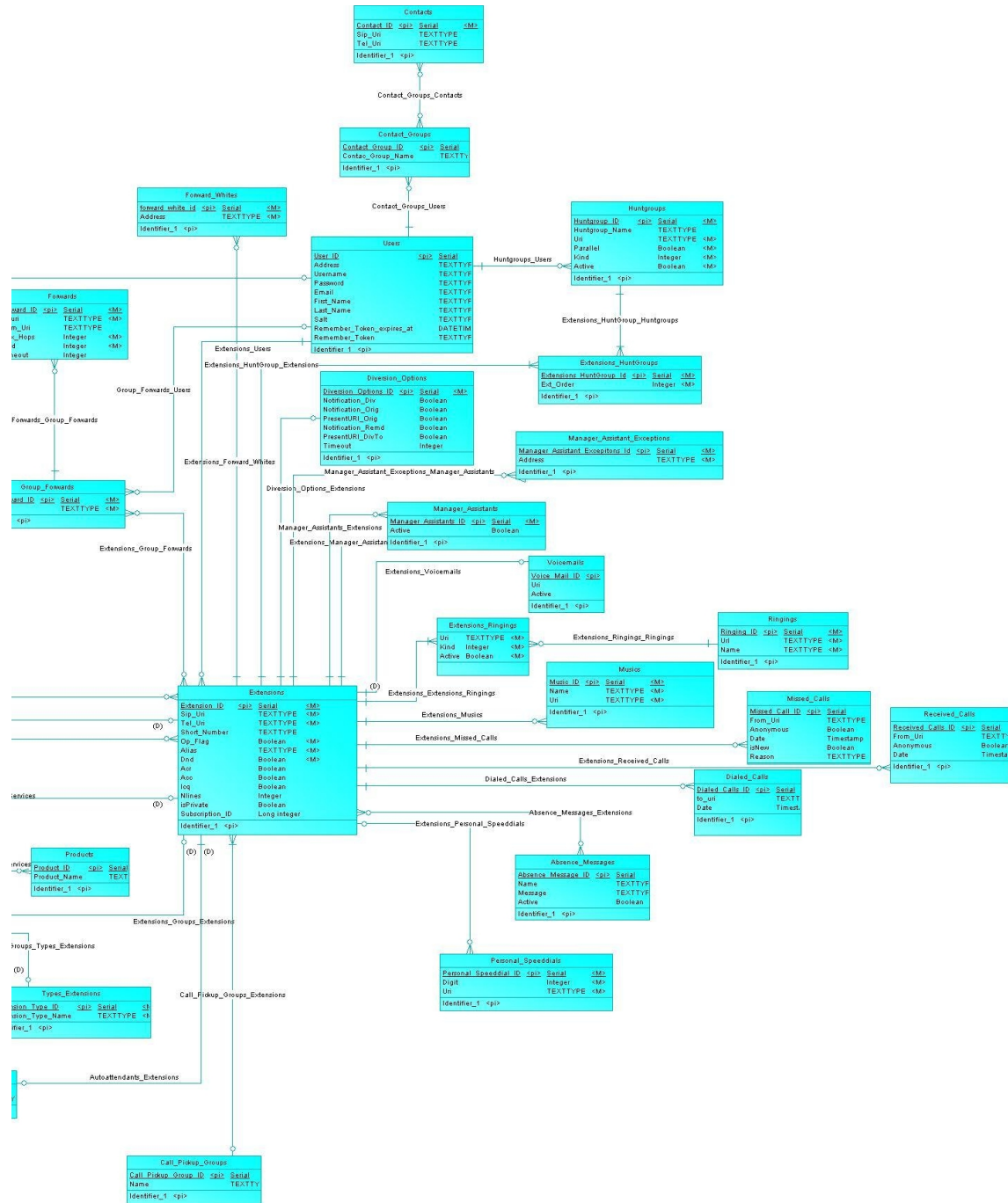


Figura D.2: Primeira Versão Modelo de Dados - Parte 2/2



# Modelos de Dados

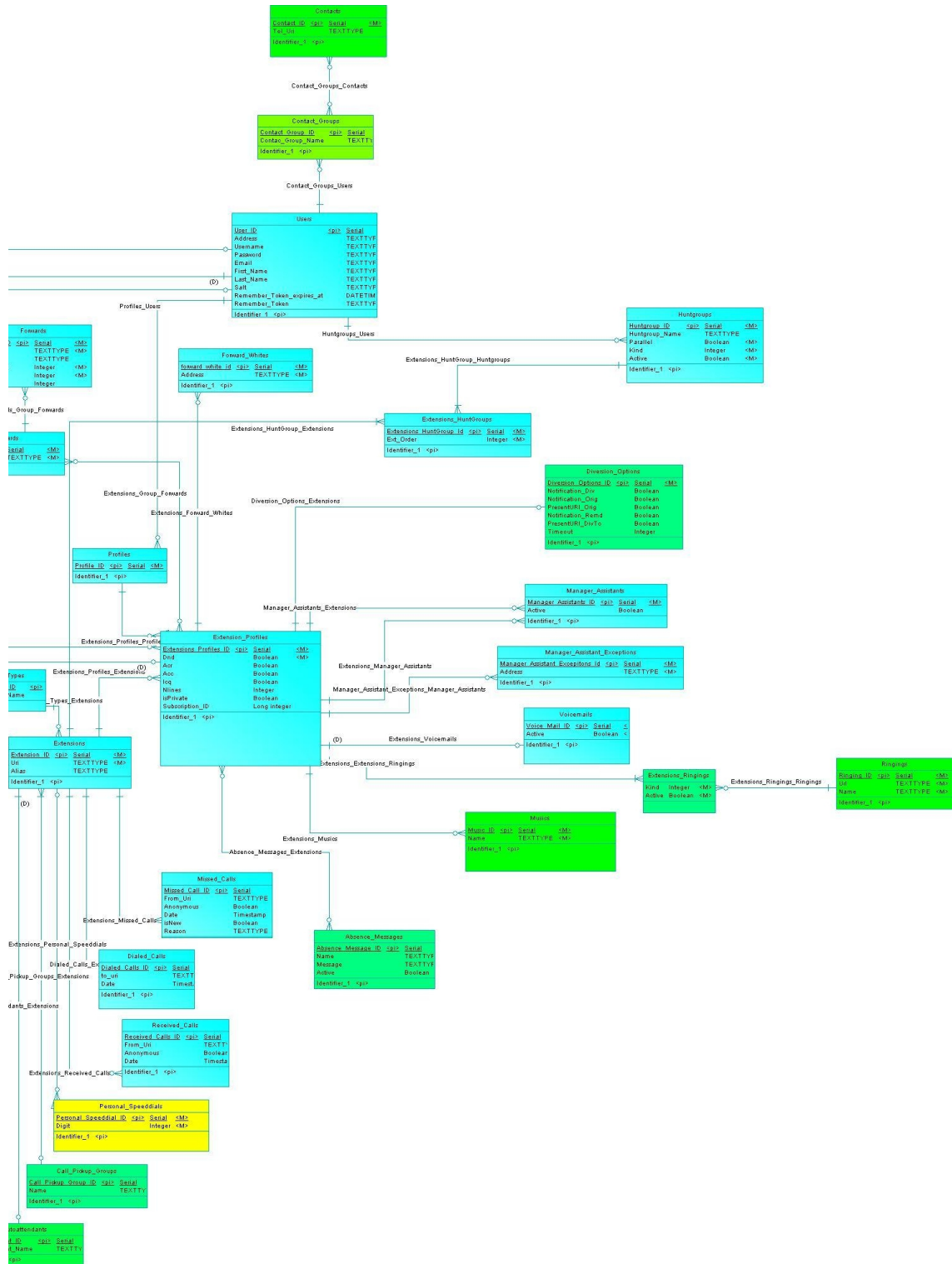


Figura D.4: Segunda Versão Modelo de Dados - Parte 2/2