2013

Instituto Politécnico de Coimbra

INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA

Desenvolvimento de Aplicação de Rich Communications para Android

MESTRADO EM INFORMÁTICA E SISTEMAS

AUTOR | Rúben Estrela

ORIENTADORES | Prof. Doutor João Cunha | ISEC
| Eng.º Tiago Leitão | WIT Software

Coimbra, dezembro 2013

Instituto Politécnico de Coimbra

Instituto Superior de Engenharia de Coimbra

Departamento de Engenharia Informática e de Sistemas

Mestrado em Informática e Sistemas
Estágio/Projeto Industrial
Relatório Final

Desenvolvimento de Aplicação de Rich Communications para Android

Ruben Leandro Cruz Estrela

Orientadores:

Prof. Doutor João Cunha, ISEC

Eng.º Tiago Leitão, WIT-Software

Coimbra, dezembro, 2013

"Learn from yesterday, live for today, hope for tomorrow.

The important thing is not to stop questioning."

Albert Einstein

Aos meus pais, Adail e Dália, a quem devo tudo pela ajuda dada ao longo deste percurso que é a vida, à minha namorada Adalgisa, pelo amor e paciência demonstrada durante esta jornada.

A vocês, dedico

AGRADECIMENTOS

Ao Professor Doutor João Cunha, pela orientação, acompanhamento e conselhos transmitidos ao longo de todo o estágio.

Ao Engenheiro Tiago Leitão, pela disponibilidade demonstrada para acompanhar o meu progresso neste projeto.

Aos meus colegas da equipa de desenvolvimento, Alex Santos, Miguel Nunes e Manuel Nunes, pelo excelente ambiente criado durante todo o projeto.

À WIT-Software, pela oportunidade de trabalhar neste projeto e pelas excelentes condições criadas para que este estágio fosse bem-sucedido.

Aos meus pais e irmão, pelo amor, apoio e incentivo na realização desta etapa da minha vida.

À minha namorada Adalgisa, que me tem acompanhado nos últimos 4 anos, mesmo nos momentos mais complicados.

A todos os meus amigos, pela amizade e força transmitida durante todo este percurso.

A todos o meu muito obrigado.

RESUMO

Com a emergência de *smartphones* no mercado de telecomunicações, é cada vez maior o número de novos meios para comunicação. Com a disponibilização de melhores condições de acesso à Internet através dos *smartphones*, o número de aplicações capazes de fornecer novos serviços de comunicação, para além dos tradicionais serviços de voz e SMS, torna os consumidores cada vez mais exigentes. Esta nova realidade está a colocar em causa a supremacia das operadoras de telecomunicações, no que diz respeito ao fornecimento de serviços.

Por forma a iniciar o processo de integração do RCS (*Rich Communication Suite*) numa operadora de telecomunicações de referência a nível europeu, a WIT-Software deu início a um projeto colaborativo para o desenvolvimento de novas soluções RCS. Esta parceria tem como objetivo o desenvolvimento de uma solução RCS para a plataforma Android, capaz de ser integrado na rede IMS (*Internet Protocol Multimedia Subsystem*) administrada pela operadora. A solução pretende integrar um conjunto de serviços descritos na especificação RCS, assim como garantir a interoperabilidade destes em diferentes redes IMS, com suporte para este tipo de serviços.

O presente documento descreve o trabalho realizado no âmbito do estágio integrado no Mestrado Informática e Sistemas, Ramo Desenvolvimento de Software, lecionado no Instituto Superior de Engenharia de Coimbra. O estágio decorreu na empresa WIT-Software, tendo a duração de 9 meses. Como estagiário, fiz parte de uma equipa Scrum responsável pelo desenvolvimento da solução RCS para Android. Entre as atividades onde estive diretamente envolvido, constam a programação de módulos em Java, a criação e execução de testes unitários, o planeamento de tarefas, a interpretação e discussão de requisitos.

PALAVRAS-CHAVE

Android,
GSMA,
IMS,
Interoperabilidade,
joyn,
Mobile
RCS,
RCS-e,
Telco,
Telecomunicações,

GLOSSÁRIO

Termo	Descrição
2G	2nd Generation of Global System for Mobile Communications (GSM)
3G	3rd Generation of Global System for Mobile Communications (GSM)
3GPP	3rd Generation Partnership Project
4G	4th Generation of Global System for Mobile Communications (GSM)
AIDL	Android Interface Definition Language
API	Application Programming Interface
AS	Application Server
BA	Broadband Access
bps	Bits por segundo (usado com Mbps: Mega-, kbps: kilo-)
CPIM	Common Profile for Instant Messaging
CPM	Converged IP Messaging
CS	Circuit Switched
CSCF	Call Session Control Function
DoD	Definition of Done
EAB	Enhanced Address Book
FT	File Transfer
GC	Group Chat
GSM	Global System for Mobile Communications
GSMA	GSM Association
HDTV	High Definition Television

HSS	Home Subscriber Server
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol Secure
IARI	IMS Application Reference Identifier
ID	IDentifier
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMDN	Instant Message Disposition Notification
IMEI	International Mobile Station Equipment Identity
IMS	Internet Protocol Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPC	Inter-process communication
ISP	Internet Service Provider
JVM	Java Virtual Machine
LTE	Long Term Evolution
MMS	Multimedia Message Service
MMTel	MultiMedia TELephony
MNO	Mobile Network Operator
MPEG	Moving Pictures Experts Group
MSISDN	Mobile Subscriber Integrated Services Digital Network Number
MSRP	Message Session Relay Protocol
NNI	Network to Network Interface
OEM	Original Equipment Manufacturer
OMA	Open Mobile Alliance
OS	Operating System
OTT	Application/Service Over-the-Top
PC	Personal Computer
PO	Product Owner
PS	Packet Switched
PSTN	Public Switched Telephone Network
RCS	Rich Communication Suite
RCS-e	RCS enhanced
RFC	Request For Comments
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
SIM	Subscriber Identity Module
SIMPLE	Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions

SIP	Session Initiation Protocol
SM	Scrum Master
SMS	Short Message Service
SP	Service Provider
SPAM	Mensagens eletrónicas não desejadas
SQA	Software Quality Assurance
TCP	Transmission Control Protocol
UE	User Equipment
UI	User Interface
UX	User Experience
VCEG	Video Coding Experts Group
VM	Virtual Machine
VoIP	Voice over IP
WWW	World-Wide Web
XMS	SMS/MMS service
xSIM	Referência genérica para diferentes tipos de SIM

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

1. INTRODUÇÃO

Neste capítulo será apresentado o enquadramento, motivação e objetivos do estágio, assim como a apresentação da empresa responsável pelo mesmo. Será incluída também uma breve contextualização da iniciativa RCS (*Rich Communication Suite*).

1.1. Apresentação do Estágio

O estágio foi integrado num processo colaborativo entre a WIT-Software e uma operadora de telecomunicações de referência a nível europeu, para o desenvolvimento de uma nova solução RCS para a plataforma Android.

O RCS consiste numa iniciativa da GSMA (*Global System for Mobile Communications Association*) com o objetivo de facilitar a introdução de novos serviços de comunicação nos mercados de comunicações móveis. Esta conta com um forte apoio da indústria de telecomunicações, havendo já um vasto número de empresas a apostar cada vez mais na introdução de soluções RCS no mercado, nomeadamente através da disponibilização de clientes RCS.

1.2. Apresentação da Organização

A WIT-Software S.A., é uma empresa especializada no desenvolvimento de *software* na área das telecomunicações. Fundada em 2001, a empresa está estabelecida nos principais pontos do país, em Coimbra (sede), Lisboa, Porto e Leiria, e tem filiais em Reading (Reino Unido) e San José (Califórnia, EUA). A empresa conta neste momento com clientes em 15 países diferentes incluindo alguns dos maiores operadores como Vodafone, Telefónica, Orange, Deutsche Telekom e TeliaSonera, aos quais fornece serviços nas suas 3 unidades de negócio:

- **Telco**: Unidade de negócios responsável pelo fornecimento de soluções para comunicações através de dispositivos móveis a operadoras de todo o mundo. Esta unidade de negócio oferece soluções convergentes baseadas em IMS para VoIP (fixo/móvel), mensagens (SMS, MMS e IM), RCS e serviços de telefonia multimédia (MMTel). A gama de soluções inclui neste momento aplicações cliente RCS (para iPhone, Android, iPad, *tablets*, PC e navegadores da Web) e um RCS

Application Server para fornecimento de funcionalidades RCS em redes IMS e pré-IMS.

- **Mobile**: Unidade de negócios centrada no desenvolvimento de aplicações para múltiplas plataformas móveis (Android, iOS, Windows Phone, Blackberry, J2ME, Symbian), para área da banca, grupos de *media*, entre outros.

- **TV**: Unidade de Negócios responsável pelo desenvolvimento de *software* para operadores de cabo e IPTV, através de uma plataforma de *widgets* TV, comunicações convergentes, soluções TV *Everywhere*, entre outras.

No início de 2012, a WIT-Software ganhou um concurso internacional para o desenvolvimento de clientes RCS de referência para Android e iOS. A WIT-Software tem vindo a desenvolver soluções RCS desde a primeira versão da norma, estando totalmente focada no seu desenvolvimento.

1.3. Motivação

A *suite* RCS tem vindo a evoluir desde a sua criação em 2007, acompanhando as tendências do mercado e projetando novas abordagens. O apoio desta iniciativa é cada vez maior, sendo os principais intervenientes as operadoras de telecomunicações, que tem vindo sofrer perdas consideráveis nas receitas provenientes dos serviços de voz e mensagens. Para colmatar este decréscimo de receitas a GSMA e cinco operadoras europeias (Vodafone, Telefónica, Orange, Telecom Italia e Deutsche Telekom) desenvolveram a especificação RCS-e (*RCS enhanced*), baseada na especificação RCS, por forma a facilitar a sua rápida integração. Com esta nova especificação foi criada a marca joyn™, iniciando a propagação de soluções RCS entre as operadoras móveis a nível mundial.

Sendo a WIT-Software a fornecedora oficial de aplicações cliente para as plataformas Android e iOS, esta desenvolveu uma gama de clientes para ambas as plataformas, as quais já acreditadas pela GSMA e atualmente disponíveis em mercados internacionais, com total interoperabilidade entre as maiores operadoras de telecomunicações a nível Europeu.

Com a introdução de soluções RCS-e em Espanha, a WIT-Software obteve a oportunidade de se afirmar no desenvolvimento deste tipo de soluções.

Através da cooperação com uma operadora de telecomunicações de referência a nível europeu, a WIT-Software pretende desenvolver uma nova aplicação cliente RCS-e, com total interoperabilidade nas redes RCS existentes.

1.4. Objetivos

O estágio teve como principal objetivo a implementação de uma aplicação de comunicações Android de acordo com as especificações RCS (RCS 5 e RCS-e), para que esta fosse integrada com sucesso numa rede IMS controlada por uma operadora de telecomunicações.

A aplicação pretendida deverá ser totalmente compatível com as especificações RCS, contendo como principais módulos:

- ***Enhanced Address Book* (EAB)** – Módulo responsável pela sincronização de contactos existentes nos dispositivos (lista nativa), com informação relativa aos serviços RCS;
- ***Enhanced Voice Call*** – Módulo que confere aos utilizadores a possibilidade de partilha imagens e vídeos em tempo real no decorrer de uma chamada de voz, proporcionando assim uma extensão ao tradicional serviço de chamadas;
- ***Enhanced Messaging*** – Módulo baseado em IM (*Instant Messaging*) para troca de mensagens instantâneas com um ou mais contactos em simultâneo, com possibilidade de partilhar de arquivos (imagens, vídeos, etc.);
- ***Geolocation Share*** – Módulo de partilha de localizações geográficas;
- ***VoIP Calls*** – Módulo de chamadas de voz e vídeo baseados em IP.

No decorrer deste estágio, a entidade responsável pelo projeto teria também como principal objetivo obter acreditação da solução perante a GSMA, de forma a formalizar a sua compatibilidade com a especificação RCS e consequente entrada na lista de operadoras acreditadas.

1.5. Estrutura do documento

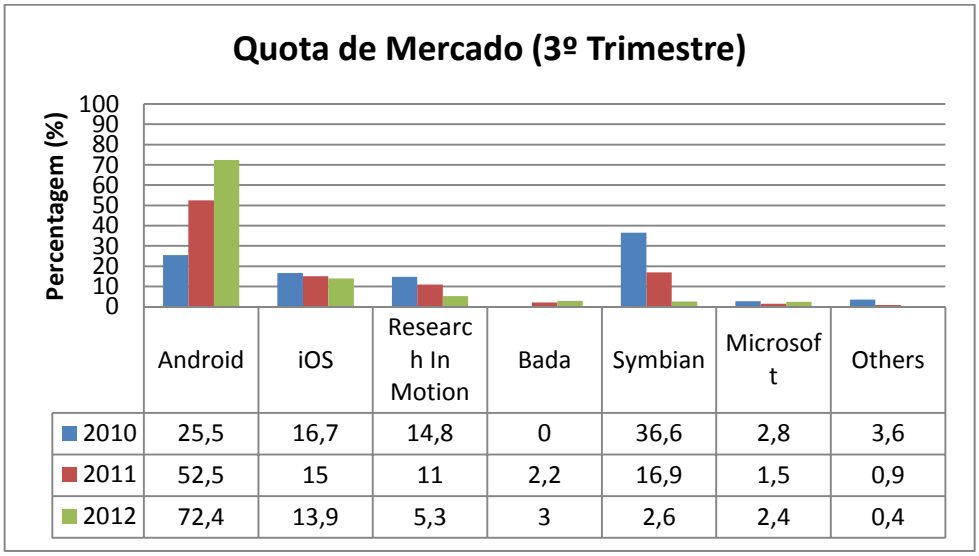
Para além do capítulo de introdução, este relatório está organizado da seguinte forma:

- **Capítulo 2 – Comunicações Móveis - Evolução e Tendências de Futuro**: O objetivo deste capítulo é apresentar e descrever uma análise do mercado de comunicações móveis e a evolução da norma RCS.
- **Capítulo 3 – Requisitos**: Neste capítulo é feita a descrição dos requisitos do projeto. É apresentada ainda uma breve descrição da evolução dos requisitos e seu enquadramento na evolução do RCS.
- **Capítulo 4 – Arquitetura e Tecnologias**: Neste capítulo são apresentadas as principais tecnologias utilizadas na arquitetura da solução desenvolvida, assim como a descrição de todo o trabalho realizado no decorrer do estágio.
- **Capítulo 5 – Metodologia de Trabalho**: Neste capítulo é apresentado o plano de desenvolvimento do estágio, incluindo a metodologia de trabalho, o processo de desenvolvimento e os papéis desempenhados.
- **Capítulo 6 – Conclusões**: Este capítulo apresenta as conclusões finais relativamente ao estágio e a análise os objetivos alcançados.

2. COMUNICAÇÕES MÓVEIS - EVOLUÇÃO E TENDÊNCIAS DE FUTURO

De acordo com a *Gartner Inc* [32], no terceiro trimestre de 2012, o total de dispositivos móveis vendidos a nível mundial terá alcançado os 428 milhões de unidades, com o mercado dos *smartphones* a continuar a sua expansão alcançando 169.2 milhões de unidades vendidas.

Na Europa a penetração de *smartphones* é cada vez maior. Segundo a *comScore* [34], existe agora uma penetração de 53.7%, com aproximadamente 130 milhões de pessoas a usar *smartphones*, em que 15.5% também possuem um dipositivo *tablet*, mais 9.3% se comparado com o ano de 2011. Este crescimento ajuda a demostrar a adoção de outros dispositivos móveis para além dos *smartphones*.



	Android	iOS	Research In Motion	Bada	Symbian	Microsoft	Others
2010	25,5	16,7	14,8	0	36,6	2,8	3,6
2011	52,5	15	11	2,2	16,9	1,5	0,9
2012	72,4	13,9	5,3	3	2,6	2,4	0,4

Figura 1 - Vendas globais de dispositivos móveis por sistema operativo no 3º Trimestre de 2010 a 2012 (fonte: [32])

Como é possível verificar na Figura 1, o sistema operativo Android terá alcançado em 2012 72.4% do mercado de dispositivos móveis, com um aumento de 19.9% se comparado com igual período de 2011 [32]. Em Setembro de 2012, a Google anunciou os 500 milhões de dispositivos Android ativos a nível mundial, estimando 1.3 milhões de novos dispositivos a cada dia, existindo cerca de 675 mil aplicações disponíveis na sua loja *online* (Google Play) e 25 mil milhões de instalações (*downloads*) até à data [33].

2.1. Comunicações Móveis

O aumento da disponibilidade de dados na banda larga móvel está a causar uma mudança na natureza das comunicações, mudanças que estão a tornar-se cada vez mais visíveis.

Estudos sugerem que no Reino Unido [35], assim como em outros países europeus, o modo como as pessoas comunicam tem vindo a sofrer constantes alterações, sendo um dos exemplos disso mesmo, o decréscimo do volume de chamadas de voz de telefonia nos últimos cinco anos. Contrariando esta tendência estão as chamadas VoIP (*Voice over IP*), tendo sido estimado o aumento do uso deste tipo de chamadas, de 16% em 2010 para 22% em 2012, o que poderá ser um forte indicador na visualização de mudança na utilização de serviços de voz.

Nos últimos anos tem-se verificado o aumento de outras formas de comunicação "não-voz", sendo o *messaging* talvez uma das mais relevantes.

No Reino Unido, de 2010 a 2012 verificou-se um aumento de 5% no que diz respeito ao uso de SMS; desde 2006 que o número de SMS enviados por pessoa por mês, terá vindo a aumentar em média 20% por ano. Este crescimento poderá ser explicado pela aposta das operadoras móveis, em disponibilizar cada vez mais tarifários que incluam o envio de mensagens de texto de forma ilimitada.

Assim como os SMS, outros tipos de comunicações baseados em texto tem vindo a sofrer um aumento ao longo dos últimos anos. De acordo com o *Ofcom Technology Tracker* [35], o número de adultos a usar o *email* ou redes sociais como meio de comunicação, tem vindo aumentar de forma significativa nos últimos dois anos.

Com o aumento de *smartphones*, a frequência de utilização do *email* e das redes sociais é ainda maior, pois torna a sua utilização mais rápida e acessível. Esta mobilidade e rapidez de comunicação através dos *smartphones* tem levado aos serviços de IM, a terem cada vez mais sucesso em termos de utilização, seja através de serviços nativos (BlackBerry Messenger, iMessage) disponibilizados pelos fabricantes, ou através de serviços OTT (*over-the-top*) multiplataforma (Skype, WhatsApp, Viber), sendo estimado que em 2012 cerca de 50% dos utilizadores de *smartphones* sejam também utilizadores de

um ou mais serviços de IM [35]. Um exemplo desta nova realidade é a corrosão que está acontecer numa das maiores fontes de receitas a nível mundial para as operadoras móveis, o SMS, onde é estimado que a sua participação nas receitas desça dos 63,5% obtidos em 2011 a menos de 50% nos próximos anos [36].

Entre 2008 e 2009 começaram a surgir os serviços designados '*over-the-top*' (OTT), sendo este termo usado uma das primeiras vezes para descrever serviços de vídeo emergentes tais como Netflix ou Hulu, que fornecem serviços para *streaming* de vídeo através de uma ligação à Internet sem qualquer dependência ao fornecedor de acesso à mesma. Nos Estados Unidos o seu lançamento desafiou empresas tais como a ComCast e AT&T, que estariam a preparar a disponibilização de serviços de vídeo *on-demand* [37].

Os serviços OTT podem ser definidos como serviços disponibilizados através da Internet diretamente ao utilizador final, independentemente do fornecedor de acesso à Internet (ISP) que possui e controla a rede.

De acordo com um relatório de mercado da *Analysys Mason* de 2012 [40], a penetração dos serviços OTT no mercado Europeu é cada vez mais forte. Segundo este relatório, em Espanha 63% dos utilizadores de *smartphones* usam os OTT como alternativa de *messaging*, sendo que 87% desses utilizadores utilizam a aplicação "WhatsApp".

No que diz respeito a chamadas voz, o relatório descreve que em Espanha quase 12% dos utilizadores de *smartphones* usam serviços OTT pala realizar chamadas VoIP, sendo que 83% desses utilizadores utilizam o serviço "Skype". Em países como o Reino Unido, França e Alemanha a utilização do "Skype" situa-se ainda acima dos 90%.

No que diz respeito às comunicações móveis, o sucesso das OTT deve-se à sua usabilidade. As aplicações OTT para *smartphones* são de fácil instalação e bastante simples em termos de utilização. Estas utilizam informações existentes nos dispositivos (ex. MSISDN, *Address-Book*), como base para comunicação através dos seus serviços. Outros fatores de sucesso são a sua utilização ser "grátis" (utilizando comunicações *IP-to-IP*) e a disponibilidade multiplataforma, dando ainda mais mobilidade de comunicação aos seus utilizadores.

Existe atualmente uma vasta gama de aplicações OTTs para *smartphones*, mas tendo em conta fatores relativos à sua maturidade ou o número de utilizadores ativos, é possível destacar algumas das mais relevantes no mercado atual:

- **Facebook**: Rede social lançada em 2004, que conta com mil milhões de utilizadores ativos. Atualmente a comunicação com base nesta rede social está disponível para praticamente todas as plataformas móveis, tendo funcionalidades tais como *Chat*, *Group Chat*, partilha de conteúdos e ainda partilha de localização.

- **Skype**: Lançado em 2003, e adquirido pela Microsoft em 2011, o "Skype" foi desde início um dos serviços mais revolucionários no que diz respeito a serviços VoIP, e estima-se que existam atualmente 600 milhões de utilizadores registados. Disponível nas principais plataformas móveis, o "Skype" conta com funcionalidades tais como *Chat*, *Group Chat*, partilha de conteúdos, chamadas de voz e vídeo, entre outras.

- **WhatsApp**: Aplicação multiplataforma de IM para *smartphones*. Além de mensagens de texto, os utilizadores podem enviar conteúdos tais como imagens, vídeos ou áudio. Estima-se que existam 250 milhões de utilizadores a enviar aproximadamente 10 mil milhões de mensagens por dia.

- **Viber**: Aplicação multiplataforma para envio de mensagens instantâneas e realizar chamadas VoIP através de *smartphones*. Desenvolvido pela Viber Media, a aplicação foi inicialmente lançada a 2 de Dezembro de 2010, para o iPhone da Apple e atualmente está disponível também para outras plataformas móveis, tais como, Android, Windows Phone e BlackBerry. Atualmente estima-se que tenha atingido os 175 milhões de utilizadores.

No mercado de aplicações Android, existe uma grande proliferação de aplicações OTT com funcionalidades para comunicação. No início deste estágio foi possível comparar algumas destas aplicações, tendo em conta a sua relevância no mercado Android.

Esta comparação foi feita com base num conjunto de características, a seguir descritas:

- ***Voice Calls***: Capacidade de realizar chamadas VoIP com um ou mais contactos;
- ***Video Calls***: Possibilidade de realizar chamadas de vídeo;
- ***Chat***: Dispõe serviço de IM para um destinatário (*single chat*) ou para múltiplos destinatários (*Group Chat*);
- ***Chat Messages***: Tipos de mensagens de *chat* disponíveis ao utilizador, para além das mensagens de texto (ex. localização, imagens, vídeo, áudio, etc.);
- ***Phonebook Sync***: Capacidade sincronização contactos existentes na lista de contactos do telefone;
- ***Network Address Book***: Capacidade de sincronização a uma lista de contactos, salvaguardada num servidor externo, acessível através de qualquer dispositivo;
- ***Blacklist***: O utilizador poderá através da aplicação gerir os contactos que pretende bloquear tentativas de comunicação dos mesmos;
- ***Social Sync***: A aplicação sincroniza dados da aplicação tais como credenciais de autenticação ou lista de contactos, relativa a uma ou mais redes socias;
- ***User Rating***: Avaliação (1 a 5) dos utilizadores, publicada na plataforma de aplicações Android (Google Play);
- ***Installations***: Número de instalações feitas a partir da plataforma Google Play;
- ***Platform***: Plataformas suportadas pela aplicação, onde é possível aos utilizadores acederem com a mesma informação de conta.

Na Tabela 1 e Tabela 2 são apresentadas algumas das aplicações com maior foco na utilização de IM para conversação, chamadas VoIP e vídeo.

As informações relativas às aplicações em análise nas tabelas de comparação poderão sofrer atualizações, devendo ser consideradas apenas como referência.

Tabela 1 - Matriz comparativa de aplicações OTTs: Características

Name	*Installations*	*User Rating*	*Voice Call*		*Video Call*		*Chat*		*Chat Messages*				*Phonebook Sync*	*Network Address Book*	*BlackList*	*Social Sycn*
			1-1	1-N	1-1	1-N	1-1	1-N	*File Transfer*	*Share Location*	*Voice Message*	*Drawing*				
Facebook	100.000.000 - 500.000.000	4,4	X				X		X	X	X			X	X	X
Skype	100.000.000 - 500.000.000	4,0	X		X		X	X	X					X		X
WhatsApp	100.000.000 - 500.000.000	4,6					X	X	X	X	X		X		X	
KakaoTalk	50.000.000 - 100.000.000	4,5	X	X			X	X	X		X		X			
LINE	50.000.000 - 100.000.000	4,1	X				X	X	X	X	X	X		X		X
Tango	50.000.000 - 100.000.000	4,3	X		X		X		X		X		X	X		X
Viber	50.000.000 - 100.000.000	4,4	X				X	X	X	X			X			
ChatON	10.000.000 - 50.000.000	4,1					X	X	X	X	X		X	X		
Kik	10.000.000 - 50.000.000	4,4					X	X	X			X	X	X		
ooVoo	10.000.000 - 50.000.000	4,2	X	X	X	X	X	X					X	X		X
WeChat	10.000.000 - 50.000.000	4,4	X		X		X	X	X	X	X		X	X		X
Voxer	10,000,000 - 50,000,000	4,4					X	X	X	X	X		X	X		X
Fring	5.000.000 - 10.000.000	3,8	X	X	X	X	X						X			
textPlus	5.000.000 - 10.000.000	4,2	X				X	X	X		X		X	X	X	X
Cubie	1.000.000 - 5.000.000	4,4					X	X	X	X	X	X	X	X		X
FreePP	1.000.000 - 5.000.000	4,3	X				X	X	X		X	X	X			X
Hike	1.000.000 - 5.000.000	4,1					X	X	X	X	X		X	X	X	X
Yuilop	1.000.000 - 5.000.000	4,3	X				X	X	X	X	X		X			X
Text Me!	1.000.000 - 5.000.000	4,2	X		X		X	X	X	X	X		X	X		X
Aire	100,000 - 500,000	4,0	X		X		X	X	X	X	X		X	X	X	X
Vippie	50,000 - 100,000	4,1	X	X	X	X	X	X	X		X		X	X		

Tabela 2 - Matriz comparativa de aplicações OTTs: Plataformas

Name	*Platform*									
	Mobile						*Desktop & Web*			
	Android	*iOS*	*Windows Phone*	*BlackBerry*	*Bada*	*Symbian*	*Web*	*Windows*	*Linux*	*Mac*
Facebook	X	X		X			X			
Skype	X	X	X					X	X	X
WhatsApp	X	X	X	X	X					
KakaoTalk	X	X	X		X					
LINE	X	X	X	X				X		X
Tango	X	X	X					X		
Viber	X	X	X	X	X	X				
ChatON	X	X	X	X	X		X	X		X
Kik	X	X	X	X		X				
ooVoo	X	X					X	X		X
WeChat	X	X	X	X		X	X			
Voxer	X	X								
Fring	X	X				X				
textPlus	X	X	X			X				
Cubie	X	X								
FreePP	X	X	X			X				
Hike	X	X	X	X		X				
Yuilop	X	X								
Text Me!	X	X	X							
Aire	X	X						X		X
Vippie	X	X	X	X		X				

2.2. Indústria de Telecomunicações Móveis

A indústria de comunicações móveis é um dos maiores e mais importantes sectores de negócio com 6 mil milhões de utilizadores [41].

Este sector consegue gerar 740 mil milhões de euros anualmente, através de serviços móveis, onde 80% provêm dos serviços tradicionais como chamadas de voz e SMS.

Os serviços móveis continuam a crescer. Nos últimos cinco anos o número de utilizadores aumentou em média 17% por ano, estimando-se que 86% da população mundial use telemóveis. Este crescimento deve-se principalmente a mercados emergentes, que representam 70% do mercado mundial. Enquanto nos continentes Africano e Asiático as telecomunicações móveis contam com cada vez mais subscrições de serviços, na Europa e América do Norte o crescimento é cada vez menor.

No que diz respeito a receitas geradas pelos serviços disponibilizados pelas operadoras, é possível afirmar que existe atualmente uma quebra de mercado, a que as operadoras estão bastante atentas. Na Europa estima-se que as receitas geradas por serviços tradicionais como chamadas de voz e SMS venham a sofrer um forte declínio até 2017, devido à penetração de serviços OTT [40]. De acordo com um relatório da *Ovum* [42], o crescimento de serviços alternativos como OTTs de *messaging*, irá provocar um declínio de receitas das operadoras de 42 mil milhões de euros até 2016.

Hoje em dia as redes móveis são tipicamente descritas como uma combinação de redes de 2ª geração (2G) para os tradicionais serviços de chamadas de voz e mensagens de texto (SMS), e serviços de 3ª geração (3G) para serviços de acesso à Internet com maior largura de banda (com velocidades de acesso a rondar os 43Mbps). A 4ª geração (4G), também conhecido por "evolução a longo-termo" (LTE) começa agora a ser uma realidade em alguns países, onde são fornecidos acessos máximos de 150Mbps [41].

Com o aumento da cobertura 3G/4G, é cada vez mais importante que as operadoras disponibilizem serviços inovadores que enriqueçam ainda mais as opções dos seus clientes, mas ao mesmo tempo mitigar os problemas criados pelos serviços OTT. A disponibilização de serviços RCS é a resposta das operadoras à cada vez maior adesão por

parte dos seus clientes a aplicações OTT. O desenvolvimento destes serviços baseia-se em três abordagens: "RCS/joyn", "Telco OTT" e parcerias:

- **RCS/joyn**: O 'RCS/joyn' é a resposta oficial da GSMA á penetração de mercado das OTTs, incluindo características tais como, *Chat*, *Group Chat*, partilha de conteúdos, etc., através da integração com a lista de contactos dos dispositivos. A disponibilização deste tipo de soluções para *smartphones*, será feita através de uma distribuição "nativa" (pré-instalados nos dispositivos vendidos pelas operadoras) ou através das plataformas *online* como 'Google Play' ou 'Apple Store'.

- **Telco OTT**: Outra abordagem são as 'Telco OTT', que são basicamente soluções segundo um modelo OTT não abrangidas pelos padrões 'RCS/joyn', nomeadamente na convergência de funcionalidades. Estas podem ser disponibilizadas através do desenvolvimento interno (normalmente com a ajuda de um fornecedor especializado), ou através de soluções de "marca branca" (ex. "Libon" da Orange, "TU Me/TU Go" da Telefónica, entre outros).

- **Parcerias**: A terceira abordagem atualmente adotada pelas operadoras é a parceria com as empresas como "WhatsApp", que fornecem serviços de *messaging* alternativos ao SMS. Esta abordagem requer que as operadoras revejam a sua estratégia de negócio, nomeadamente nos custos de utilização dos serviços das parcerias.

As maiores operadoras mundiais como Vodafone, Telefónica, Orange ou Deutsche Telekom, estão cada vez mais ativas na adoção de soluções "RCS/joyn", como principal meio de combate contra a "migração" de clientes para serviços OTT. Esta luta conjunta faz-se denotar com o aumento de mercado capaz de suportar o *standard* RCS, sendo exemplo disso a Espanha e a Alemanha os primeiros países a lançar serviços RCS [43][44], com interoperabilidade entre ambos os mercados. O principal foco das operadoras é garantir uma evolução "país-a-país" de forma estável, para evitar problemas de interoperabilidade. Apesar destes esforços por parte das operadoras, há quem acredite que esta tecnologia tem vindo a demorar muito tempo para chegar ao mercado, o que

pode comprometer o envolvimento por parte dos clientes e perda de importantes canais de comunicação operador-cliente.

O mercado é cada vez mais feroz, e todos os dias surgem notícias de novos *players* ou anúncios realizados por *players* bem enraizados no dia-a-dia de todos nós, como por exemplo o Facebook [45] que, para além do serviço de *messaging* atualmente disponível por todo o mundo, começou em 2013 a disponibilização de chamadas VoIP aos utilizadores do Reino Unido (à semelhança do que já acontecia nos Estados Unidos e Canadá). Este tipo de notícias obriga as operadoras a serem ainda mais rápidas a apostar na evolução e implementação de soluções 'RCS/joyn', para que esta seja mais incisiva na mitigação dos problemas criados pelas OTTs.

2.3. Rentabilização das Infraestruturas de Comunicação

Ao longo dos anos as operadoras de telecomunicação móveis desenvolveram infraestruturas para suporte a serviços VoIP. Com a necessidade de disponibilizar novos serviços, as operadoras móveis iniciaram a integração de redes IMS, aumentando a complexidade das suas infraestruturas de comunicação. O acesso à infraestrutura de comunicação é assegurado pelo protocolo SIP, sendo este o protocolo amplamente usado pela indústria de telecomunicações (*telco*).

A iniciativa RCS tem como objetivo principal a disponibilização de novos serviços para comunicação, capazes de rentabilizar toda infraestrutura de comunicação. Para que isto seja possível, é necessário a implementação de clientes RCS capazes de disponibilizar esses mesmos serviços nos terminais dos utilizadores.

Apesar de existir uma especificação RCS para convergência de implementações de clientes RCS, esta encontra-se limitada aos requisitos impostos pela arquitetura das infraestruturas de comunicação existentes. A utilização do protocolo SIP para utilização de serviços RCS é visto atualmente como um possível problema, devido ao peso do protocolo no processamento da comunicação entre os terminais RCS e a rede. Apesar de existiram possíveis soluções, nomeadamente para terminais Android, este problema ainda não poderá ser devidamente mitigado devido às características de toda a infraestrutura envolvida.

3. REQUISITOS

Neste capítulo será apresentado um resumo dos requisitos do projeto, enquadrados com a norma RCS.

É possível encontrar informações mais detalhadas relativamente aos requisitos aqui descritos na documentação oficial RCS e OMA, contendo uma definição mais aprofundada dos requisitos funcionais e os requisitos técnicos necessários para a implementação dos módulos incluídos neste capítulo.

3.1. Enquadramento

A especificação RCS foi iniciada através da colaboração de um grupo de entidades da área das telecomunicações, tendo como principal objetivo a adoção de aplicações e serviços interoperáveis e convergentes, por forma a enriquecer as comunicações móveis. A iniciativa RCS inclui operadores de rede, fornecedores rede e fornecedores de dispositivos.

O desenvolvimento da iniciativa RCS baseia-se num processo iterativo, que pretende desenvolver conjuntos consistentes de diretrizes de implementação, tais como, casos de uso, demonstrações e experimentações, tendo como foco a interoperabilidade, normas e especificações existentes.

O trabalho desenvolvido em torno da iniciativa RCS encontra-se dividido numa sequência de versões, publicadas em *Releases*:

- ***Release 1*:** Encontra-se descrito todo o núcleo de serviços RCS, abrangendo o enriquecimento da lista de contactos (*Address-Book*), o enriquecimento da troca de mensagens e o enriquecimento das chamadas [11].

- ***Release 2*:** São apresentados melhoramentos dos serviços da *Release* anterior, permitindo o acesso aos mesmos através de banda larga (*broadband*), aumentado a gama de dispositivos para comunicação [12].

- ***Release 3*:** São consolidados serviços descritos na *Release 2* e adicionados melhoramentos, nomeadamente o IMS, possibilitando aos utilizadores o uso da banda larga (BA) para comunicação através de dispositivos não-móveis [13].

- ***Release 4*:** São consolidados serviços descritos na *Release 3*, adicionando melhorias no serviço de partilha de conteúdos, na sincronização da lista de contactos de rede (NAB), no serviço de presença e na extensão RCS para uso de números fixos. Nesta *Release* foi também incluindo o suporte de redes LTE (*Long Term Evolution*), possibilitando melhorias na qualidade dos serviços RCS, tais como voz, vídeo e partilha de conteúdos. Em termos de serviços de *messaging* através de IP, descritos nas versões anteriores, foram adicionadas funcionalidades para suporte a utilizadores *legacy*, possibilitando a inclusão de utilizadores não-RCS para sessões de *chat* [14].

- ***Release 5*:** Concentra-se na disponibilização de uma *framework* adaptável e interoperável, para disponibilização de serviços RCS com suporte a todas as versões anteriores. Elaborada sobre os fundamentos descritos nas *Releases* RCS anteriores, esta sucede todas elas como ponto de convergência. Esta especificação inclui todos os serviços descritos na especificação RCS-e, assim como a maioria das funcionalidades descritas nas versões RCS 1-4. Além de estender serviços já descritos em versões anteriores, a *Release 5* inclui também novos serviços tais como chamadas de voz sobre IP (*IP Voice Call*), chamadas de vídeo sobre IP (*IP Video Call*) e partilha de localização (*Location Sharing*). Foram também feitos alguns melhoramentos em serviços já existentes como, *Chat 1-to-1*, *Group Chat* e *File Transfer*. Na Figura 2 são esquematizados serviços disponíveis na *Release 5* da especificação RCS, estando a negrito os serviços adicionados nesta versão e a itálico os serviços que sofreram melhoramentos [17].

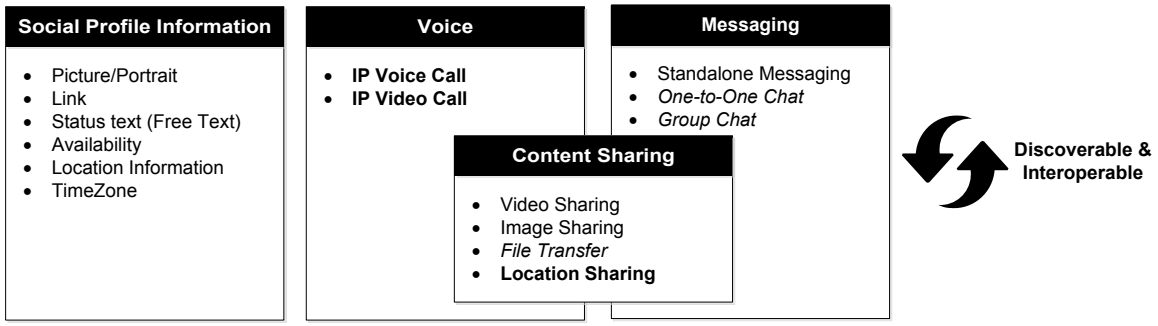


Figura 2 - Serviços RCS

(fonte: [17])

Subconjuntos RCS

Os requisitos incluídos na especificação RCS encontram-se em constante evolução, existindo outros subconjuntos de requisitos, incluídos em versões desenvolvidas sobre as especificações RCS:

- ***RCS-e*:** A versão RCS-e [16] foi criada com base nos serviços de comunicação descritos na *Release* da especificação RCS. Elaborado segundo princípios de interoperabilidade no ecossistema das operadoras móveis, esta especificação tem como objetivo acelerar a introdução de serviços RCS no mercado das operadoras de redes móveis (MNO). Esta versão tem com principal foco, o aumento do número de serviços disponibilizados para além dos serviços tradicionais de voz, SMS e MMS. Atualmente esta versão encontra-se disponibilizada em produtos com a marca joyn™.

- ***RCS Release 5.1***: A *Release 5.1* [18] é uma evolução da *Release 5.0*, incluindo diversos melhoramentos nas definições de serviços descritos na versão anterior.

- ***RCS-e joyn Hot Fixes***: Tendo como base a especificação RCS-e, o *RCS-e joyn Hot Fixes* [19] é um conjunto de melhoramentos na implementação de funcionalidades RCS em soluções joyn™.

- ***joyn Blackbird***: O *joyn Blackbird* [20] é uma evolução do RCS-e, tendo como base a especificação RCS 5.1, incluindo também requisitos funcionais descritos no *RCS-e joyn Hot Fixes*.

3.2. Requisitos Funcionais

3.2.1. Visão Global

A disponibilização dos serviços RCS nos terminais dos utilizadores baseia-se na integração de requisitos funcionais, para adaptação do terminal segundo os seguintes pontos de interação:

- **Lista de Contactos e Registo de Chamadas**: O utilizador deverá ser capaz de aceder aos serviços de IM/Chat e FT (*File Transfer*) através de informações de contactos existentes nos terminais, assim como identificar o suporte para serviços RCS.

- **Aplicação de Chat**: O utilizador deve ser capaz de aceder a um *chat* a partir da sua lista de aplicações, sendo que este poderá optar por iniciar *chat* de 1-para-1 ou uma conversação de grupo (*Group Chat*). Deverá ser possível ao utilizador visualizar o histórico de conversações assim como continuar conversações anteriormente iniciadas.

- **Explorador de Ficheiros, Galeria de Conteúdos e Câmara**: Para usufruir do serviço de FT, deverá ser dada ao utilizador a possibilidade de partilhar ficheiros e/ou conteúdos existentes nos seus terminais. Esta possibilidade de partilha deverá ser combinada com interação através de uma sessão de conversação ou durante uma chamada de voz.

- **Ecrã de Conversação**: Durante uma sessão de conversação, um utilizador poderá adicionar outros participantes à conversação, assim como partilhar conteúdos com os participantes. Esta partilha deverá ocorrer numa sessão separada da sessão de *chat* já estabelecida.

- **Ecrã de Chamada**: Para enriquecimento das chamadas de voz, o utilizador deverá conseguir optar pela partilha de conteúdos multimédia vídeo e imagem no decorrer de uma chamada. Toda a disposição destas opções de partilha deve complementar as funcionalidades de chamadas existentes nos terminais do utilizador.

3.2.2. Módulo 'Messaging'

Nesta secção serão descritos os serviços para conversação, sendo incluído uma breve apresentação das suas funcionalidades chave.

3.2.2.1. *Standalone Messaging*

O serviço 'Standalone Messaging' consiste num serviço de *messaging* que pretende incluir a troca de mensagens de texto e conteúdos multimédia baseando-se na plataforma IMS, substituindo os serviços de SMS e MMS. A utilização deste serviço pretende remover as limitações dos serviços xMS, tais como, número máximo de caracteres, tipo de conteúdo ou falhas nas notificações de entrega a utilizadores com múltiplos dispositivos. As mensagens deverão ser trocadas sob o *standard* OMA CPM (*Open Mobile Alliance - Converged IP Messaging*).

3.2.2.2. *One-to-One Chat*

O *chat* '1-para-1' é um serviço de conversação que permite a troca de IM entre dois utilizadores. A troca de mensagens deverá suportar os *standards* OMA SIMPLE IM e OMA CPM, com interoperabilidade entre os dois. Este serviço é um dos serviços base que deverá estar disponível para qualquer utilizador registado como RCS. São apresentadas em seguida as funcionalidades incluídas para este serviço:

Initiate a Chat Um utilizador RCS (A) inicia uma conversação selecionando um dos seus contactos (B) através da lista de contactos do terminal ou através do ecrã de contactos da aplicação RCS.

Answer a Chat Um utilizador B ao receber um convite para conversação poderá ter a possibilidade de responder ao mesmo convite e estabelecer uma conversação 1-para-1.

Exchange Messages in an Established Chat

Assim que exista uma sessão de IM estabelecida, a troca de mensagens entre os utilizadores deverá ser feita através do protocolo MSRP, contendo as notificações de estado IMDN (*Instant Message Disposition Notification*) para cada mensagem trocada. Estes estados deverão conter os estados '*delivered*' e '*displayed*'.

Display and Store Local History

As mensagens trocadas durante uma conversação deverão ser guardadas no terminal, contendo informações tais como data e estado da mensagem. Todas as mensagens de conversação com um contacto deverão ser apresentadas como um segmento, respeitando a ordem temporal das mensagens.

Leave Composing Chat Window

O utilizador deverá ter a possibilidade de deixar o ecrã de uma conversação sem terminar a mesma, continuando a receber novas mensagens. Deve ser ainda possível ao utilizador regressar à mesma conversação, através de qualquer um dos pontos de entrada disponibilizados (histórico de conversações, lista de contactos do terminal, entre outros).

Display "Is Composing" Notification

Quando qualquer um dos intervenientes da conversação inicia a digitação de uma mensagem, deverá ser enviado uma notificação '*Is Composing*' que o utilizador está a digitar uma mensagem. As notificações de '*Is Composing*' recebidas, deverão ser apresentados no ecrã de conversação, e removidas do ecrã assim que seja recebida uma nova mensagem ou a duração das mesmas ultrapasse um determinado tempo limite.

Close/Re-Open Chat O utilizador poderá fechar e/ou reabrir um ecrã de uma conversação, visualizando as mensagens trocadas anteriormente. A nível protocolar uma conversação será iniciada ou fechada consoante o seu estado (terminada devido a inatividade, interrupções abruptas, entre outros). O início ou a finalização de uma conversação deverá ser totalmente transparente para o utilizador, não devendo este ser notificado de qualquer transição.

Support Store & Forward Mode O cliente RCS deverá suportar a receção de mensagens enviadas durante um período de inatividade (*offline*). Assim que este estiver novamente *online*, as mensagens recebidas deverão ser salvaguardadas no terminal como parte da conversação, sem que seja necessário estabelecer uma conversação com o remetente das mensagens.

Switch to Group Chat O utilizador poderá adicionar mais participantes a uma conversação já estabelecida iniciando uma conversação em grupo.

File Transfer Within 1-to-1 Chat Durante uma conversação 1-para-1, deverá ser possível a qualquer um dos participantes iniciar transferências de ficheiros a partir do ecrã de conversação. A transferência deve ser feita através do protocolo MSRP, iniciado através de um convite de sessão SIP. Qualquer transferência só poderá ser iniciada após aceitação do destinatário.

Spam/Blacklist Filter Deverá ser possível ao utilizador classificar mensagens recebidas como SPAM. A classificação de novas mensagens deverá ser feita através da classificação de contactos como 'bloqueados', passando estes a pertencer à lista de contactos a que serão ignoradas notificações sobre novas mensagens, ou convites para conversação.

Support Emoticons Deve ser possível ao utilizador enviar/receber texto com representação gráfica (*Emoticon*). Os códigos suportados deverão respeitar a lista de *emoticons* incluída na especificação RCS.

Perform Chat Message Size Limitations Para reduzir a complexidade a nível protocolar e evitar potenciais transições TCP, deverá ser cumprida a recomendação de limitação do tamanho de mensagem enviado durante uma conversação 1-para-1. No caso de o utilizador tentar enviar uma mensagem acima do limite estabelecido, este deve ser informado que a mensagem não poderá ser enviada devido ao tamanho da mesma.

Support User Alias (Display Name) O utilizador poderá configurar o nome de exibição para uma conversação, que será enviado na sua iniciação. Qualquer nome de exibição (*Alias*) recebido deverá apenas ser usado quando o contacto de conversação não estiver presente na lista de contactos do terminal, respeitando o nome de exibição (*Display Name*) existente no mesmo.

3.2.2.3. *Group Chat*

O 'Group Chat' é um serviço de conversação de grupo, que permite a troca de mensagens instantâneas (IM) entre múltiplos utilizadores. A utilização deste serviço deve permitir a iniciação de uma conversação com quaisquer contactos registados como RCS:

Initiate a Chat O utilizador RCS (A) deverá conseguir iniciar uma conversação, selecionado um ou mais contactos (até que seja atingido o limite máximo de participantes, estabelecido pela plataforma RCS), a partir da lista de contactos presentes no terminal, registados como RCS e com capacidades de conversação (IM).

Leave Composing Chat Window O utilizador deverá ter a possibilidade de deixar o ecrã de uma conversação sem terminar a mesma, continuando a receber novas mensagens. Deve ser ainda possível o utilizador regressar à mesma conversação através do histórico de conversações.

Leave Chat Deve ser possível abandonar uma conversação explicitamente, parando de receber novas mensagens. O utilizador deverá conseguir identificar que uma conversação foi ou não abandonada através do histórico de conversações ou no ecrã de conversação.

Define Chat Subject Ao iniciar uma conversação de grupo o utilizador poderá definir um assunto (*Subject*) para a conversação. Este "assunto" deverá ser apresentado em todos os participantes da conversação.

Exchange Messages in an Established Chat

Assim que exista uma sessão de IM estabelecida a troca de mensagens entre os utilizadores deverá ser feita através do protocolo MSRP, contendo as notificações de estado IMDN para cada mensagem trocada. Estes estados deverão conter os estados '*delivered*' e '*displayed*'.

Display and Store Local History

As mensagens trocadas durante uma conversação deverão ser guardadas no terminal contendo informações tais como data e estado da mensagem. Todas as mensagens de conversação de grupo deverão ser apresentadas como um segmento respeitando a ordem temporal das mensagens.

Display "Is Composing" Notification

Quando qualquer um dos intervenientes da conversação inicia a digitação de uma mensagem, deverá ser enviada uma notificação '*Is Composing*' que o utilizador está a digitar uma mensagem. As notificações de '*Is Composing*' recebidas, deverão ser mostradas no ecrã de conversação, devendo estas ser removidas do ecrã assim que seja recebida uma nova mensagem ou a duração das mesmas ultrapasse um tempo limite estabelecido.

Close/Re-Open Chat

O utilizador poderá fechar e/ou reabrir um ecrã de uma conversação, visualizando as mensagens trocadas anteriormente. A nível protocolar uma conversação será iniciada ou fechada consoante o seu estado (terminada devido a inatividade, interrupções abruptas, entre outros). O início ou a finalização de uma conversação deverá ser totalmente transparente para o utilizador, não devendo este ser notificado de qualquer transição.

Spam/Blacklist Filter	Deverá ser possível iniciar uma conversação com um contacto anteriormente bloqueado, sendo sua a decisão de prosseguir com a conversação. As mensagens trocadas durante esta conversação não deverão ser classificadas como SPAM, devido à decisão do utilizador recetor.
Support Emoticons	Deve ser possível ao utilizador enviar/receber texto com representação gráfica (*Emoticon*). Os códigos suportados deverão respeitar a lista de *emoticons* incluída na especificação RCS.
Perform Chat Message Size Limitations	Para reduzir a complexidade a nível protocolar e evitar potenciais transições TCP, deverá ser cumprida a recomendação de limitação do tamanho de mensagem enviado durante uma conversação. No caso de o utilizador tentar enviar uma mensagem acima do limite estabelecido, este deve ser informado que a mensagem não poderá ser enviada devido ao tamanho da mesma.
Support User Alias (Display Name)	O utilizador poderá configurar o nome de exibição para uma conversação que será enviado na sua iniciação. Qualquer nome de exibição (*Alias*) recebido deverá apenas ser usado quando o contacto de conversação não estiver presente na lista de contactos do terminal, respeitando o nome de exibição (*Display Name*) existente no mesmo.
Display Chat Participants	O utilizador deverá conseguir visualizar a lista de participantes presentes na conversação, assim como o estado (*online* ou *offline*).

Add a Participant	A lista de participantes em conversação poderá sofrer atualizações através da adição de novos participantes. Deve ser possível a qualquer participante de conversação estabelecida adicionar novos participantes, até que seja atingindo o número máximo possível.
Diplay Conference Events	No decorrer de uma conversação deverão ser apresentados eventos relativos à atualização da mesma. Deverão existir notificações para novos participantes adicionados à conversação ou quando um participante abandona a mesma.
Support Store & Forward Mode	O cliente RCS deverá suportar a receção de mensagens enviadas durante um período de inatividade (*offline*). Assim após restabelecer a conectividade, as mensagens recebidas deverão ser salvaguardadas no terminal como parte da conversação, sem que seja necessário estabelecer uma conversação com o remetente das mensagens.

3.2.3. Módulo 'Content Sharing'

Nesta secção serão descritos os serviços para partilha de conteúdos, sendo incluído uma breve apresentação das suas funcionalidades chave.

3.2.3.1. *Video Sharing*

A funcionalidade de "Video Sharing" consiste na partilha em tempo real de vídeo, possibilitando ao utilizador RCS a partilha de um vídeo através do seu terminal:

- Usando uma camara frontal ("Eu")
- Usando camara traseira ("O que eu vejo")
- *Streaming* de um ficheiro de vídeo

Esta funcionalidade deverá estar disponível ao utilizador RCS, seguindo os seguintes requisitos:

Share During a Voice Call O utilizador RCS (A) deverá ter a possibilidade de partilhar um vídeo em tempo real com utilizador RCS (B) assim que ambos tenham uma chamada de voz estabelecida e ambos possuam suporte para partilha de vídeo. A partilha deverá ser feita através de uma ligação de dados (3G ou WiFi), usando o protocolo RTP (*Real-time Transport Protocol*) para *streaming* dos dados de vídeo, sendo possível a reprodução em tempo real no terminal de todos os intervenientes, de forma unidirecional ou bidirecional. A partilha poderá ser terminada pelo cancelamento por parte de um dos intervenientes ou com o fim da chamada de voz.

Share Without Voice Call Um utilizador poderá iniciar uma partilha de vídeo, à semelhança de uma chamada de voz, mas sem que haja qualquer chamada voz estabelecida. Ao contrário da partilha durante uma chamada, a sessão de partilha de vídeo fora de uma chamada de voz deverá ser apenas unidirecional.

Receive Invites Deverá ser possível ao utilizador receber convites para a partilha de vídeo. Os convites deverão ser apresentados ao utilizador sobre qualquer ecrã existente, possibilitando a aceitação ou rejeição do convite.

3.2.3.2. *Image Sharing*

A funcionalidade de "Image Sharing" consiste na partilha de imagens, possibilitando ao utilizador RCS a partilha de uma imagem através do seu terminal. Esta funcionalidade estará disponível ao utilizador RCS sempre que este esteja numa chamada de voz, seguindo os seguintes requisitos:

Share During a Voice Call O utilizador RCS (A) deverá ter a possibilidade de partilhar uma imagem com utilizador RCS (B) assim que ambos tenham uma chamada de voz estabelecida e ambos possuam suporte para partilha de imagens (*image share*). A partilha deverá ser feita através de uma ligação de dados (3G ou WiFi), usando o protocolo MSRP para transferência dados da imagem, sendo possível visualizar o progresso da transferência no terminal de todos os intervenientes. As partilhas serão feitas de forma unidirecional. A partilha poderá ser terminada pelo cancelamento por parte de um dos intervenientes ou com o fim da chamada de voz.

Receive Invites Deverá ser possível ao utilizador receber convites para a partilha de imagens. Os convites deverão ser apresentados ao utilizador sobre qualquer ecrã existente, possibilitando a aceitação ou rejeição do convite.

3.2.3.3. *Location Sharing*

A funcionalidade de "Location Sharing" consiste na partilha de uma geolocalização, possibilitando ao utilizador RCS partilhar a sua própria localização através da informação disponível do seu terminal. Esta funcionalidade estará disponível ao utilizador RCS sempre que existir suporte para a mesma, seguindo os requisitos abaixo descritos:

Share During a Voice Call O utilizador RCS (A) deverá ter a possibilidade de partilhar uma geolocalização com utilizador RCS (B) assim que ambos tenham uma chamada de voz estabelecida e ambos possuam suporte para partilha de geolocalização (*location share*). A partilha deverá ser feita através de uma ligação de dados (3G ou WiFi), usando o protocolo MSRP para transferência dos dados da localização. No final da transferência, deverá ser possível ao utilizador RCS (B) visualizar a localização enviada por (A) diretamente no mapa. A partilha poderá ser terminada pelo cancelamento por parte de um dos intervenientes ou com o fim da chamada de voz.

Share from 1-to-1 Chat O utilizador deverá conseguir enviar geolocalizações (posição atual ou outra localização) a partir de um ecrã de conversação. Após seleção da localização definida pelo utilizador RCS (A), a localização deverá ser enviada para o utilizador RCS (B). As informações sobre a transferência deverão ser guardadas no histórico local, sendo possível visualizar informação sobre a mesma no ecrã de conversação.

Share from Address-Book Deverá ser possível ao utilizador selecionar um contacto RCS da lista de contactos do terminal, como destinatário da partilha da geolocalização. Após seleção da localização definida pelo utilizador RCS (A), o utilizador deverá ser redirecionado para o ecrã de conversação com o utilizador RCS (B). As informações sobre a transferência deverão ser guardadas no histórico local, sendo possível visualizar informação sobre a mesma no ecrã de conversação.

3.2.3.4. *File Transfer*

O serviço "File Transfer" (FT) é um serviço que permite ao utilizador RCS a partilha de ficheiros entre um ou mais contactos RCS de forma instantânea. Este serviço depende também de requisitos externos como largura de banda ou espaço disponível no recetor da transferência, que deverão ser tidos em conta na sua disponibilização. São apresentadas em seguida as funcionalidades incluídas para este serviço:

Initiate Transfer from Address Book/Call-log Deverá ser possível ao utilizador selecionar um contacto RCS da lista de contactos do terminal, como destinatário de um ou mais ficheiros. Após seleção dos ficheiros por parte do utilizador, as informações sobre as transferências deverão ser guardadas no histórico local, sendo possível visualizar informação sobre a mesma no ecrã de conversação com o contacto RCS selecionado.

Initiate Transfer from Media Gallery/File Browser O utilizador poderá explorar os ficheiros locais através de aplicações nativas e selecionar um ou mais ficheiros para partilha dos mesmos com um ou mais contactos RCS. As informações sobre as transferências deverão ser guardadas no histórico local de conversações.

Initiate Transfer from Camera application	O utilizador poderá partilhar uma imagem ou vídeo obtido através da câmara do terminal, e iniciar a partilha dos mesmos com um ou mais contactos RCS. As informações sobre as transferências deverão ser guardadas no histórico local de conversações.
Initiate Transfer from IM/Chat Screen	O utilizador deverá conseguir iniciar a escolha de um ou mais ficheiros para partilha a partir de um ecrã de conversação. Após seleção por parte do utilizador, as informações sobre as transferências deverão ser guardadas no histórico local, sendo possível visualizar informações sobre a mesma no ecrã de conversação.
Receive File Transfers	Deverá ser possível ao utilizador RCS (A) receber transferência de ficheiros de um utilizador RCS (B) – não bloqueado – como parte de uma conversação. Ao receber a transferência, o utilizador RCS (A) deverá ser notificado sobre a mesma. Este deverá ter ainda a capacidade de aceitar/rejeitar a transferência ou configurar a autoaceitação de todas as transferências realizadas a partir de qualquer utilizador RCS (B).
Support File Thumbnails	No caso de transferência de imagens ou vídeos, deverá ser possível aos utilizadores recetores da transferência, receber uma pré-visualização do ficheiro original, por forma a facilitar a decisão de aceitar/rejeitar a transferência do ficheiro original.

Support Contact Card Push Deverá ser possível ao utilizador RCS (A) partilhar informações sobre os contactos existentes no seu terminal. As informações a partilhar deverão estar incluídas num ficheiro vCard (*Contact Card*). O ficheiro vCard deverá ser transferido à semelhança de ficheiros de outros formatos. As informações sobre a transferência deverão ser guardadas no histórico local, sendo possível visualizar informações sobre a mesma no ecrã de conversação.

Support Black-List/Spam Filter Deverá ser possível ao utilizador bloquear a transferência de ficheiros a contactos 'bloqueados'. Sendo as transferências bloqueadas no início do processo de transferência, estas devem ser ignoradas, não devendo estar presentes no histórico de conversação ou fazer parte das mensagens classificadas como SPAM.

3.2.4. Módulo 'Voice'

Nesta secção serão descritos os serviços VoIP, sendo incluído uma breve apresentação das suas funcionalidades chave.

3.2.4.1. *IP Voice Call*

O serviço "IP Voice Call" consiste na possibilidade de estabelecer chamadas VoIP nos terminais de um utilizador RCS. Este serviço terá a capacidade de interagir com as chamadas de voz PSTN (*Public Switched Telephone Network*). Esta funcionalidade estará disponível ao utilizador RCS sempre que este possua suporte para a mesma:

Start Call from 1-to-1 Chat No ecrã de conversação 1-para-1 deverá ser disponibilizado a possibilidade de iniciar uma chamada de voz sobre IP entre os intervenientes da sessão de conversação (*chat*). Os utilizadores deverão ser redirecionados para um ecrã de chamada, semelhante ao existente para chamadas de voz através de PSTN.

Start Call from Address-Book Deverá ser possível ao utilizador selecionar um contacto RCS da lista de contactos do terminal com suporte para '*IP Voice Calls*', para estabelecer uma chamada VoIP. Os utilizadores deverão ser redirecionados para um ecrã de chamada, semelhante à existente para chamadas de voz através de PSTN.

Receive Invite Deverá ser possível ao utilizador receber convites para a iniciar uma chamada de VoIP. Os convites deverão ser apresentados ao utilizador sobre qualquer ecrã apresentado, possibilitando aceitar ou rejeitar o convite.

3.2.4.2. *IP Video Call*

O serviço "IP Video Call" consiste na possibilidade de estabelecer chamadas de vídeo através de IP, nos terminais de um utilizador RCS. Este serviço terá a capacidade de interagir com as chamadas de voz PSTN. O serviço estará disponível ao utilizador RCS sempre que este possua suporte para a mesma:

Direct Launch O utilizador deverá ter a possibilidade de iniciar uma nova chamada de vídeo, se nenhuma chamada de voz foi anteriormente estabelecida.

Upgrade to IP Video Call O utilizador deverá ter a possibilidade de estender uma chamada VoIP já estabelecida entre contactos RCS, para uma chamada de voz e vídeo.

Receive Invite Deverá ser possível ao utilizador receber convites para a iniciar uma chamada de vídeo. Os convites deverão ser apresentados ao utilizador sobre qualquer ecrã apresentado, possibilitando aceitar ou rejeitar o convite.

3.2.5. Módulo 'Social Presence'

O módulo "Social Presence" pretende incluir um conjunto de informações sobre os contactos de um qualquer utilizador RCS. Este serviço possibilita a publicação de informações, tais como:

- ***Availability***: Indicação da disposição do utilizador;
- ***Portrait Icon***: Imagem de apresentação (ex. uma foto ou imagem fornecida pelo próprio utilizador);
- ***Free Text***: Nota textual e com possibilidade de adicionar *emoticons*;
- ***Favourite Link***: *Hyperlink* de um *site* pessoal/favorito;
- ***Timestamp***: Data da última atualização do perfil;
- ***Geolocation***: Descreve a localização do utilizador.

As informações publicadas para cada utilizador RCS fazem parte de um perfil de presença, que deverá ser configurável pelo utilizador e visível na lista de contactos de cada contacto em que o utilizador RCS está relacionado.

As informações de "Social Presence" deverão estar sincronizadas com a lista de contactos existentes no terminal, sem sobrepor informações existentes no mesmo (ex. Nome de Exibição).

3.2.6. Módulo 'Capabilities Discovery'

O módulo "Capability Discovery" consiste na disponibilização de um serviço de sincronização de informação sobre suporte de serviços RCS para os quais determinado contacto estará disponível ou indisponível. Este serviço deverá basear-se em informações provenientes do protocolo de sinalização SIP (via SIP OPTIONS) ou do serviço de "Social Presence". Por forma a manter este tipo de informação o mais sincronizada possível, é necessário a inclusão de um mecanismo de *polling* para descoberta de novos contactos RCS e atualização do estado dos contactos já detetados como sendo RCS, segundo a lista de contactos (*Address-Book*) do terminal.

Para que um utilizador RCS consiga facilmente detetar os serviços disponíveis para qualquer contacto RCS existente no seu terminal, este módulo deverá disponibilizar pontos de informação sobre os serviços suportados nos diferentes ecrãs de iteração com um qualquer contacto:

- ***Address-Book/Contact Details***: "Chat", "File Transfer", "Location Sharing", "IP Voice Call"
- ***Call Screen***: "Video Sharing", "Image Sharing", "Location Sharing"
- ***IM/1-to-1 chat screen***: "File Transfer", "Location Sharing", "IP Voice Call"

3.3. Outros Requisitos

Nesta secção serão apresentados os principais requisitos não-funcionais da solução desenvolvida. Os identificadores usados na lista destes requisitos (RP, RU, RS) não seguem nenhuma ordem em específico, sendo meramente descritivos.

3.3.1. Requisitos Protocolares

RP-1: O cliente RCS deverá usar o protocolo SIP para execução de pedidos a uma rede IMS.

RP-2: O cliente RCS deverá usar o protocolo MSRP para troca de mensagens (e/ou ficheiros) numa sessão de conversação.

RP-3: O cliente RCS deverá usar o protocolo RTP para transmissão de vídeo *(vídeo sharing)* entre clientes RCS.

3.3.2. Requisitos de Usabilidade

RU-1: O cliente RCS deverá carregar o histórico de uma conversação em menos de 3 segundos após o utilizador entrar no ecrã de conversação.

RU-2: O cliente RCS deverá disponibilizar ao utilizador as últimas mensagens trocadas para cada conversação em menos de 5 segundos.

RU-3: O cliente RCS deverá disponibilizar uma réplica da lista de contactos existentes no terminal em menos de 5 segundos.

3.3.3. Requisitos de Segurança

RS-1: O utilizador deverá ser alertado sobre as permissões necessárias para utilização do cliente RCS no terminal.

RS-2: As configurações RCS existentes no terminal deverão estar encriptadas com uma chave simétrica.

3.3.4. Atributos de Qualidade

Disponibilidade: O cliente RCS deverá estar disponível no terminal de cada utilizador em 99% do tempo de atividade do terminal.

Robustez: Na ocorrência de uma quebra na ligação de dados entre o cliente e o serviço RCS externo. O cliente deverá conseguir restabelecer ligação ao serviço assim que esteja disponível uma nova ligação de dados, retomando as operações pendentes de confirmação perante o serviço, não sendo necessária a intervenção por parte do utilizador.

Usabilidade: O cliente deverá estar implementado de acordo com as diretrizes visuais Android, cumprindo em 99% das vezes as diretrizes relativas aos fluxos entre ecrãs de funcionalidades.

3.5. Roadmap

Os requisitos deste projeto foram alinhados com a *Release 5* da especificação RCS e todas as especificações RCS-e publicadas.

O principal objetivo do projeto seria a implementação de um novo cliente RCS, contendo os módulos de *Messaging* e *Content Sharing* e *Capability Discovery.* Devido ao plano inicial do projeto, os módulos *Social Presence* e *Voice*, apesar de estarem comtemplados na *Release 5* não foram incluídos nos desenvolvimentos, devido ao fato destes não se enquadrarem no tipo de cliente RCS pretendido pela entidade cliente.

Módulo	**Sub-Módulo**	**Integração (Não/Sim)**	
Messaging	*Standalone Messaging*	•	
	One-to-One Chat		•
	Group Chat		•
Content Sharing	*Video Sharing*		•
	Image Sharing		•
	Location Sharing		•
	File Transfer		•
Voice	*IP Voice Call*	•	
	IP Video Call	•	
Social Presence		•	
Capabilities Discovery			•

Tabela 3 - Tabela de integração de requisitos

Para além dos requisitos inicialmente previstos, outros requisitos foram adicionados ao longo do projeto, sendo estes extensões de funcionalidades já incluídas no planeamento inicial.

No capítulo seguinte serão descritas todas as tecnologias envolvidas no desenvolvimento dos requisitos anteriormente descritos. Será também apresentado a arquitetura geral da aplicação assim como a arquitetura de cada módulo integrado.

4. ARQUITETURA E TECNOLOGIAS

Neste capítulo será feita a contextualização da arquitetura da solução desenvolvida durante o período de estágio. Esta contextualização inclui também uma visão de alto nível da infraestrutura RCS no qual a solução será integrada. No final do capítulo é feita a descrição do trabalho realizado incluindo o meu nível de participação no desenvolvimento da solução RCS.

4.1. Protocolos e Tecnologias

Para a implementação da solução RCS para Android foi necessário ter em consideração um conjunto de tecnologias e protocolos.

4.1.1. Protocolos

Nesta secção serão descritos os protocolos utilizados na solução RCS. Estes podem ser analisados de forna mais detalhada nos RFC (*Request For Comments*) disponibilizados pela IETF (*Internet Engineering Task Force*).

HTTP – *Hypertext Transfer Protocol*

O protocolo HTTP consiste num protocolo de comunicação utilizado em sistemas de hipermédia, distribuídos e colaborativos [31]. Este é usado desde 1990 como protocolo base para a comunicação de dados WWW (World-Wide Web).

MSRP – *Message Session Relay Protocol*

O protocolo MSRP consiste num protocolo aplicacional para transmissão de mensagens baseadas em sessões. Apesar de ainda estar em desenvolvimento pela IETF, existem diversos produtos comerciais a utilizarem sua versão inicial, para disponibilização de *Instant Messaging* e *Image Sharing* entre dispositivos móveis [29].

RTP – *Real-time Transport Protocol*

Protocolo aplicacional para transmissão de dados em tempo real, atualmente usado como protocolo base para implementações de voz e vídeo sobre IP [24].

SIP/SDP – *Session Initiation Protocol/Session Description Protocol*

O SIP [23] é um protocolo aplicacional de sinalização, definido pela IETF para a criação e gestão de sessões sobre uma rede IP. O termo "sessão" refere-se a um plano de comunicação – de voz, vídeo, etc. – entre um conjunto de participantes. A sessão poderá

ser definida através de um formato SDP (*Session Description Protocol)* [22], usado para negociação dos dados de comunicação (ex. tipo de codificação de vídeo a transmitir).

4.1.2. Tecnologias

Nesta secção serão descritas todas as tecnologias utilizadas para disponibilização da solução RCS.

IMS – *Internet Protocol Multimedia Subsystem*

O IMS é uma tecnologia definida pela 3GPP (*3rd Generation Partnership Project*) e suportada por fornecedores de equipamentos de rede e serviços. O sistema IMS permite unificar aplicações baseadas no protocolo SIP, por forma a interligar serviços tradicionais de telefonia a outros tipos de serviços (ex. IM). A arquitetura IMS é definida por três camadas [8]:

- ***Transport and Endpoint***: Unifica transporte e comunicação provenientes de formatos analógicos, digitas ou banda-larga, através dos protocolos RTP e SIP, utilizando para isso *gateways* de comunicação e sinalização.
- ***Session and Control***: Orquestra ligações lógicas entre vários outros elementos da rede. Possibilita o registo de *end-points*, o encaminhamento de mensagens SIP e coordenação de recursos para sinalização e comunicação. Esta camada contém dois dos elementos mais importantes numa rede IMS, o *Call Session Control Function* (CSCF) e a base de dados do *Home Subscriber Server* (HSS). Este último mantem o perfil de serviço para cada utilizador, incluindo informações relativas ao registo, preferências, lista de contactos, entre outros.
- ***Application Services***: Responsáveis pela execução de funcionalidades sobre as sessões subscritas, controlando os estados das chamadas (sessões).

ANDROID

O Android é um sistema operativo constituído por um conjunto de componentes de *software*, em que a arquitetura carateriza-se pela divisão dos componentes em cinco secções, distribuídas em quatro camadas primárias (ver Figura 3):

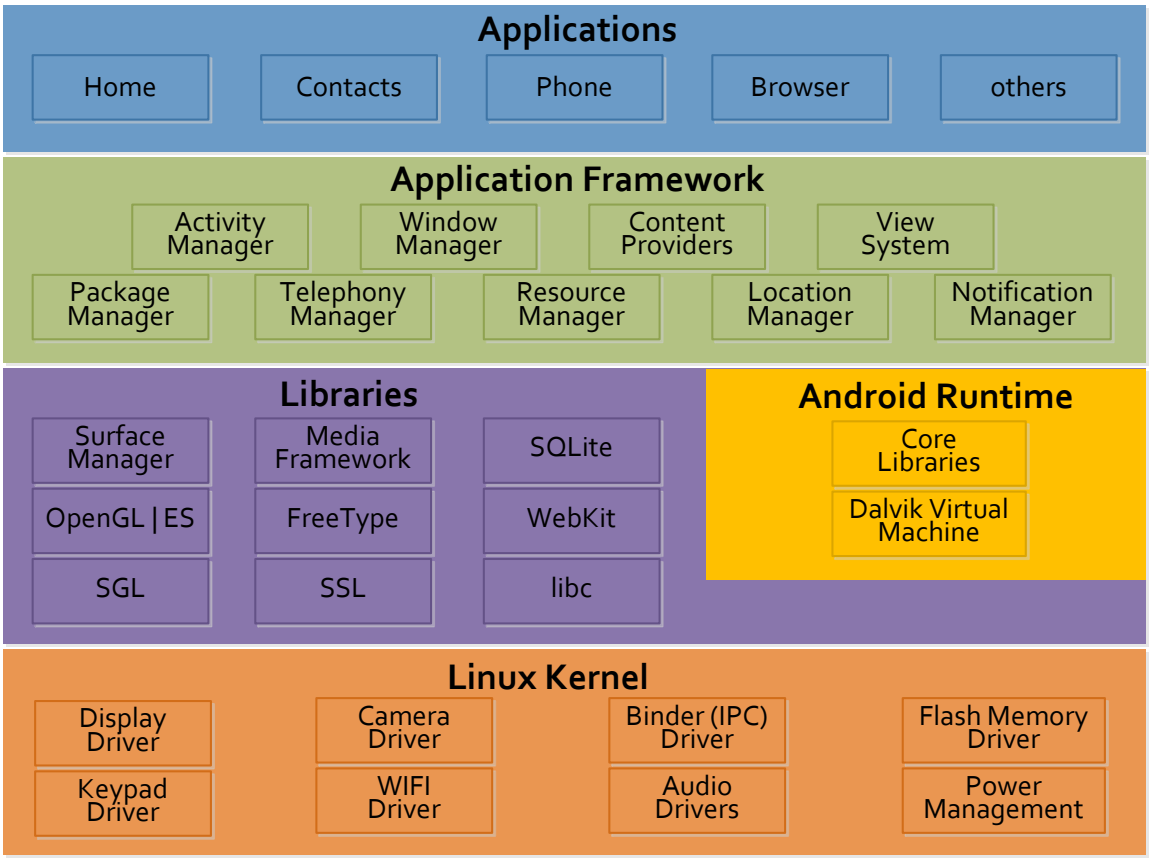


Figura 3- Arquitetura Android

- ***Linux Kernel***: Núcleo Linux contendo um conjunto de componente para gestão do sistema, incluindo a gestão de processos e memória, interface para todo o tipo de componentes *hardware*, tais como, componentes de rede, *keypad*, ecrã, entre outros.

- ***Libraries***: Conjunto de bibliotecas contendo módulos base para as funcionalidades do sistema. Entre estas bibliotecas é possível encontrar o SQLite (base de dados para salvaguarda de dados aplicacionais), o WebKit (motor de navegação *web*), bibliotecas para reprodução/gravação de áudio e vídeo, bibliotecas SSL, entre outras.

- ***Android Runtime***: Contém os componentes chave da arquitetura do sistema, como uma JVM especialmente desenhada e otimizada para o sistema Android, conhecida como Dalvik VM, que permite a cada aplicação Android a execução em processos diferenciados, recorrendo a funcionalidade do núcleo Linux para gestão de memória e *multi-threading*, intrínsecos na linguagem Java. É também disponibilizado um conjunto de bibliotecas núcleo para o desenvolvimento de aplicações sobre o sistema Android, baseadas na linguagem de programação Java.

- ***Application Framework***: Camada de acesso *high-level* a serviços do sistema. A disponibilização desta camada é baseada em Java, permitindo aos programadores de aplicações Android aceder aos serviços de sistema através das suas aplicações.

- ***Applications***: Esta camada representa a camada onde são instaladas todas aplicações desenvolvidas para utilização da plataforma Android.

H264

O H264 é um *standard* para codificação de vídeo desenvolvido pela VCEG (*ITU-T Video Coding Experts Group*) em conjunto com a MPEG (*ISO/IEC Moving Picture Experts Group*). Publicado em 2003, é atualmente usado para *streaming* de vídeos através da Internet mas também para transmissão HDTV terrestre e por satélite.

Registado sobre diversas patentes, em Agosto de 2010 foi anunciado pela entidade MPEG LA, responsável pelo licenciamento de patentes deste *standard*, que este poderá ser utilizado de forma gratuita para codificação de vídeos transmitidos através da Internet para utilizadores finais (*end users*).

JUnit

Plataforma *open-source* para automatização de testes unitários em Java [61].

4.2. Arquitetura da Infraestrutura

A arquitetura de uma infraestrutura RCS inclui obrigatoriamente o elemento IMS, para comunicação ponto-a-ponto entre clientes RCS.

A Figura 4 ilustra de forma simplificada uma infraestrutura RCS, podendo esta sofrer alterações dependendo das abordagens de implementação escolhidas pelos fornecedores de serviços RCS. Esta figura apresenta também um exemplo de comunicação entre dois fornecedores de serviços RCS.

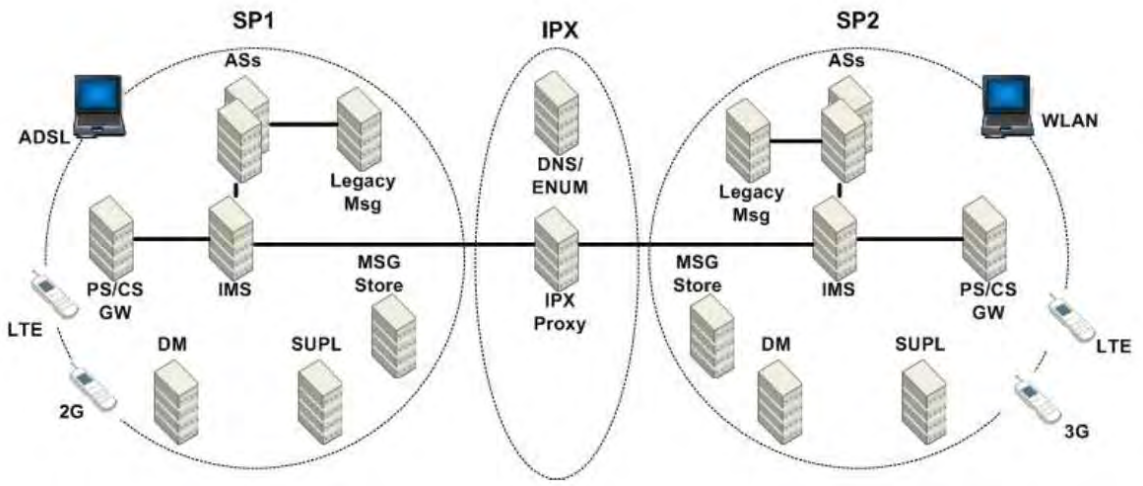


Figura 4 - Exemplo simplificado da arquitetura RCS

(fonte: [17])

As especificidades de cada elemento existente na Figura 4, assim com as suas principais responsabilidades, encontram-se detalhadas em qualquer especificação publicada pela GSMA para definição do *standard* RCS.

4.3. Arquitetura da Solução

Neste subcapítulo é feita a descrição da arquitetura da solução RCS desenvolvida, assim como dos seus principais componentes.

4.3.1. Contexto

A solução RCS implementada consiste numa aplicação Android disponibilizada para instalação em terminais Android com a versão do sistema operativo igual ou superior a 2.3 (Gingerbread). A instalação da aplicação será feita pelo utilizador do terminal Android, sendo da responsabilidade do mesmo, aceitar ou rejeitar as permissões necessárias para que a solução RCS consiga ser integrada com o sistema nativo, assim como com outras aplicações instaladas. Após a instalação da aplicação no terminal, esta estará disponível para utilização, seja para novos utilizadores (não-RCS) ou para utilizadores já registados nos serviços RCS. A identificação de um utilizador é feita com base na informação do IMSI (*International Mobile Subscriber Identity*) existente no terminal. Por forma a utilizar os serviços RCS, a aplicação irá comunicar com serviços externos, disponíveis através do acesso à Internet. Dos serviços externos utlizados pela aplicação, consta a rede IMS no qual o utilizador deverá estar registado para poder utilizar os serviços de RCS de IM, *Sharing* ou *Rich Call* (serviços de *sharing* disponibilizados durante a chamada de voz). Para além da rede IMS, são utilizados serviços Google para disponibilização de informações de geolocalização. A Figura 5 demostra de forma simplificada, o contexto da arquitetura da solução RCS implementada.

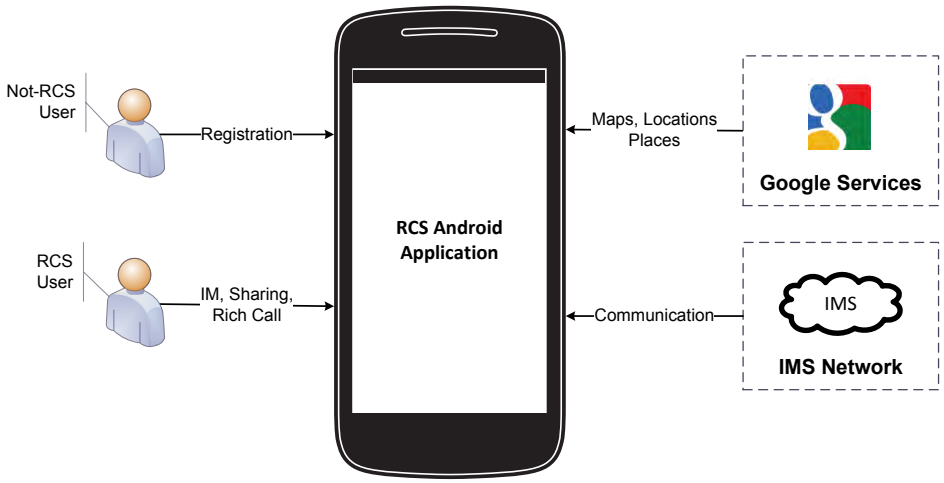


Figura 5 - Diagrama de Contexto

4.3.2. Módulos

A implementação da solução é baseada na interligação de três diferentes módulos, existindo para cada um deles diferentes níveis de integração. Na Figura 6 é ilustrada de forma simplificada a arquitetura da solução a nível modular.

RCS Application

Este módulo representa a camada onde se encontra toda a lógica de interface com o utilizador, disponibilizadas através de ecrãs de funcionalidades designadas por '*Activity*'. É também neste módulo que é feita a ligação da interface com o núcleo de comunicação RCS, designado de 'RCS API'. Esta comunicação é feita através do mecanismo IPC (*Inter-process Communication*), executadas na interface Android (AIDL).

Por forma a utilizar determinadas funcionalidades de geolocalização, este módulo inclui também a integração de chamadas aos serviços Google através de HTTPS, obtendo de forma segura dados como localizações, mapas, entre outros.

RCS API

O 'RCS API' consiste na camada onde é assegurado todo o processo de comunicação RCS. É neste núcleo que se encontra toda o processo protocolar que permite ligar a aplicação Android à rede IMS através do protocolo de sinalização SIP, assim como transmitir dados para a mesma rede através dos protocolos de transmissão MSRP, RTP ou HTTP. Toda a informação gerada no processo de comunicação, assim como configurações necessárias à mesma, são salvaguardadas neste módulo, acessível através das chamadas sobre AIDLs.

Para além de toda a lógica protocolar, este módulo possui também a responsabilidade de integrar e sincronizar os estados dos serviços RCS com os dados existentes na lista de contactos do terminal Android (*Address-Book*).

Android System

Este módulo representa todo o ambiente nativo existente no terminal onde a solução RCS é instalada, desde *providers* (base de dados com conteúdos nativos), a aplicações e serviços disponibilizados nativamente nas diferentes versões da plataforma Android e/ou pelo fabricante do terminal.

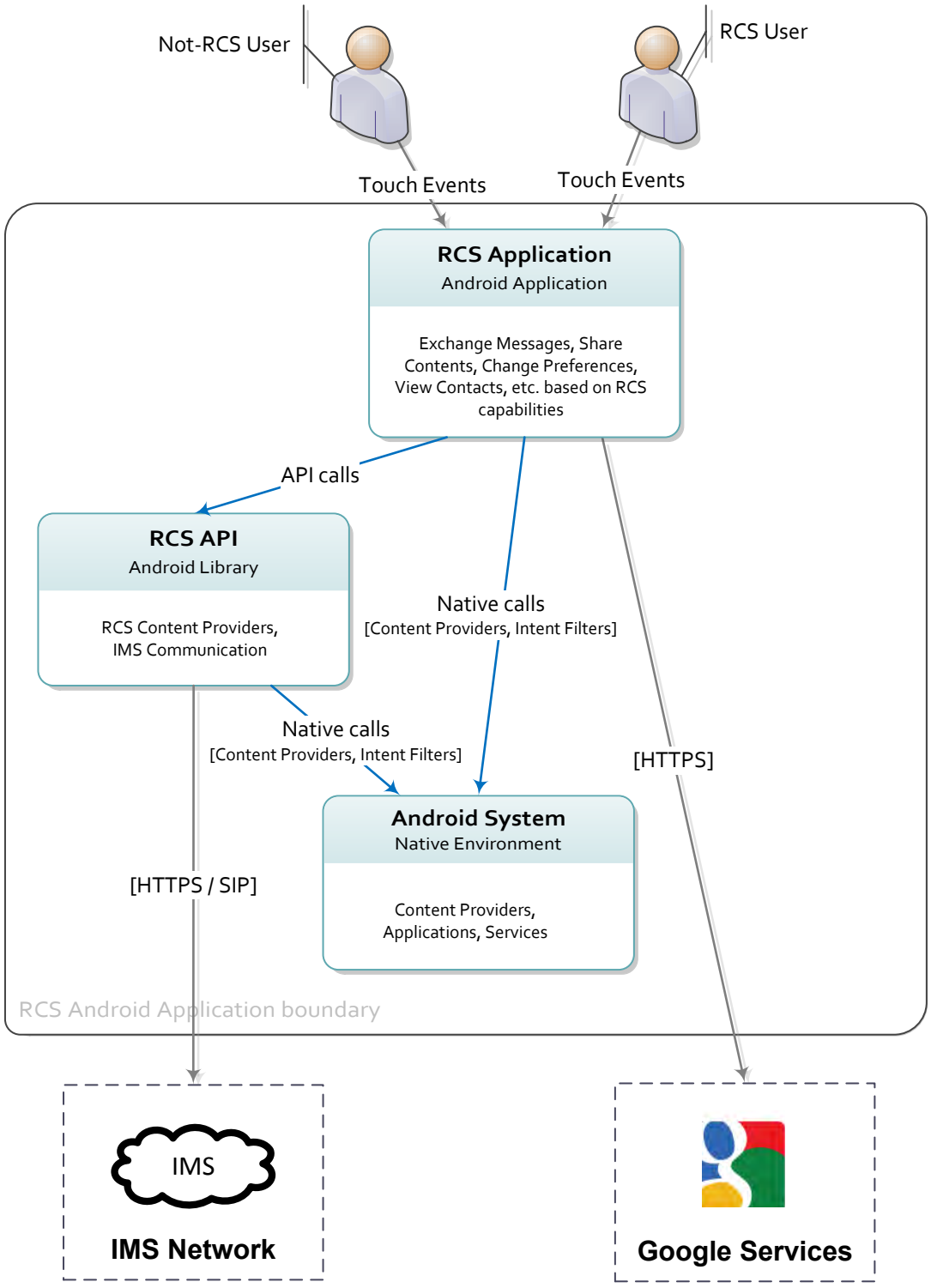


Figura 6 - Diagrama de Módulos

(simplificado)

4.3.3. Componentes

A solução integra um conjunto de componentes, distribuídos através dos módulos 'RCS Application' e o módulo 'RCS API' descritos na secção anterior e ilustrados na Figura 6.

RCS API

O módulo "RCS API" foi integrado com base na *stack* RCS para Android, disponibilizada pela Orange através do seu projeto *open-source* [53].

A implementação deste módulo é feita através de cinco diferentes componentes, ilustrados de forma simplificada na Figura 7. Entre os componentes constam:

- ***Service API***: Representa o componente responsável pela disponibilização de APIs para acesso a sessões de comunicação geridas pelo 'IMS Core' e conteúdos salvaguardados nas bases de dados locais (*providers*). Este componente inclui:
    - *Terms & Conditions API*: Controlo sobre a aceitação ou rejeição dos termos e condições apresentados ao utilizador.
    - *Capability API*: Acesso a informações sobre os serviços disponíveis para cada contacto existente na lista de contactos nativa, permitindo ainda requisitar novas informações sobre esses mesmos contactos.
    - *Contacts API*: Gestão de contactos RCS e acesso a informações existentes na lista de contactos nativa.
    - *Presence API*: Partilha, subscrição e publicação de dados de *Social Presence*.
    - *Rich Call API*: Controlo de ações de *Image Sharing* e *Video Sharing* durante uma chamada de voz CS (*Circuit Switched*).
    - *IP Call API*: Gestão de chamadas VoIP.
    - *Messaging API*: Controlo das ações de envio e receção de mensagens em conversações 1-para-1 ou grupo.

- *Media API*: Disponibilização de funcionalidade "Media Player/Renderer".
- *Events Log API*: Gestão central de todo o histórico de registos gerados através da plataforma (*Chat*, *File Transfer*, *Rich Call*) e possibilidade de agregação de dados do sistema nativo (Registos de Chamadas, SMS, MMS).

- ***Address-Book Synchronizer***: Componente responsável pela sincronização contínua dos dados existentes na lista de contactos nativa, com a lista de contactos RCS existentes na plataforma RCS. A sincronização é feita periodicamente ou quando detetadas alterações na lista de contactos nativa (ex. novo contacto, edição de informação de contacto, etc.), através de '*Content Observers*' registados para observação do *provider* de contactos nativo [54].

- ***IMS Core***: Componente nuclear de todo o processo de comunicação. É através deste componente que é feita todo a gestão de sessões de conversação, controlo protocolar, registo e redirecionamento de eventos de comunicação, iniciação de serviços locais para execução de tarefas de registo na rede IMS e inicialização de APIs de serviço.

- ***Platform Providers***: - Este componente representa todos os *providers* de conteúdos disponibilizados através do 'RCS API'. Estes *providers* são baseados em base de dados SQLite [55], estando atualmente disponíveis 6 diferentes *providers*:
  - *Rich Address-Book Provider*: Contém a lista de contactos RCS e o estado dos serviços suportados por cada contacto. Contém ainda registos de agregação para sincronização dos contactos RCS com as informações existentes na lista de contactos nativa.

- *Event Log Provider*: Contém todos os registos gerados sobre a plataforma (*Chat*, *File Transfer*, *Rich Call*), assim com a possibilidade de agregação de dados do sistema nativo (Registos de Chamadas, SMS, MMS).
- *IP Call Provider*: Contém todo o histórico de chamadas sobre IP realizadas através da plataforma.
- *Rich Messaging Provider*: Contém todo o histórico de mensagens geradas durante conversações entre contactos RCS. Estas mensagens incluem simples mensagens de texto, mensagens de geolocalização ou mensagens com informações relacionadas com a transferência de ficheiros.
- *RCS Settings Provider*: Contém configurações RCS fornecidas pela rede RCS e pelas preferências do utilizador.
- *Rich Call Provider*: Contém todo o histórico de ações de partilha (*Image Share* e *Video Share*), realizados durante as chamadas de voz.

- ***SIP Stack***: Biblioteca Java contendo a implementação do protocolo SIP.

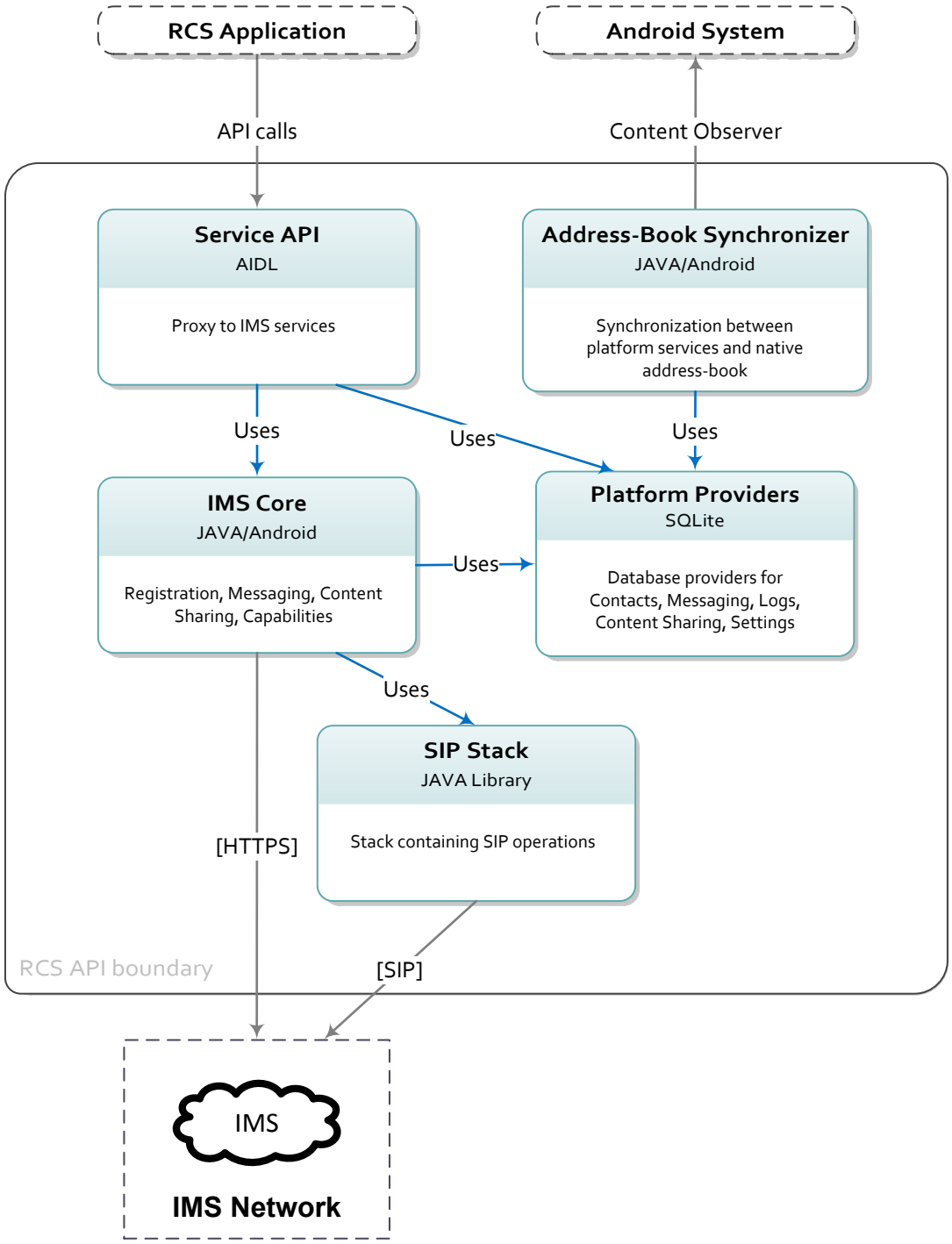


Figura 7 - Diagrama de componentes 'RCS API'

(simplificado)

RCS Application

No módulo "RCS Application" encontra-se implementada toda a lógica de interface com o utilizador, assim como a interligação das interfaces com a camada de comunicação RCS API. A implementação deste módulo é feita através de quatro diferentes componentes, ilustrados de forma simplificada na Figura 8. Entre os componentes constam:

- ***Activities***: Representa o componente responsável pela disponibilização de ecrãs (*Activity*) contendo toda a lógica funcional da aplicação. Dependendo do número de ações disponíveis ao utilizador, cada ecrã apresenta diferentes níveis de complexidade. Os ecrãs atualmente incluídos neste componente são:
    - *History Activity* (ver Figura 9): Ecrã central da aplicação. É neste ecrã que é disponibilizado todo o histórico de conversações realizadas pelo utilizador, contendo informações sobre as últimas mensagens (recebida/enviada). Através da iteração com este ecrã, o utilizador deverá conseguir reentrar numa conversação anteriormente iniciada.
    - *Contacts Activity* (ver Figura 9)*:* Ecrã contendo uma cópia da lista de contactos existentes no terminal. Para além da disponibilização de informações chave como o nome e número de cada contacto existente na lista nativa, são também disponibilizadas informações sobre o estado de cada utilizador (RCS ou não-RCS). Através da seleção de contactos, o utilizador deverá conseguir iniciar conversações com um ou múltiplos contactos (*Single Chat/Group Chat*).
    - *Spam History Activity:* Neste ecrã é disponibilizado todo o histórico de mensagens enviadas por contactos bloqueados (*black-list*). Estas mensagens são designadas como SPAM.

- *Spam Folder Activity:* Ecrã de visualização de histórico de mensagens enviadas por um contacto bloqueado e marcadas como mensagens SPAM.

- *Single Chat Activity* (ver Figura 10)*:* Ecrã de conversação entre o utilizador e um determinado contacto RCS. Através deste ecrã o utilizador poderá visualizar o histórico de mensagens trocadas durante a conversação, assim como visualizar o estado de cada mensagem enviada (*sent*, *queued*, *delivered*, *displayed* ou *failed*). Para além da troca de simples mensagens de texto, o utilizador terá também a possibilidade de partilhar todo o tipo de conteúdos (imagens, vídeos, etc.) e localizações, dependendo dos serviços suportados pelo contacto destino.

- *Group Chat Activity:* Ecrã de conversação entre o utilizador e um grupo de contactos RCS. Através deste ecrã o utilizador poderá visualizar o histórico de eventos de conferência ("Participante X entrou na conversação" ou "Participante X saiu da conversação") ou mensagens trocadas durante a conversação, assim como visualizar o estado de cada mensagem enviada (*sent*, *queued*, *delivered*, *displayed* ou *failed*). Neste tipo de conversação apenas é possível enviar e receber simples mensagens de texto. O ecrã disponibiliza também informações sobre os participantes envolvidos na conversação, a possibilidade de adicionar novos participantes à conversação e sair explicitamente da conversação em curso.

- *File Transfer Activity:* Ecrã intermédio para controlo sobre o ficheiro a partilhar com um determinado contato. O controlo inclui a verificação de um conjunto de regras determinadas pela rede IMS (ex. tamanho máximo para transferência).

- *Location Activity:* Ecrã de definição de geolocalização a partilhar como um determinado contacto RCS. A definição de uma geolocalização a partilhar, cruza informações sobre a localização do terminal com as informações requisitadas pelo utilizador.
- *Settings Activity:* Ecrã de configurações RCS, configuráveis com base nas preferências do utilizador.
- *Personalization Activity:* Ecrã de personalização visual, desde a definição do tamanho de texto das mensagens à definição do fundo de ecrã (*wallpaper*) utilizado nos ecrãs de conversação.
- *Image Share Activity:* Ecrã de partilha de imagens durante uma chamada de voz.
- *Video Share Activity:* Ecrã de partilha de vídeo em tempo real durante uma chamada de voz.
- *Location Share Activity:* Ecrã de partilha de geolocalizações durante uma chamada de voz.

- ***Managers***: Este componente representa todas as entidades para acesso a determinados dados que são constantemente utlizados nos ecrãs da aplicação. A implementação destas entidades é baseada no modelo *Singleton* [56], contendo mapeamentos de informações em *caches* locais, por forma a melhorar os tempos de acesso a informações salvaguardados nos *providers* nativos ou do RCS API.

- ***Communication Service***: Este componente consiste num serviço Android funcionando como interface cliente-servidor, sendo o cliente os ecrãs da aplicação (*Activity*) e o servidor o módulo RCS API. Este ponto intermédio permite controlar as tarefas a executar sobre a camada de comunicação (RCS API), assim como tratar informações dessa mesma camada (ex. disponibilizar mensagem recebidas no ecrã de conversação) para *Activities* e *Managers*.

- ***Receivers***: Representa a integração de entidades para recção de ações enviadas via *Intent* [57] por componentes do sistema operativo Android ou por um componente da aplicação. Estes recetores encontram-se integrados noutros componentes da aplicação (ex. *Managers*, *Activities*, entre outros).

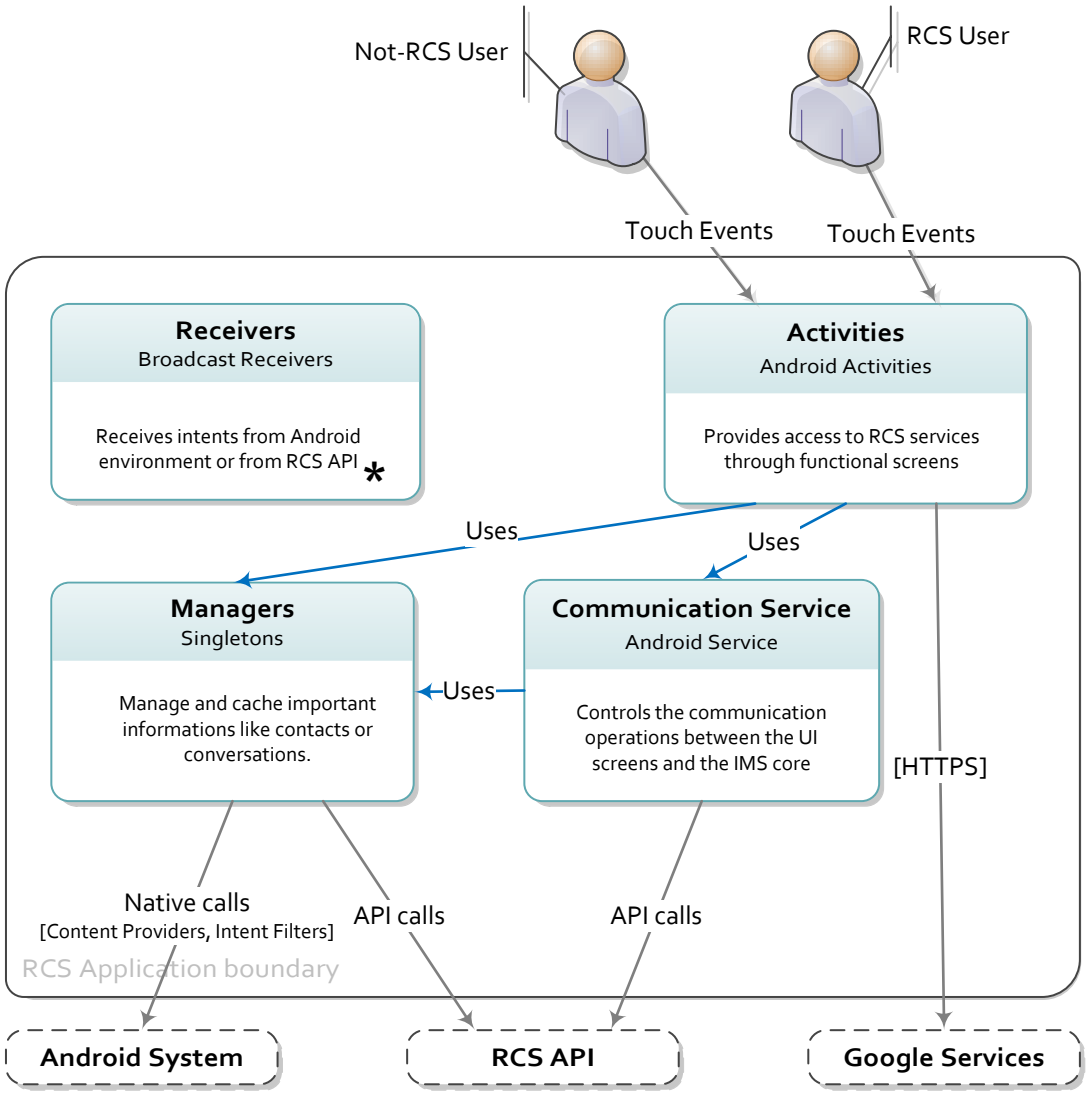


Figura 8 - Diagrama de componentes 'RCS Application' (simplificado)

4.5. Trabalho Realizado

A arquitetura da solução implementada durante o período do estágio, desenvolveu-se segundo o *roadmap* previsto na Tabela 3, tendo sido desenvolvidas a maioria das funcionalidades previstas. De seguida são apresentadas algumas informações sobre o meu envolvimento no desenvolvimento, seja através de participações de maior relevância (ex. desenvolvimento de um novo módulo) ou de menor relevância (ex. correção de *issues*)

Requisitos Funcionais RCS

Dos requisitos funcionais RCS (ver secção 0), foram implementadas com sucesso a maioria das funcionalidades requeridas para o módulo de '*Messaging*', o módulo de '*Content Sharing*' e o módulo '*Capability Discovery*'. O progresso atingindo para cada um destes módulos é descrito na Tabela 4, Tabela 5 e Tabela 6 respetivamente.

Sub-Módulo	Funcionalidade	Implementado	Nível Participação
One-to-One Chat	*Initiate a Chat*	Sim	Baixo
	Answer a Chat	Sim	Baixo
	Exchange Messages in an Established Chat	Sim	Baixo
	Display and Store Local History	Sim	Alto
	Leave Composing Chat Window	Sim	Baixo
	Display "Is Composing" Notification	Sim	Baixo
	Close/Re-Open Chat	Sim	Baixo
	Support Store & Forward Mode	Sim	Alto
	Switch to Group Chat	Sim	Alto
	File Transfer Within 1-to-1 Chat	Sim	Médio
	Spam/Blacklist Filter	Sim	Alto
	Support Emoticons	Sim	Alto
	Perform Chat Message Size Limitations	Sim	Baixo
	Support User Alias	Sim	Alto
Group Chat	*Initiate a Chat*	Sim	Alto
	Leave Composing Chat Window	Sim	Médio

	Leave Chat	Sim	Alto
	Define Chat Subject	Sim	Alto
	Exchange Messages in an Established Chat	Sim	Médio
	Display and Store Local History	Sim	Alto
	Display "Is Composing" Notification	Sim	Alto
	Close/Re-Open Chat	Sim	Alto
	Spam/Blacklist Filter	Sim	Médio
	Support Emoticons	Sim	Médio
	Perform Chat Message Size Limitations	Sim	Baixo
	Support User Alias	Sim	Alto
	Display Chat Participants	Sim	Alto
	Add a Participant	Sim	Alto
	Diplay Conference Events	Sim	Médio
	Support Store & Forward Mode	Não	-

Tabela 4 - Implementação requisitos funcionais 'Messaging'

Das funcionalidades de '*Messaging*' não foi possível a implementação da funcionalidade 'Store & Forward' em conversações de grupo, devido à falta de suporte por parte da rede IMS no período de desenvolvimento deste estágio.

No que diz respeito à participação no desenvolvimento deste módulo, o maior nível de contribuição deu-se no sub-módulo de *Group Chat*. Foram realizadas também algumas contribuições no módulo de conversações *One-to-One*, nomeadamente no controlo de contactos bloqueados para conversação (Black-List/SPAM), na disponibilização do histórico de conversações e no suporte de mensagens 'Store & Forward'.

Sub-Módulo	Funcionalidade	Implementado	Nível Participação
Video Sharing	*Share During a Voice Call*	Sim	Médio
	Share Without Voice Call	Não	-
	Receive Invites	Sim	Médio
Image Sharing	*Share During a Voice Call*	Sim	Médio
	Receive Invites	Sim	Médio
Location Sharing	*Share During a Voice Call*	Sim	Alto
	Share from 1-to-1 Chat	Sim	Alto
	Share from Address-Book	Sim	Baixo
File Transfer	*Initiate Transfer from Address Book/Call-log*	Sim	Baixo
	Initiate Transfer from Media Gallery/File Browser	Sim	Baixo
	Initiate Transfer from Camera application	Sim	Baixo
	Initiate Transfer from IM/Chat Screen	Sim	Baixo
	Receive File Transfers	Sim	Baixo
	Support File Thumbnails	Sim	Baixo
	Support Contact Card Push	Sim	Baixo
	Support Black-List/Spam Filter	Sim	Baixo

Tabela 5 - Implementação requisitos funcionais ‘Content Sharing’

Das funcionalidades de ‘*Content Sharing*’ não foi possível a implementação da funcionalidade ‘*Video Sharing*’ sem uma chamada estabelecida devido às prioridades de implementação impostas.

No que diz respeito à participação no desenvolvimento deste módulo, o maior nível de contribuição deu-se no sub-módulo de ‘*Location Sharing*’. Foram realizadas contribuições significativas para os sub-módulos de ‘*Video Sharing*’ e ‘*Image Sharing*’; e contribuições menos significativas para as funcionalidades de ‘*File Transfer*’.

Funcionalidade	Implementado	Nível Participação
Mecanismo de '*Discovery*'	Sim	Baixo
Mecanismo de '*Pooling*'	Sim	Médio

Tabela 6 - Implementação requisitos funcionais 'Capabilities Discovery'

Para o módulo de '*Capabilities Discovery*' foi integrado todos os mecanismos previstos nos requisitos, havendo um nível de contribuição maior no mecanismo de '*Pooling*'.

Outros Requisitos Funcionais

Para além dos requisitos funcionais inicialmente previstos na especificação RCS (ver secção 0), foram integradas funcionalidades não contempladas pela especificação RCS.

Das funcionalidades desenvolvidas é possível destacar dois módulos de partilha de conteúdos, designados de '*Stickers*' e 'Drawings'. Estes dois tipos de conteúdos são criados através da solução implementada, sendo a sua partilha feita segundo diretrizes de '*File Transfer*' descritas na secção 3.2.3.4. Na Tabela 7 é apresentado o progresso da implementação dos requisitos e o meu nível de participação.

Funcionalidade	Implementado	Nível Participação
Share 'Stickers'	Sim	Alto
Share 'Drawings'	Sim	Baixo

Tabela 7 - Implementação de requisitos funcionais
não contemplados na especificação RCS

Na Figura 9 e Figura 10 é possível visualizar alguns dos ecrãs de funcionalidades implementados.

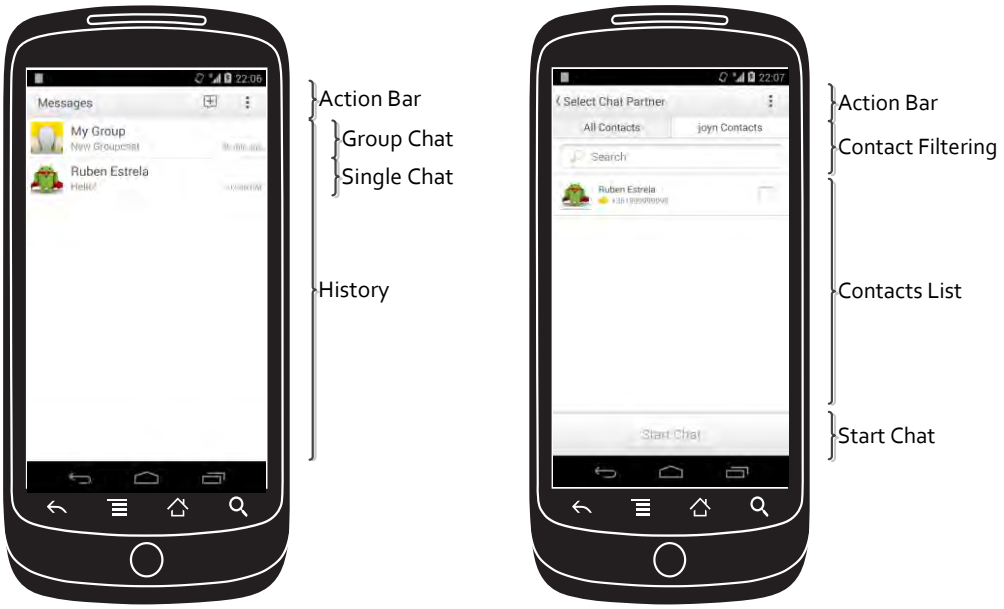


Figura 9 - Ecrã 'History' e ‘Contacts’

(à esquerda e à direita respetivamente)

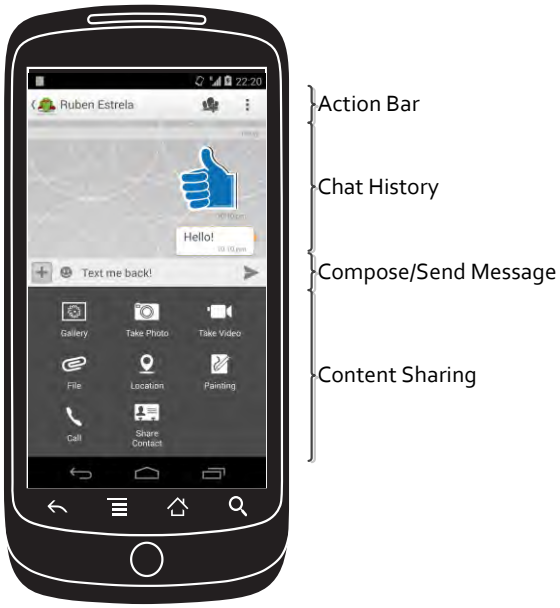


Figura 10 - Ecrã ‘Single Chat’

Requisitos Não Funcionais

No que diz a requisitos não funcionais (ver secção 3.3), foram implementados com sucesso todos os pontos referidos. Os pontos referentes à 'Segurança' e ao 'Desempenho' foram considerados de grande importância devido aos atributos de qualidade requeridos para a implementação da solução. Quanto aos requisitos protocolares, foram cumpridas todas as diretrizes necessárias para submissão da solução à entidade de acreditação de soluções RCS [58].

No capítulo seguinte será descrito a metodologia de trabalho usada no decorrer do estágio e as responsabilidades no processo de desenvolvimento.

5. METODOLOGIA DE TRABALHO

Neste capítulo é abordada a metodologia de trabalho usada para o desenvolvimento do *software* no decorrer do projeto de estágio. Serão abordadas noções base sobre a metodologia usada assim como o enquadramento na mesma neste projeto.

5.1. Enquadramento

A metodologia usada ao longo deste projeto foi a metodologia ágil Scrum, sendo esta metodologia requerida especificamente pela entidade cliente.

O Scrum [48] é uma de *framework* de desenvolvimento ágil, usada para gerir o desenvolvimento de um produto de *software* através de procedimentos iterativos. Estes procedimentos envolvem práticas de engenharia, em que os seus resultados finais são apresentados em pequenas etapas. A monotorização do desenvolvimento realizando entre as etapas é feito através de ciclos diários e semanais, onde é obtido o *feedback* sobre o progresso dos desenvolvimentos realizados.

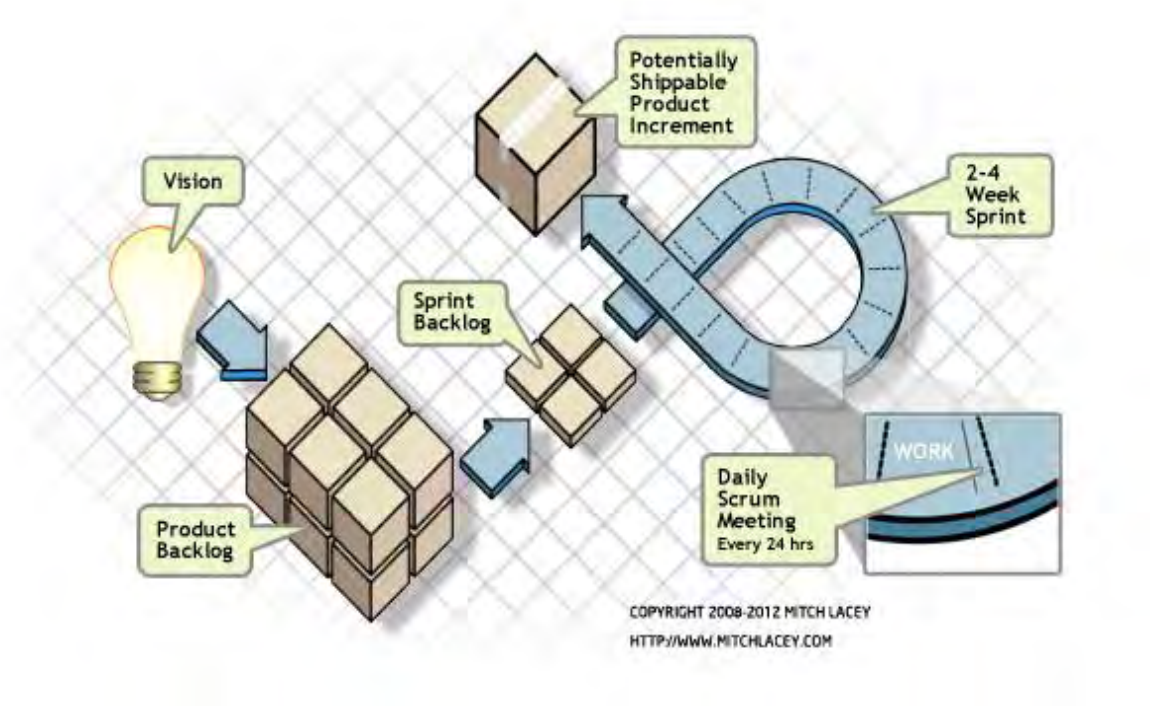


Figura 11 - Plataforma Scrum

(fonte:[59])

Esta metodologia foi projetada para projetos com constantes alterações nos requisitos iniciais, assim como para suportar requisitos emergentes. Sendo as fases de desenvolvimento caraterizadas por períodos de curta duração, os requisitos são ordenados pelo nível importância para o cliente final. Assim, o cliente final poderá visualizar mais rapidamente o produto com as principais funcionalidades implementadas, verificando desde cedo se estas contemplam os seus requisitos antes de o produto estar completamente finalizado. A Figura 11 ilustra a aplicação do Scrum no processo de desenvolvimento de um produto. A plataforma Scrum inclui conceitos chave, tais como:

- ***Scrum Master*** (SM) – Responsável por manter as práticas Scrum.
- ***Product Owner*** (PO) – Proprietário do projeto e representante dos *stakeholders*.
- ***Scrum Team*** – Grupo de trabalho que irá analisar, projetar, implementar e testar as funcionalidades do produto.
- ***Product Backlog*** – Plano de trabalho inicialmente planeado com as funcionalidades que deverão ser desenvolvidas para o produto. Este plano é normalmente discutido com o cliente e poderá sofrer alterações ao longo do tempo. As funcionalidades incluídas neste planeamento são ordenadas pela prioridade e características das mesmas (tarefas relacionadas, dificuldade, observações, etc.).
- ***Sprint*** – Período de tempo entre a reunião de planeamento (*Planning Meeting*) e a reunião de retrospeção (*Retrospective Meeting*). Na reunião de planeamento é discutido o plano de desenvolvimento do próximo *Sprint* (*Sprint Backlog*) e na reunião de retrospeção é feita a análise dos resultados obtidos, assim como os problemas encontrados durante o *Sprint*. O período tempo usado para um Sprint é normalmente curto (uma a duas semanas), durante o qual são também realizadas reuniões de atualização diária (*Daily Scrum Meetings*). Estas reuniões diárias são reuniões de curta duração (cerca de quinze minutos), e têm como objetivo o acompanhamento do trabalho realizado e também resolver problemas que poderão estar a afetar o processo de desenvolvimento.

- ***Sprint Backlog*** – Lista de tarefas a que a equipa Scrum está comprometida a completar no decorrer de um *Sprint*. As funcionalidades implementadas durante o *Sprint* são baseadas no *Product Backlog*, sendo o seu dimensionamento feito tendo em conta o tempo de Sprint. O *Sprint Backlog* é discutido no primeiro dia do *Sprint*, entre a equipa e o *Scrum Master*, não podendo ser alterado até ao final do mesmo.

5.2. Processo

O uso da metodologia Scrum foi requerido desde inicio pela entidade cliente do projeto, sendo o processo de desenvolvimento do mesmo adaptado para a utilização da metodologia de uma forma rigorosa, obedecendo a todas as regras e práticas.

Não tendo qualquer experiência neste tipo metodologia e projeto, a adaptação ao processo de desenvolvimento criou algumas dificuldades iniciais, que levaram a alguns atrasos em deferentes etapas do processo.

O processo inclui as etapas designadas como: ‘Planeamento’, ‘Desenvolvimento’, ‘Testes’, ‘Validação’ e ‘Entrega’.

Na etapa ‘Planeamento’ decorriam reuniões antes do início de uma *Sprint*, sendo analisados as prioridades do *Product Owner* em termos de funcionalidades a implementar.

Na etapa ‘Desenvolvimento’ eram realizadas as implementações das funcionalidades incluídas no *Sprint*, sendo feito em simultâneo, a projeção da arquitetura do produto de forma incremental.

A etapa ‘Testes’ decorreria imediatamente após a finalização da etapa ‘Desenvolvimento’, incluindo a execução de testes sobre as funcionalidades implementadas.

A etapa ‘Validação’ era caracterização pela execução de tarefas de qualidade, sendo uma delas a realização de revisões de código (*code-review*), feitas após o desenvolvimento de qualquer tarefa de desenvolvimento, por forma a monitorização o código criado durante as etapas de ‘Desenvolvimento’.

Estas três últimas etapas eram parte da definição de uma tarefa como 'terminada', designada de *Defenition of Done* (DoD) e ilustrada na Figura 12.

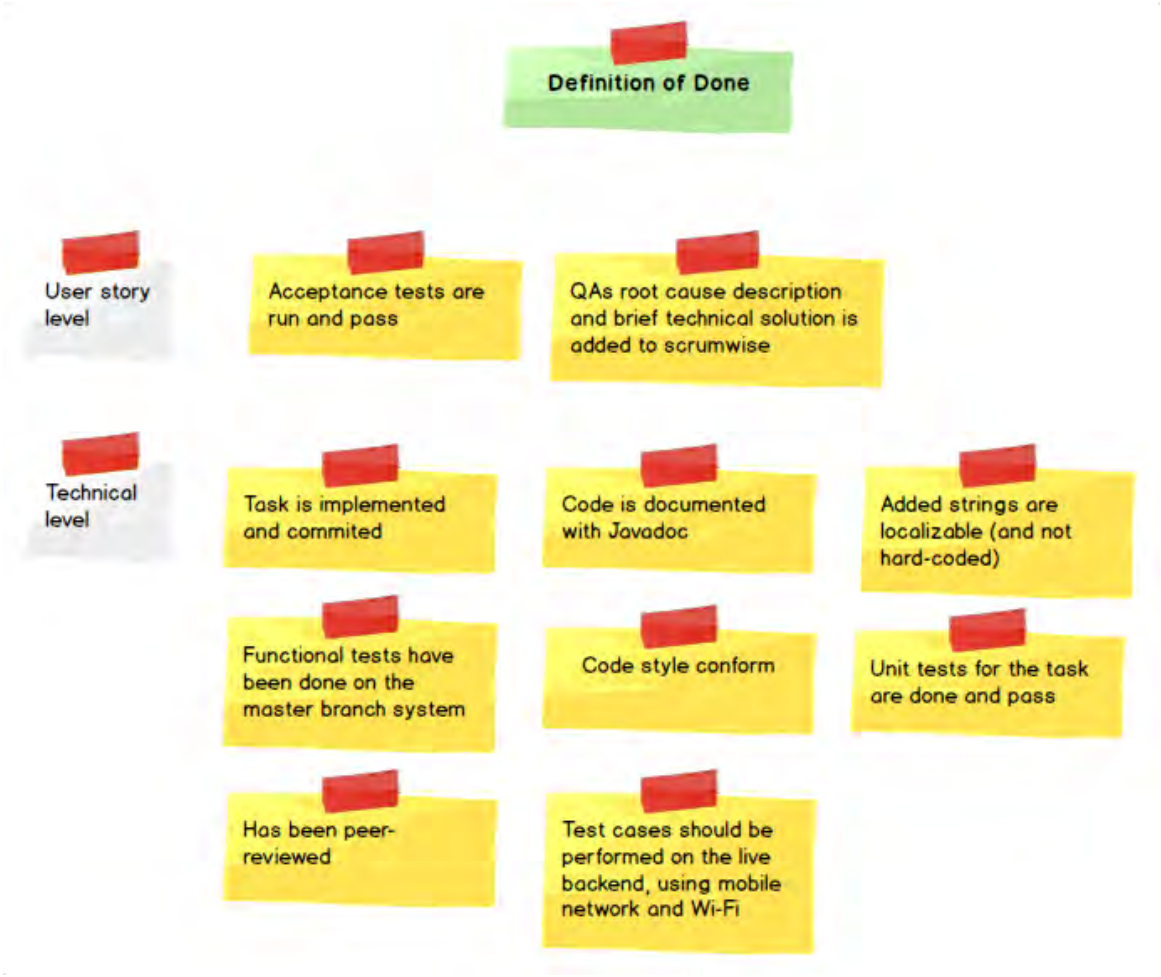


Figura 12 - Definition of Done

A última etapa o processo de desenvolvimento é a 'Entrega', sendo realizado na reunião final da *Sprint* (*Retrospective Meeting*), onde seriam apresentados todos os resultados alcançados a toda equipa do projeto, incluído os PO, sendo este último o responsável pela aprovação ou reprovação dos resultados.

5.3. Equipa

A distribuição da equipa Scrum seguiu moldes definidos pela entidade cliente, contendo elementos da WIT-Software e da própria entidade.

No que diz respeito aos papéis de gestão, foi definido pela entidade cliente nomear dois PO responsáveis pelo projeto, sendo um responsável pela área de negócio do projeto e outro responsável pela área técnica. Ambos têm igual responsabilidade de comunicação com a equipa no que diz respeito a divulgação das prioridades do projeto, assim como os parâmetros de aprovação de todos os desenvolvimentos, através da lista de objetivos que constitui o *Product Backlog*.

Outro dos papéis da equipa é o SM, tendo este como principal papel, a organização da equipa. A entidade cliente atribuiu este papel a um elemento da WIT-Software por forma facilitar a gestão da equipa em Portugal, assegurando que todos os pedidos dos PO seriam corretamente executados. Esta gestão foi feita sempre de forma passiva, uma vez que a equipa auto-organizou-se durante todo o desenvolvimento. O SM teve um papel importante na organização de todos os eventos realizados ao longo do projeto, tais como as reuniões diárias (*Daily Meetings*) e as reuniões quinzenais (*Sprint Meetings*).

O último dos papéis é "a equipa", constituída por elementos que irão garantir o cumprimento de todos os objetivos definidos previamente pelos PO e presentes nos *Sprint Backlogs*. Neste projeto, a equipa foi criada numa base de colaboração de desenvolvimentos por parte WIT, mas também pela empresa cliente, estando ambas as partes totalmente sincronizadas em termos de objetivos a atingir em cada período de desenvolvimento.

A equipa Scrum foi estruturada em 10 elementos, estando incluídos 6 elementos pertencentes à unidade *telco* da WIT-Software e 4 elementos da responsabilidade da empresa cliente. A equipa é composta por elementos da área de gestão, desenvolvimento e qualidade (SQA), como é possível visualizar na Figura 13.

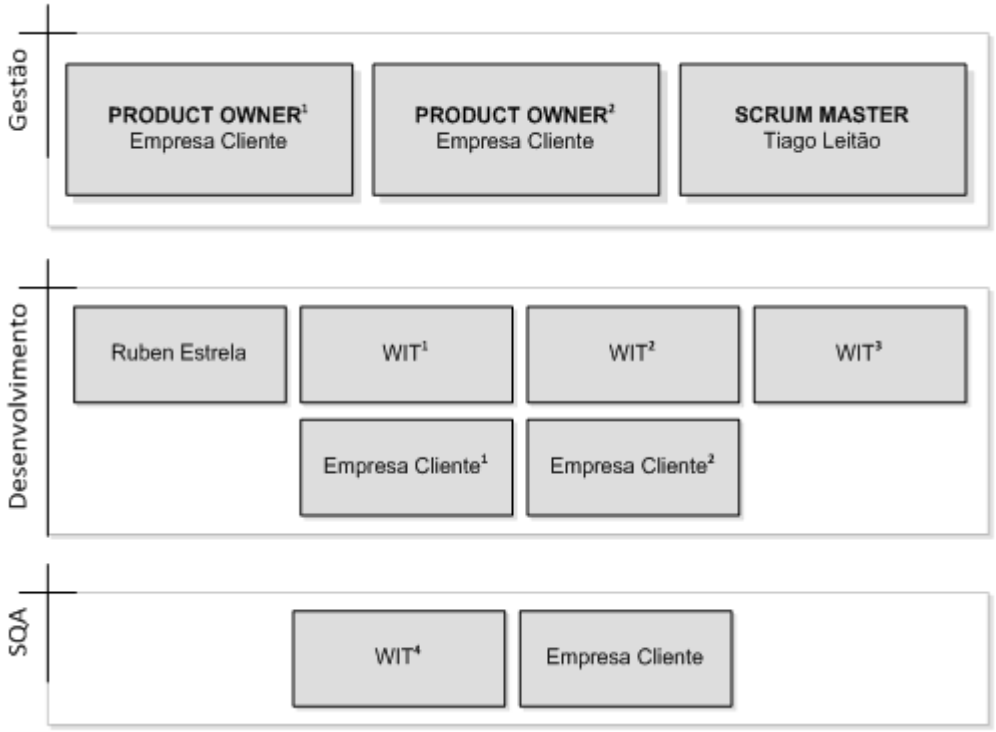


Figura 13 - Estrutura da Equipa Scrum

O meu papel nesta equipa resume-se à minha integração na equipa de desenvolvimento, sendo responsável por tarefas relacionadas com a codificação e testes.

5.4. Comunicação

A comunicação foi um dos fatores que criaram mais riscos no projeto, devido às características do mesmo. As limitações existentes em termos geográficos levaram a adaptação da equipa assim como a metodologia. Assim foi definido pela equipa o uso de teleconferências e videoconferências como via de comunicação para as reuniões diárias e semanais respetivamente.

Toda a comunicação foi feita em inglês, sendo muitas vezes difícil discutir questões mais específicas ou técnicas, uma vez que não era estabelecido um contacto presencial. Não tendo qualquer experiência neste tipo de comunicação durante um processo de desenvolvimento foi-me possível constatar que nem sempre é fácil definir claramente todos os conceitos pretendidos.

5.5. Planeamento

Os planeamentos dos *Sprints* basearam-se em três etapas. Numa primeira etapa os PO do projeto definem o *roadmap* do projeto contendo todos as funcionalidades descritas na norma RCS a serem incluídas num novo *Sprint*. A lista (*Product Backlog*) contém as funcionalidades pretendidas, possíveis melhorias e correções a incluir em futuras versões da solução. Os itens incluídos nesta lista eram constituídos por uma descrição, uma correlação com a especiação RCS e uma estimativa de esforço, sendo a sua ordenação feita com base nas prioridades dos PO. De modo a clarificar melhor cada item, sempre que necessário seria realizado um *Grooming* do *Product Backlog*. Esta seria uma atividade de colaboração entre os PO e a equipa de desenvolvimento na qual seriam adicionados detalhes, feitas estimativas de esforço e avaliação das prioridades dos itens. As estimativas elaboradas representavam o esforço necessário para alcançar o objetivo final, previamente definido no *Product Backlog*, refletidas num valor acordado por toda a equipa.

Devido às limitações geográficas, neste projeto as estimativas foram realizadas através de uma ferramenta online de *Planning Poker* [49], onde cada participante teria à sua disposição um baralho de cartas virtual para votação, cujos valores eram de 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100 e ?. O valor base para as estimativas teria como referência a complexidade, o risco ou o tempo de implementação, sendo estimado tipicamente a 0 ou ½ ponto, tarefas de fácil e rápida execução (*no-brainer*). Tarefas de maior peso seriam estimadas de 1-20 pontos, em que 1 ponto de estimativa representaria um dia de esforço da equipa para completar o objetivo final. Durante as atividades de estimativa, cada participante seleciona um destes valores para estimar o esforço para realização de uma determinada '*User Story*', proposta pelos PO. O período de votação só termina após a concordância de um valor, caso contrário a equipa continuaria o processo de votação até que haja completo consenso. Se numa das estimativas a um qualquer item do *Product Backlog*, fosse atingido um valor superior ou igual a 20 pontos, este seria mitigado por toda a equipa, por forma a criar novos itens com menor nível de complexidade.

Na Figura 14 é possível visualizar um exemplo do *Product Backlog* do projeto.

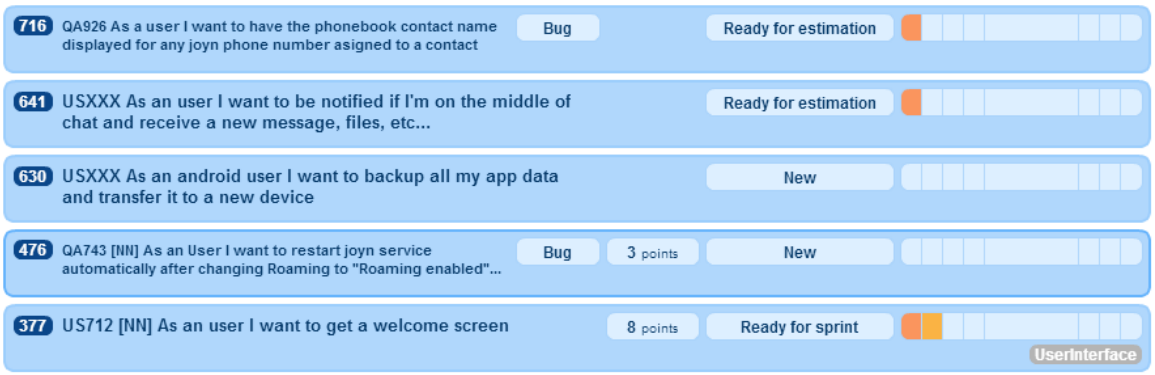


Figura 14 - Product Backlog

A segunda etapa de planeamento consistia no planeamento de trabalhos para um novo ciclo de desenvolvimento (*Sprint*), incluído um novo incremento de funcionalidades para o projeto. Este planeamento ocorria na fase de *Sprint Planning* e idealmente toda equipa estaria presente, incluindo os PO e o SM. O planeamento seria conduzido com o auxílio dos PO por forma a serem estabelecidas os objetivos para o novo *Sprint*, sendo estes os responsáveis por definir quais são as *user stories* prioritárias, criando a lista *user stories* a desenvolver (*Sprint Backlog*). Com base na seleção realizada pelos PO, a equipa procedia à aprovação do *Sprint Backlog*, assim como a velocidade prevista. Esta velocidade seria estabelecida através da soma dos pontos de estimativa de cada item presente no *Sprint Backlog*. Após aprovação da velocidade e de todo o *Sprint Backlog*, a fase de planeamento entre cliente e equipa de desenvolvimento estaria terminada, sendo iniciando de imediato uma nova etapa de planeamento. Na Figura 15 é possível visualizar parte de um *Sprint Backlog* de um dos *Sprints* do projeto.

Figura 15 - Sprint Backlog

A terceira etapa do planeamento consistia no planeamento de tarefas que seriam necessárias para completar cada item do *Sprint Backlog*. Nesta etapa apenas os elementos da equipa de desenvolvimento estariam presentes, sendo estes os responsáveis pela criação e definição de tarefas, assim como a distribuição de pontos entre as tarefas criadas para cada item do *Sprint Backlog*. Este processo era realizado de uma forma não formal, tendo a equipa total liberdade para definir prioridades entre as tarefas criadas. Por forma a organizar todo o processo de criação de tarefas, foi-me dada a responsabilidade de orientar estas atividades de planeamento, para todos os itens incluídos nos *Sprint Backlogs*. Como estas atividades eram realizadas antes do início do desenvolvimento, para alguns itens o processo de criação de tarefas poderia ser bastante complexo, sendo apenas possível criar tarefas inicias para orientação, passando a ser da responsabilidade dos envolvidos pela implementação do item, a criação de tarefas mais específicas e mais detalhadas.

Após terminar a terceira etapa de planeamento a fase de planeamento de um *Sprint* seria dado como terminado, sendo iniciado de seguida um novo ciclo de desenvolvimento.

5.6. Ciclos de Desenvolvimento

O progresso de trabalho na implementação Scrum é baseado num conjunto de ciclos de desenvolvimento designados de *Sprints*. Cada ciclo tem a duração ideal de 2 ou 4 semanas. Neste projeto o período temporal destes ciclos foi de 2 semanas e em circunstâncias específicas como períodos festivos (ex.: Natal, Páscoa, etc.), 1 semana.

Durante cada ciclo de desenvolvimento foram realizados eventos Scrum – as reuniões diárias (*Daily Meeting*) - e eventos relacionando com SQA – *code-review*:

- ***Daily Meetings***: Reuniões diárias, realizadas durante a manhã, teriam uma duração limitada a 15 minutos. Estas teriam como propósito sincronizar o trabalho diário da equipa através da comunicação a cada 24 horas. Para tal cada elemento da equipa respondia a três questões chave:
    - O que foi realizado desde a última reunião?
    - O que será feito até à próxima reunião?
    - Que obstáculos existem ou poderão existir?

  Estas reuniões eram controladas pelo SM do projeto. Os PO do projeto poderiam também estar presentes dependendo da disponibilidade dos mesmos.

- ***Code-Review***: Os eventos de *code-review* consistiam em reuniões de inspeção de código-fonte para monitoração das alterações ao código do projeto. Estas inspeções eram realizadas por um elemento da equipa de desenvolvimento, não responsável pelo desenvolvimento do código sob inspeção e os elementos responsáveis pelos desenvolvimentos. O processo de inspeção de código poderia ter uma duração de 1 a 8 horas, dependendo da complexidade das alterações ou dimensão da funcionalidade implementada.

Durante todo o ciclo cada elemento da equipa era responsável pela atualização do estado e progresso das suas tarefas. Na Figura 16 é possível visualizar um exemplo de uma dessas tarefas no contexto de um dos *Sprints* do projeto.

Figura 16 - Sprint Task Board

O progresso de cada ciclo (*Sprint*) era calculado com base na diferença do número de pontos completados e o número de pontos das tarefas a realizar, distribuído pelo número de dias até ao final do ciclo. Este tipo de informação era disponibilizado através de gráficos designados de *Sprint Burndown Chart*. Na Figura 17 é possível visualizar um *Burndown Chart* de um dos *Sprints* do projeto.

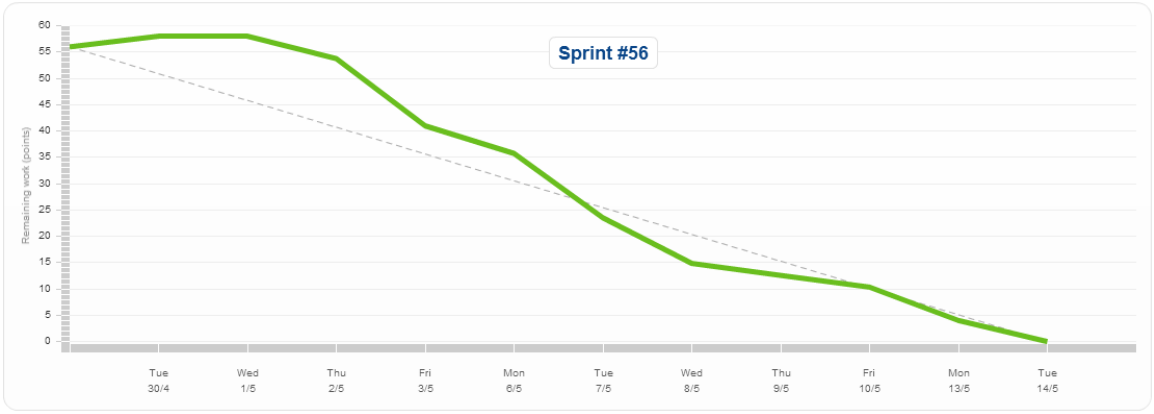


Figura 17 - Sprint Burndown Chart

5.7. Testes e Validação

Durante todo o processo de desenvolvimento, foram incluídas diversas atividades relacionadas com a garantia de qualidade de cada funcionalidade integrada. Entre elas, estão atividades de teste e atividades de validação:

- **Testes**: Os testes consistem em atividades relacionadas com a automatização e execução de testes funcionais (*black-box*) sobre cada funcionalidade integrada. Estas atividades seriam da total responsabilidade da equipa de desenvolvimento.

- **Validação**: As atividades de validação consistem em atividades realizadas após uma funcionalidade ser dada como terminada e testada por parte da equipa de desenvolvimento. Estas incluiriam a validação dos critérios de aceitação de cada funcionalidade, segundo a especificação RCS e critérios estabelecido pelos PO do projeto. Estas atividades seriam maioritariamente da responsabilidade da equipa de SQA.

Durante este projeto, foi realizado um estudo sobre ferramentas para automatização de testes funcionais sobre a plataforma Android. Com base no resultado desse mesmo estudo, foi possível integrar a ferramenta *open-source* Robotium [52]. Os resultados do estudo encontram-se mais detalhados no Anexo A.

5.8. Monitorização

Para monitorização do estado do projeto e progresso dos desenvolvimentos para cada ciclo desenvolvimento foi definido pela equipa, a utilização de uma ferramenta *online* por forma a eliminar as limitações geográficas, assim como melhorar o processo de registo e atualização dos artefactos do projeto. A ferramenta utilizada foi o Scrumwise [50] que consiste num portal *web* acessível 24h/dia a cada elemento do projeto, tendo como principais funcionalidades, gerir o *Product Backlog*, gerir o *Sprint Backlog*, gerir o *Sprint Task Board* e visualizar o progresso no *Sprint Burndown Chart*.

5.9. Gestão de Informação

A gestão da informação durante o projeto foi feita através da utilização de um sistema de gestão de versões SVN [51] para salvaguarda de artefactos e código-fonte e a plataforma Scrumwise [50] para salvaguarda de informações sobre a evolução de requisitos do projeto.

Este projeto sofreu um processo evolutivo dividido em etapas. Para cada uma das etapas era gerada uma nova *release* (REL), organizada no SVN através de um *branch* próprio, salvaguardando o código-fonte dessa *release* e não afetando a *release* em desenvolvimento. Na Figura 18 é ilustrada a evolução das *releases* criadas no decorrer do estágio.

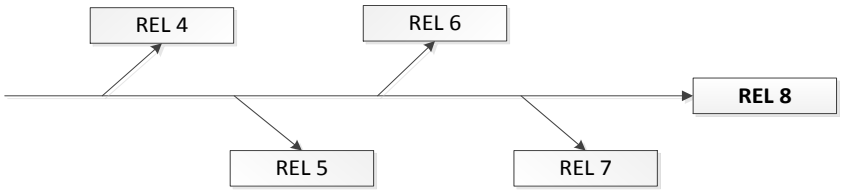


Figura 18 - Fluxo de versões do projeto

6. CONCLUSÕES

Este capítulo expõe uma análise dos resultados alcançados durante o desenvolvimento do cliente RCS para Android. De seguida são descritas algumas das dificuldades sentidas durante o desenvolvimento do estágio. O capítulo termina com uma referência sobre trabalho futuro relativamente à evolução do projeto.

6.1. Revisão do trabalho realizado

No decorrer do período de estágio foi desenvolvida com sucesso uma aplicação Android segundo o *standard* com capacidade de ser integrada em redes IMS com suporte ao mesmo *standard*.

Esta nova solução RCS pretende satisfazer a procura de um novo cliente RCS para a plataforma Android, capaz de ser utilizado na rede de uma operadora de telecomunicações de referência a nível europeu. Foram estabelecidos elementos chave para o sucesso do projeto, nomeadamente, assegurar a disponibilização de uma solução contendo funcionalidades para conversação e partilha de conteúdos; e garantir a acreditação (GSMA) da solução desenvolvida. Em termos gerais, é possível afirmar que os objetivos delineados para o período de estágio foram alcançados com sucesso, existindo uma versão da solução já acreditada e disponível na plataforma oficial de aplicações Android (Google Play).

No ponto de vista do estágio, é necessário dar especial ênfase ao nível de conhecimento adquirido ao longo do mesmo, através da aprendizagem de protocolos e tecnologias, da integração em equipas de desenvolvimento adaptadas à metodologia Scrum e do estudo do *standard* RCS como ponto de partida para a disponibilização de novos serviços de comunicação.

6.2. Dificuldades

Eram esperados diversos tipos de dificuldades no decorrer deste estágio, tanto a nível de metodologia de trabalho como a nível técnico.

Uma das principais dificuldades com que me vi confrontado foi a minha inexperiência na metodologia de desenvolvimento utilizada. O Scrum apesar de facilitar o

processo de desenvolvimento com a sua abordagem ágil requer também alguma experiência profissional no que diz respeito à comunicação interpessoal, por forma a facilitar o processo de planeamento de tarefas e a sua implementação. Foi-me possível experienciar por diversas vezes a importância de uma boa comunicação dentro da equipa de desenvolvimento. As barreiras geográficas a que a equipa estava sujeita, criavam por vezes grandes dificuldades no processo de planeamento de tarefas, tanto na definição das tarefas como na organização de prioridades e dependências das mesmas. Nem sempre foi possível alcançar o consenso, sendo por vezes obrigado a questionar determinadas prioridades ou escolhas para implementação de algumas das funcionalidades da solução.

O progresso da solução era acompanhado de perto pelos responsáveis do projeto, sendo a comunicação um fator constante. Apesar desta mais-valia, esta criava algumas dificuldades a longo prazo devido à falta de suporte documental, que ao existir seria capaz de facilitar a discussão de escolhas feitas no passado, através de uma linha cronológica dessas mesmas escolhas. Uma das áreas onde me foi requerido um maior nível de esforço devido à falta dessa documentação de suporte, foram as questões de *User Experience* (UX), de extrema importância no desenvolvimento deste tipo de soluções. Muitas das vezes a prototipagem era feita quase em simultâneo com a implementação, o que apesar de facilitar as decisões dos responsáveis do projeto, criavam algum esforço adicional na fase de implementação, nomeadamente em questões de UX não compatíveis com a plataforma Android.

A nível técnico, fui confrontado com diferentes dificuldades, derivadas da complexidade do projeto mas também pela minha inexperiência em todas as tecnologias envolvidas. Como seria de esperar, o desconhecimento dos protocolos usados requereu um estudo mais aprofundado, por forma a compreender os requisitos funcionais da especificação RCS.

O desenvolvimento de uma arquitetura para implementação da solução foi outro dos grandes desafios deste projeto. Devido à minha inexperiência no desenvolvimento de aplicações para a plataforma Android, algumas das minhas escolhas para o desenvolvimento de determinados módulos estavam bastantes limitadas ao meu nível de conhecimento da arquitetura Android, sendo por vezes necessário um estudo aprofundado

de determinados componentes. Durante esses estudos era muitas vezes confrontado com a necessidade de estes serem mais específicos, o que seria muitas vezes difícil visto que a documentação disponibilizada para alguns dos componentes da arquitetura Android não se encontrarem devidamente documentados. Estas falhas na documentação oficial obrigavam ao estudo do próprio código dos componentes, disponibilizada de forma não-oficial, tendo em conta as várias versões do sistema operativo Android.

Outra das dificuldades sentidas ao longo de todo o estágio foi a capacidade de garantir a estabilidade das funcionalidades desenvolvidas, nomeadamente através da execução de testes funcionais. Sendo as execuções de testes feitas inicialmente de forma manual, constatei que a integração de uma ferramenta para automatização dos testes pode ser um processo bastante complexo.

Apesar das dificuldades descritas anteriormente, foi extremamente gratificante verificar que todas elas foram ultrapassadas com sucesso, graças ao esforço pessoal e à ajuda da equipa no qual estive integrado no decorrer do estágio.

6.3. Trabalho futuro

A especificação RCS está em constante evolução, existindo constantes atualizações através do lançamento de novas versões ou através de *subsets* específicos para soluções cliente, contendo melhorias às funcionalidades RCS disponibilizadas.

Para que a implementação atual continue compatível com a norma é necessário que esta acompanhe a evolução, através da integração de futuros serviços RCS ou aplicando as melhorias sugeridas.

Graças ao sucesso da solução desenvolvida durante o período de estágio, foi já iniciado o planeamento dos próximos passos para tornar a solução compatível com a especificação "*joyn Blackbird"*, que inclui entre outras coisas a integração dos serviços xMS.

7. REFERÊNCIAS BIBLIOGRÁFICAS

As referências utilizadas ao longo deste relatório são à frente designadas. É feita a segmentação por 4 categorias: Artigos científicos, Livros, Normas e Sites Web.

Artigos científicos

[1] Lin, M. & Arias, J. (2011), "Rich Communication Suite: The Challenge and Opportunity for MNOs"

[2] Henry, K., Liu, Q. & Pasquereau, S. (2009), "Rich Communication Suite: A convergent multimedia communication service over IMS"

[3] Thanh, D. et al. (2008), "Towards a Uniform IMS Client on Heterogeneous Devices"

[4] Lee, A. (2010), "Application Creation for IMS Systems Through Macro-Enablers and Web 2.0 Technologies"

[5] Dorbes, G. & Hue, C. (2009), "Beyond RCS: Capitalizing on address book to embrace multimedia services"

[6] Trivedi, N. & Jain, A. (2013), "Implementation Challenges in Rich Communication Suite-enhanced (RCS-e)"

Livros

[7] Wuthnow, M., Shih, J. & Stafford, M. (2009), "IMS: A New Model for Blending Applications"

[8] Poikselka, M. et al (2006), "The IMS: IP Multimedia Concepts and Services"

[9] Milano, D. (2012), "Android Application Testing Guide"

[10] Hanzo, L., Cherriman, P. & Streit, J. (2007). "Video Compression and Communications"

Normas

[11] GSMA (2010), "RCS Release 1 Functional Description", Version 1.2

[12] GSMA (2011), "RCS Release 2 Functional Description", Version 2.0

[13] GSMA (2011), "RCS Release 3 Functional Description", Version 2.0

[14] GSMA (2011), "RCS Release 4 Functional Description", Version 1.0

[15] GSMA (2011), "RCS Release 2 Functional Description", Version 2.0

[16] GSMA (2012), "RCS-e - Advanced Communications: Services and Client Specification", Version 1.2.2

[17] GSMA (2012), "RCS 5.0: Services and Client Specification", Version 1.0

[18] GSMA (2012), "RCS 5.1: Services and Client Specification", Version 1.0

[19] GSMA (2012), "RCS-e joyn Hot Fixes: User Experience Guidance Document", Version 1.2

[20] GSMA (2012), "joyn Blackbird Product Definition Document", Version 0.1

[21] IETF RFC 3261 (2002), "SIP: Session Initiation Protocol"

[22] IETF RFC 3264 (2002), "An Offer/Answer Model with the Session Description Protocol (SDP)"

[23] IETF RFC 3428 (2002), "Session Initiation Protocol (SIP) Extension for Instant Messaging"

[24] IETF RFC 3550 (2003), "RTP: A Transport Protocol for Real-Time Applications"

[25] IETF RFC 3862 (2004), "Common Presence and Instant Messaging (CPIM): Message Format"

[26] IETF RFC 3966 (2004), "The tel URI for Telephone Numbers"

[27] IETF RFC 3984 (2005), "RTP Payload Format for H.264 Video"

[28] IETF RFC 3994 (2005), "Indication of Message Composition for Instant Messaging"

[29] IETF RFC 4975 (2007), "The Message Session Relay Protocol (MSRP)"

[30] IETF RFC 5438 (2009), "Instant Message Disposition Notification (IMDN)"

[31] IETF RFC 1945 (1996), "Hypertext Transfer Protocol -- HTTP/1.0"

Sites Web

[32] Gartner, "Gartner Says Worldwide Sales of Mobile Phones Declined 3 Percent in Third Quarter of 2012; Smartphone Sales Increased 47 Percent", http://www.gartner.com/newsroom/id/2237315

[33] Android Official Blog, http://officialandroid.blogspot.pt

[34] comScore, "15.5 percent of European Smartphone Owners Have a Tablet", http://www.comscoredatamine.com/2012/11/15-5-percent-of-european-smartphone-owners-have-a-tablet

[35] OfCom, "Communications Market Report 2012", http://stakeholders.ofcom.org.uk/binaries/research/cmr/cmr12/CMR_UK_2012.pdf

[36] PORTIORESEARCH, "Mobile Messaging Futures 2012-2016", http://www.portioresearch.com/en/reports/current-portfolio/mobile-messaging-futures-2012-2016.aspx

[37] Disruptive Telephony, "What is an Over-The-Top (OTT) Application or Service?", http://www.disruptivetelephony.com/2012/07/what-is-an-over-the-top-ott-application-or-service-a-brief-explanation.html

[38] Business Insider, "Why 2011 Is Being Called The Year Of 'The Cable Cut'", http://articles.businessinsider.com/2010-12-30/tech/30022400_1_internet-video-ott-video-services

[39] The Guardian, "WhatsApp: the new text messaging?", http://www.guardian.co.uk/technology/shortcuts/2012/dec/04/whatsapp-new-text-messaging

[40] analysys mason, "Quantifying the impact of OTT communications services in Western Europe", http://www.analysysmason.com/Research/Content/Reports/voice-and-messaging-Western-Europe-Dec2012-RDMV0

[41] Vodafone, "Annual Report 2012", http://www.vodafone.com/content/dam/vodafone/investors/annual_reports/Vodafone_Annual_Report_12.pdf

[42] FierceWireless, "Ovum figures indicate that operators will lose US$54bn by 2016 due to smartphone messaging apps", http://www.fiercewireless.com/europe/press-releases/ovum-figures-indicate-operators-will-lose-us54bn-2016-due-smartphone-messag-0

[43] GSMA, "Spanish Mobile Operators Launch Nationwide joyn Services", http://www.gsma.com/rcs/spanish-mobile-operators-launch-nationwide-joyn-services

[44] FierceWireless, "Vodafone Germany leads rivals with Joyn launch to battle OTT messaging apps", http://www.fiercewireless.com/europe/story/vodafone-germany-leads-rivals-joyn-launch-battle-ott-messaging-apps/2012-08-29

[45] TechCrunch, "Facebook Says VoIP Calling Will Be Added To Its Messenger iOS App In The U.K. Today", http://techcrunch.com/2013/03/25/facebook-messenger-uk-gets-voip

[46] WIT-Software, http://www.wit-software.com

[47] GSMA, "Rich Communications", http://www.gsma.com/rcs

[48] Scrum Alliance, http://scrumalliance.org

[49] Planning Poker, http://www.planningpoker.com

[50] Scrumwise, http://www.scrumwise.com

[51] Apache Subversion, http://subversion.apache.org

[52] Robotium, https://code.google.com/p/robotium

[53] android-rcs-ims-stack, https://code.google.com/p/android-rcs-ims-stack

[54] Android Contacts Provider, http://developer.android.com/guide/topics/providers/contacts-provider.html

[55] SQLite, http://www.sqlite.org

[56] Singleton Pattern, http://www.oodesign.com/singleton-pattern.html

[57] Broadcast Receiver, http://developer.android.com/reference/android/content/BroadcastReceiver.html

[58] GSMA – RCS – Interoperability & Testing – Accreditation, http://www.gsma.com/rcs/interoperability-testing/accreditation

[59] mitchlacev, http://www.mitchlacey.com/

[60] Dialogic, "Architecture and Benefits of IMS", http://www.dialogic.com/~/media/products/docs/whitepapers/11297-ims-arch-benefits-wp.pdf

[61] JUnit, http://junit.org

8. ANEXOS

A seguinte listagem identifica os anexos complementares a este documento:

- Anexo A. Especificação *Rich Communication Suite*
- Anexo B. Automatização de testes funcionais na plataforma Android

ANEXO A. Especificação *Rich Communication Suite*

O conteúdo deste anexo encontra-se disponibilizado em formato digital.

ANEXO B. Automatização de testes funcionais na plataforma Android

O conteúdo deste anexo pretende descrever os resultados obtidos através do estudo de ferramentas *open-source* para suporte na execução de testes funcionais sobre a plataforma Android.

B.1. Ferramentas *open-source*

Existem atualmente algumas ferramentas *open-source* para automatização de testes funcionais em Android, com convergência na execução de testes em três tipos de funcionalidades: nativas, *web-based* e híbridos (funcionalidades nativas com funcionalidades *web-based* incorporadas).

Nesta secção são apresentadas algumas das ferramentas encontradas durante o trabalho de investigação, sendo feita uma breve descrição de cada uma das ferramentas e suas características. As ferramentas abaixo descritas não foram escolhidas com base em nenhum critério específico, mas sim pela convergência de opiniões encontradas, no que pode ser descrito como "ferramentas de teste úteis" para suporte ao desenvolvido de aplicações Android.

B.1.1 Activity Instrumentation

O Android SDK disponibiliza uma avançada *framework* de testes, baseada na *framework* de testes JUnit e na instrumentação de código. A instrumentação consiste na capacidade de monitorizar e diagnosticar uma aplicação através da inserção de código de monitorização, técnicas de depuração (*debug*), contadores de performance e registos (*logs*) de eventos no código, que permite também a medição do desemprenho e controlar comportamentos.

Os testes criados para esta *framework*, são baseados em testes unitários validados através da funcionalidade *standard* "falhou/passou" do JUnit, para avaliação dos resultados da execução.

A utilização desta *framework* requer conhecimentos em programação Java, assim como alguma experiência no desenvolvimento de testes unitários com a *framework* JUnit,

por forma a facilitar os testes às funcionalidades implementadas, através do acesso ao código-fonte da aplicação.

Uma característica fundamental da *framework* de instrumentação Android são as classes elaboradas para cada componente tipo, incluindo métodos próprios para lidar com o ciclo de vida de cada componente e métodos para inserção de objetos de simulação (*mocks*).

Até ao momento a *framework* de instrumentação Android inclui classes de instrumentação para execução de testes em componente tais como, 'Atividades', 'Serviços' e 'Base de dados de conteúdos'.

B.1.2 Monkeyrunner

O *monkeyrunner* é uma ferramenta disponível no Android SDK, contendo um ambiente próprio para desenvolvimento de programas, fora do código-fonte das aplicações Android, capazes de executar diferentes tarefas associadas aos casos de testes.

Com o *monkeyrunner* é possível controlar dispositivos e emuladores Android, através de uma workstation que envia comandos e eventos específicos através da sua interface (API). As execuções de comandos são feitas através de programas desenvolvidos com a linguagem de programação Python.

O *monkeyrunner* foi projetado para testar aplicações e dispositivos a nível funcional, e correr conjuntos de testes unitários, sendo possível estender o seu uso para outros fins que se possam vir a desejar, tais como criar novos módulos desenvolvidos em Java e usá-los para estender a API do *monkeyrunner* com novas funcionalidades ou alterar as existentes.

B.1.3 Robotium

O Robotium foi desenvolvido para execução de casos de testes sobre aplicações Android, tendo como principal objetivo a criação de testes funcionais automatizados. Desenvolvido sobre a plataforma de instrumentação Android, o Robotium permite a criação de cenários de testes funcionais, sistema e aceitação.

Esta ferramenta permite simular comportamentos semelhantes ao de um normal utilizador, tais como mudar a orientação do ecrã, digitar texto ou até mesmo simular o toque no ecrã. É possível a utilização de componentes de testes unitários e de regressão, sendo da responsabilidade dos programadores o nível de rigor a aplicar aos testes desenvolvidos.

Tendo a instrumentação Android e a plataforma JUnit como base, é possível refinar os critérios de "passou/falhou" para cada caso de teste, através da informação proveniente do acesso direto ao código-fonte.

B.1.4 MonkeyTalk

O MonkeyTalk é uma ferramenta para execução de testes funcionais, concebida para ser utilizada sobre aplicações de diferentes plataformas, entre elas a plataforma Android.

Através desta ferramenta é possível criar vários tipos de testes nativos, web e híbrido, de forma automatizada, sobre dispositivos reais – sem alterações de configurações originais do fabricante – mas também sobre simuladores.

A ferramenta é composta por três componentes: scripts MonkeyTalk – contêm ações de teste a executar sobre uma aplicação alvo, e que podem ser construídos através de comandos MonkeyTalk ou através de código JavaScript; o IDE MonkeyTalk – editor para gravação e reprodução de scripts de teste; e o agente MonkeyTalk – biblioteca que ao ser adicionada ao ficheiro binário da aplicação alvo, permite gravar ou reproduzir comandos MonkeyTalk diretamente na aplicação.

Por se tratar de uma ferramenta de testes multiplataforma, os mesmos scripts podem ser executados em diferentes plataformas (ex. Android e iOS), tornando mais uniforme a execução de testes a funcionalidades.

B.1.5 Calabash

O Calabash consiste numa biblioteca desenvolvida para suportar a plataforma Android (existindo uma versão iOS) com base na plataforma de testes *Cucumber*. O Cucumber foi desenvolvido para criar e executar especificações de *software*, através de testes funcionais escritos em texto simples. Sendo uma extensão do Cucumber, o

Calabash permite que os testes criados sejam criados numa linguagem natural, com termos e conceitos do domínio de negócio.

Os testes criados para esta ferramenta (tal como no Cucumber) são criados com base em "cenários" e "passos" (steps) desses mesmos cenários, que deverão ser executados para "aprovação" de um determinado caso de teste. Estes steps podem ser criados com base na linguagem Ruby ou qualquer outra linguagem baseada na JVM.

B.2. Estudo Comparativo

Nesta secção serão apresentados os resultados obtidos através da experimentação e investigação mais detalhada sobre as capacidades de cada uma das ferramentas apresentadas na secção anterior.

B.2.1 Métricas de Avaliação

Por forma a facilitar a comparação das características de cada ferramenta, estas foram analisadas e comparadas tendo em conta um conjunto de métricas, que poderão auxiliar a escolha da ferramenta mais viável, num contexto empresarial. As cinco métricas utilizadas, incluem: 'Usabilidade', 'Simplicidade de Instalação', 'Complexidade de Criação de Cenários', 'Flexibilidade de Execução' e 'Detalhes de Execução', abrangendo um conjunto de fatores que foram tidos em conta no processo de comparação das ferramentas:

Definição da métrica 'Usabilidade'

Fator	Definição
Ambiente Gráfico	A ferramenta dispõe um ambiente gráfico (GUI) próprio, ou é parte integrante de outra plataforma
Suporte Técnico	Suporte técnico (ex. Online) para qualquer problema na utilização da ferramenta.
Nível Conhecimento Técnico	Nível de conhecimentos técnicos seja eles de programação ou configuração, para criação e execução de casos de teste
Documentação de Suporte	A ferramenta dispõe de documentação (ex. Online) com a terminologia usada, comandos, etc.

Definição da métrica 'Simplicidade de Instalação'

Fator	Definição
Ambiente de Instalação	Plataforma de instalação da ferramenta, para posterior utilização
Ambiente de Desenvolvimento	Dependência da ferramenta a um ambiente de desenvolvimento Android (SDK)

Tipo de Instalação	Tipo de instalação realizada para tornar a ferramenta totalmente operacional
Tutorial de Instalação	Documentação de suporte à instalação (tutorial)

Definição da métrica 'Complexidade de Criação de Cenários'

Fator	Definição
Criação de Cenários	Métodos utilizados na criação de cenários de teste
Gravação / Reprodução de Cenários	Minimização do tempo alocado à criação dos *scripts* de teste, incluído a possibilita de criação automática de cenários de teste, sem que seja necessário alocar recursos específicos para a criação dos *scripts*
Conhecimentos de Programação	Necessidade dos responsáveis pela criação de testes, possuírem ou não conhecimentos de programação

Definição da métrica 'Flexibilidade de Execução'

Fator	Definição
Ambiente de Execução	Capacidade de execução sobre dispositivos Android (local/remoto)
Instrumentação	Necessidade de instrumentação de código-fonte para execução dos testes pretendidos
Interoperabilidade	*Scripts* de testes com suporte multiplataforma (ex. Android e iOS)

Definição da métrica 'Detalhes de Execução'

Fator	Definição
Produção de Relatórios de Execução	Produção automática de relatórios detalhados com informação de cada teste executado.
Capturas de Estado	Capacidade de aumentar o nível de detalhe de execução, através de capturas de ecrã durante a execução

B.2.2 Resultados da Avaliação

Os resultados obtidos através da avaliação das ferramentas/plataformas de teste, segundo as métricas descritas na secção anterior, serão detalhadas abaixo:

Usabilidade: No que diz respeito ao fator descrito como 'Ambiente Gráfico', das ferramentas em estudo, apenas a ferramenta *MonkeyTalk* apresenta um ambiente gráfico próprio, sendo mais fácil a sua utilização sem qualquer dependência de plataformas secundárias, tais como ambientes de desenvolvimento específicos (ex. Eclipse). Para o 'Suporte Técnico', todas as plataformas não-integrantes no Android SDK, apresentam bom suporte técnico, ou seja, apenas *Activity Instrumentation* e *monkeyrunner* não apresentam um suporte técnico direto. Em termos de conhecimentos técnicos para utilização das ferramentas é possível afirmar que apenas duas, *MonkeyTalk* e *Calabash*, não requerem avançados conhecimentos técnicos de programação ou configuração, para

que seja possível a criação e execução de casos de teste. Em termos de 'Documentação de Suporte', é possível verificar que atualmente as ferramentas estão bem documentadas *online*, sendo o nível de documentação existente proporcional ao nível de maturidade da ferramenta.

Fator	***(AI)***	***(MR)***	***(RB)***	***(MT)***	***(CL)***
Ambiente Gráfico	Não	Não	Não	Sim	Não
Suporte Técnico	Não	Não	Sim	Sim	Sim
Nível Conhecimento Técnico	Alto	Médio	Alto	Baixo	Baixo
Documentação de Suporte	Sim	Sim	Sim	Sim	Sim

Simplicidade de Instalação: Em termos de 'Ambiente de Instalação', atualmente todas as ferramentas analisadas, possui versões para as plataformas Windows, Mac e Linux. No fator 'Ambiente de Desenvolvimento', verifica-se que a única ferramenta que não requer um ambiente de desenvolvimento para a sua utilização é a ferramenta MonkeyTalk. No que diz respeito ao processo de instalação foram analisados dois fatores em paralelo, o 'Tipo de Instalação' e 'Tutorial de Instalação'. No caso do primeiro fator, as ferramentas MonkeyTalk e Calabash apresentam uma instalação própria, independente de qualquer outra plataforma e sem necessidade de acesso ao código-fonte das aplicações alvo. A ferramenta Robotium é composta por um ficheiro binário que deverá ser incluído num projeto de teste com acesso ao código-fonte de cada aplicação alvo. Das ferramentas Activity Instrumentation e monkeyrunner, apenas a primeira requer acesso ao código-fonte das aplicações. No que diz respeito ao segundo dos dois fatores analisados no processo de instalação - 'Tutorial de Instalação' - foi possível constatar que apenas as ferramentas Robotium, MonkeyTalk e Calabash apresentam algum nível de suporte para o processo de instalação, assim como disponibilização de esclarecimento de dúvidas/problemas que possam surgir no decorrer de todo o processo.

Fator	***(AI)***	***(MR)***	***(RB)***	***(MT)***	***(CL)***
Ambiente de Instalação	Windows/Mac/Linux	Windows/Mac/Linux	Windows/Mac/Linux	Windows/Mac/Linux	Windows/Mac/Linux
Ambiente de Desenvolvimento	Sim	Sim	Sim	Não	Sim
Tipo de Instalação	Integrada no	Integrada no	Inclusão de	Executável	*Plugin*

	SDK	SDK	Binário		(Cumcumber)
Tutorial de Instalação	Não	Não	Sim	Sim	Sim

Complexidade de Criação de Cenários: Perante o fator 'Criação de Cenários', foi possível constatar que a maior parte das ferramentas apresentadas é baseada na criação de scripts. Estes scripts podem ser produzidos sem qualquer conhecimento do código-fonte das aplicações alvo, as exceções são Activity Instrumentation e Robotium que requerem avançados conhecimentos da plataforma Android assim como conhecimentos mínimos do código-fonte da aplicação alvo de testes. Para o fator 'Gravação / Reprodução de Cenários', apenas a ferramenta MonkeyTalk apresenta a possibilidade de gravação de cenários de teste, para serem reproduzidos como testes funcionais em execuções posteriores. Em termos de 'Conhecimentos de Programação', foi possível constatar que na sua maioria, as ferramentas necessitam de conhecimentos em programação, tais como, Java, Pyton, Javascript ou Ruby. Apenas com a ferramenta Calabash existe a possibilidade de criar testes funcionais, sem qualquer conhecimento em programação, isto porque, os scripts são baseados em linguagem natural, não havendo qualquer dependência a uma linguagem de programação.

Fator	***(AI)***	***(MR)***	***(RB)***	***(MT)***	***(CL)***
Criação de Cenários	Testes unitários	*Scripts*	Testes unitários	*Scripts*	*Scripts/Linguag em Natural*
Gravação / Reprodução de Cenários	Não	Não	Não	Sim	Não
Conhecimentos de Pro-gramação	Obrigatório (Java)	Obrigatório (Pyton)	Obrigatório (Java)	Opcional (Javascript)	Opcional (Ruby)

Flexibilidade de Execução: Em termos de 'Ambiente de Execução', todas as ferramentas analisadas suportam a execução local, isto é, a execução dos testes é feita sobre dispositivos físicos conetados fisicamente à ferramenta de teste; a abordagem remota, isto é, dispositivos conectados à ferramenta de teste em rede através de uma rede local ou através da Internet, é atualmente suportado pelas ferramentas monkeyrunner, MonkeyTalk e Calabash, sendo que as duas últimas apresentam suporte a serviços externos, com capacidade de seleção de dispositivos de testes (*smartphones* e/ou *tablets*) localizados em serviços *cloud*. O fator 'Instrumentação' verifica-se que apenas a ferramenta MonkeyTalk, que requer obrigatoriamente a instrumentação do código-fonte devido à necessidade de incluir um agente, que irá estabelecer uma ligação entre a

ferramenta e a aplicação, durante a execução da mesma. Em relação ao fator 'Interoperabilidade', apenas as ferramentas MonkeyTalk e Calabash permitem a execução de scripts de teste em diferentes plataformas, havendo nas restantes ferramentas analisadas uma total dependência à arquitetura Android, sendo os seus scripts totalmente adaptados à mesma.

Fator	***(AI)***	***(MR)***	***(RB)***	***(MT)***	***(CL)***
Ambiente de Execução	Local	Local/Remoto	Local	Local/Remoto	Local/Remoto
Instrumentação	Opcional	Não	Opcional	Obrigatório	Não
Interoperabilidade	Não	Não	Não	Sim	Sim

Detalhes de Execução: A 'Produção de Relatórios de Execução' é atualmente suportada de forma automática pelas ferramentas MonkeyTalk e Calabash, produzindo automaticamente relatórios detalhados com informação de cada teste executado. As restantes ferramentas apesar de também possuírem suporte para produção de relatórios, requerem uma configuração mais avançada de ferramentas secundárias. As 'Capturas de Estado' são atualmente suportadas pelas ferramentas monkeyrunner, Robotium, MonkeyTalk e Calabash, sendo possível através de configuração automatizar as capturas quando os cenários de testes falham a sua execução. No caso das ferramentas MonkeyTalk e Calabash as capturas de ecrã em caso de falha são automaticamente incluídas nos relatórios produzidos no final das execuções.

Fator	***(AI)***	***(MR)***	***(RB)***	***(MT)***	***(CL)***
Produção de Relatórios de Execução	Não	Não	Não	Sim	Sim
Capturas de Estado	Não	Sim	Sim	Sim	Sim