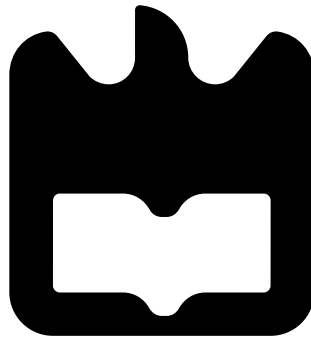




**Daniel Filipe  
Nunes Marques**

**Comunicações unificadas em dispositivos móveis  
Mobile Unified Communication**



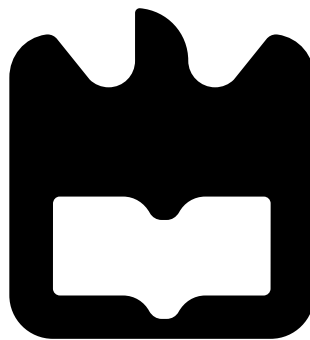




**Daniel Filipe Nunes  
Marques**

**Comunicações unificadas em dispositivos móveis  
Mobile Unified Communication**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Diogo Nuno Pereira Gomes, Professor Auxiliar Convidado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor Rui Luís Aguiar, Professor Associado com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.





**o júri**

presidente

**Doutor José Carlos da Silva Neves**

Professor Catedrático, Universidade de Aveiro

vogais

**Doutor Diogo Nuno Pereira Gomes**

Professor Auxiliar Convidado da Universidade de Aveiro (orientador)

**Doutor Rui Luís Andrade Aguiar**

Professor Associado com agregação da Universidade de Aveiro (co-orientador)

**Doutor Pedro Miguel Naia Neves**

Investigador, PT Inovação



## **agradecimentos**

Aproveito para agradecer à minha família e especialmente aos meus pais Manuel Marques e Aurora Nunes, como também ao meu irmão Tiago Marques por todo o apoio constante em toda minha vida e em particular no meu percurso académico, no qual sem eles muito difícilmente teria atingido este desfecho.

Quero ainda agradecer a todos os meus amigos que fizeram parte desta experiência académica, no entanto tenho de realçar os meus amigos mais próximos que me acompanharam sempre em todos os momentos.

Por último quero ainda agradecer ao meu orientador Diogo Gomes pelo suporte e paciência demonstrada durante o desenvolvimento da minha dissertação.





**Palavras-chave**

Computação Móvel, Android, Serviços, Telecomunicações, Usabilidade, .

**Resumo**

Com a constante evolução tecnológica, estão frequentemente a surgir novos métodos de comunicação. Acompanhar a evolução das telecomunicações não é um processo fácil, sendo que a criação de novas aplicações e serviços leva ao conseqüente desaproveitamento do seu potencial, por via da falta de familiaridade do utilizador com os mesmos. A abordagem desta dissertação tem como foco explorar esta noção ao reunir os serviços mais recentes num só protótipo de aplicação móvel. As funcionalidades pretendidas para o protótipo estão divididas em três vertentes tecnológicas: realização de chamadas de voz, envio de mensagens e recolha de dados pessoais. Os protocolos SIP, XMPP, SMTP e IMAP são alguns dos quais serviram para a implementação dos serviços, utilizando interfaces de programação incorporadas no sistema operativo Android. Ao integrar estes elementos, o resultado obtido foi o de uma aplicação de telecomunicações versátil e completa. Apesar dos problemas de ergonomia encontrados, o protótipo final alcançou os objectivos pretendidos, recolhendo bons resultados nos testes de usabilidade efectuados, despertando também o interesse de alguns utilizadores pela inovação oferecida face às aplicações tradicionais que possuem apenas um serviço de comunicações.



**Keywords**

Mobile Computing, Android, Services, Telecommunications, Usability.

**Abstract**

With the constant evolution of technology, new ways of communication are created everyday. Keeping current with this evolution in telecommunications isn't an easy process: the creation of new applications and services, and the associated lack of familiarity with the new technology and applications, frequently leads to a sub-optimal use. This dissertation is focused on exploring this notion by gathering the most recent communication services on a single mobile application prototype. The target functionality for this prototype is divided in three technological features: voice call service, messaging service and personal data retrieval. The protocols used are SIP, XMPP, SMTP and IMAP, making use of native or third party libraries in the Android OS. By integrating these elements, the resulting prototype was robust and versatile, allowing for a more unified user experience. Despite some ergonomic problems, the final prototype achieved the desired results in the usability tests performed and raised interest from some users due to this unified experience – something which is not present in traditional, single service, communication applications.



# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>v</b>
<b>Lista de Tabelas</b>	<b>vii</b>
<b>Lista de Acrónimos</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contexto e motivação . . . . .	1
1.2 Objectivos . . . . .	1
1.3 Estrutura da dissertação . . . . .	2
<b>2 Estado da arte</b>	<b>3</b>
2.1 VoIP . . . . .	3
2.1.1 Skype . . . . .	3
2.1.2 Twilio . . . . .	5
2.2 Protocolo SIP . . . . .	6
2.3 3GPP IMS . . . . .	8
2.4 Comunicações unificadas . . . . .	10
2.4.1 RIM . . . . .	11
Blackberry MVS . . . . .	12
2.4.2 Nortel e Avaya . . . . .	12
2.4.3 Web 2.0 . . . . .	12
Youtube . . . . .	13
eBay / Amazon . . . . .	13
2.4.4 Social Networks . . . . .	13
Facebook . . . . .	14
LinkedIn . . . . .	14
2.5 Dispositivos móveis . . . . .	14
2.5.1 iMessage . . . . .	14
2.5.2 Windows Phone Messaging . . . . .	15
2.5.3 Google+ Hangouts . . . . .	15
2.5.4 Multiplataforma . . . . .	15
Viber . . . . .	15
WhatsApp . . . . .	16

<b>3</b>	<b>Requisitos do sistema</b>	<b>17</b>
3.1	Desenvolvimento	17
3.1.1	iOS	19
3.1.2	Windows Phone 7	19
3.1.3	Android	20
3.2	Estruturais	21
3.3	Funcionais	23
3.3.1	Registo do Utilizador	24
	Pre-Condições:	24
	Diagrama de Sequência	25
3.3.2	Configuração dos Serviços	25
	Pre-Condições:	25
3.3.3	Gestão de Contas	26
	Pre-Condições:	26
3.3.4	Sincronização de Dados	26
	Pre-Condições:	26
	Diagrama de Sequência	27
3.3.5	Chamadas de voz	27
	Pre-Condições:	27
	Diagrama de Sequência	28
3.3.6	Mensagens de texto	28
	Pré-Condições:	28
	Diagrama de Sequência	29
3.3.7	Registo de serviços efectuados	30
	Pré-Condições:	30
<b>4</b>	<b>Arquitectura do sistema</b>	<b>31</b>
4.1	Estrutura física	31
4.2	Base de Dados	31
4.2.1	Evolução do Sistema	31
4.2.2	Solução	34
<b>5</b>	<b>Implementação</b>	<b>41</b>
5.1	Modelo gráfico do protótipo	41
5.1.1	Maqueta gráfica	41
5.1.2	Interfaces gráficas	44
5.2	APIs do protótipo	50
5.2.1	SIP Manager	50
5.2.2	Facebook SDKs e APIs	50
5.2.3	LinkedIn API	51
5.2.4	XMPP	51
5.2.5	EMAIL	52
5.3	Implementações incompletas	52
5.3.1	Implementação VoIP	52
5.3.2	face.com API	53
5.3.3	Unificação de Contactos	54

<b>6 Usabilidade do protótipo</b>	<b>55</b>
6.1 Tarefas realizadas . . . . .	55
6.2 Resultados iniciais . . . . .	56
6.2.1 Análise global . . . . .	56
6.2.2 Análise detalhada de tarefas . . . . .	58
6.2.3 Observações Iniciais . . . . .	59
6.3 Resultados após melhoramento . . . . .	59
6.3.1 Análise global . . . . .	59
6.3.2 Observações finais . . . . .	60
<b>7 Conclusão</b>	<b>61</b>
7.1 Trabalho futuro . . . . .	62
<b>Bibliografia</b>	<b>63</b>





# Lista de Figuras

2.1	SILK Random Loss [10]	4
2.2	Twilio : Exemplo de funcionamento	5
2.3	Twiml: Exemplo de utilização [12]	6
2.4	Estrutura da mensagem SIP	6
2.5	Request-Line	6
2.6	Status-Line	7
2.7	Header SIP	7
2.8	Processo de Autenticação	8
2.9	Arquitectura IMS - Visão Geral	9
2.10	Arquitectura IMS	10
3.1	Exemplo : Programação baseada em eventos [24]	17
3.2	Arquitectura MVC	18
3.3	Percentagem mundial de Smartphones [25,26]	18
3.4	Ciclo de vida da actividade	21
3.5	Arquitectura Estrutural	22
3.6	Cenário de Uso : Gestão de Contas	24
3.7	Cenário de Uso : Serviços de Comunicação	24
3.8	Registo do Utilizador e Configuração	25
3.9	Sincronização e unificação de contactos	27
3.10	Serviço de Chamadas de voz	28
3.11	Serviço de mensagens de texto	29
4.1	Primeira versão	32
4.2	Segunda versão	33
4.3	Versão Final	34
5.1	Ecrã principal	41
5.2	Ecrã serviços	42
5.3	Ecrã configuração	42
5.4	Ecrãs de contactos	43
5.5	Ecrã do histórico de serviços	43
5.6	Ecrã de principal	44
5.7	Ecrã de configuração do sistema	45
5.8	StateChart: Fluxo da aplicação, parte 1	45
5.9	Ecrãs de contactos	46
5.10	Informações do contacto	47

5.11	StateChart: Fluxo da aplicação, parte 2 . . . . .	48
5.12	Ecrã de serviços 1 . . . . .	48
5.13	Ecrã de serviços 2 . . . . .	49
5.14	Mensagens recebidas . . . . .	50
5.15	Estruturas das Pacotes . . . . .	53
6.1	Gráfico de duração das tarefas . . . . .	57
6.2	Gráfico de erros por tarefa . . . . .	57
6.3	Gráfico de satisfação . . . . .	58
6.4	Gráfico da duração de execução das tarefas - segunda experiência . . . . .	60

# Lista de Tabelas

2.1	Ecrãs de configuração . . . . .	11
3.1	Windows Phone 7 . . . . .	20
3.2	Requisitos funcionais. . . . .	23
3.3	Ecrãs de configuração . . . . .	26
4.1	Tabela Person . . . . .	35
4.2	Tabela User . . . . .	35
4.3	Tabela Contacts . . . . .	35
4.4	Tabela Phone . . . . .	35
4.5	Tabela VoIPProfile . . . . .	35
4.6	Tabela ServerVoIP . . . . .	36
4.7	Tabela Account . . . . .	36
4.8	Tabela AccountUser . . . . .	36
4.9	Tabela AccountContacts . . . . .	36
4.10	Tabela Address . . . . .	37
4.11	Tabela Mail . . . . .	37
4.12	Tabela XMPP . . . . .	37
4.13	Tabela Server . . . . .	38
4.14	Tabela ServicesData . . . . .	38
4.15	Tabela Message . . . . .	38
4.16	Tabela Call . . . . .	39
4.17	Tabela Group . . . . .	39
4.18	Tabela GroupMembers . . . . .	39
5.1	Análise na unificação de dados . . . . .	54



# Lista de Acrónimos

<b>VOIP</b>	Voice over Internet Protocol
<b>XMPP</b>	Voice over Internet Protocol
<b>SIP</b>	Session Initiation Protocol
<b>IMS</b>	IP Multimedia Subsystem
<b>IVR</b>	Interactive voice response
<b>IM</b>	Instant Messaging
<b>CSCF</b>	Call Session Control Function
<b>P-CSCF</b>	Proxy-CSCF
<b>I-CSCF</b>	Interrogating-CSCF
<b>S-CSCF</b>	Serving-CSCF
<b>HSS</b>	Home Subscriber Server
<b>AS</b>	Application Server
<b>3GPP/3GPP2</b>	Third Generation Partnership Project
<b>MVC</b>	Model-View-Controller
<b>MVS</b>	Model Voice System
<b>RIM</b>	Research in Motion
<b>PDA</b>	Personal Digital Assistant
<b>BS</b>	Base Station
<b>MSC</b>	Mobile Switching Center
<b>Twiml</b>	Twilio Markup Language
<b>REST</b>	Representational State Transfer
<b>MD5</b>	Message-Digest algorithm 5
<b>SaaS</b>	Software as a service



# Capítulo 1

## Introdução

### 1.1 Contexto e motivação

Com o recente aumento do número de dispositivos de comunicações móveis em uso, a procura de novas aplicações de suporte a vários serviços aumentou também de forma substancial. Como resultado disto foram criados novos serviços de comunicações disponibilizados de forma gratuita, necessitando unicamente de acesso a redes Wi-Fi ou 3G. Pela análise deste crescimento repentino, pressupõe-se que mais serviços se podem desenvolver para estes dispositivos, oferecendo aos utilizadores novos meios de comunicação ou até novos serviços que podem mudar as formas de comunicação actuais. A constante evolução tecnológica nas telecomunicações não foi o único aspecto a ter em conta: a importância das redes sociais aumentou imenso e veio ser outro grande factor no método de desenvolvimento das telecomunicações actuais, evoluindo a componente social através da disponibilização de novos métodos de comunicação e partilha multimédia. Torna-se importante interligar estes novos serviços com os actuais meios comunicação de forma transparente, sob uma interface comum para o utilizador, que possa melhorar a experiência de utilização.

### 1.2 Objectivos

Oferecer ao utilizador uma aplicação móvel com vários serviços de telecomunicações, recorrendo à reutilização de software existente e ao desenvolvimento de métodos que interliguem o seu funcionamento. O protótipo a desenvolver tem como objectivo reunir alguns dos serviços e disponibilizá-los de forma cómoda e simples. É pretendido que o protótipo obtenha informações suplementares sobre os contactos do utilizador em tempo real, através das redes sociais. Formulando um questionário sobre os tipos de serviços de telecomunicações e redes sociais que são os mais populares, deverá ser possível obter informações sobre os utilizadores e as suas rotinas de comunicação com outras pessoas. De maneira geral, pretende-se realizar os seis passos apresentados de seguida e assim obter o máximo de informação relativo a esta experiência.

- Identificar os principais problemas no desenvolvimento de aplicações para dispositivos móveis.
- Identificar os principais serviços de comunicação actuais e novos serviços úteis num futuro próximo.

- Estudar qual a forma adequada para apresentação dos serviços e conteúdos num dispositivo móvel.
- Desenvolver um protótipo que tire partido apenas de recursos OpenSource para uma implementação de comunicações unificadas.
- Avaliar o protótipo num ambiente com utilizadores que usem os mesmos serviços de forma disjunta.
- Recolher ilações sobre a potencialidade da utilização destes serviços.

### 1.3 Estrutura da dissertação

Este documento encontra-se dividido em 7 capítulos.

- **Capítulo 1:** Introdução - Refere o contexto do trabalho que será desenvolvido neste documento, motivação que o levou a desenvolver e qual o tipo de estrutura definida.
- **Capítulo 2:** Estado da Arte - Resumo de várias tecnologias que estão interligadas nas comunicações unificadas, desde o estudo de clientes VoIP, arquitectura IMS, conceito de comunicações unificadas, aos sistemas actuais de web 2.0 e ainda as soluções de comunicações unificadas dos sistemas operativos móveis mais importantes.
- **Capítulo 3:** Requisitos - Aspectos funcionais e estrutura física necessária para o funcionamento de todo o sistema.
- **Capítulo 4:** Arquitectura - Evolução do modelo do sistema presente na aplicação e todos os detalhes e decisões efectuadas sobre cada componente presente no sistema.
- **Capítulo 5:** Implementação - Processo de desenvolvimento da aplicação desde os vários protótipos iniciais, *design* do sistema, funcionalidades dos serviços e aspectos de usabilidade.
- **Capítulo 6:** Resultados - Realização de algumas tarefas de modo a recolher resultados de usabilidade e a obtenção de ideias para posterior modificação do protótipo desenvolvido.
- **Capítulo 7:** Conclusão - Descrição das metas atingidas, obstáculos encontrados e o conhecimento adquirido através do desenvolvimento do protótipo.



## Capítulo 2

# Estado da arte

### 2.1 VoIP

Com a evolução dos meios de comunicação ao longo dos anos, a necessidade de diminuir os gastos envolvidos nos serviços de voz e multimédia tornou-se um aspecto importante.

A solução encontrada passa pela utilização do protocolo VoIP de forma a se conseguir atingir esta meta. No entanto, a utilização desta nova tecnologia veio possibilitar a criação de mais serviços para além da simples conversa telefónica entre duas pessoas.

Actualmente as várias aplicações presentes no mercado contêm serviços como chamadas telefónicas entre clientes *Skype* ou entre a rede pública de telefones, *instant messaging*, videoconferência e notificação de presença.

Actualmente, empresas como a *Vocaltec*, *Skype*, *Twilio* e *Voxeo*, baseadas em serviços de comunicações e multimédia, seguem abordagens diferentes quanto ao protocolo VoIP e, como tal, houve a necessidade de analisar as alternativas existentes. Começou-se por analisar primeiro o *Skype*, devido ao seu enorme impacto no crescimento do protocolo desde o início e apesar de o seu criador ter sido a *Vocaltec* [1].

#### 2.1.1 Skype

O *Skype* é considerado um cliente VoIP e segue uma abordagem diferente das outras implementações de cliente, já que utiliza apenas ligações ponto a ponto [2, 3]. No entanto, este não segue a típica abordagem descentralizada, segundo a arquitectura do *Skype* [4], possui servidores dedicados exclusivamente para guardar nomes, palavras-passe e verificar credenciais dos utilizadores.

Vários estudos foram efectuados ao longo dos últimos anos de maneira a conseguir entender todo o seu processo de funcionamento, devido ao facto de o *Skype* ser uma aplicação de renome [5].

O *Skype* pode-se estruturar em 3 entidades distintas [6, 7]:

- Os nós finais: são os próprios clientes e estão ligados através da aplicação fornecida pelo *Skype*.
- Os super nós: consiste num nó da rede que possui as características necessárias para o correcto funcionamento do serviço, tais como: um endereço IP público, alto processamento de CPU e grande largura de banda.

- O servidor de login: que fornece serviços de autenticação aos utilizadores no sistema.

Através dos vários estudos [6,7], verificou-se que os super nós têm grande importância em todo o sistema, desde o registo dos utilizadores no servidor de autenticação à manutenção da tabela de indexação dos vários utilizadores activos. Porém, este tipo de estrutura levou a que actualmente a própria *Skype* adquirisse a sua própria estrutura de super nós, passando assim a fornecer um melhor serviço aos seus utilizadores e precavendo novos transtornos nos seus serviços, como aconteceu no passado [8].

O *Skype* permite ainda ao cliente o uso dos servidores *SkypeOut* e *SkypeIn*. Como tal, a partir destes serviços é possível receber ou realizar chamadas conjuntamente com a rede pública de telefones. Este tipo de serviço é, no entanto, efectuado de forma centralizada. Contudo esta questão não é tida em conta, não contradizendo a afirmação de que o *Skype* utiliza uma estrutura ponto a ponto [4]. Outro aspecto importante é a forte evolução no desenvolvimento de *CODECS* até esta altura. A utilização do CODEC de som GIPS iSAC, possibilita que as ligações de som sejam geridas pelo próprio CODEC, consoante a largura de banda da rede entre os dois nós [9]. Mais tarde foi adoptado o codec *SVOPC*, sendo recentemente substituído pelo codec *SILK*.

Após alguns estudos efectuados [9–11], confirmou-se que com a adição de parâmetros como *Sampling Rate* (8-24 kHz), *Packet Rate* (20ms, entre 1 a 5 frames) e *Bit Rate* (entre 6 e 40 kbps) no codificador, permitindo superar os antecessores, como é possível verificar na figura seguinte.

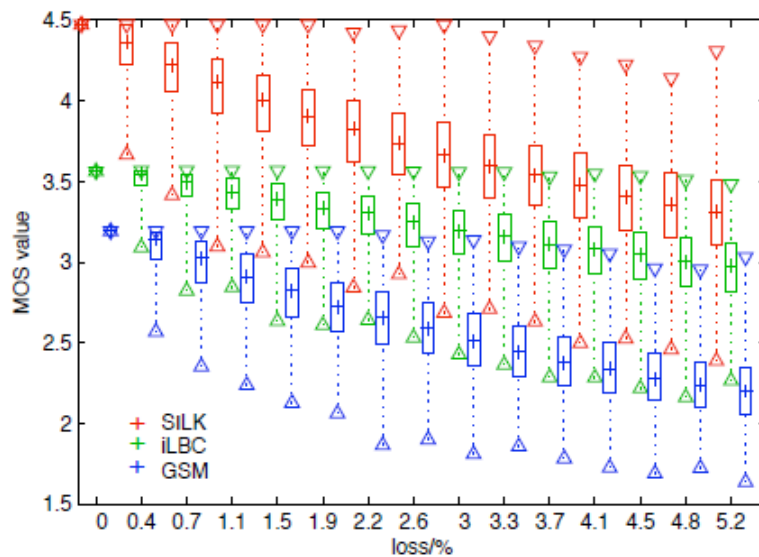


Figura 2.1: SILK Random Loss [10]

O *Skype* tornou-se muito relevante no mundo das telecomunicações pelo facto do conceito VoIP criado ser actualmente impossível de bloquear através de *firewalls* ou *NAT* traversal [7]. Vários estudos demonstram que o *Skype* é, efectivamente, umas das poucas excepções. Alguns destes estudos levam a acreditar que cada nó presente na rede possui mecanismos similares aos utilizados pelo protocolo STUN [7].

## 2.1.2 Twilio

A *Twilio* é uma empresa recente que obteve um grande crescimento nos últimos dois anos. Este aumento da empresa no mundo das telecomunicações deveu-se ao facto da alteração da visão de como efectuar o seu serviço VoIP(figura 2.2). Com a abordagem centrada na utilização de serviços através da *Web*, adoptando o modelo SaaS (Software as a Service), oferece uma maior abstracção da estrutura física de todo o sistema de modo a disponibilizar os seus serviços de forma mais simples.



Figura 2.2: Twilio : Exemplo de funcionamento

*Twilio Client* é uma interface de programação desenvolvida pela empresa para disponibilizar aos seus clientes todo o seu leque de produtos. Actualmente é possível criar novos serviços através deste cliente, tanto para dispositivos móveis Android e iOS ou por *Web*. Na vertente *Web* oferecem diversas bibliotecas, como por exemplo: *PHP*, *Java*, *Ruby* e *Perl* [12]. Outro aspecto importante nesta interface de programação é a linguagem desenvolvida pela própria empresa, denominada por TwiML. TwiML é uma linguagem de *markup*(Figura 2.3) com o intuito de efectuar a gestão dos serviços, permitindo ao utilizador realizar uma configuração prévia na qual o sistema responde consoante essas indicações. Na figura seguinte é registada uma mensagem no servidor, a qual é efectuada em resposta a uma chamada telefónica. Deste modo, é elaborada uma mensagem de voz pelo servidor e, após essa mensagem, a chamada é gravada durante 20 segundos.

```

1 | <?xml version="1.0" encoding="UTF-8"?>
2 | <Response>
3 |   <Say voice="woman">Please leave a message after the tone.</Say>
4 |   <Record maxLength="20" />
5 | </Response>

```

Figura 2.3: TwiML: Exemplo de utilização [12]

## 2.2 Protocolo SIP

O SIP é um protocolo cliente-servidor de sinalização de comunicação e é normalmente utilizado para efectuar a gestão de sessões multimédia. Como o HTTP, é baseado em mensagens de texto sendo um padrão especificado pela *IETF* e descrito no RFC 3261.

A estrutura da mensagem genérica descrita pelo protocolo está dividida em três partes:



Figura 2.4: Estrutura da mensagem SIP

O primeiro componente *Start-Line* serve para distinguir entre dois tipos de mensagens: solicitações ou respostas. A distinção entre solicitações e respostas é feita através do formato da *Start-Line* - solicitações usam o formato *Request-Line*, ao passo que as respostas usam o formato *Status-Line*. A mensagem contém ainda um ou mais cabeçalhos(*Header*), seguindo-se uma mudança de linha obrigatória, representado na figura 2.4 pelo conjunto de caracteres *CRLF* e uma mensagem de texto.



Figura 2.5: Request-Line

Na mensagem de solicitação, o campo *Method* é pode assumir um dos seis valores: *Register*, *Invite*, *Ack*, *Cancel*, *Bye* e *Options*. O primeiro método é necessário para efectuar solicitações de "Registo do utilizador", os métodos *Invite*, *Ack* e *Cancel* são métodos de configuração da sessão e o método *Bye* serve para terminar a sessão. Finalmente, o método *Options* é usado para inquirir o servidor sobre as suas características.

O campo *Request-URI* é um identificador do tipo *SIP URI* ou *SIPS URI*, que é utilizado para endereçar o utilizador ou serviço da solicitação e pode ser representado da seguinte forma:

sip:user:password@host:port;uri-parameters?headers

O campo *SIP-Version* tem o propósito de indicar qual a versão do protocolo SIP a utilizar. Cada caracter *SP* é obrigatório entre cada campo, assim como os caracteres *CRLF* no final. Como se pode verificar, este esquema é igualmente necessário na mensagem de resposta:

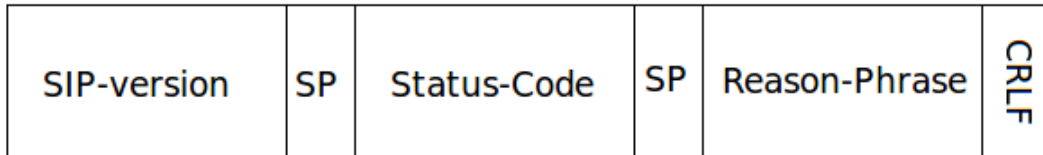


Figura 2.6: Status-Line

A resposta é constituída por três campos: *SIP-Version*, *Status-Code* - constituído por três dígitos - e o *Reason-Phrase*. O *Reason-Phrase* armazena uma mensagem com o intuito de informar o tipo de resposta que o utilizador recebe.

Os campos *Status-Code* e *Reason-Phrase* seguem a numeração listada:

- 1xx: Provisional - mensagem indicativa do processamento do pedido.
- 2xx: Success - mensagem indicativa de sucesso;
- 3xx: Redirection - mensagem indicativa da necessidade de mais ordens para efectuar o pedido.
- 4xx: Client Error - mensagem gerada quando um servidor não consegue efectuar o pedido com sucesso ou a mensagem contém algum erro.
- 5xx: Server Error - mensagem gerada pelo servidor quando ocorre um erro que, aparentemente, deveria ser aceite com sucesso.
- 6xx: Global Failure - mensagem de erro quando o pedido não é processado em nenhum servidor.

No cabeçalho, presente na mensagem genérica, também é necessário obedecer a algumas regras que são apresentadas na figura 2.7:

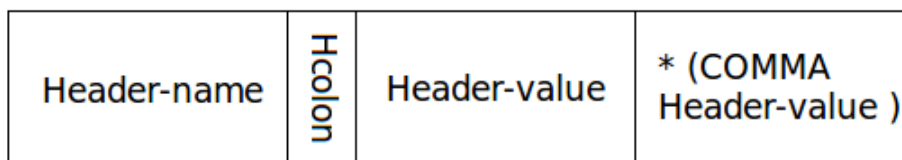


Figura 2.7: Header SIP

É necessário o nome do campo do cabeçalho, seguido pelo caracter dois pontos ':' e um ou mais valores do campo, podendo valores adicionais serem separados por vírgulas. Quando se pretende colocar mais que um campo no cabeçalho, é necessário separá-los por linhas. Delimita-se o cabeçalho, e o início do *Message-Body*, através dos caracteres *CRLF*.

Segue a troca de fluxos de mensagens efectuado entre o cliente e o servidor. O processo de autenticação por parte do utilizador é efectuado sempre da forma como é apresentado na figura 2.8:

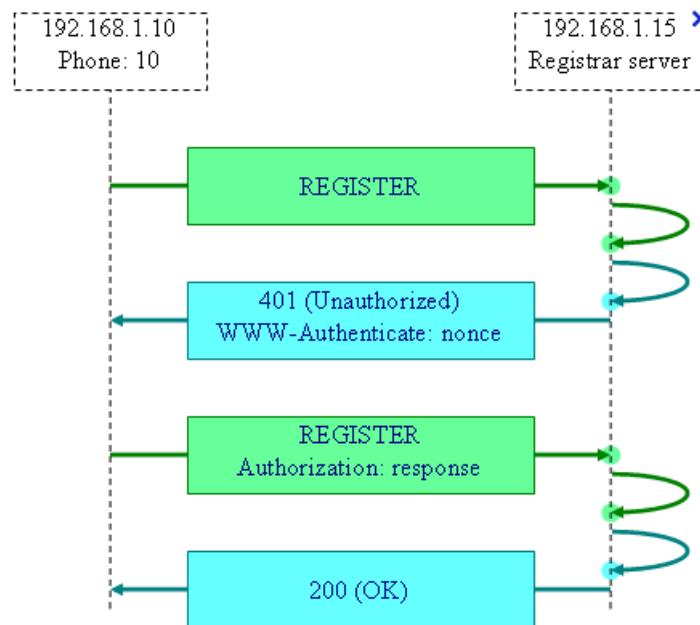


Figura 2.8: Processo de Autenticação

Após a realização da autenticação entre o cliente e o servidor SIP e, a partir deste momento, serviço fica pronto para realizar ou receber chamadas de voz. O cliente que pretende efectuar a chamada efectua o envio de uma mensagem *INVITE* para o servidor que a re-envia para o outro participante, este responde com uma mensagem de *Trying* e *Ringing*, de modo a aceitar o pedido. De seguida o cliente efectua o envio de uma mensagem de *ACK* e a partir desse momento ambos os participantes iniciam um sessão multimédia, sendo esta terminada por qualquer um dos participantes através do envio de uma mensagem *Bye*.

## 2.3 3GPP IMS

O IMS é um *standard* reconhecido internacionalmente, especificado inicialmente pela 3GPP/3GPP2, que define de forma genérica a arquitectura que permite oferecer voz sobre IP (VoIP) e serviços multimédia [13]. Os serviços baseados no IMS permitem aos utilizadores uma comunicação *person-to-person* ou *person-to-content* de variados modos, tais como voz, imagem, texto ou vídeo, bem como uma qualquer combinação destes.

Para os operadores o IMS baseia-se numa arquitectura por camadas horizontais (Figura 2.9),

onde serviços e funções mais comuns podem ser reutilizados por múltiplas aplicações. Esta arquitectura horizontal também permite a interoperabilidade, o roaming, e é portador de camadas de controlo, segurança e cobrança [14,15].

Esta arquitectura horizontal permite às operadoras afastarem-se da tradicional forma de criação de serviços, com sobreposições funcionais ao nível de presença, grupo, encaminhamento, cobrança e provisionamento [16].

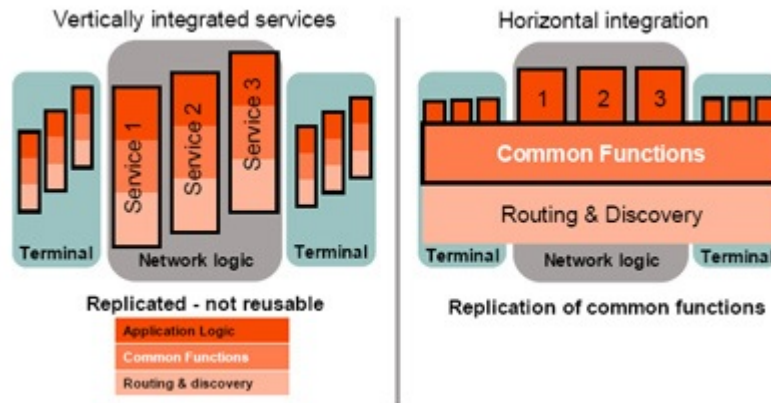


Figura 2.9: Arquitectura IMS - Visão Geral

A camada de aplicação engloba aplicações e servidores de conteúdo para oferecer ao utilizador serviços de valor acrescentado. Serviços genéricos, tais como os definidos no *standard IMS* (tais como presença, grupo etc) [14,16], são implementados como serviço num servidor de aplicações SIP. Arquitectura contém a camada de controlo que engloba os servidores de controlo para gestão das chamadas e do estabelecimento de sessões. O mais importante nesta camada é o CSCF, também conhecido como servidor SIP. Esta camada contém ainda uma série de funções de suporte como provisionamento, cobrança e gestão. A intercomunicação entre os operadores e outros tipo de redes é feito através de *gateways*. A camada de rede inclui os *routers* e *switches*, que formam o *backbone* da rede de acesso.

O HSS é o servidor onde está alojada a base de dados que contém os dados dos utilizadores e a respectiva informação de forma a ser possível gerir as suas chamadas e sessões. Quando um utilizador se regista num domínio IMS, o seu perfil é descarregado do HSS e o S-CSCF é o responsável por efectuar a gestão das sessões dos utilizadores. Como tal, é neste que se efectua o registo do protocolo SIP, que pode ser conhecido como *SIP server*. É ele que reencaminha para os pedidos SIP para os AS. Podem existir vários numa rede, de forma a fazer o balanceamento da rede de forma automática.

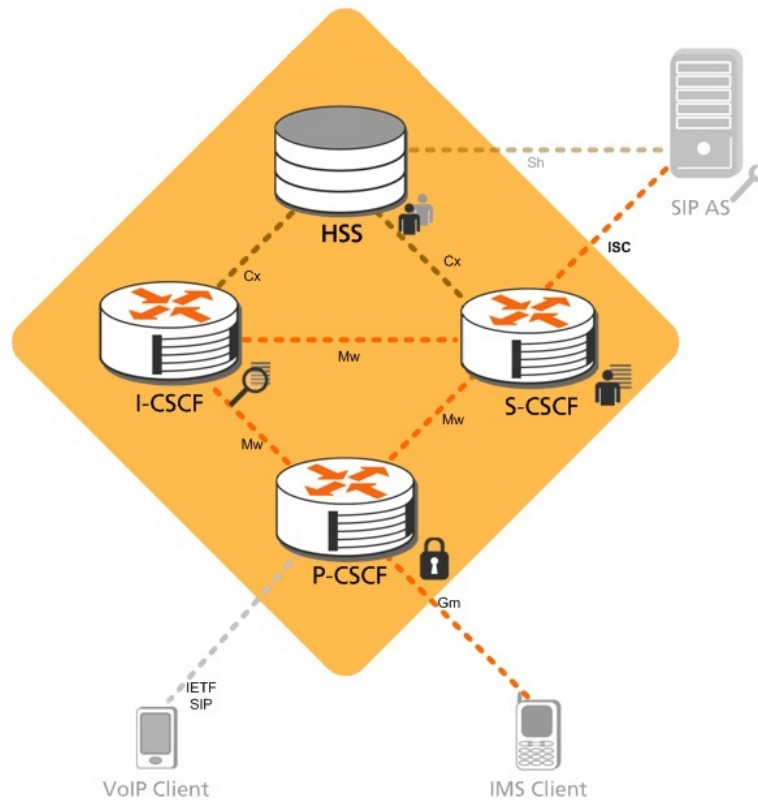


Figura 2.10: Arquitectura IMS

O P-CSCF actua como um *proxy* que encaminha os pedidos dos clientes para o I-CSCF. Podem existir vários na mesma rede e, até estarem na rede visitada, são o ponto de entrada para os clientes. O I-CSCF é o ponto de contacto externo para outros operadores, pois é ele que interroga o HSS para saber o endereço do S-CSCF, a fim de poder fazer o encaminhamento dos pedidos recebidos.

## 2.4 Comunicações unificadas

O conceito de comunicações unificadas teve origem na década de 80, na sequência da junção dos serviços de mensagens de voz, resposta interactiva de voz (IVR) e correio electrónico. Desde então que este conceito cresceu e, no início dos anos 90, a globalização dos telemóveis elevou a importância deste conceito no futuro das telecomunicações e no aparecimento de novos serviços.

A partir desta altura, novos serviços como o serviço de mensagens texto, fax, mensagens instantâneas, chamadas telefónicas através de IP, videoconferência e presença, foram alvos de novas implementações por parte da comunidade. Como referido anteriormente na secção 2.1, as chamadas telefónicas através de IP são um serviço bastante importante actualmente. No entanto, o serviço de presença permite auxiliar outros serviços, tornando-se influente no modo de funcionamento de todos. Com a integração do serviço de presença nos conceitos de VoIP e IM, é dado ao utilizador uma resposta sobre a disponibilidade dos seus contactos em tempo real.

Actualmente, o aumento do número de serviços tem sido cada vez maior, principalmente



serviços *Web*, devido ao aparecimento de cada vez mais dispositivos móveis que permitem o acesso a redes Wi-Fi ou 3G. Conforme foi divulgado através de UCStrategies [17], as comunicações unificadas encontram-se divididas em duas vertentes: utilizador (UC-U) e empresarial (UC-B). UC-U é a vertente referente ao melhoramento significativo da comunicação do utilizador no seu quotidiano, com o principal objectivo de simplificar todo o funcionamento dos diversos serviços disponíveis. Quanto ao UC-B é direccionado ao melhoramento dos processos de gestão e comunicação empresariais, aumentando a produtividade dos seus funcionários e dos seus serviços internos.

Para uma melhor distinção e enquadramento sobre como qualificar os mais importantes serviços de comunicações unificadas, foram realizados alguns estudos de análise. Foram definidas 5 áreas na aplicação [17] e depois retiradas as informações presentes na seguinte tabela: 2.1.

<b>Funcionalidade</b>	<b>Impacto Empresarial</b>	<b>Benefícios</b>
Gestão de Contactos. Aumento drástico da primeira chamada.	Aumenta a satisfação do cliente.	Redução do ciclo de vendas.
Acesso a informação em qualquer lugar.  Identificação de recursos e resolução do problema.	Entrega de informação apenas quando necessária. Sem utilização de tempo e recursos.  Acesso imediato a recursos especializado. Acompanhamento total no projecto.	Redução de custos de suporte. Evita chamadas desnecessárias. Aceleração de decisões.  Elimina atrasos. Automatiza tarefas administrativas. Aumento de receitas. Velocidades das transacções.
Aceleração colaborativa.  Especificação da informação e aspectos comuns.	Acelerar a conclusão do projecto mais facilmente. Melhoramento no acesso a informação.  Serviços de comunicação integrada consoante a sua necessidade.	Elimina atrasos. Melhora a qualidade. Redução no custo das reuniões.  Aumento da aceleração e precisão dos melhores métodos de comunicação.

Tabela 2.1: Ecrãs de configuração

Examinando todos estes aspectos é notório que o mundo empresarial beneficiará com o uso das metodologias apresentadas pelas comunicações unificadas. No entanto, como não é um modelo genérico para todas as empresas, é necessário efectuar alterações dependendo do tipo de empresa.

#### 2.4.1 RIM

A RIM é uma empresa de fabrico de equipamentos de telecomunicações e neste momento é mais conhecida pela sua gama de produtos BlackBerry. A empresa obteve algum desse sucesso devido à inovação do serviço de *push-to-talk* presente na sua primeira geração de dispositivos. A partir desse momento a RIM alargou a sua área de negócios e actualmente possui o sistema

Blackberry MVS interligado à sua gama de produtos, disponibilizando assim novos serviços para todos os seus dispositivos móveis.

### **Blackberry MVS**

O sistema Blackberry MVS é um sistema de comunicações unificado que permite ao cliente interligar vários dispositivos como os telefones de trabalho e pessoais num só dispositivo, tornando possível a realização de chamadas através de qualquer um deles. É ainda possível transferir chamadas para outro número, efectuar teleconferências e realizar chamadas para as redes fixas tradicionais através de acesso Wi-Fi ou às redes de dados móveis.

### **2.4.2 Nortel e Avaya**

A *Nortel*, actualmente parte da *Avaya*, foi uma das empresas mais importantes na criação dos sistemas de telecomunicações contemporâneos. Inicialmente era apenas uma empresa de fabrico de telefones mas alguns anos mais tarde, e após algumas transformações no final da década de 70, optaram por uma solução direccionada às comunicações unificadas. O principal objectivo era disponibilizar às empresas estes serviços de forma a melhorar as comunicações entre os seus funcionários, oferecendo também um custo reduzido. Com o recente aparecimento dos *smartphones*, a Nortel desenvolveu os seus serviços de forma a interligar estes dispositivos com os seus telefones móveis tradicionais [18]. Actualmente é a aplicação *Avaya OneX Mobile* que disponibiliza ao utilizador este tipo de funcionalidades:

- Gestão de contactos,
- Chamadas de voz,
- *Voice Mail*
- Redireccionamento de chamadas para dispositivos móveis,
- Realização de chamadas a partir de dispositivo móvel com identificação de outro telefone,
- Gestão de chamadas com prioridade,
- Teleconferência.

### **2.4.3 Web 2.0**

O conceito inicial da internet (*Web 1.0*) disponibilizava apenas conteúdos e serviços estáticos, não permitindo uma interacção mais dinâmica e integrada com as necessidades do utilizador. A globalização e a crise *dot com* [19] originou uma revolução de conceitos que deu origem à *Web* moderna. Através de novas tecnologias, a *Web* deixou de seguir o modelo de conteúdo estático usado até então.

O termo Web 2.0 foi mencionado inicialmente em 1999 por Darcy DiNucci [20] mas só em 2004, numa apresentação efectuada por Tim O'Reilly [21, 22], é que o conceito começou a ser considerado com relevância.

Como já foi referido anteriormente, o nome *Web 2.0* é a "denominação" na qual a Internet se

encontra actualmente, onde o conteúdo é gerado de forma dinâmica e onde os seus utilizadores têm um papel preponderante. No entanto, Tim O'Reilly enumera algumas características importantes a ter em consideração como [19]:

- Criação de serviços e não de software que necessite de novas actualizações.
- Depositar confiança nos utilizadores como criadores de conteúdos.
- Criação de conteúdos e melhoramento ao longo do tempo.
- Software disponível em vários dispositivos.
- Interfaces leves e atractivas.
- Análise do modelo de negócio

Com base neste pressuposto é possível analisar algumas das empresas com sucesso na actualidade tais como o Youtube, eBay, Amazon, Facebook, LinkedIn, Blogger e Flickr, interpretando as características destas relativas aos princípios referidos anteriormente.

## **Youtube**

O *Youtube* é fundamentalmente uma página de partilha de vídeos. Actualmente disponibiliza funcionalidades para além da possibilidade de carregar vídeos como :criação de listas de reprodução, opiniões, avaliações, categorizações e partilha de vídeos com amigos. Com base nestas mesmas características, a própria empresa recolhe informação relativa ao utilizador, como hábitos e preferências, de forma a tornar a experiência do utilizador mais atractiva. O modelo de negócio actual é a utilização do serviço *Google Ads* para popularizar os vídeos dos utilizadores ou, em alguns países, a utilização de anúncios de marcas internacionais.

## **eBay / Amazon**

Estas páginas de venda de produtos tiram partido do sistema de comentários para avaliar os vendedores, proporcionando aos utilizadores uma maior segurança na compra de produtos. Com base na análise feita ao histórico de compras é possível sugerir novos produtos ao utilizador e assim melhorar a experiência e cativá-lo para um uso mais frequente. O método explorado por este tipo de negócios é efectuado exclusivamente através de comissões sobre as vendas.

### **2.4.4 Social Networks**

Com o crescimento do conceito *Web 2.0*, a exigência de novos métodos de comunicação e a convergência de ambos levou ao aparecimento das *redes sociais*. Actualmente são muito os utilizadores que tiram vantagem deste serviço pelo facto de simplificarem a sua forma de comunicar e permitirem uma alargada partilha de conteúdos pessoais. No entanto, existem diversas abordagens quanto ao tipo de redes sociais, devido à necessidade de separação de conceitos como lazer, pessoais ou profissionais.

## Facebook

Facebook é uma rede social direccionada para o uso pessoal de cada utilizador. Das várias funcionalidades disponibilizadas pela plataforma é necessário salientar os diversos métodos de comunicação oferecidos através do Facebook, estando divididos em três tipos de mensagens. O método mais comum é a utilização do *chat* de IM para efectuar uma conversa entre dois ou mais utilizadores de forma privada. Para além desta simples troca de mensagens o Facebook disponibiliza muitas outras funcionalidades úteis ao utilizador, tais como:

- Criação de eventos.
- Chamadas de voz.
- Criação ou participação em grupos.
- Desenvolvimento dos dados pessoais.
- Secção de filmes e música.
- Upload de fotografias ou vídeos.
- Área de agregação de notícias sobre páginas ou grupos de interesse do utilizador ou contactos a ele adicionado.

## LinkedIn

A rede social LinkedIn, tem como objectivo oferecer aos seus utilizadores uma forma de contacto com o mundo empresarial. Os utilizadores têm assim uma forma de expor as suas informações profissionais, receber recomendações profissionais de colegas de trabalho ao longo da sua profissão e ainda possibilita às empresas contactar os utilizadores de forma mais simples.

## 2.5 Dispositivos móveis

Após a adopção em massa dos telemóveis, o aparecimento do conceito *personal data assistant (PDA)* e o enorme crescimento da Internet no início dos anos 90, surge então a ideia da unir ambas as tecnologias num dispositivo que junta estas funcionalidades: o *smartphone*. Actualmente existem diversas plataformas para os dispositivos móveis, tais como: *iOS*, *Android*, *Windows Phone*, *Bada* e *Symbian*. As três plataformas mais utilizadas, *iOS*, *Android* e *Windows Phone*, são utilizadas em cerca de 80% do mercado mundial (figura 3.3), tornando assim essencial verificar qual a abordagem às comunicações móveis cada sistema operativo oferece ao utilizador.

### 2.5.1 iMessage

O *iMessage* é o serviço criado pela *Apple* que integra os serviços de IM e SMS. Para utilizar o serviço é necessário criar uma conta no sistema informático da *Apple* para identificação do utilizador. Após associação através da configuração do número telefónico ou correio electrónico na conta *Apple*, o serviço fica activo no dispositivo. O utilizador pode então efectuar o envio grátis de mensagem de texto/multimédia para dispositivos da *Apple*, como o *iPhone*, *iPad* e *Mac*, quando realizado através da internet.

### 2.5.2 Windows Phone Messaging

O *Messaging* é aplicação utilizada nos dispositivos Windows Phone para comunicar com os seus contacto através de texto. A aplicação disponibiliza a associação de serviços de texto através do *Facebook*, *Twitter*, *LinkedIn*, *Windows Live* e *e-mail*, para além do envio de SMS. O envio deste tipo de mensagens não é realizado de forma automática: o utilizador precisa de inserir manualmente o tipo de serviço do contacto para qual pretende enviar a mensagem. A interface de mensagens de texto é categorizada por dois tipos: um ecrã único para o envio de SMS/MMS e uma interface de *chat* para o envio de IMs através do Facebook ou Windows Live, onde se pode também verificar a disponibilidade dos utilizadores em tempo real.

### 2.5.3 Google+ Hangouts

*Google+ Hangouts* é a solução da Google para efectuar serviços de comunicações áudio, vídeo e texto a partir das aplicações *Web* da empresa, como o *GMail* e *Google Plus*. Com o *Hangouts* é possível realizar videoconferências com até dez utilizadores, de forma gratuita, sendo possível publicar as sessões no *Youtube* para posterior partilha nas redes sociais. A Google promove também o serviço como uma forma de qualquer pessoa poder efectuar transmissões em directo, à semelhança de outros serviços de transmissão de áudio e vídeo em directo como o *Ustream* e *justin.tv*. Além do suporte via aplicações *Web*, é suportado nativamente nos dispositivos móveis baseados em sistemas *Android* e *iOS*.

Contrariamente aos serviços anteriores da Google, o *Google+ Hangouts* não utiliza o protocolo aberto XMPP mas um protocolo proprietário da Google. A interface de programação, lançada recentemente, é apenas orientada ao desenvolvimento de aplicações *Web* que complementem os serviços já suportados. Devido a estas limitações, não foi considerada uma implementação do serviço no protótipo de aplicação desenvolvido.

### 2.5.4 Multiplataforma

É também necessário analisar algumas das aplicações que já integram alguns serviços de texto e voz entre dispositivos móveis que possuam os sistemas operativos *Android*, *iOS*, *Windows Phone* e *BlackBerry*. Através de um inquérito sobre os hábitos de comunicação dos utilizadores, foram mencionadas as aplicações *Viber* e *WhatsApp*. Decidiu-se analisar ambas as aplicações para perceber quais as suas mais valias para os utilizadores.

#### Viber

É uma aplicação que integra os serviços de voz e mensagem de texto de forma gratuita. Ao iniciar aplicação, é obrigatório que o utilizador realize o seu registo no sistema. É requerido o número de telefone e, após recepção de uma mensagem de texto, realiza-se a verificação do registo através da inserção de um código de acesso. A aplicação é capaz de utilizar redes Wi-Fi ou 3G para suportar os serviços entre utilizadores da aplicação, sendo então gratuitos, ou através de operadores telefónicos caso o contacto não seja utilizador do *Viber*. Para a realização do serviço VoIP é necessário uma ligação com largura de banda mínima de 32kbit/s [23]. Para além dos serviços mencionados, aplicação permite promover contactos a favoritos e guarda no dispositivo toda a informação relativa as chamadas e mensagens de texto para posterior consulta.

## **WhatsApp**

É uma simples aplicação destinada ao envio de mensagens de texto através de uma ligação Wi-Fi ou 3G, disponibilizando ainda a inclusão de dados como fotografias, vídeos e localização. Aplicação oferece a opção de efectuar conversas em grupo e envio de mensagens "Broadcast", que servem para difundir a mesma mensagem para várias pessoas sem que tenham acesso aos vários destinatários.

## Capítulo 3

# Requisitos do sistema

### 3.1 Desenvolvimento

Como referido anteriormente, existem várias plataformas no mercado sobre as quais se pode desenvolver uma aplicação móvel. Estas seguem o paradigma de programação baseada em eventos, no qual o utilizador é o principal interveniente no funcionamento do sistema. Este tipo de paradigma é visto como um sistema de grafo [24], em que as transições entre cada grafo é representada através de uma interação do utilizador no sistema. Um exemplo deste sistema encontra-se representado na figura seguinte:

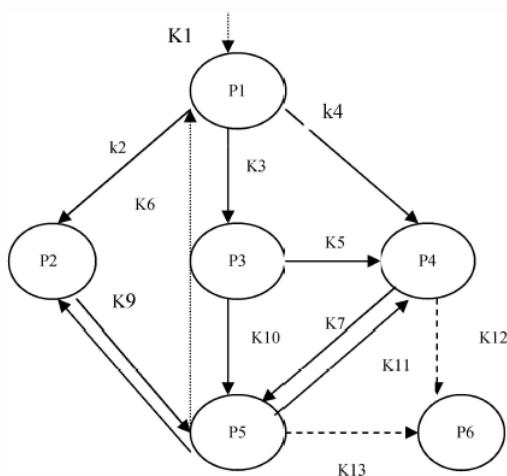


Figura 3.1: Exemplo : Programação baseada em eventos [24]

O padrão de arquitectura de software utilizado para o desenvolvimento de aplicações nos dispositivos móveis modernos é o *MVC*. Este padrão, largamente usado também em *frameworks* Web 2.0, encontra-se dividido em três componentes:

- Modelo: Classe de persistência de dados da aplicação. Pode ter uma ou mais associações para outro modelo. Devido às regras do modelo MVC, deve ser apenas actualizado através do seu controlador e informar o mesmo quando os dados forem guardados.
- Vista: Janela de visualização de dados e interacção com o utilizador. Este componente

encontra-se totalmente desagregado do modelo, utilizando o controlador para fazer a gestão das interações com o utilizador.

- Controlador: Classe que interliga um ou mais modelos e vistas, sendo responsável por monitorizar as alterações nas vistas e actualizar os modelos. O controlador também tem o papel de definir o ciclo de vida dos outros componentes.

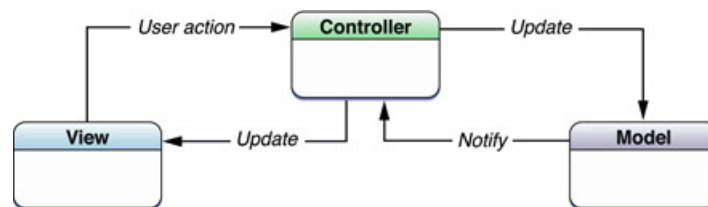


Figura 3.2: Arquitectura MVC

Neste momento, ao comparar o mercado dos *smartphones*, verifica-se que o sistema operativo Android possui alguma vantagem relativamente ao iOS devido ao alargado número de fabricantes e ao preço mais baixo. Como podemos ver no gráfico de vendas, o mercado de *smartphones* Android é muito superior aos restantes concorrentes:

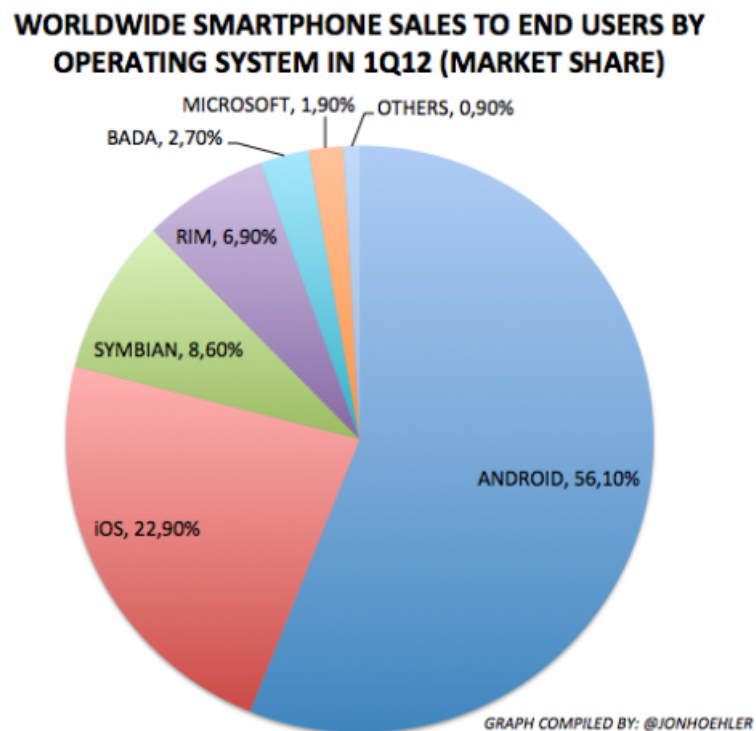


Figura 3.3: Percentagem mundial de Smartphones [25, 26]



### 3.1.1 iOS

Sistema operativa utilizado pela Apple nos seus dispositivos móveis, a plataforma de desenvolvimento adoptada é o *Cocoa* e *Cocoa Touch*, usando como linguagem o *Objective-C* [27]. Estas linguagens estão integradas no ambiente de desenvolvimento desenvolvido pela Apple, denominado por *Xcode*. As *frameworks* Cocoa e Cocoa Touch, que consistem num conjunto de bibliotecas e interfaces de programação, oferecem ao programador um método para o desenvolvimento de aplicações. Também são as mesmas utilizadas para a criação do sistema operativo e usam os mesmos padrões [27]. A *framework* contém funcionalidade para:

- gestão de conteúdos de audio e vídeo
- métodos de acesso a bibliotecas gráficas como OpenGL e Quartz
- Acesso à biblioteca SQLite
- Ruby
- Python
- AppleScript

### 3.1.2 Windows Phone 7

O Windows Phone é o novo sistema operativo da Microsoft que veio substituir o *Windows Mobile*. O .NET continua presente como plataforma de desenvolvimento mas, no entanto, as aplicações criadas no Windows Mobile não são compatíveis com o novo sistema operativo. Como tal, o desenvolvimento da interface de novas aplicações pode ser feito da mesma forma que o seu antecessor, utilizando a camada gráfica *Windows Forms* [28,29]. Do ponto de vista do programador, o novo sistema Windows Phone trouxe duas novas plataformas de desenvolvimento: a plataforma XNA, para criação de jogos, e a plataforma Silverlight, para aplicações móveis e Web. A tabela 3.1 expõe as diferentes aplicações de cada uma das plataformas do sistema:

Silverlight	XNA Framework	XNA Framework e Silverlight
Quando se pretende uma aplicação baseada no paradigma de programação por eventos e usando uma interface através da XAML.	Desenvolver jogos com alto desempenho.	Unir o paradigma de programação de eventos de uma aplicação com o desenvolvimento de jogos
Criar um interface Web agradável ao utilizador de forma rápida.	Criar jogos 2D ou 3D de modo rápido.	Criar jogos 2D ou 3D incorporando uma aspectos presentes numa interface Web.
Utilizar controladores presentes no Windows Phone.	Efectuar a gestão de aspectos gráficos como, modelos, malhas, sprites, texturas, efeitos, terrenos, ou animações da pipeline presente na XNA.	Incorporar os controladores na gestão dos vários aspectos gráficos.
Incorporar vídeos dentro de uma aplicação.		Utilizar a renderização de texto através do Silverlight, em vez de texto com sprites.
Utilizar os controladores HTML.		Criar uma página com o Silverlight incorporando efeitos gráficos através da plataforma XNA.

Tabela 3.1: Windows Phone 7

### 3.1.3 Android

O Android é um sistema operativo, criado especificamente para dispositivos móveis, que usa o *Linux kernel*. Consiste numa arquitectura organizada em 5 camadas: *kernel*, bibliotecas, *runtime*, *frameworks* e aplicações.

A camada do *kernel* efectua a gestão de memória, processos e controladores necessários para instalação nos dispositivos. A camada de *bibliotecas*, possui suporte para C, C++ e Java. O Java é a principal plataforma usada no sistema operativo Android, fazendo parte da camada de *runtime*, através do uso da máquina virtual *Dalvik* para executar a gestão de recursos e aplicações, possibilitando uma melhor abstracção da micro-arquitectura do dispositivo. A camada de *frameworks* fornece a reutilização de componentes para aplicações, disponibilizando à camada superior componentes como fragmentos, vista, botões, caixas de texto, imagens e listas suspensas, entre outros.

O primeiro conceito que se apreende aquando da programação em Android é o conceito de actividade. Cada aplicação é criada inicialmente a partir de uma única actividade principal, sendo esta reponsável pela execução das novas actividades que ficam em execução no ecrã do dispositivo. No entanto, as actividades criadas a partir da actividade principal ficam no sistema pela ordem na qual foram criadas e como tal são terminadas pela mesma ordem. Cada actividade possui um estado interno, como é possível verificar na diagrama seguinte, esta característica possibilita uma maior flexibilidade e consequente simplificação das várias

funcionalidades pretendidas.

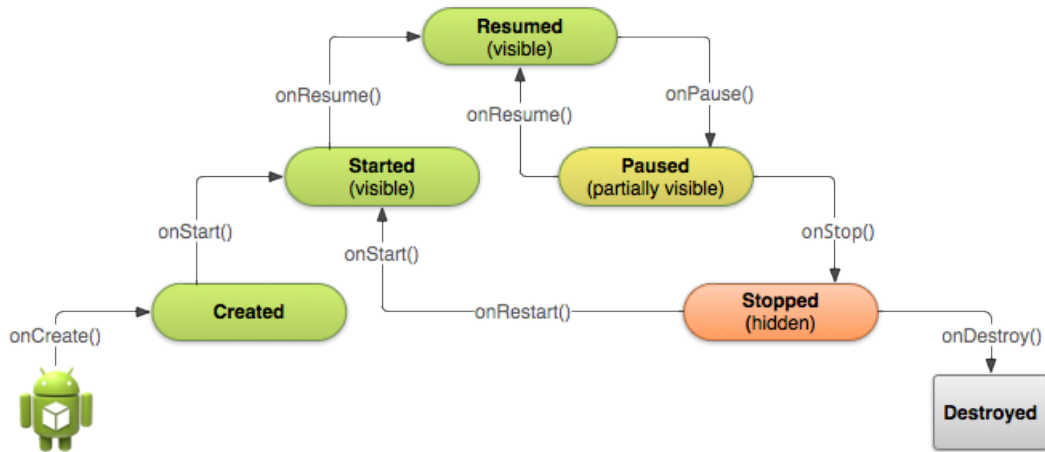


Figura 3.4: Ciclo de vida da actividade

Um exemplo concreto do conceito actividade é interface inicial do dispositivo Android que, aquando da execução de uma nova aplicação, é colocada em segundo plano para que uma nova aplicação(actividade) possa ser visualizada.

A interface gráfica do sistema operativo Android é efectuada de duas formas: utilizando ficheiros XML ou executado de forma dinâmica através de código Java. No entanto ambos os métodos podem ser utilizados para oferecer uma maior flexibilidade à aplicação.

Em diversos casos, é necessária interligação entre actividades, sendo estes dados transmitidos através de estruturas de dados *Intent*. Estas actividades podem ainda transmitir dados para *Broadcast Receivers*, que recebem as estruturas de dados da classe *Intent*.

A disponibilização ou recepção de dados para o exterior de um aplicação é possível quando a própria contém um *Content Provider*. Deste modo, quando se pretende que uma aplicação Android autorize o acesso externo de dados, estes podem ser guardados de forma segura em ficheiros ou num base de dados SQLite. É então necessário recorrer ao uso do *Content Provider*, já que o sistema operativo Android não disponibiliza nenhuma outra funcionalidade para o efeito.

Um conceito importante ao efectuar o desenvolvimento de uma aplicação é a configuração de um ficheiro XML, que está presente obrigatoriamente em todas as aplicações, denominado por *AndroidManifest*. Neste ficheiro estão presentes todos os caminhos físicos das diversas actividades presentes na aplicação, como também todas as permissões de sistema a que aplicação terá acesso.

## 3.2 Estruturais

Dado que um grande número de smartphones suportam o sistema operativo Android, aliado à não obrigatoriedade de adquirir licenças e hardware para o desenvolvimento do protótipo, optou-se pelo sistema operativo da Google para o desenvolvimento do protótipo da aplicação móvel.

Sendo o principal objectivo desta dissertação estudar a unificação de diversos serviços de

comunicação, foram seleccionados como principais serviços as chamadas de voz e mensagens de texto - os serviços de e-mail e redes sociais foram relegados para segundo plano. Para o desenvolvimento dessas funcionalidades, são necessários servidores para interligar os utilizadores a cada serviço. O acesso à rede telefónica é fornecido pela operadora telefónica presente no dispositivo móvel e assim o mecanismo de realização de chamadas telefónicas através do *smartphone* é relativamente fácil. Para os serviços de chamadas VoIP é necessário um servidor SIP que forneça os mecanismos necessários ao registo de utilizadores e o encaminhamento dos pedidos para a realização da chamada. O serviço de mensagens de texto é efectuado da forma tradicional, utilizando as infraestruturas da operadora e as *frameworks* do Android. Quanto à implementação do sistema de mensagens instantâneas não seria possível enviá-las através do servidor SIP, devido às limitações apresentadas pela interface de programação adoptada pelo Android(SIP Manager). De forma a ultrapassar este obstáculo, optou-se por escolher uma nova interface de programação que efectua o envio de mensagens de texto através do protocolo XMPP, sendo necessário um servidor adicional para o efeito.

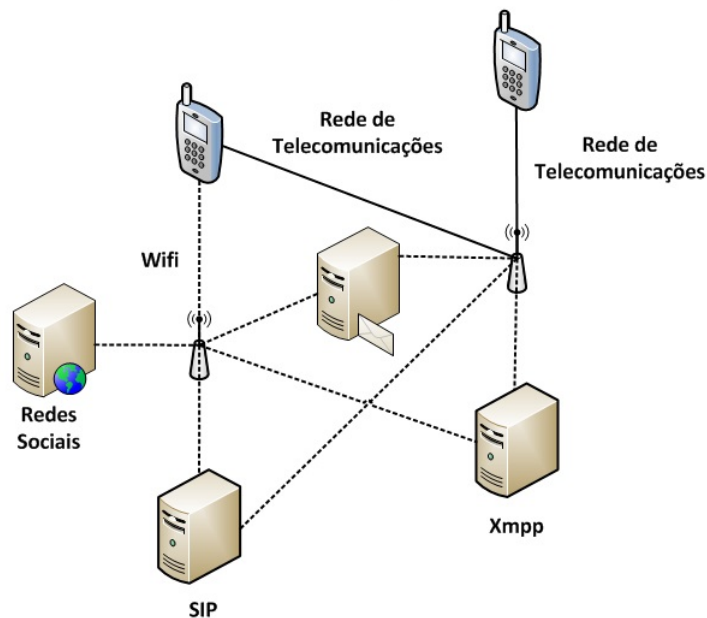


Figura 3.5: Arquitectura Estrutural

Como está presente na figura 3.5, é ainda necessário que o sistema tenha acesso a pelo menos um servidor de correio electrónico para que aplicação efectue o envio e recepção de mensagens electrónicas. O acesso aos servidores das redes sociais é também importante para a recolha de dados associados ao utilizador ou para adicionar um sistema de IM que é disponibilizado por estes servidores.

### 3.3 Funcionais

Ao delinear as funcionalidades do protótipo, iniciou-se uma execução gradual de tarefas básicas como o registo de dados do utilizador para armazenamento no dispositivo. Esta informação é necessária de modo a permitir simplificar a utilização da aplicação, como no caso da utilização do serviço de correio electrónico. Para descrever melhor todo este processo e outras funcionalidades que o protótipo oferece ao utilizador, é apresentado na figura 3.6 e figura 3.7 alguns possíveis cenários aos quais o utilizador têm acesso no protótipo. Estes cenários são posteriormente analisados de forma pormenorizada, com o auxílio de pré-condições, informação necessária e diagramas. Antes de detalhar estas funcionalidades, são apresentados na tabela 3.2 os requisitos e uma breve descrição associada a cada funcionalidade do sistema:

Requisitos	Descrição
	<b>Dados</b>
<b>Configuração</b>	Preenchimento da configuração dos serviços.
<b>Sincronização</b>	Processamento de dados externos do utilizador.
<b>Consistência</b>	Consistência de dados do dispositivo e da aplicação.
	<b>Serviços</b>
<b>Chamadas</b>	Comunicação através de chamadas telefónica e VoIP.
<b>Mensagens</b>	Envio de mensagens através do protocolo XMPP, SMS ou pelas redes sociais.
<b>Correio Electrónico</b>	Gestor de mensagens.
	<b>Interface</b>
<b>Configuração</b>	Pagerview com 5 layouts num só ecrã.
<b>Serviços</b>	Pagerview com 4 layouts num só ecrã, com todos os serviços disponíveis ao mesmo tempo.
<b>Visualização dos Contactos</b>	Lista de contactos, grupos e favoritos do utilizador.
<b>Visualização de Notícias</b>	Lista de informações disponibilizadas pelas redes sociais.
<b>Histórico de Serviços</b>	Lista associada a cada contacto, com o histórico de serviços efectuados.

Tabela 3.2: Requisitos funcionais.

O cenário de utilização presente na figura 3.6 pretende demonstrar algumas das primeiras acções que o utilizador efectua até obter acesso aos serviços do protótipo. Dos quatro cenários presentes na figura, o cenário 3.3.3 e 3.3.2 podem ser ignorados na primeira interacção, sendo os restantes obrigatórios. No caso dos dois cenários referidos serem ignorados na fase inicial, o utilizador não têm acesso aos dados externos relativas às redes sociais e os serviços de comunicação disponíveis são apenas o de SMS e chamada telefónica.

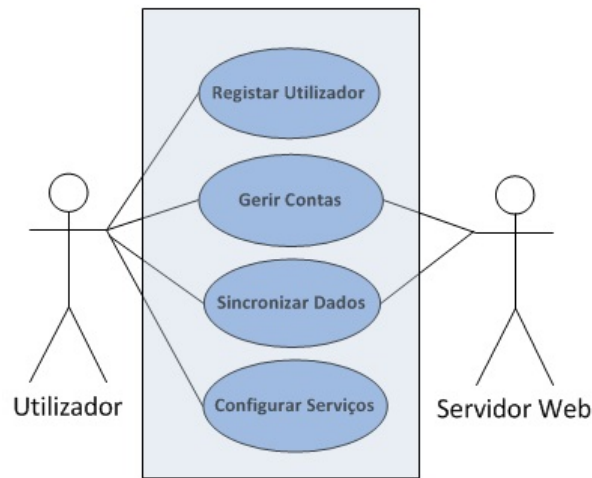


Figura 3.6: Cenário de Uso : Gestão de Contas

Os cenários dos vários serviços oferecidos pelo protótipo - figura 3.7 - representam todos os serviços de comunicação desenvolvidos. Os dois primeiros cenários são relativos ao serviço de chamadas de voz, seguindo-se três cenários de mensagens de texto e, por último, o correspondente ao acesso dos servidores das redes sociais, onde é feita a recolha e visualização de dados ou ainda o envio de mensagens para as redes sociais.

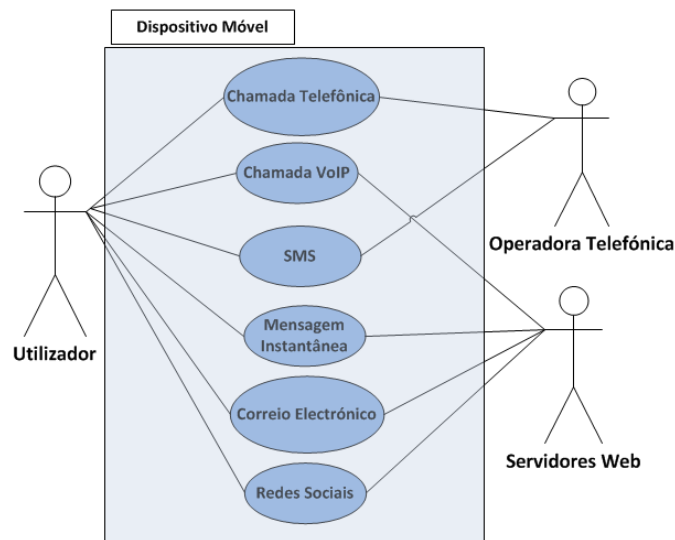


Figura 3.7: Cenário de Uso : Serviços de Comunicação

### 3.3.1 Registo do Utilizador

#### Pre-Condições:

- Inexistente

Este cenário é o primeiro passo para efectuar um dos requisitos do protótipo. O utilizador executa-o no primeiro contacto com a aplicação, preenchendo obrigatoriamente os campos pessoais:

- Nome
- Data de Nascimento
- Correio Electrónico
- Morada

Após a validação destes campos, o utilizador tem a opção de preencher mais informação que acrescenta ao protótipo os serviços de comunicação gratuitos, como chamadas VoIP, mensagens instantâneas e correio electrónico. Caso prossiga para a configuração dos restantes dados, o utilizador está automaticamente a efectuar o segundo cenário e deve preencher todos os campos presentes nos próximos ecrãs.

### Diagrama de Sequência

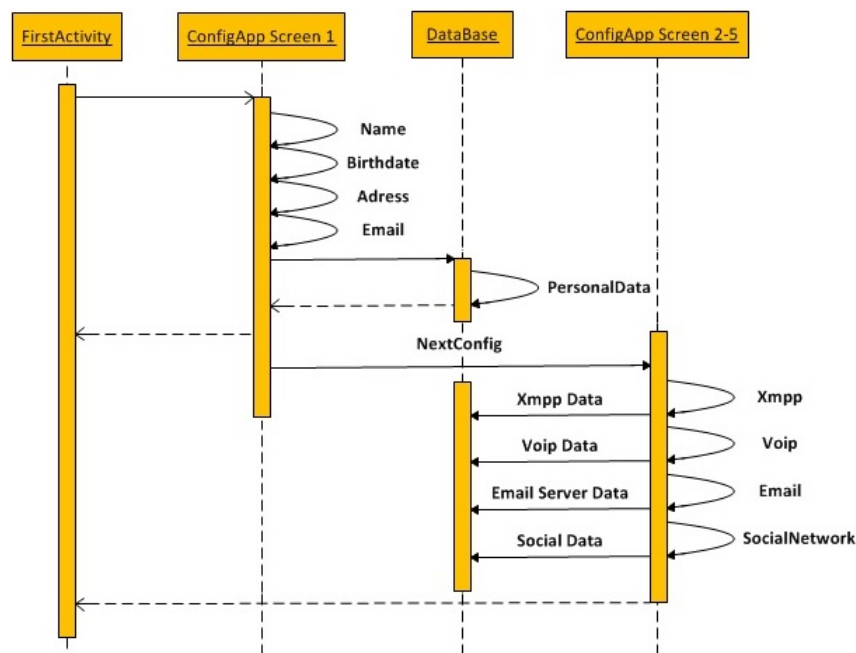


Figura 3.8: Registo do Utilizador e Configuração

### 3.3.2 Configuração dos Serviços

#### Pre-Condições:

- Registo do utilizador

O cenário de configuração dos serviços é apresentado ao utilizador quando este inicia pela primeira vez a aplicação, logo após o registo. Pode, posteriormente, usar um botão de

configurações para alterá-las, sendo este de fácil acesso nos vários ecrãs da aplicação. As respectivas configurações são exibidas de forma sequencial ao utilizador, através quatro ecrãs representado cada serviço:

<b>Perfil VoIP</b>	<b>Servidor VoIP</b>	<b>Servidor Mail</b>	<b>Servidor XMPP</b>
<ul style="list-style-type: none"> <li>• Extension</li> <li>• Password</li> <li>• Realm</li> </ul>	<ul style="list-style-type: none"> <li>• Name</li> <li>• IP</li> <li>• Description</li> </ul>	<ul style="list-style-type: none"> <li>• Username</li> <li>• Password</li> <li>• SMTP Host</li> <li>• SMTP Port</li> <li>• IMAP Host</li> <li>• IMAP Port</li> <li>• URL</li> </ul>	<ul style="list-style-type: none"> <li>• Username</li> <li>• Password</li> <li>• Host</li> <li>• Port</li> </ul>

Tabela 3.3: Ecrãs de configuração

Os campos acima referidos são necessários para o funcionamento dos serviços, sendo requerido ao utilizador que efectue a correcção dos campos com erro.

### 3.3.3 Gestão de Contas

#### Pre-Condições:

- Registo do utilizador
- Acesso a redes sem fios

Cenário básico de gestão de contas associadas ao utilizador. São possíveis, no máximo, gerir quatro tipos de contas:

- Manual : conta mais simples presente na aplicação.
- Android : conta que associada aos dados já existentes que foram recolhidos do *smartphone*.
- Facebook : conta associada à conta de Facebook do utilizador
- LinkedIn : conta associada à conta de LinkedIn do utilizador

Cada conta suporta a associação de dados específicos como, um fotografia, um ou mais endereços de e-mail, moradas residenciais e números telefónicos.

### 3.3.4 Sincronização de Dados

#### Pre-Condições:

- Registo do utilizador
- Configuração das Contas
- Acesso a redes sem fios



O último cenário, representado pela figura 3.6, oferece ao utilizador a possibilidade de automaticamente processar os dados de cada conta associada. Como referido no cenário de gestão de conta (3.3.3), existem quatro contas em que três contêm informações úteis sobre o utilizador, como os seus contactos e toda a informação pessoal associada. Todo o processamento de dados associado a cada conta é feito em plano de fundo, não sendo necessário aguardar por uma acção do utilizador. Este processamento e a metodologia da unificação de contactos é demonstrado de forma simples no diagrama 3.9:

### Diagrama de Sequência

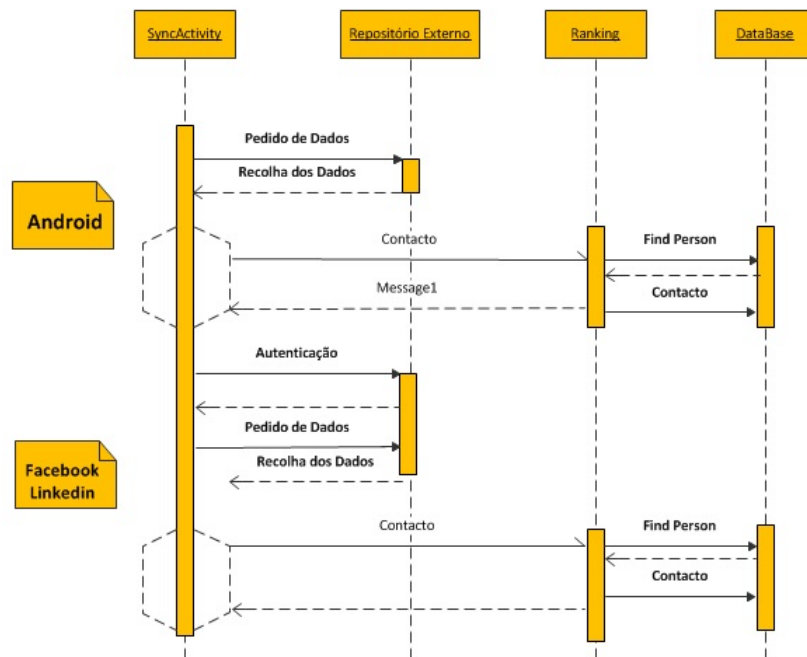


Figura 3.9: Sincronização e unificação de contactos

### 3.3.5 Chamadas de voz

#### Pre-Condições:

##### Gerais

- Registo do utilizador
- Dados sincronizados

##### Serviço VoIP

- Pre-configuração do perfil VoIP
- Pre-configuração do servidor VoIP
- Contacto precisa de estar online
- Acesso a redes sem fios

## Serviço Telefónico

- Cartão telefónico
- Crédito no cartão

Este cenário engloba os dois serviços de voz presentes na protótipo a ser desenvolvido. Em ambos os casos é necessário escolher qual o contacto para o qual é realizado o serviço de voz. Porém, para efectuar este tipo de serviço, é possível acedê-lo através de duas formas distintas. A primeira forma é através do ecrã de serviços, no qual o utilizador deve digitar o nome do contacto e seleccionar o tipo de serviço de voz que pretende. O utilizador poderá também aceder ao mesmo serviço caso prima o botão de contactos, seleccione o contacto e, ao visualizar as suas informações pessoais, pode então seleccionar qual o serviço de voz que pretende utilizar.

## Diagrama de Sequência

A figura 3.10 demonstra as interacções do sistema para os serviços de voz:

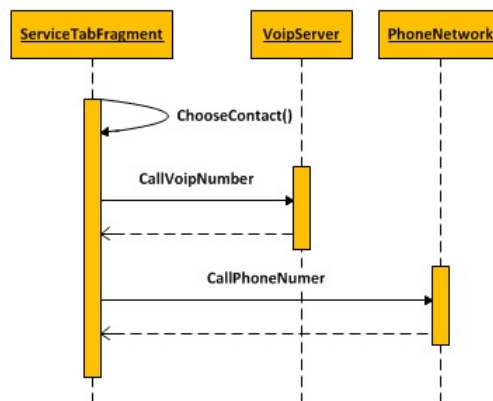


Figura 3.10: Serviço de Chamadas de voz

### 3.3.6 Mensagens de texto

#### Pré-Condições:

##### Gerais

- Registo do utilizador
- Dados sincronizados

##### Serviço XMPP

- Pré-configuração do perfil XMPP
- Contacto precisa de estar online
- Acesso a redes sem fios

##### Serviço e-mail

- Pré-configuração do servidor mail
- Servidor com protocolo SMTP e IMAP activo
- Acesso a redes sem fios

### Serviço Telefónico

- Cartão telefónico
- Crédito no cartão

### Serviço de redes sociais

- Pré-configuração da conta
- Permissões da rede social
- Acesso a redes sem fios

Tal como o cenário anterior, este também engloba outros serviços de mensagens, atingindo um total de quatro. O acesso a este cenário é relativamente semelhante ao referido na subsecção anterior(3.3.5) mas, ao aceder aos ecrãs de serviços, é necessário seleccionar o serviço de mensagem de texto.

### Diagrama de Sequência

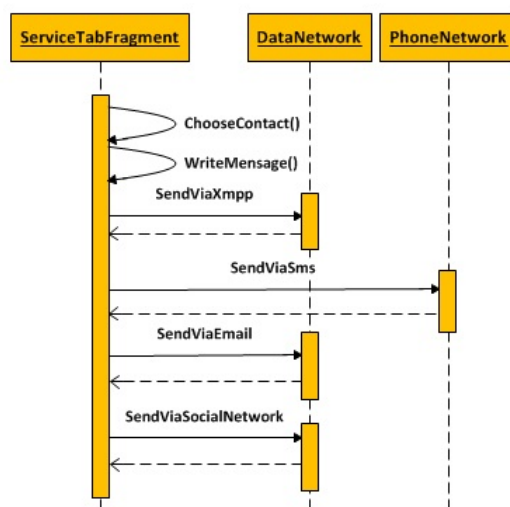


Figura 3.11: Serviço de mensagens de texto

### **3.3.7 Registo de serviços efectuados**

#### **Pré-Condições:**

##### **Gerais**

- Registo do utilizador
- Dados sincronizados

##### **Registo de serviços**

- Pré-configuração dos serviços

Este cenário é responsável por oferecer ao utilizador todo o histórico de serviços efectuados. Com esta funcionalidade, o sistema disponibiliza todas as informações relacionadas entre o utilizador e os seus contactos originando um histórico global.

## Capítulo 4

# Arquitectura do sistema

### 4.1 Estrutura física

A visão geral e conceptual do protótipo é demonstrada pela figura 3.5, este sistema tem como ponto central o *smartphone*. No entanto, as restantes estruturas são parte integrante dos requisitos do sistema para os serviços presentes no protótipo.

O dispositivo móvel incorpora o protótipo desenvolvido que funciona unicamente como cliente dos vários serviços implementados. Este comunica através de Wi-Fi ou através da rede de telecomunicações. Os quatro servidores presentes na figura 3.5 podem ser divididos em duas categorias, pelo facto dos servidores das redes sociais serem conceptualmente únicos. Os restantes são servidores externos, ficando dependentes do tipo de configuração estabelecida por parte do utilizador. Desta forma, o utilizador tem acesso a inúmeros servidores externos, simplificando e multifacetando os serviços de comunicação.

### 4.2 Base de Dados

#### 4.2.1 Evolução do Sistema

Ao abordar alguns aspectos iniciais sobre quais os serviços prioritários a desenvolver, tomou-se a decisão de começar pelo desenvolvimento do serviço VoIP e guardar os dados mais importantes do utilizador como nome, morado, número telefónico e data de nascimento. Foi definido também que a aplicação apenas terá um único utilizador (figura 4.1), representado pelas tabelas delineadas a preto. O utilizador irá precisar de inserir alguns dados pessoais, dados relativos ao seu perfil VoIP e do servidor ao qual se pretende ligar. Nesta etapa foi também delineado que é necessário guardar os dados de autenticação das várias contas de redes sociais que o utilizador possua.

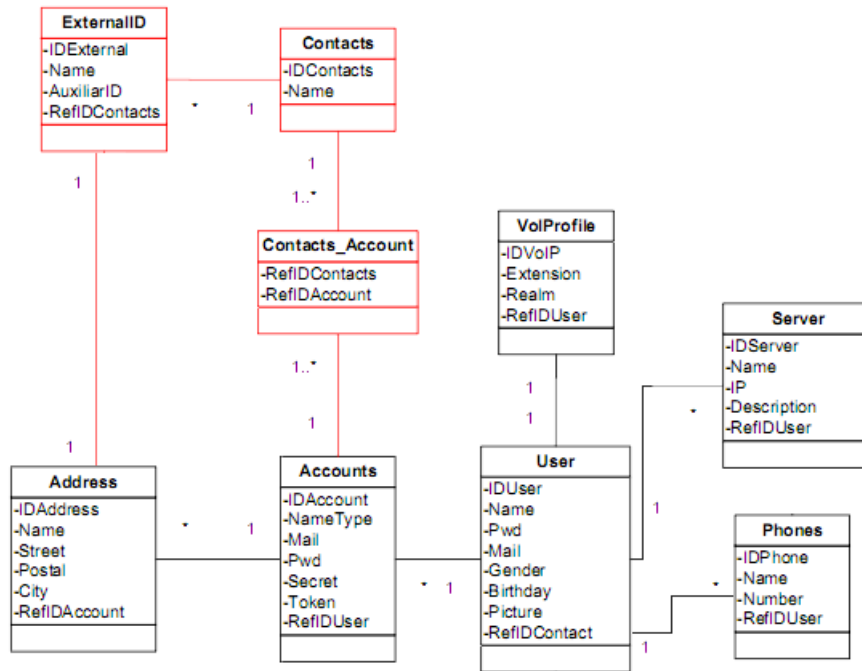


Figura 4.1: Primeira versão

Nesta primeira abordagem teve-se em conta que todos os contactos associados ao utilizador seriam acedidos em tempo real, tanto no acesso aos contactos do dispositivo móvel ou contactos associados através das suas contas externas. De forma a evitar o constante acesso às redes de dados para obter informação do utilizador, foram acrescentadas novas classes (delineadas a vermelho) no diagrama 4.2.

Pela adição das novas classes e de mais requisitos do sistema, seguiu-se uma reformulação do sistema como apresentado no diagrama 4.2, obtendo uma arquitectura mais complexa e, como consequência, uma aplicação mais completa do que a inicialmente previsto.

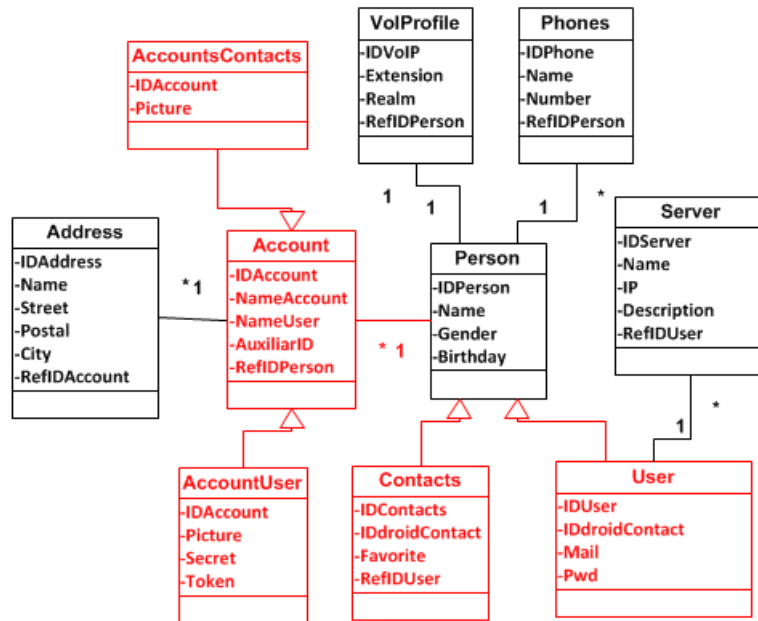


Figura 4.2: Segunda versão

O nome da classe "User" foi alterado para "Person", na qual existe agora duas generalizações (delineadas a vermelho), diferenciando o tipo associado a "Person". Foi reformulada também a classe "Conta", que com esta alteração distingue que tipo de conta deve representar e os dados relativo à conta.

Ao desenvolver o sistema deparou-se com alguns problemas já referidos anteriormente, o que veio trazer a necessidade de incluir novas classes no sistema para que corrigir essas falhas. Após estas correcções foram ainda analisados quais os dados necessários para a criação de uma nova classe relativa ao servidor de correio electrónico.

O diagrama 4.3 representa o estado final do sistema, tendo sido criada a classe "Mail", associada à conta do "User". Esta classe é bastante importante por ser a que sofre associações de novas classes, servidor de XMPP e *e-mail*. Com cada ocorrência da classe é possível associar um servidor de correio electrónico ou de mensagens instantâneas distinto. Com este aspecto adicional, o sistema torna-se flexível no acesso a diferentes tipos de servidores.

A classe "ServicesData", e respectivas generalizações, acrescentam ao sistema forma de manter um histórico de chamadas ou mensagens de texto realizadas na aplicação. Outra alteração é adição de uma classe permitindo a formação de grupos por parte do utilizador, oferecendo-lhe um forma cómoda para enviar mensagens e usufruir de conversas em grupo. Por ultimo ocorreu um acréscimo de um atributo na classe "Contacto", dando ao utilizador uma diferenciação sobre os contactos com que mais interage.

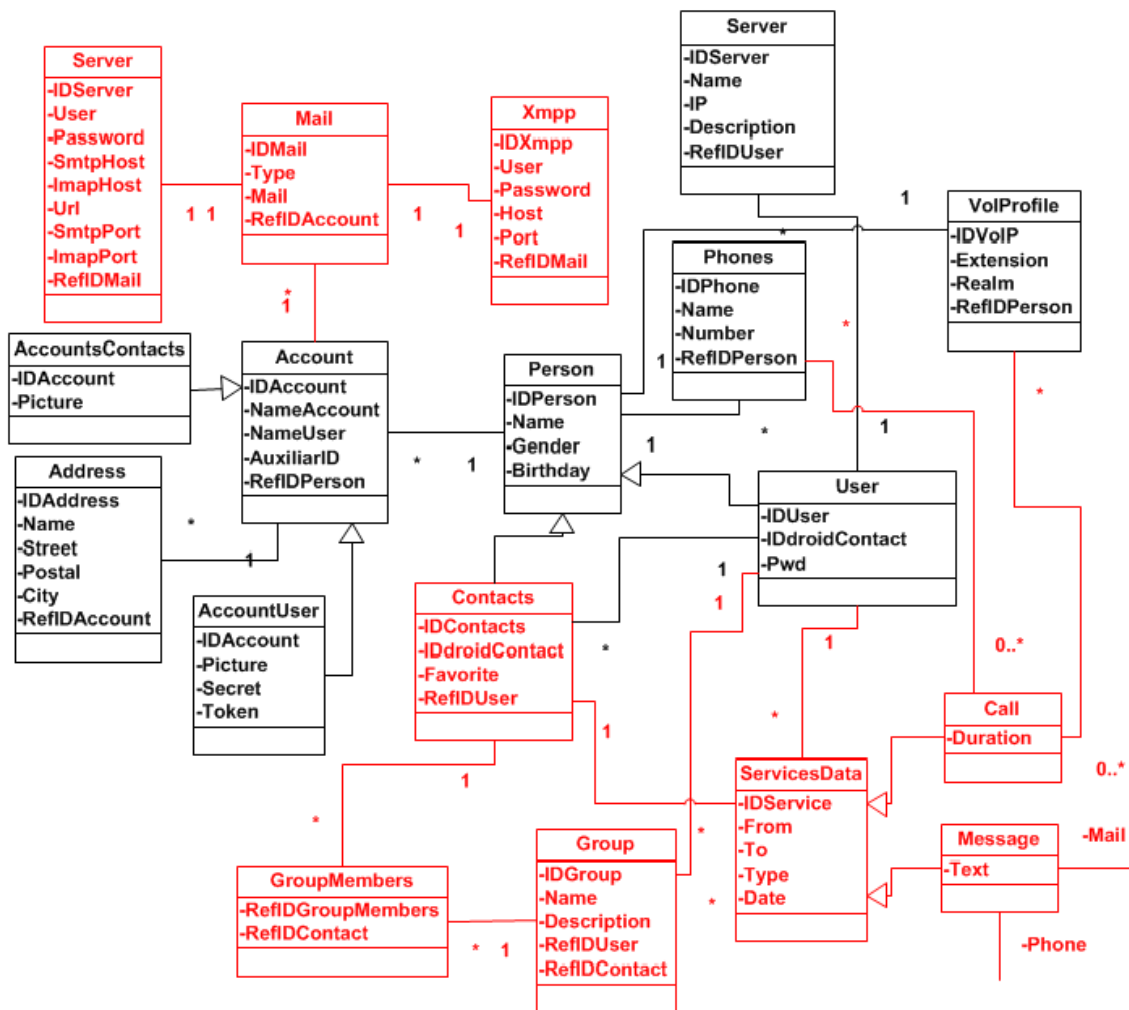


Figura 4.3: Versão Final

## 4.2.2 Solução

A classe "Person" é a classe principal que tem associada duas generalizações. Esta é a classe central da arquitectura e é fundamental para a gestão dos conteúdos pessoais associados ao utilizador e aos vários contactos presentes no sistema. Ambas as generalizações são úteis para diferenciar quais conteúdos associados a ela e manter a arquitectura modular.

A classe "User", em conjunto com a classe "Person", é considerado como o actor do sistema. Esta contém o atributo "IDdroidContact", que tem o objectivo de guardar o ID do utilizador presente na base de dados interna do dispositivo móvel, enquanto a "Password" serve para guardar a palavra-chave de autenticação.



<b>IDPessoa</b>	ID associado na base de dados.
<b>Name</b>	Nome Completo.
<b>Gender</b>	Género da Pessoa. 'M'(Masculino) ou 'F'(Feminino).
<b>BirthDay</b>	Data de Nascimento da Pessoa.

Tabela 4.1: Tabela Person

<b>IDUser</b>	ID do utilizador associado.
<b>IDdroidContact</b>	ID da base de dados interna do Android.
<b>Password</b>	Palavra-passe da aplicação.

Tabela 4.2: Tabela User

"Contacts" é a classe responsável pela informação das pessoas associadas ao utilizador. Esta classe contém também o atributo "IDdroidContact", presente também na classe "User", e o atributo "Favorite" que sinaliza a distinção do contacto.

<b>IDContact</b>	ID do contacto associado.
<b>IDdroidContact</b>	ID da base de dados interna do Android.
<b>Favorite</b>	Caracterizar o contacto como favorito.
<b>RefIDUser</b>	ID da Pessoa associada.

Tabela 4.3: Tabela Contacts

A classe "Phone" e "VoIPProfile" servem para guardar os dados relacionados com os serviços de voz. As duas possuem o número de contacto de ambos os serviços, na primeira com o nome "Number" e na segunda com o nome "Extension". Na tabela 4.5 encontra-se o atributo "Realm", com intenção de guardar o endereço do servidor ao qual o perfil pertence.

<b>IDPhone</b>	ID do telefone associado.
<b>Name</b>	Nome do tipo de número associado.
<b>Number</b>	Número de telefone.
<b>RefIDPerson</b>	ID da "Pessoa" associada.

Tabela 4.4: Tabela Phone

<b>IDVoIP</b>	ID do perfil VoIP associado.
<b>Extension</b>	Número de telefone associado ao perfil VoIP.
<b>Realm</b>	Domínio do perfil.
<b>RefIDPerson</b>	ID da "Pessoa" associada.

Tabela 4.5: Tabela VoIPProfile

O serviço VoIP contém ainda a tabela 4.6, a fim do utilizador poder configurar o servidor VoIP no qual registou o seu perfil. O atributo "Name" é indicativo do nome do servidor, não sendo relevante para realizar a conexão. O mesmo é verdade para o campo descrição, sobrando então o atributo IP que é o único importante para a efectuar a ligação ao servidor.

<b>IDServer</b>	ID do servidor VoIP associado.
<b>Name</b>	Nome do servidor VoIP.
<b>IP</b>	IP do servidor VoIP.
<b>Description</b>	Descrição breve do servidor.
<b>RefIDUser</b>	ID do Utilizador associado.

Tabela 4.6: Tabela ServerVoIP

Tal como a classe "Person", a classe "Account" possui também ela duas generalizações, contendo os campos necessários para guardar a informação sobre as diferentes contas que o utilizador controla. Esta tem associada a si endereços postais e endereços de correio electrónico, para que toda a informação relacionada com o utilizador fique completa e sem replicação de dados.

<b>IDAccount</b>	ID da Conta associada.
<b>NameAccount</b>	Nome da conta externa.
<b>NameUser</b>	Nome que o utilizador tem associado na conta.
<b>AuxiliarID</b>	Identificador do utilizador na conta externa.
<b>RefIDPessoa</b>	ID da Pessoa associada.

Tabela 4.7: Tabela Account

A tabela "AccountUser" e "AccountContacts" têm o intuito de distinguir o tipo de conta associada a cada pessoa. Ambas possuem objectivos diferentes no sistema, especialmente a "AccountUser", que contém campos de configuração para as redes sociais. No caso da tabela "AccountUser" 4.8, os campos "Secret" e "Token" são recolhidos sobre a rede social configurada e são importantes para o armazenamento de códigos de autenticação. Os restantes campos são de chaves primárias e estrangeiras.

Quanto à tabela relacionada com a "AccountContacts", não contém nenhuma informação extra, servindo simplesmente para distinção da conta:

<b>IDAccount</b>	ID da Conta associada.
<b>Picture</b>	Fotografia da conta externa.
<b>Secret</b>	1º Código de acesso a conta externa.
<b>Token</b>	2º Código de acesso a conta externa.
<b>RefIDPessoa</b>	ID da Pessoa associada.

Tabela 4.8: Tabela AccountUser

<b>IDAccount</b>	ID da Conta associada.
<b>Picture</b>	Fotografia da conta externa.

Tabela 4.9: Tabela AccountContacts

A classe "Address" tem o objectivo de não replicar endereços, sendo constituída pelos campos "Rua", "Cidade" e "Código Postal". Esta classe efectua a distinção do tipo de endereço, de forma a clarificar qual o de trabalho ou residencial. A classe encontra-se apenas associada

à classe "Conta" para que, quando é criado um novo endereço, o sistema de unificação de contactos verifica se o endereço já se encontra no sistema, possibilitando também associar a um contacto já presente no protótipo.

<b>IDAddress</b>	ID do endereço da conta.
<b>Name</b>	Tipo do endereço.
<b>Street</b>	Rua do endereço.
<b>Postal</b>	Código Postal.
<b>City</b>	Nome da Cidade.
<b>RefIDAccount</b>	ID da Conta associada.

Tabela 4.10: Tabela Address

Nas três tabelas seguintes, representam as configurações dos serviços de XMPP e de correio electrónico. A tabela "Mail" 4.11 é associada à tabela "Account" e cada uma delas contém uma configuração associada do serviço de correio electrónico e/ou de XMPP. Como tal, é possível ao utilizador configurar mais do que um correio electrónico e/ou serviço de XMPP. A tabela "Mail" 4.11 para além do endereço de correio electrónico, inclui ainda o campo "Type" para diferenciar o seu género, assim o utilizador tem a opção de enviar mensagens direccionadas exclusivamente através do seu endereço de trabalho ou pessoal.

A tabela "XMPP" 4.12 contém os campos necessários para a configuração do serviço de XMPP, usando o campo "User" e "Password" para autenticação no servidor e os campos "Host" e "Port" para a ligação ao servidor associado ao correio electrónico. A tabela "Server" 4.13 é destinada à configuração do servidor do correio electrónico e, para além do campos "User" e "Password", inclui campos de configuração específicos de "Host" e "Port" para os protocolos IMAP e SMTP.

<b>IDMail</b>	ID da correio electrónico.
<b>Type</b>	Tipo do correio electrónico.
<b>Mail</b>	Endereço do correio electrónico.
<b>RefIDAccount</b>	ID da Conta associada.

Tabela 4.11: Tabela Mail

<b>IDXmpp</b>	ID da configuração XMPP.
<b>User</b>	Nome do utilizador.
<b>Password</b>	Palavra-passe.
<b>Host</b>	Endereço do servidor.
<b>Port</b>	Porta do servidor.
<b>RefIDMail</b>	ID do correio electrónico associado.

Tabela 4.12: Tabela XMPP

<b>IDServer</b>	ID do servidor de correio electrónico.
<b>User</b>	Nome do utilizador.
<b>Password</b>	Palavra-passe.
<b>Smtphost</b>	Endereço SMTP do correio electrónico.
<b>Smtpport</b>	Porta SMTP do correio electrónico.
<b>imapHost</b>	Endereço IMAP do correio electrónico.
<b>imapPort</b>	Porta IMAP do correio electrónico.
<b>Url</b>	Domínio do servidor.
<b>RefIDMail</b>	ID do correio electrónico associado.

Tabela 4.13: Tabela Server

As próximas tabelas têm o objectivo de permitir ao utilizador guardar um histórico de todas as comunicações efectuadas pelo utilizador. Com estas informações, o sistema oferece ao utilizador uma forma de visualizar os conteúdos localmente e até sugerir o melhor método de contacto com base na resposta às suas comunicações.

Os campos presentes na tabela "ServicesData" 4.14 guarda o remetente, destinatários, data e subtipo de serviço efectuado, no qual o campo destinatário pode, ou não, ser uma única entidade.

<b>IDServiceData</b>	ID associados ao serviço.
<b>Type</b>	SubTipo do serviço.
<b>Date</b>	Data realização do serviço.
<b>From</b>	ID Remetente do serviço.
<b>To</b>	ID Destinatário do serviço.

Tabela 4.14: Tabela ServicesData

As próximas classes são generalizações da classe anterior. A tabela "Message" 4.15 contém informação relativa a serviços relacionados com mensagens enviadas, utilizando o campo "Text" para guardar qual a mensagem enviada. Quanto à tabela "Call" 4.16, esta possui o campo "Duration" para guardar a duração da chamada e o campo "Status" para manter o registo do estado da chamada, ou seja: se foi realizada com sucesso, se estava ocupado, ou até mesmo desligado.

<b>IDMessage</b>	ID associado ao serviço.
<b>Text</b>	Texto da mensagem.
<b>RefIDMail</b>	ID do correio electrónico associado.
<b>RefIDPhone</b>	ID do telefone associado.

Tabela 4.15: Tabela Message

<b>IDCall</b>	ID associados ao serviço.
<b>Duration</b>	Duração da chamada.
<b>Status</b>	Estado da chamada.
<b>RefIDMail</b>	ID do correio electrónico associado.
<b>RefIDPhone</b>	ID do telefone associado.

Tabela 4.16: Tabela Call

As tabelas "Group" e "GroupMembers", oferecem ao utilizador acesso à gestão de grupos, oferecendo a funcionalidade de envio de mensagens globais para os vários contactos associados ao grupo, evitando escolher cada utilizador de forma individual.

<b>IDGroup</b>	ID do grupo.
<b>Name</b>	Nome do grupo.
<b>Description</b>	Descrição do grupo.
<b>RefIDUser</b>	ID do utilizador associado.

Tabela 4.17: Tabela Group

<b>RefIDGroup</b>	ID do Grupo associado.
<b>RefIDContacto</b>	ID do contacto associado.

Tabela 4.18: Tabela GroupMembers



# Capítulo 5

## Implementação

Este capítulo encontra-se dividido em três secções, de modo a clarificar as etapas do desenvolvimento do protótipo. Após a abordagem inicial de concepção do modelo do protótipo, seguiu-se o desenvolvimento do modelo gráfico da aplicação e a implementação da funcionalidade pretendida sobre as interfaces de programação.

### 5.1 Modelo gráfico do protótipo

Nesta secção é abordado o modelo gráfico do protótipo, começando pelos requisitos iniciais, de modo a que as interações entre o protótipo e o utilizador sejam ergonómicas. Após esse esboço inicial segue-se a implementação do modelo gráfico, mostrando ao longo da implementação as alterações ocorridas aquando da sua transição entre maquete e produto final.

#### 5.1.1 Maqueta gráfica

Ao criar a interface gráfica do protótipo, foi projectado o ecrã principal de forma a manter uniformidade por toda a aplicação. Com este aspecto em mente, o ecrã principal possui o logotipo da aplicação em destaque e dois botões importantes: configurações e contactos. Neste ecrã estão ainda presentes os botões de autenticação e registo. No caso do botão de autenticação, este é visível apenas se o utilizador activar a opção de segurança, que requer autenticação ao iniciar a aplicação. O botão de registo é colocado abaixo do botão de autenticação, devido a ser menos utilizado 5.1.:

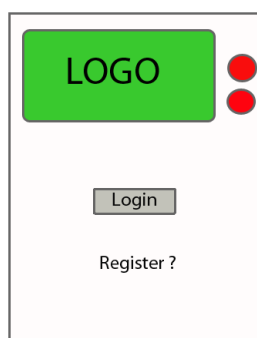


Figura 5.1: Ecrã principal

O segundo ecrã é desenhado de modo a reunir todos dos serviços presentes no protótipo. Com a implementação dos vários serviços efectuada de forma individual, foi necessário projectar cuidadosamente a interface deste ecrã de modo a que a agregação de todos os serviços fosse facilmente acessível. O resultado final é visível na figura "Ecrã de serviços" 5.2. A interface foi desenhada para conter os vários serviços da aplicação através de uma barra de tarefas incorporada. Através dos diferentes ícones é possível distinguir qual o serviço a utilizar: texto, voz, redes sociais ou consulta de serviços(notícias). A motivação para esta abordagem deve-se à forma pela qual os serviços são utilizados: no botão da rede social é apresentada a interface para o utilizador actualizar o seu estado, algo que não é partilhado com outros serviços. São ainda inseridos na barra de tarefas o botão de telefone e das notícias.



Figura 5.2: Ecrã serviços

Devido ao sistema não ser auto-configurável, após o registo do utilizador existe a necessidade de desenvolver no protótipo uma interface para a configuração manual dos serviços. Para este efeito, seguiu-se a abordagem anterior da implementação do ecrã de serviços, usando uma barra de tarefas com botões de rápido acesso.

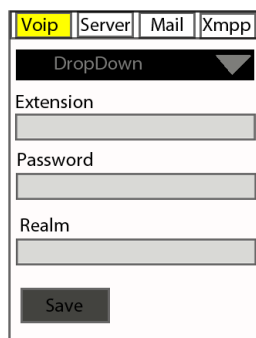


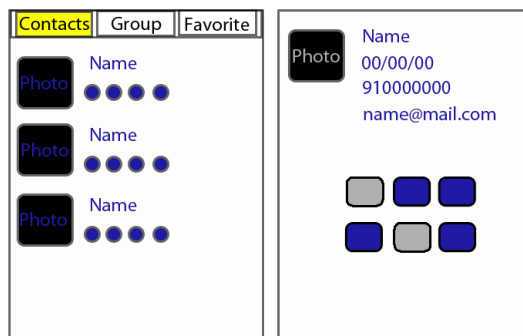
Figura 5.3: Ecrã configuração

A elaboração de uma lista, como representada na imagem esquerda da figura 5.4a, permite ao utilizador o acesso directo aos contactos. A necessidade de criar esta interface deve-se ao protótipo oferecer as informações sobre os seus contactos e disponibilizar a escolha do serviço a usar. Para além do acesso aos contactos, é disponibilizado ainda o acesso aos grupos e contactos classificados como favoritos.

Ao desenhar a interface 5.4b, pretendeu-se colocar o mínimo de informação possível na lista, de modo a oferecer ao utilizador apenas os conteúdos de forma simplificada. Com este objectivo,



apenas se inclui o nome do utilizador, serviço de chamada de voz, SMS e ícone para proceder para o perfil do contacto.



(a) Ecrã contactos (b) Ecrã do perfil

Figura 5.4: Ecrãs de contactos

A última interface a ser projectada é representada na figura 5.5. Este ecrã foi desenhado para conter as informações relativas a todos os serviços efectuados por parte do utilizador. Desta forma, é possível ter acesso ao registo dos serviços em qualquer momento e consultá-los para recordar quais as últimas conversas que manteve. Esta é uma interface na qual é apenas colocada a fotografia da pessoa que efectuou os serviços, o seu respectivo nome e mensagem associada a este serviço.

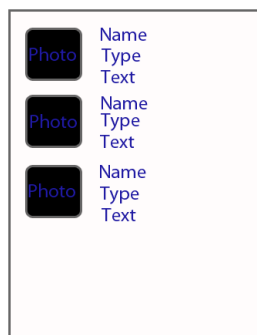


Figura 5.5: Ecrã do histórico de serviços

### 5.1.2 Interfaces gráficas

A concepção inicial do visual do protótipo sofreu alterações durante todo o processo de desenvolvimento, mantendo-se apenas o ecrã de apresentação. Este está presente na figura 5.6 e permaneceu inalterável ao longo do seu desenvolvimento, com o objectivo do mecanismo de autenticação estar claramente visível. No entanto, é possível ao utilizador desactivar este campo através da funcionalidade de autenticação automática.

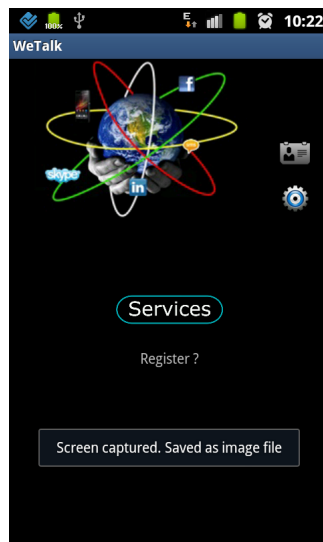


Figura 5.6: Ecrã de principal

Este primeiro ecrã contém também o acesso ao registo do utilizador no sistema através do botão "Register?". Para alterar esta informação relativa à autenticação (automaticamente configurada aquando de um registo no sistema), é possível alterar os dados no ecrã de configuração de sistema figura 5.7. Neste ecrã mostra-se o cenário de configuração de dados do sistema por parte do utilizador. Como é possível constatar, o ecrã contém cinco configurações apresentadas através de um *pageview* com cinco *views*. Desta forma, não é necessário conceber cinco actividades distintas, simplificando a interface da aplicação.

A mudança de *views* pode ser efectuada através de duas formas: pressionando a aba de cada serviço a configurar ou através da acção de "deslize" da *view*, sendo esta automaticamente alterada para a *view* da configuração seguinte.

De forma a simplificar a configuração de parâmetros dos serviços, os valores configurados são apresentadas automaticamente nas caixas de texto. Caso existe alguma configuração prévia, cada configuração só é guardada quando o botão "Guardar" é activado pelo utilizador na respectiva *view*.

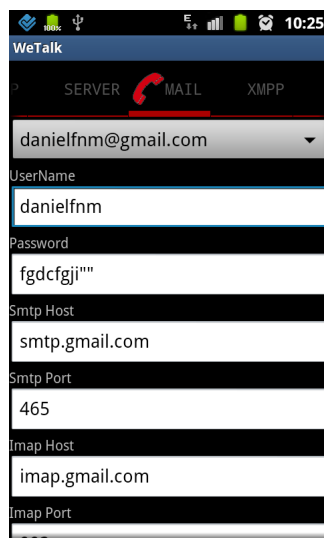


Figura 5.7: Ecrã de configuração do sistema

Após a configuração do utilizador, é possível usar o ecrã principal para três acções distintas. O diagrama 5.8 transmite uma melhor exposição das interações nos restantes ecrãs do sistema. Este diagrama oferece apenas metade dessas interações, focando-se apenas na configuração de dados e visualização - as restantes interações são abordadas posteriormente.

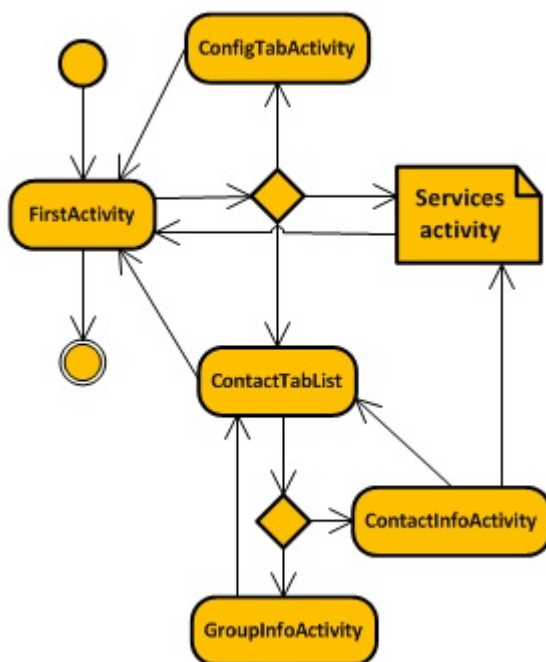
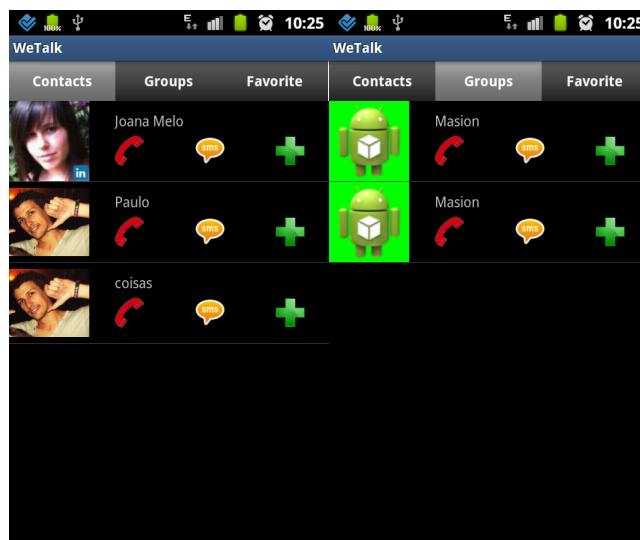


Figura 5.8: StateChart: Fluxo da aplicação, parte 1

No ecrã criado especificamente para a visualização dos contactos do utilizador foram acrescentadas duas abas extra para representar também os grupos criados e os contactos favoritos do utilizador. Estes dados, como é possível verificar na figura 5.9a, são expostos através de três *list views*, que contêm a fotografia e o nome associado ao contacto ou grupo em questão. Ao pressionar o contacto é apresentado um novo ecrã (figura 5.10) com os dados do contacto. Estes dados pessoais são obtidos pelo acesso às contas do utilizador, enquanto os serviços são apresentados apenas se configurados. No entanto, é possível outra interacção a partir deste ecrã: ao pressionar o contacto de forma prolongada, a interface automaticamente activa a edição de dados do contacto.



(a) Vista de contactos

(b) Vista de grupos

Figura 5.9: Ecrãs de contactos

O ecrã mostra os dados relativos ao contacto seleccionado pelo utilizador, como fotografia, nome, data de nascimento, número telefónico, correio electrónico, morada e serviços disponíveis. Os serviços são apresentados através de imagens, em que cada imagem tem dois estados de exibição para distinguir entre os serviços do contacto que estão *online* ou *offline*. Neste ecrã é possível, através da activação dos ícones, realizar qualquer serviço de comunicações com o contacto em questão.

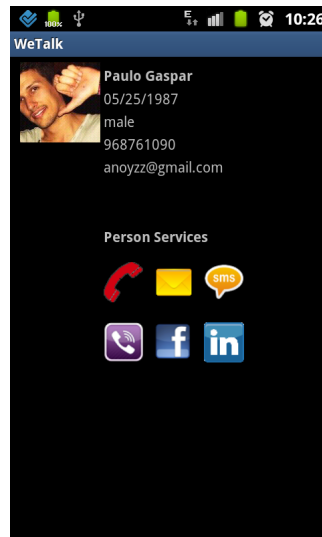


Figura 5.10: Informações do contacto

A segunda parte das interacções do protótipo são apresentadas no diagrama 5.11, que aborda detalhadamente todos os restantes serviços presentes na aplicação. A bifurcação no diagrama representa as quatro *views* criadas para a apresentação dos serviços de comunicação, gestão de *e-mail* e mural das redes sociais. Tal como na figura 5.7, este ecrã também contém um *pageview* que serve para simplificar as acções do utilizador. As acções feitas a partir de cada *view* são efectuadas após a interacção com o botão do serviço. Os mecanismos de validação de dados estão presentes em algumas das *views*, para garantir o bom funcionamento dos serviços.

Outro aspecto a ter em conta: quando o utilizador pretende efectuar algum serviço, é a verificação de dados do contacto que avisa o utilizador se o contacto contém os serviços de comunicação. Caso contrário, notifica o utilizador da impossibilidade de realizar essa acção.

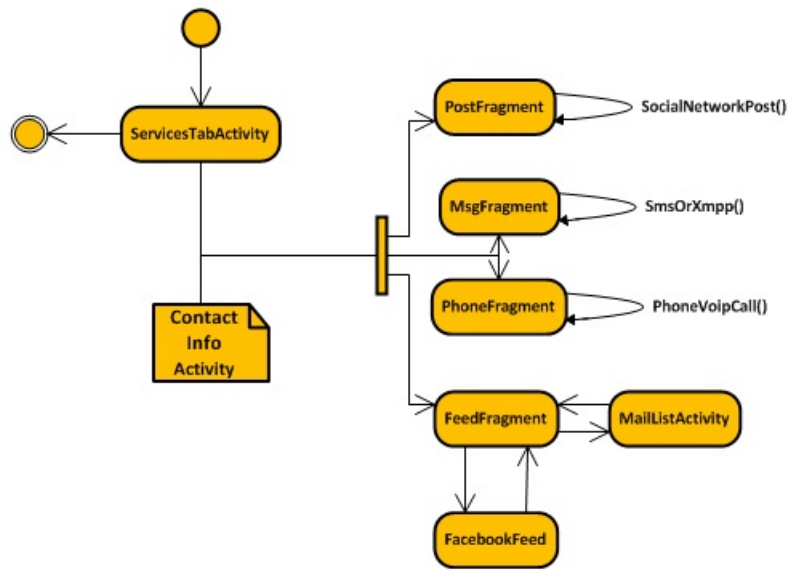
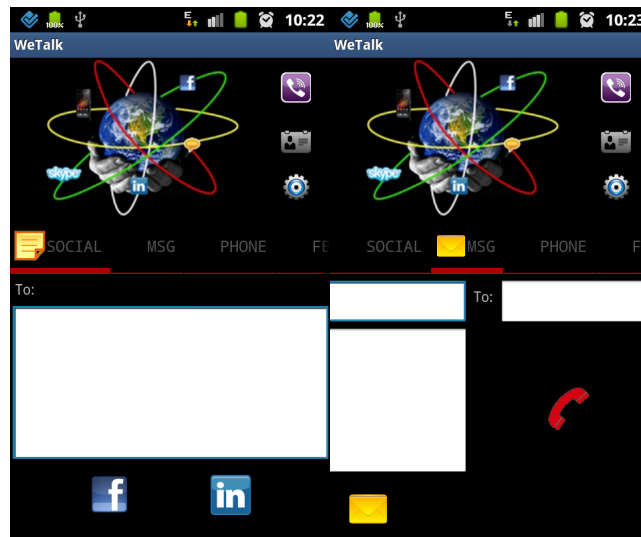


Figura 5.11: StateChart: Fluxo da aplicação, parte 2

A primeira interface, apresentada na figura 5.12a, é relativa à actualização do estado do mural nas redes sociais. A segunda e terceira interfaces são responsáveis por efectuar o serviço de texto e voz. É possível verificar na figura 5.12b como é efectuada a transição entre os ecrãs dos serviços, através da acção de "arrastar" o ecrã para a esquerda ou direita, na horizontal. A reacção da interface da aplicação é imediata à interacção do utilizador com o ecrã.



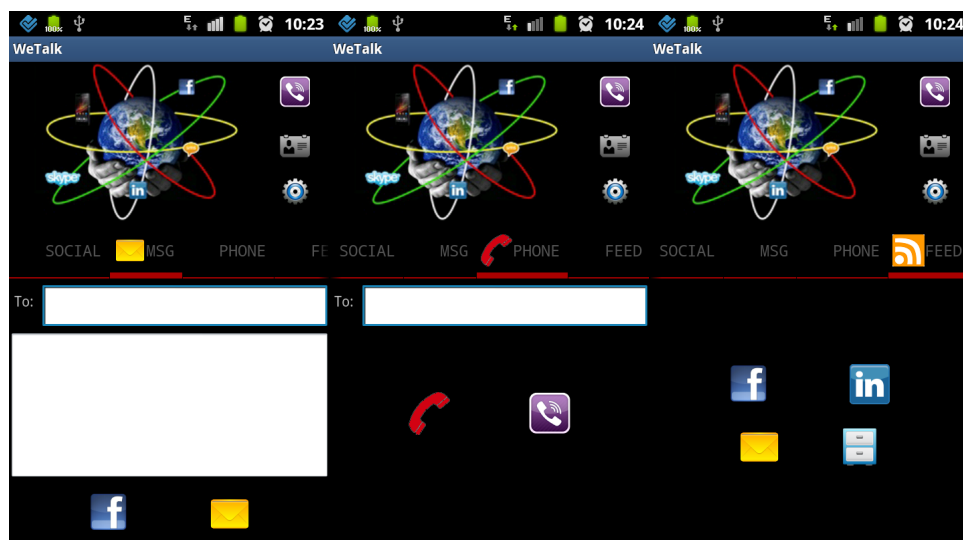
(a) Interacção com redes sociais (b) Navegação entre views

Figura 5.12: Ecrã de serviços 1

A próxima figura (5.13) representa as interfaces dos serviços de mensagens, telefone e

notícias. A primeira interface inclui o campo destinatário, um campo para a mensagem de texto e botões relativos aos serviços que são usados para o envio da mensagem. A interface representada na figura 5.13b também contém um campo para inserção do destinatário como na interface anterior e ambas possuem a funcionalidade de *Auto Complete*, de modo a evitar erros na introdução do nome. A interface possui dois botões relativos aos serviços telefónicos: chamada normal e chamada VoIP.

A restante interface (5.13c), oferece uma forma de verificar de conteúdos, mais especificamente acesso aos seus dados do Facebook, LinkedIn, correio electrónico e o histórico de mensagens.



(a) Serviço de Mensagens (b) Serviço de Chamadas (c) Mural de Notícias e Email

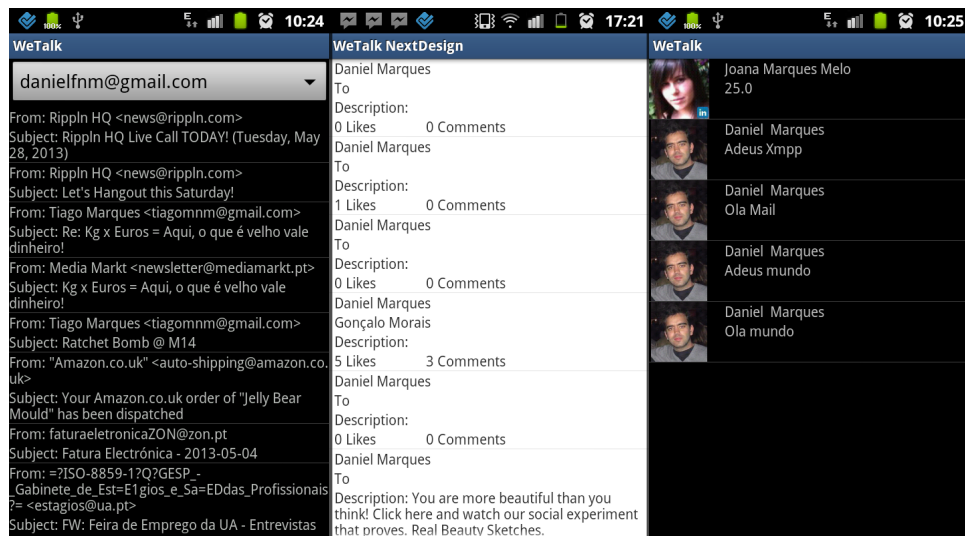
Figura 5.13: Ecrã de serviços 2

As três interfaces da figura 5.14, representam os conteúdos apresentados pelos serviços referidos na figura 5.11.

A visualização das mensagens electrónicas é feita através de uma *Listview*, mostrando ao utilizador as mais recentes primeiro. Para cada mensagem presente na interface gráfica, é mostrado o nome do remetente e o assunto, de modo a otimizar o número de mensagens que é possível exibir nestes ecrãs de *smartphones*. Ao apresentar as mensagens, a aplicação utiliza uma *drop down* para que o utilizador aceda facilmente entre os vários endereços de correios electrónicos configurados. Desta forma, o utilizador tem acesso a esses conteúdos separadamente, para uma melhor disposição da informação.

A segunda interface possui as mensagens publicadas pelo utilizador ou amigos no seu mural do Facebook. Na apresentação de cada mensagem é possível verificar a sua descrição, qual o remetente e destinatário, número de comentários associados e o número de "Gosto" que contém.

A última interface inclui registos de todas as mensagens de texto e serviços de voz efectuados, obtendo assim o histórico global de utilização.



(a) Mensagens do correio (b) Mural pessoal (c) Histórico dos serviços electrónico

Figura 5.14: Mensagens recebidas

## 5.2 APIs do protótipo

### 5.2.1 SIP Manager

Ao analisar as opções existentes para a plataforma Android, adoptou-se esta interface de programação nativa de telefonia, presente a partir da versão 2.3 do sistema operativo, para realização de chamadas VoIP.

Com a adopção desta interface é possível efectuar também toda a gestão de chamadas, sem que seja necessário tratar os passos do protocolo SIP de forma manual. Para utilizar esta interface de programação torna-se necessário inserir os dados do serviço SIP na aplicação. Esta acção cria um perfil que é consultado sempre que se pretenda efectuar uma chamada VoIP.

O uso desta interface de programação na aplicação inicia-se através da implementação de uma actividade para a gestão das sessões entre o utilizador e servidor. Esta actividade começa por verificar se existe a conectividade de dados, a fim de poder iniciar a conexão aos servidores configurados pelo utilizador. Na aplicação é criada um *Broadcast Receiver* para receber as várias notificações de chamada através da interface SIP Manager, despoletando a actividade responsável por mostrar o ecrã de chamada em espera.

### 5.2.2 Facebook SDKs e APIs

Para o suporte de serviços *Facebook* foram utilizadas dois *software development kits*: *Easy Facebook Android SDK (EFBSDK)* e *Facebook SDK for Android (FBSDK)* - o SDK oficial do Facebook. Adoptou-se principalmente o uso do EFBSDK, devido à sua simplicidade, mas foi também necessário adoptar também o FBSDK, devido a lacunas do EFBSDK. A maior falha do EFBSDK prende-se com a falta de funcionalidade do envio de mensagens para a caixa de correio de outro contacto.



O desenvolvimento inicial efectuado com o SDK oficial do Facebook é simples, já que é reutilizado todo o código fornecido pela empresa para efectuar a autenticação, o acesso aos dados do utilizador, e a gestão de sessões do utilizador. Após o primeiro registo do utilizador, o *token* de acesso do utilizador é armazenado na base de dados local da aplicação por um determinado período de tempo. Assim, é possível que o acesso aos serviços seja feito de forma mais rápida.

Quanto ao acesso aos dados, é possível realizá-lo através de três formas:

- Servidor REST.
- Graph API.
- URLs.

O acesso aos dados do utilizador é feito através do FBSDK, que utiliza a Graph API para efectuar o envio das mensagens de texto ou recolha de dados. Como a informação recebida está no formato JSON, é necessário realizar o tratamento de dados e criar uma estrutura de dados que organiza essa informação de forma mais adequada ao sistema desenvolvido. Ao interagir com esses dados, verificou-se a inexistência da separação do tipo de dados, não havendo distinção entre uma publicação de texto e uma publicação com conteúdo multimédia, como uma actualização da fotografia de perfil, o que dificulta todo o processo de tratamento de dados.

O EFBSDK contém os seus próprios objectos, realizando o processo de *parsing* dos objectos JSON recebidos de forma transparente para o programador. Este SDK mostrou-se muito mais adequado ao modelo de acesso aos dados do utilizador, nomeadamente as suas publicações e contactos, contrastando com o FBSDK.

Em ambos os casos, a questão de privacidade de conteúdos é realizada do mesmo modo, pois os dados do utilizador não são facultados de um único modo, mas apenas accionando as várias permissões pretendidas para aplicação. Com esta implicação, o programador não tem acesso a todos os dados de forma directa, precisando da autorização do utilizador sempre que é necessária a actualização de permissões.

### 5.2.3 LinkedIn API

Analisando a metodologia de autenticação para acesso aos dados do LinkedIn, verificou-se a necessidade de utilização da classe command-line interface (CLI) para realizar a configuração. Esta é criada com base na chave de acesso e chave secreta, presentes no perfil de desenvolvimento da aplicação do LinkedIn, e através da verificação de um URL, necessário para a finalização do pedido de autenticação. Ao realizar o acesso ao URL fornecido pelo *browser* presente no Android, é necessário recolher o código de segurança e proceder então à autenticação através da junção desse código com os restantes dados do utilizador. Após estas etapas, o servidor fornece o acesso aos dados do utilizador e só necessita de realizar novamente estes passos de autenticação quando a sessão for terminada pelo servidor.

### 5.2.4 XMPP

Com as limitações da interface de programação *SIP Manager*, que não possibilitando o envio de mensagens de texto entre os utilizadores, é necessária a utilização de uma nova interface chamada *Smack*, que utiliza o protocolo XMPP.

Este protocolo de comunicação é focado no envio de mensagens de texto através de IP, podendo ser utilizado através de TCP, HTTP ou WebSockets. Todo o processo de implementação na aplicação é relativamente simples, sendo a etapa inicial de análise da API relativamente curta, visto que este processo envolveu utilizar o serviço XMPP da Google. Posteriormente, o sistema estendeu-se ao suporte de várias configurações de outros servidores não-Google, usando a base de dados local para armazenamento de todas as configurações. Outras funcionalidades, para além do envio de mensagens, como o estado actual do contacto, são também disponibilizadas pela *Smack*.

### 5.2.5 EMAIL

Para a implementação do suporte de serviços de correio electrónico foi escolhida a interface de programação *JavaMail*. Adoptou-se esta interface de programação pelo facto de já ser uma versão estável e multiplataforma que torna possível uma abstracção dos protocolos de e-mail utilizados para comunicação com o servidor de e-mail.

A recepção das mensagens electrónicas é efectuada pelos protocolos POP ou IMAP, em detrimento do protocolo SMTP [30]. O envio de mensagens é feito através do protocolo SMTP. Para o desenvolvimento da funcionalidade de envio de e-mail, foram efectuados quatro passos:

- Leitura da configuração presente na base de dados.
- Autenticação no servidor através do protocolo SMTP.
- Construção e envio da mensagem electrónica.
- Registo da mensagem na base de dados.

Para a recepção de mensagens de correio electrónico foi usado o protocolo IMAP, escolhido em detrimento do protocolo POP3 pelo facto de o IMAP4 incluir novas funcionalidades [30]. A metodologia típica *download and delete*, inerente ao protocolo POP3, não é adequado ao modelo actual de utilização de serviços de e-mail - as mensagens devem ficar sempre armazenadas no servidor, a fim de que todos os dispositivos do utilizador tenham acesso à mesma informação em simultâneo [31].

## 5.3 Implementações incompletas

Durante a fase de pesquisa e desenvolvimento das tecnologias a suportar no protótipo, foram abordadas quatro implementações de suporte a serviços: VoIP, *face.com* API, unificação de contactos e *Twitter* API.

### 5.3.1 Implementação VoIP

Numa primeira análise do protótipo, tomou-se a iniciativa de desenvolver uma implementação própria do protocolo SIP para possível comercialização da aplicação. Com base na análise efectuada na secção 2.2 sobre o protocolo, abordou-se inicialmente a implementação ao desenvolver o protocolo com a divisão em quatro componentes, visto que o pacote *Header*

foi necessita de suportar cinquenta tipos de campos de cabeçalho diferentes. O pacote *Message* inclui os algoritmos de construção da mensagem genérica e para efectuar a autenticação MD5, que é necessária para a autenticação do cliente no servidor. O pacote *Parser* serve para decompor as mensagens recebidas pelo cliente e processar as mensagens de solicitação ou resposta. O pacote *Transport* será o pacote responsável pelos algoritmos de envio das mensagens através do protocolo TCP ou UDP.

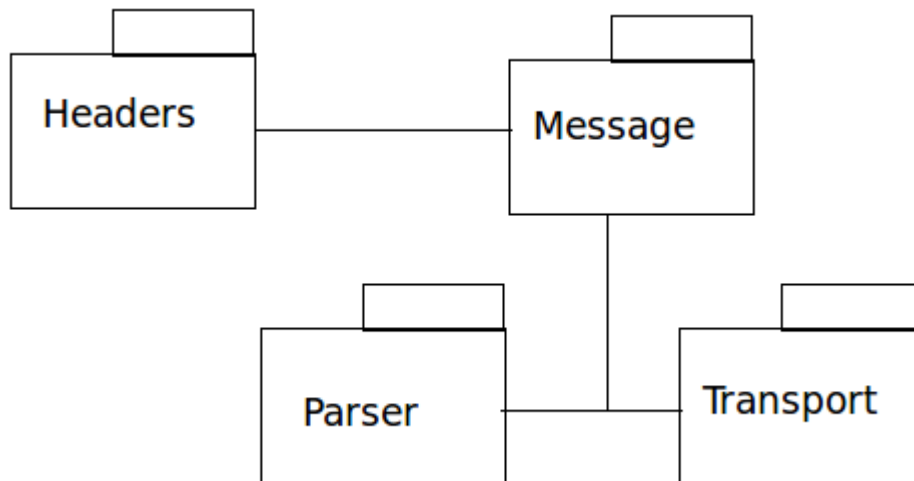


Figura 5.15: Estruturas das Pacotes

Com algum trabalho desenvolvido até então, chegou-se à conclusão que o processo de implementação de uma interface SIP seria demasiado demorado. Seria ainda necessário desenvolver suporte para diversos tipos de mensagens que fazem parte do protocolo, além de toda a componente multimédia necessária para a realização de chamadas telefónicas e para o envio de mensagens de texto. Visto que o trabalho que se pretende desenvolver não é unicamente a realização de uma chamada através de IP, foi decidido parar o desenvolvimento efectuado até então e explorar outras alternativas.

### 5.3.2 face.com API

Ao efectuar o sistema de unificação de contactos foram consideradas várias optimizações a realizar, do ponto de vista funcional, para o sistema. Uma ideia seria utilizar o reconhecimento facial a partir das fotografias dos contactos do utilizador, detectando as várias contas e unir todas as contas da mesma pessoa. Com adopção da tecnologia de reconhecimento facial da empresa *face.com*, obteve-se algum sucesso neste objectivo.

API é complexa, sendo capaz de detectar as expressões faciais e retornar informações sobre a imagem em questão, como por exemplo: olhos, nariz, boca, se possui óculos ou se está a sorrir. Todas estas informações podem ser exploradas mas a interface de programação apenas serviu para o objectivo de detecção e reconhecimento facial dos contactos.

Uma das desvantagem da utilização desta API é a necessidade de estabelecer uma ligação de dados constante com o servidor *face.com*, não podendo o reconhecimento facial ser processado localmente. A implementação pede a verificação da imagem ao servidor externo, recebendo as estimativas encontradas relativamente às imagens que foram previamente processados e,

caso esta seja superior a 90%, o sistema aceita a imagem como semelhante, retornando qual a imagem no servidor que é semelhante à processada. Como existem diversas contas associadas ao utilizador, o tratamento das imagens e a sua actualização no servidor é feito apenas após toda a unificação de dados. Esta é realizada através de uma *thread* assíncrona, para não afectar a experiência do utilizador.

Após implementação desta API, a empresa *face.com* foi comprada pelo Facebook, deixando de disponibilizar os seus serviços ao público em geral. Como tal, este código deixou de fazer parte da versão final do protótipo da aplicação.

### 5.3.3 Unificação de Contactos

A implementação da unificação de contactos está dependente de serviços de terceiros que nem sempre deixam obter toda a informação dos utilizadores via API: o correio electrónico, número telefónico ou número de identificação pessoal. A abordagem inicial para obter os dados para estudo engloba a análise de campos como nome, morada, correio electrónico, número telefónico e data de nascimento. O primeiro aspecto a considerar foi definir qual o valor que o sistema aceita como sendo de sucesso, sendo atribuído o valor *1,00* a sucesso. Após analisar os vários parâmetros obtidos, foram escolhidos como os principais o correio electrónico e o número telefónico. O nome e data de nascimento são utilizados para o cálculo do valor final mas são considerados como secundários.

Ao analisar os dados para a unificação, não se descartam dados que não sejam iguais. Estes dados são marcados com um valor inferior ao fixado inicialmente, na tentativa de não descartar dados que possam ser importantes e que podem levar a uma unificação incorrecta de alguns contactos.

Nome do Campo	Análise detalhada
Nome da pessoa	Análise individual a cada nome
Número telefónico	Não realizada
e-mail	Analisado apenas o nome de utilizador.
Morada	Análise da rua e cidade.
Data de Nascimento	Não realizada.

Tabela 5.1: Análise na unificação de dados

A implementação descrita ficou inserida no protótipo, mas não se encontra a funcionar do modo pretendido porque cada serviço não fornece os dados pessoais necessários. Como consequência disto, existe alguma duplicação de contactos no sistema. Com a inclusão da API *face.com* o sistema de unificação ultrapassava este problema completamente.

## Capítulo 6

# Usabilidade do protótipo

Neste capítulo são efectuadas tarefas de estudo de usabilidade para testar e poder retirar conclusões sobre a implementação do protótipo desenvolvido. Foram analisados os dados considerados mais relevantes para o estudo.

### 6.1 Tarefas realizadas

Para envolver o utilizador no estudo do protótipo, foram delineadas algumas tarefas que oferecem contacto com as várias interfaces da aplicação. Todos os utilizadores executaram as tarefas num telemóvel Android idêntico e, na preparação do ambiente de teste, foram inseridos manualmente contactos e dados relativos aos serviços utilizados na aplicação, de forma a acelerar todo o processo de estudo. Alguns dados foram propositadamente mal configurados, de forma a serem corrigidos pelo utilizador durante o estudo.

O método empírico foi escolhido para o teste, pois segue a abordagem de análise do comportamento do utilizador no protótipo, efectuando tarefas, questionários e experiências em contacto com a aplicação. No teste de usabilidade foram analisadas as tarefas consoante três parâmetros: tempo de execução, número de erros na execução e dificuldade associada à tarefa. Ao iniciar o protótipo o utilizador depara-se a interface inicial da aplicação (figura 5.6).

- **Tarefa 1:** Entrar na Interface de serviços e navegar entre o menu de serviços.

Pretende-se oferecer uma tarefa relativamente fácil e verificar como o utilizador interage com barra de tarefas. Deste modo, é dado ao utilizador tempo para se familiarizar com a aplicação. Pretende-se também avaliar se os utilizadores familiarizados com dispositivos de ecrã táctil efectuam a escolha de serviços através de uma acção de "deslize" do ecrã ou se apenas usam a barra de tarefas.

- **Tarefa 2:** Actualizar o estado do Facebook.
- **Tarefa 3:** Realizar o envio SMS.

Estas tarefas passam pela utilização de serviços no protótipo, sendo que nesta etapa é verificado quais os erros que podem cometer e se estes acedem aos serviços através da interface da tarefa anterior ou encontram um modo alternativo de a realizar.

- **Tarefa 4:** Editar as configurações do servidor SIP, alterando o IP para o endereço "192.168.0.122" no perfil XMPP do utilizador e adicionar ".com" no campo *Host*. Após a actualização dos valores, deve efectuar uma chamada através do serviço VoIP.

A quarta tarefa aumentou a dificuldade de execução ao requerer interacção com outro ecrã do protótipo. Pretende-se testar se utilizador consegue realizar uma re-configuração de dados e efectuar uma chamada VoIP.

- **Tarefa 5:** Verificar o perfil de um contacto e os últimos serviços realizados entre ambos pela aplicação.

Esta tarefa pretende que o utilizador explore a sua área de contactos e verifique a existência de outra alternativa na realização dos serviços do protótipo.

- **Tarefa 6:** Ler as mensagens de correio electrónico e efectuar a resposta a uma mensagem.

Por último, pretende-se que o utilizador aceda às suas mensagens de correio electrónico e efectue uma resposta. Com isto, pretende-se saber se o utilizador consegue navegar sem dificuldade entre as interfaces do protótipo da aplicação, nomeadamente voltar ao ecrã inicial da selecção de serviços.

## 6.2 Resultados iniciais

### 6.2.1 Análise global

Devido ao protótipo ser direccionado a qualquer utilizador que contenha um dispositivo móvel, não foi colocada nenhuma restrição à amostra de utilizadores escolhida. O teste de usabilidade foi efectuado por dez pessoas, três mulheres e sete homens, que disseram que têm ou já tiveram um dispositivo móvel com sistema operativo *Android* ou *iOS*. Anteriormente, também já usaram serviços VoIP ou XMPP.

Ao iniciar o teste de usabilidade, foi explicado o tipo de protótipo que se desenvolveu sem ser divulgada qualquer informação relativa às interfaces do protótipo. Durante a realização das tarefas propostas, esperava-se que estas demorassem entre 10 e 50 segundos, dependendo do grau de dificuldade de cada tarefa. O gráfico 6.1 mostra que, na maioria das tarefas, essa meta foi atingida: apenas 32% do total das tarefas foram superiores a 60 segundos. Nas três primeiras tarefas estes dados são consistentes em todos os utilizadores mas verificamos que nas três últimas alguns demoraram bastante mais do que 60 segundos, contrariando os pressupostos que uma boa experiência de utilização deve seguir.

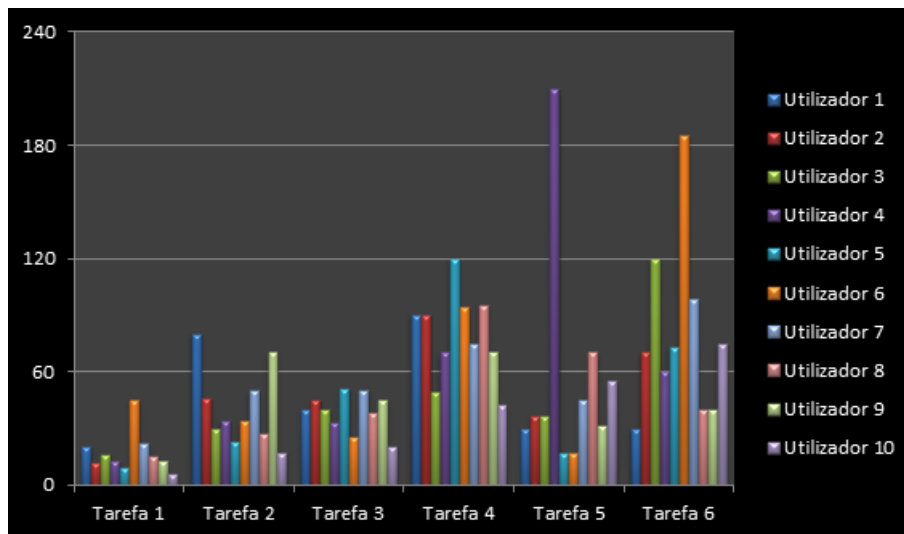


Figura 6.1: Gráfico de duração das tarefas

Ao analisar os dados relativos aos erros efectuados em cada tarefa, verificou-se que os utilizadores cometem mais erros pelo facto de não conseguirem distinguir o local correcto para a acção desejada. Alguns utilizadores limitaram-se a recorrer a um método de "tentativa e erro" para encontrar a acção correcta, o que os prejudicou bastante na execução da tarefa. Pelo contrário, os utilizadores que tentaram explorar os ecrãs do protótipo conseguiram resultados mais satisfatórios, reduzindo em cerca de 50% os erros cometidos quando comparado com os utilizadores que não realizaram a tarefa desta forma.

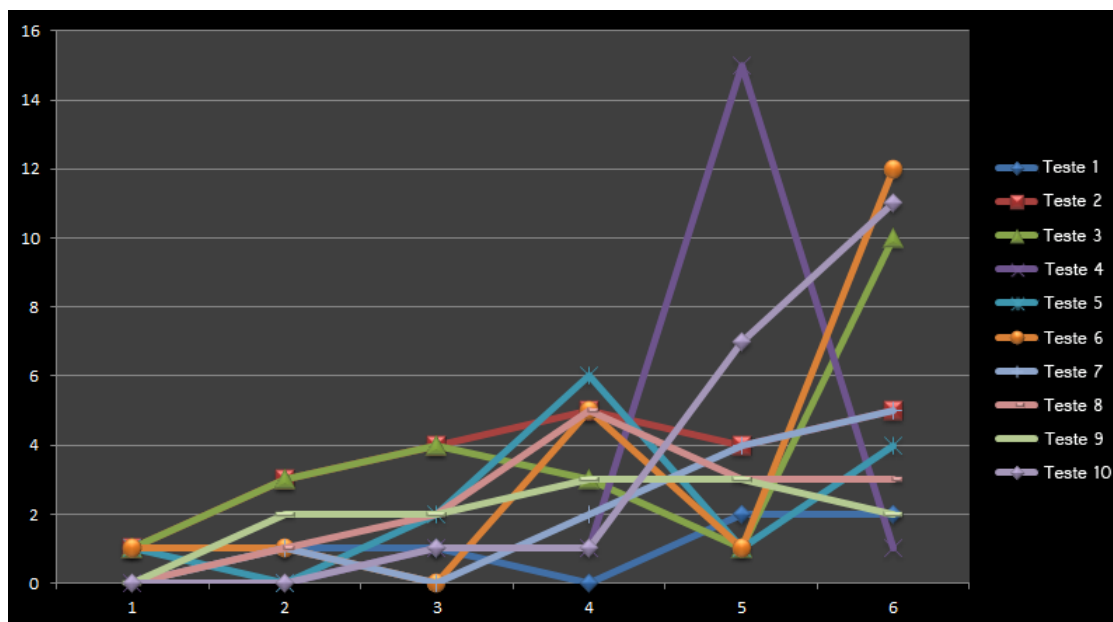


Figura 6.2: Gráfico de erros por tarefa

Entre tarefas foi pedido ao utilizador uma avaliação quanto ao grau de dificuldade da tarefa e de satisfação global do protótipo, sendo estas avaliações efectuadas numa escala de um a

cinco, com o valor *um* correspondendo a "muito fácil" e *cinco* a "muito difícil". O resultado obtido foi satisfatório, como demonstrado no gráfico de satisfação (figura 6.3), sendo obtido apenas 9% de avaliações "difícil", 32% para "médio", 46% "fácil" e 13% "muito fácil". Os utilizadores classificaram a satisfação global em 50% como uma "boa" e 50% como "razoável".

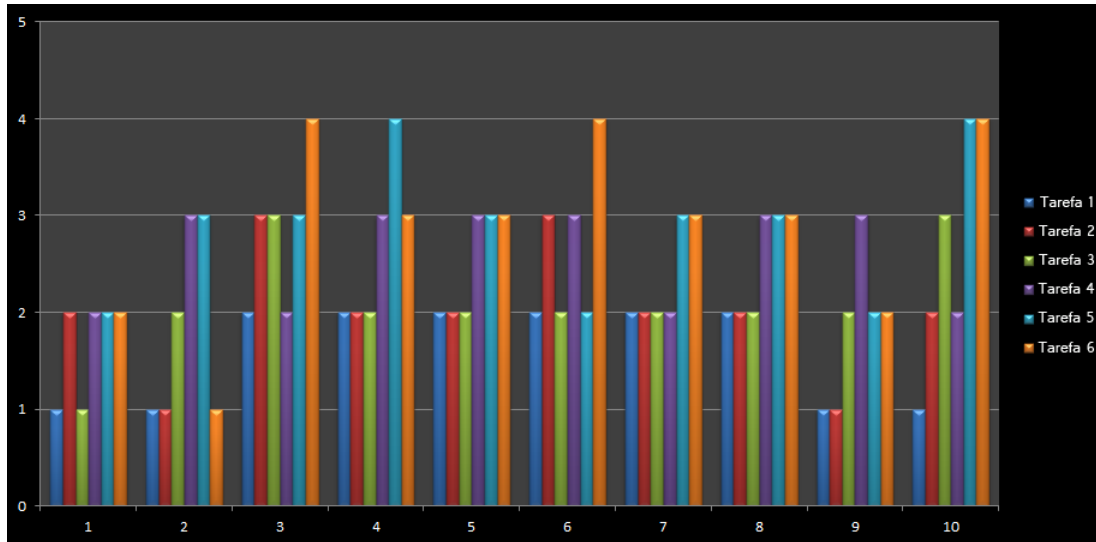


Figura 6.3: Gráfico de satisfação

## 6.2.2 Análise detalhada de tarefas

Ao finalizar os dez testes, verificou-se que algumas tarefas não foram compreendidas da melhor forma. Neste sentido, será detalhado quais as percepções recolhidas na realização das tarefas e de que forma era esperado que o utilizador as realizá-se.

- **Tarefa 1:** O utilizador acede à interface de serviços sem cometer nenhum erro mas 90% dos utilizadores utilizaram a barra de tarefas de serviços para navegar entre eles. Seria de esperar que uma percentagem maior dos utilizadores utilizassem a acção de "deslize" do ecrã para navegação entre painéis de serviços.
- **Tarefa 2:** Os utilizadores tiveram a tendência de ignorar a caixa de texto onde deveriam inserir a actualização do seu estado. Em vez de efectuarem esse passo, alguns deles carregavam no botão do *Facebook* e eram obrigados a voltar a efectuar toda a interacção de forma correcta. Conclui-se que os utilizadores estão mais habituados a realizar uma activação de um botão antes de poder inserir texto.
- **Tarefa 3:** Nesta tarefa o utilizador já compreendeu o mecanismo da tarefa anterior e cumpre esta terceira tarefa de forma normal, sem que realize um grande número de erros. É de salientar que apenas um sujeito utilizou a interface da vista de contactos para enviar a mensagem SMS. Um dos principais problemas que os sujeitos tiveram com o teste foi o ícone escolhido para a funcionalidade de *Correio Electrónico* que, por ser igual ao ícone da barra de tarefas, induziu alguns em erro.
- **Tarefa 4:** Pela quarta tarefa ser mais extensa que as anteriores, os utilizadores despenderam mais tempo nesta do que nas anteriores. A análise recolhida levou à conclusão de



que os utilizadores não reparavam nos botões de "salvar" presentes em cada ecrã de configuração. Além deste aspecto, detectou-se que os utilizadores que estão familiarizados com o sistema operativo *iOS* não estão habituados ao botão "voltar" definido nas teclas físicas dos dispositivos *Android*. Na execução da chamada VoIP também surgiu alguma dificuldade inerente aos ícones escolhidos para a aplicação, já que a cor vermelha do ícone de chamada telefónica sugeriu ao utilizador que aquela acção seria para desligar uma chamada.

- **Tarefa 5:** Nesta tarefa, 90% dos utilizadores acedeu de forma directa ao perfil de um dos contactos. Devido ao ícone escolhido para a funcionalidade de histórico, os sujeitos obtiveram algumas dúvidas sobre qual a forma correcta de aceder à mesma.
- **Tarefa 6:** Esta foi a tarefa que mais erros de utilização provocou aos sujeitos: devido ao nome escolhido para a vista da barra de tarefas, a associação do ícone de correio electrónico à correspondente funcionalidade não foi imediata.

### 6.2.3 Observações Iniciais

O primeiro aspecto modificado no protótipo foram os ícones, pois existiam alguns utilizadores que não os associavam correctamente à funcionalidade correspondente. Alterações adicionais passam pela redefinição visual da aplicação ou pela adição de uma legenda no próprio ícone.

O segundo aspecto a melhorar é a colocação de texto informativo, acerca do formato de dados, nas caixas de texto da interface de serviços. Deste modo, é oferecido ao utilizador mais informação sobre cada componente, de forma a otimizar a curva de aprendizagem do utilizador.

O atalho *Feed* presente na barra de tarefas não é associado ao tipo de funcionalidades que implementa, motivo pelo qual se modificou para *Pessoal*. Deste modo, pretende-se que o utilizador possa mais facilmente depreender que o ecrã representa a sua área de consulta de informação dos serviços implementados.

O facto de vários utilizadores accionarem os botões de serviço - que executam a acção de envio do texto - antes de efectuarem a inserção da mensagem não foi contemplado como um problema durante a fase de conceptualização do protótipo. O modo de funcionamento requer apenas a inserção de texto seguido de uma activação do botão do serviço para o qual deve ser enviada a mensagem. Esta implementação da interface deveria de contemplar esta alteração da ordem de operações numa fase posterior.

## 6.3 Resultados após melhoramento

### 6.3.1 Análise global

Após efectuar as modificações referidas na secção "Observações Iniciais" 6.2.3, foram realizadas novamente as seis tarefas mantendo-se os mesmo parâmetros da primeira experiência, observando desse modo se estas alterações vieram melhorar ou não o desempenho do utilizador. A experiência foi efectuada com oito sujeitos, dos quais quatro não estão em contacto constante com o dispositivo de teste. Dois sujeitos utilizam um dispositivo deste género, diariamente, e dois já realizaram a primeira experiência. Esta selecção deveu-se ao facto de

se quiser analisar o protótipo com utilizadores mais exigentes e verificar também novamente a opinião dos dois utilizadores que já tinham participado na experiência anterior. Por observação dos resultados da primeira experiência, verificou-se que o tempo de realização de uma tarefa se encontra relacionado com o número de erros efectuados pelo utilizador. Deste modo, para a segunda experiência, não se consideraram os resultados relativos aos erros cometidos pelos utilizador.

O gráfico de duração da execução das tarefas 6.4 demonstra que os valores obtidos na segunda experiência foram semelhantes aos obtidos na primeira. O tempo médio requerido para execução das tarefas passou de 40 segundos para 26 segundos e a percentagem de utilizadores que realizou as tarefas em menos de 60 segundos foi mais baixa: inicialmente era de 68% mas nesta experiência atingiu os 95%.

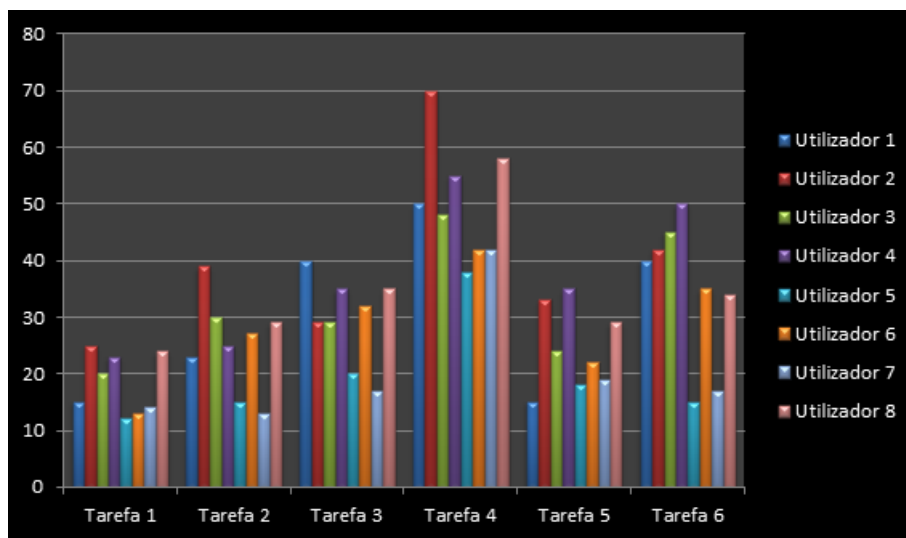


Figura 6.4: Gráfico da duração de execução das tarefas - segunda experiência

### 6.3.2 Observações finais

Pela realização da última experiência, notaram-se melhoramentos na utilização da nova implementação sendo observado que os utilizadores já não reportaram alguns dos problemas de usabilidade iniciais. Ao analisar as duas experiências, foi constatado que a implementação de um ecrã de ajuda na aplicação pode ser útil aos utilizadores experientes e ainda mais importante para os que não estão familiarizados com este tipo de aplicações.

Devido às dimensões reduzidas do ecrã de alguns dispositivos móveis, deveria-se ter removido o logotipo da aplicação da interface de serviços para otimizar a área disponível para a interacção do utilizador.

## Capítulo 7

# Conclusão

Com base nos objectivos propostos, construiu-se um protótipo funcional de aplicação para realizar serviços de telecomunicações: recolha e envio de dados de correio electrónico, envio de mensagens instantâneas, acesso a informação disponível em redes sociais, chamadas de voz e mensagens de texto.

O sistema de unificação de dados do utilizador foi implementado a partir da informação recolhida no dispositivo móvel e através das redes sociais configuradas no protótipo, ficando aquém de alguns dos objectivos iniciais desta funcionalidade 3.3.4. A reutilização de interfaces de programação incorporadas ou compatíveis com *Android* prejudicou a implementação de serviços fornecidos por terceiros, nomeadamente *Facebook* e *LinkedIn*. Os obstáculos presentes estão relacionados maioritariamente com lacunas no fornecimento de acesso aos dados do serviço, que provocaram uma diminuição da informação disponível para a automatização da criação de uma lista de contactos unificada. Contudo, foi possível consolidar a informação dos contactos existentes nos vários serviços numa lista de contactos local ao dispositivo, ainda que não tenha sido realizada de forma ideal.

Devido à quantidade de serviços suportados pela aplicação, muitos dos quais criados por entidades diferentes, a configuração da aplicação torna-se mais complexa do que o que seria desejado para os utilizadores. Esta área poderá ser melhorada posteriormente, de modo a oferecer maior facilidade na configuração dos serviços, para benefício dos utilizadores com menos capacidades técnicas. Contudo, este ponto requer constante melhoramentos na aplicação, conforme a evolução dos mesmos serviços, ou terá sempre uma maior complexidade de configuração do que as aplicações dedicadas.

Através da informação obtida nas experiências de usabilidade realizadas 6, a resposta dos utilizadores foi positiva quanto às melhorias no tempo de acesso aos serviços, revelando mais valias no conceito de unificação sobre uma interface comum. Conclui-se também que, ao não requerer uma aplicação por serviço, obtêm-se os seguintes benefícios: redução no número de acções a realizar na totalidade em comunicações, redução na curva de aprendizagem para a utilização de vários serviços e melhor desempenho do dispositivo móvel.

## 7.1 Trabalho futuro

Melhoramentos ao protótipo podem ser divididos em três fases: simplificação da configuração de serviços, criação de algoritmos de unificação das listas de contactos e melhoria da interface da aplicação.

Para obter a simplificação da configuração inicial da aplicação, será importante complementar os serviços de terceiros com um serviço próprio de VoIP e XMPP que permitisse obter a mesma funcionalidade sem requerer muito trabalho de configuração por parte do utilizador: meramente o registo nos serviços.

A criação de um algoritmo de unificação de contactos, através de dados obtidos dos vários serviços, pode necessitar de tecnologia como o reconhecimento facial dos contactos em fotografias de perfil. Como implementado através da interface de programação *face.com* 5.3.2, este tipo de tecnologia é extremamente eficaz para realizar com sucesso este tipo de operações, sendo necessário uma exploração mais aprofundada de como poderia ser implementada de raiz para ser possível a integração nesta aplicação.

Noutra fase, seria necessário melhorar a interface actual de modo a estar preparada para ser possível interagir de forma um pouco mais intuitiva, tendo como base os resultados obtidos durante as duas experiências de usabilidade realizadas 6. Seria também uma mais valia aproximar os elementos de interacção da aplicação, de modo a homogeneizar os padrões visuais da aplicação para que sejam eficazes numa futura implementação multiplataforma móvel.

# Bibliografia

- [1] Tom Keating. Internet phone release 4. *Computer Telephony Interaction Magazine*, page 28, 2007.
- [2] Gao Lisha and Luo Junzhou. Performance analysis of a p2p-based voip software. *AICT/ICIW*, 0:6, 2006.
- [3] Dongyan Zhang, Hongli Zhang, Chao Zheng, and Hongliang Yu. Identification and analysis of skype peer-to-peer traffic. *IEEE*, 0:7, 2010.
- [4] Dennis Bergstrom. An analysis of skype voip application for use in a corporate environment. page 30, 2004.
- [5] Emanuel P. Freire, Artur Ziviani, and Ronaldo M. Salles. Detecting skype flows in web traffic. *IEEE*, 0:8, 2007.
- [6] Chun-Ming Leung and Yuen-Yan Chan. Network forensic on encrypted peer-to-peer voip traffics and the detection, blocking, and prioritization of skype traffics. *Wetice*, 0:6, 2007.
- [7] Salman A. Baset and Henning G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. *IEEE*, 0:11, 204.
- [8] Brian Trammell and Dominik Schatzmann. A tale of two outages: a study of the skype network in distress. *IEEE*, 0:5, 2011.
- [9] Mohammad Goudarzi, Lingfen Sun, and Emmanuel Ifeachor. Modelling speech quality for nb and wb silk codec for voip applications. *IEEE*, 0:6, 2010.
- [10] Daniel Schlosser, Michael Jarschel, Valentin Burger, and Rastin Pries. Monitoring the user perceived quality of silk-based voice calls. *IEEE*, 0:6, 2010.
- [11] Jan Silovsky, Petr Cerva, and Jindrich Zdansky. Assessment of speaker recognition on lossy codecs used for transmission of speech. *IEEE*, 0:4, 2011.
- [12] Twilio. Twilio docs.
- [13] J. W. Nah, Y. H. Cho, S. W. Kim, and J. T. Park. Architecture for extensible mobile instant messaging and presence service over ims. *IEEE*, page 6, 2008.
- [14] Shengyao Chen and Peter Weik. Design and implementation of an extensible online charging architecture for the open ims playground. *IEEE*, page 5, 2007.

- [15] Alberto Diez Albaladejo, Fabricio Carvalho de Gouveia, Marius Iulian Corici, and Thomas Magedanz. The pcc rule in the 3gpp ims policy and charging control architecture. *IEEE*, page 5, 2008.
- [16] A. De Rogatis, R. Garufi, A.L. Robustelli, P. Adesso, and M. Longo. An automatic-payment parking service integrated within the 3g-ims architecture. *IEEE*, page 6, 2009.
- [17] Unified communication strategies.
- [18] Nortel. Nortel solution brief mobile unified communications. *Nortel*, page 8, 2008.
- [19] Tim O'Reilly. What is web 2.0: Design patterns and business models for the next generation of software. *Communications & Strategies*, page 21, Settembre 2007.
- [20] Darcy DiNucci. Fragmentedfuture. *Print*, page 3, 1999.
- [21] Toshihiko Yamakami. Mobileweb 2.0: Lessons fromweb 2.0 and past mobile internet development. *IEEE*, page 5, 2007.
- [22] He Shaohua and Wang Peilin. Web 2.0 and social learning in a digital economy. *IEEE*, page 4, 2008.
- [23] Viber. Viber support.
- [24] Akshi Kumar and Ruchi Goel. Event driven test case selection for regression testing web applications. *IEEE*, 0:7, 2012.
- [25] John Oehler. The rise of samsung, the dominance of android, Maio 2012.
- [26] Christy Pettey and Rob van der Meulen. Worldwide sales of mobile phones declined 3 percent in third quarter of 2012; smartphone sales increased 47 percent, Novembre 2012.
- [27] Dave Mark and Jeff LaMarche. *Beginning iPhone 3 Development: Exploring the iPhone SDK*. Apress, 2009.
- [28] A. Platek, W. Zabierowski, and A. Napieralski. Building winfbase application using windows forms 2.0 and c#.net. *IEEE*, 0:5, 2008.
- [29] Windows. Windows forms.
- [30] Lawrence E. Hughes. *Internet e-mail Protocols, Standards and Implementation*. Artech House Publishers, 1998.
- [31] Muhammad Anwari Leksono and Rinaldi Munir. Email client application with rabbit algorithm for android smart phone. *IEEE*, 0:6, 2012.