

Mestrado em Engenharia Informática  
Estágio  
Relatório Final

# Implementação de uma solução para melhoria da segurança rodoviária na utilização de aplicações de messaging

João David Messias Simões Magalhães  
jdmessias@student.dei.uc.pt

Orientador DEI:  
Vasco Pereira

Orientador WIT Software:  
Frederico Lopes

1 de Julho de 2016



**FCTUC** DEPARTAMENTO  
**DE ENGENHARIA INFORMÁTICA**  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA



FCTUC

**Departamento de Engenharia Informática**  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra  
Pólo II, Pinhal de Marrocos, 3030-290 Coimbra

Tel: +351239700000 | Fax: +351239701266 | [info@dei.uc.pt](mailto:info@dei.uc.pt)



**WIT Software, S.A.**  
Centro de Empresas de Taveiro  
Estrada de Condeixa, 3045-508 Taveiro, Coimbra

Tel: +351239801030 | Fax: +351239801039 | [info@wit-software.com](mailto:info@wit-software.com)

ESTAGIÁRIO:

**João David Messias Simões Magalhães**  
[jdmessias@student.dei.uc.pt](mailto:jdmessias@student.dei.uc.pt), [joao.magalhaes@wit-software.com](mailto:joao.magalhaes@wit-software.com)

JÚRIS:

**César Alexandre Domingues Teixeira**  
[cteixeir@dei.uc.pt](mailto:cteixeir@dei.uc.pt)

**Marília Pascoal Curado**  
[marilia@dei.uc.pt](mailto:marilia@dei.uc.pt)

ORIENTADOR DEI:

**Vasco Pereira**  
[vasco@dei.uc.pt](mailto:vasco@dei.uc.pt)

ORIENTADOR WIT SOFTWARE:

**Frederico Lopes**  
[frederico.lopes@wit-software.com](mailto:frederico.lopes@wit-software.com)

## Resumo

Na última década assistimos a uma rápida evolução no mundo das comunicações, os telefones móveis passaram de meros dispositivos capazes de fazer chamadas de voz e enviar mensagens a ferramenta importante na gestão diária das pessoas através do acesso global à *internet* e às aplicações móveis.

Deste crescimento resultou o aparecimento de várias aplicações de mensagens sobre IP, designadas como *Over-The-Top* (OTT), que competem com as tradicionais comunicações por SMS e MMS dos operadores de telecomunicações. Para se diferenciarem e combaterem a popularidade das OTT, as operadoras, através da GSMA, criaram a especificação *Rich Communication Services* que oferece serviços sobre IP semelhantes às aplicações OTT. O produto RCS+ da WIT Software é uma das aplicações que implementa esta especificação.

Não sendo um problema recente, a explosão do número de dispositivos móveis trouxe um aumento de acidentes na estrada, resultado da distração pelo uso do telemóvel. A tentativa de criar opções de comunicações mais seguras para condutores tem vindo a revelar-se infrutífera face ao número de vítimas registadas nos últimos anos.

O objetivo deste estágio passa por dar um contributo para a resolução deste problema, assim como por valorizar a aplicação *Android RCS+*, acrescentando um conjunto de funcionalidades que permitem uma condução segura, em paralelo com a possibilidade de comunicação.

## Palavras-Chave

*Rich Communication Services, Smartphone, Android, Segurança Rodoviária, Messaging, Comunicação, Dispositivos Móveis.*



## **Abstract**

In the last decade we witnessed a fast progress in the world of communications, mobile phones are no longer limited to making voice calls and sending messages, they become an important tool in people's lives daily management through global internet access and mobile applications.

From this growth resulted the emergence of various message over IP (Internet Protocol) applications, known as Over-The-Top (OTT), which compete with traditional SMS (Short Message Service) and MMS (Multimedia Messaging Service) communications provided by telecommunications operators. To innovate and fight the popularity of OTT, operators, through the GSMA (Groupe Spécial Mobile Association), created the Rich Communication Services specification that offers services over IP similar to OTT applications. The WIT Software product, RCS+, is one of the applications that implement this specification.

Despite not being a recent problem, the explosion in the number of mobile devices brought an increase in traffic accidents, consequence of the distractions by using cell phones while driving. Attempts to create secure communications options to use while driving has been proving to be unfruitful, confirmed by the number of victims recorded in recent years.

The goal of this internship is to contribute to the resolution of this problem and the enhancement of the Android RCS+ application, adding a set of features that allow safe driving and at the same time preserve the possibility of communication.

## **Keywords**

Rich Communication Services, Smartphone, Android, Road Safety, Messaging, Communication, Mobile Devices.



## **Agradecimentos**

Em primeiro lugar quero agradecer à minha família, em especial aos meus pais e irmãos, pelo apoio e paciência que demonstraram ao longo de toda a minha vida.

Aos meus orientadores, Frederico Lopes e Vasco Pereira, pelo saber, competência e disponibilidade. Ao João Lopes pela sua partilha de conhecimentos e à Sandrina Silva que deu cor a este projeto.

A todos os meus amigos, um profundo obrigado pela amizade e força que me deram ao longo deste tempo.

Por fim, quero agradecer à WIT Software por me ter dado a oportunidade de realizar este desafiante estágio.





# Índice

|   |           |
|---|-----------|
| <b>Capítulo 1 Introdução</b> .....                          | <b>1</b>  |
| <b>1.1. Contexto</b> .....                                  | <b>1</b>  |
| 1.1.1. Mundo das Telecomunicações .....                     | 1         |
| 1.1.2. Comunicação em condução.....                         | 2         |
| <b>1.2. Objetivos</b> .....                                 | <b>3</b>  |
| <b>1.3. Empresa</b> .....                                   | <b>3</b>  |
| <b>1.4. Planeamento</b> .....                               | <b>3</b>  |
| <b>1.5. Estrutura do Documento</b> .....                    | <b>5</b>  |
| <b>Capítulo 2 Estado da Arte</b> .....                      | <b>7</b>  |
| <b>2.1. Competidores</b> .....                              | <b>7</b>  |
| 2.1.1. Competidores Diretos .....                           | 7         |
| 2.1.2. Competidores Indiretos.....                          | 18        |
| 2.1.3. Análise Comparativa .....                            | 20        |
| <b>2.2. WIT RCS Suite</b> .....                             | <b>21</b> |
| <b>2.3. Mecanismos Text-To-Speech</b> .....                 | <b>23</b> |
| 2.3.1. IVONA Text-to-Speech HQ .....                        | 23        |
| 2.3.2. SVOX .....   | 23        |
| 2.3.3. Google Text-To-Speech .....                          | 24        |
| <b>2.4. Mecanismos de Reconhecimento de Atividade</b> ..... | <b>24</b> |
| 2.4.1. Google Activity Recognition API.....                 | 24        |
| 2.4.2. PathSense Activity Recognition .....                 | 24        |
| <b>2.5. Conceitos Android</b> .....                         | <b>25</b> |
| <b>Capítulo 3 Abordagem</b> .....                           | <b>27</b> |
| <b>3.1. Scrum vs Waterfall</b> .....                        | <b>27</b> |
| <b>3.2. Metodologia Scrum</b> .....                         | <b>28</b> |
| 3.2.1. Cargos.....  | 29        |
| 3.2.2. User Stories - Requisitos .....                      | 29        |
| 3.2.3. Definition of Done .....                             | 29        |
| 3.2.4. Product Backlog .....                                | 30        |
| 3.2.5. Sprints .....  | 31        |
| 3.2.6. Reunião de Planeamento de Sprint .....               | 34        |
| 3.2.7. Reunião Diária .....                                 | 34        |
| <b>3.3. Análise de Riscos</b> .....                         | <b>34</b> |
| <b>Capítulo 4 Arquitetura e Desenvolvimento</b> .....       | <b>37</b> |
| <b>4.1. WIT Mobile Communicator</b> .....                   | <b>38</b> |
| <b>4.2. Arquitetura Android RCS+</b> .....                  | <b>40</b> |
| 4.2.1. Módulos .....  | 40        |
| 4.2.2. Estrutura dos Módulos.....                           | 41        |
| 4.2.3. Comunicação com o RCS Stack.....                     | 42        |
| <b>4.3. Módulo de Condução</b> .....                        | <b>43</b> |
| 4.3.1. Camada de Apresentação .....                         | 43        |
| 4.3.2. Camada Lógica.....                                   | 44        |
| 4.3.3. Camada de Dados .....                                | 44        |
| <b>4.4. Funcionalidades do Módulo</b> .....                 | <b>45</b> |
| 4.4.1. Gestão de mensagens e ficheiros de áudio.....        | 45        |
| 4.4.2. Text-To-Speech .....                                 | 45        |
| 4.4.3. Detecção de Condução via Sensores.....               | 45        |
| 4.4.4. Detecção de Condução via Bluetooth .....             | 46        |
| 4.4.5. Floating Button.....                                 | 47        |
| 4.4.6. Perceção de Condução .....                           | 47        |
| 4.4.7. Opções de envio a condutor .....                     | 47        |

|  |           |
|--|-----------|
| 4.4.8. Resposta com mensagem de áudio.....       | 48        |
| 4.4.9. Gestão de dispositivos mãos-livres .....  | 48        |
| 4.4.10. Serviço de Notificações.....             | 49        |
| 4.4.11. Notificação de Início de Condução.....   | 49        |
| 4.4.12. Pontuações (Gamification).....           | 49        |
| 4.4.13. Iniciar Chamada em Condução.....         | 50        |
| 4.4.14. Indicador de Bateria.....                | 50        |
| <b>Capítulo 5 Análise de Condução .....</b>      | <b>51</b> |
| 5.1. Recolha de Dados.....                       | 52        |
| 5.2. Análise de Dados.....                       | 53        |
| 5.3. Conclusão.....                              | 57        |
| <b>Capítulo 6 Testes .....</b>                   | <b>59</b> |
| 6.1. Testes Funcionais – Protótipo.....          | 59        |
| 6.2. Teste Unitário - Algoritmo de Detecção..... | 61        |
| <b>Capítulo 7 Conclusões .....</b>               | <b>63</b> |
| 7.1. Avaliação Final .....                       | 63        |
| 7.2. Trabalho Futuro .....                       | 64        |
| <b>Referências .....</b>                         | <b>65</b> |

## Índice de Figuras

|   |    |
|---|----|
| Figura 1 - Adoção RCS .....                                       | 2  |
| Figura 2 - AT&T Logótipo.....                                     | 7  |
| Figura 3 - Car Home Ultra Logótipo .....                          | 8  |
| Figura 4 - Drive Safe.ly Logótipo .....                           | 9  |
| Figura 5 - Safe Driving Text Machine Logótipo .....               | 9  |
| Figura 6 - Drivemode Logótipo.....                                | 10 |
| Figura 7 - text-Star Logótipo.....                                | 11 |
| Figura 8 - SafeDrive rewards you .....                            | 11 |
| Figura 9 - TextDrive Logótipo .....                               | 12 |
| Figura 10 - Sprint Drive First Logótipo .....                     | 12 |
| Figura 11 - Drive Mode Logótipo .....                             | 13 |
| Figura 12 - Car Dashdroid Logótipo .....                          | 13 |
| Figura 13 - Safely Go Logótipo .....                              | 14 |
| Figura 14 - DriveSafe Mode Logótipo.....                          | 14 |
| Figura 15 - Drive Alive Lite Logótipo .....                       | 15 |
| Figura 16 - Ultimate Car Dock Logótipo .....                      | 15 |
| Figura 17 - LifeSaver Logótipo .....                              | 16 |
| Figura 18 - Dashdroid Logótipo.....                               | 16 |
| Figura 19 - Automate Logótipo .....                               | 17 |
| Figura 20 - Windows Phone Driving Mode Menu .....                 | 18 |
| Figura 21 - Android Auto Logótipo.....                            | 18 |
| Figura 22 - Cell Control Logótipo.....                            | 19 |
| Figura 23 - Apple Carplay Logótipo .....                          | 19 |
| Figura 24 - WIT RCS Suite.....                                    | 21 |
| Figura 25 - IVONA TTS Logótipo.....                               | 23 |
| Figura 26 - SVOX TTS Logótipo.....                                | 23 |
| Figura 27 - Google TTS Logótipo .....                             | 24 |
| Figura 28 - PathSense Logótipo.....                               | 24 |
| Figura 29 - Processo de Desenvolvimento Scrum .....               | 28 |
| Figura 30 - WIT RCS na infraestruturas IMS.....                   | 38 |
| Figura 31 - Arquitetura alto nível do WIT RCS Stack.....          | 39 |
| Figura 32 - Principais Módulos RCS+ .....                         | 40 |
| Figura 33 - Diagrama de Camadas Módulos.....                      | 41 |
| Figura 34 – Ciclo de vida compacto de uma Activity.....           | 42 |
| Figura 35 - Menu Principal da aplicação de recolha de dados ..... | 52 |

## Índice de Tabelas

|  |    |
|--|----|
| Tabela 1 - Comparação de Funcionalidades entre Competidores Diretos.....             | 20 |
| Tabela 2 - Sub-conjunto do Product Backlog.....                                      | 30 |
| Tabela 3 - Excerto de um ficheiro de dados.....                                      | 53 |
| Tabela 4 - Estatísticas de Dados recolhidos.....                                     | 53 |
| Tabela 5 - Exemplo de resultados do programa de análise de dados.....                | 54 |
| Tabela 6 - Atividades recolhidas pela API da Google entre 20 e 40% de confiança..... | 54 |
| Tabela 7 - Atividades recolhidas pela API da Google entre 50 e 70% de confiança..... | 55 |
| Tabela 8 - Atividades recolhidas pela PathSense API entre 20 e 40% de confiança..... | 56 |
| Tabela 9 - Atividades recolhidas pela PathSense API entre 50 e 70% de confiança..... | 56 |
| Tabela 10 - Testes Funcionais do Protótipo.....                                      | 60 |
| Tabela 11 - Resultado teste unitário ao algoritmo de deteção.....                    | 61 |

## Glossário

|                           |   |
|---------------------------|---|
| <b>API</b>                | Uma <i>Application User Interface</i> é um conjunto de métodos ou pontos de acesso que um <i>software</i> fornece para que terceiros possam utilizar as suas funcionalidades sem conhecerem os seus processos internos. |
| <b>DoD</b>                | O <i>Definition of Done</i> é um documento que especifica os requisitos que devem ser concluídos para que uma <i>User Story</i> seja dada como concluída.   |
| <b>GSMA</b>               | A <i>GSM Association</i> é um grupo de operadores de telecomunicações e empresas relacionadas, criada em 1995, cujo principal objetivo é a especificação de padrões para sistemas de comunicação móveis.                |
| <b>OTT</b>                | O termo <i>Over-the-top</i> refere-se às aplicações que fornecem conteúdos multimédia pela Internet.  |
| <b>Product Backlog</b>    | O <i>Product Backlog</i> é o repositório de funcionalidades, <i>bugs</i> ou melhorias, na forma de <i>User Stories</i> de um projeto de <i>Scrum</i> .  |
| <b>Product Owner</b>      | O <i>Product Owner</i> é a pessoa responsável no processo de <i>Scrum</i> pela gestão do <i>Product Backlog</i> .   |
| <b>RCS</b>                | <i>Rich Communication Services</i> é uma iniciativa da GSMA para inovar os serviços dos operadores de telecomunicação com o objetivo de competir com as funcionalidades das aplicações OTT.                             |
| <b>Scrum</b>              | <i>Scrum</i> é uma <i>framework</i> de desenvolvimento ágil.  |
| <b>Sprint</b>             | <i>Sprint</i> é o período de tempo em que uma equipa de desenvolvimento <i>Scrum</i> desenvolve um determinado número de tarefas, <i>User Stories</i> , definidas no <i>Sprint Backlog</i> .                            |
| <b>Velocidade (Scrum)</b> | A velocidade de uma equipa de desenvolvimento <i>Scrum</i> é a soma dos pontos concluídos num <i>Sprint</i> .   |

## Acrónimos

|             |                                   |
|-------------|-----------------------------------|
| <b>API</b>  | Application User Interface        |
| <b>APK</b>  | Android Package                   |
| <b>CMS</b>  | Central Message Store             |
| <b>DoD</b>  | Definition of Done                |
| <b>GPS</b>  | Global Positioning System         |
| <b>GSMA</b> | Groupe Speciale Mobile Associaton |
| <b>HTTP</b> | Hypertext Transfer Protocol       |
| <b>IM</b>   | Instant Messaging                 |
| <b>IMS</b>  | IP Multimedia Subsystem           |
| <b>IP</b>   | Internet Protocol                 |
| <b>IPTV</b> | Internet Protocol Television      |
| <b>MMS</b>  | Multimedia Message Service        |
| <b>MSRP</b> | Message Session Relay Protocol    |
| <b>NAB</b>  | Network Address Book              |
| <b>OTT</b>  | Over The Top                      |
| <b>RCS</b>  | Rich Communication Services       |
| <b>SBC</b>  | Session Border Control            |
| <b>SDK</b>  | Software Development Kip          |
| <b>SIP</b>  | Session Initiation Protocol       |
| <b>SMS</b>  | Short Message Service             |
| <b>UI</b>   | User Interface                    |
| <b>VoIP</b> | Voice over IP                     |
| <b>WCAS</b> | WIT Communications Server         |
| <b>WCS</b>  | WIT Communications Suite          |
| <b>WMC</b>  | WIT Mobile Communicator           |
| <b>XML</b>  | EXtensible Markup Language        |

# Capítulo 1

## Introdução

Com este documento pretende-se dar conta do trabalho realizado na empresa WIT Software no âmbito do estágio curricular do Mestrado em Engenharia Informática da Universidade de Coimbra.

O trabalho, ora apresentado, foi orientado na empresa pelo Engenheiro Frederico Lopes e no departamento de engenharia informática pelo Professor Vasco Pereira. O acompanhamento do estagiário em questões técnicas deveu-se ao Mestre João Lopes, tutor atribuído por parte da WIT Software.

O estágio decorreu nas instalações da WIT Software em Coimbra entre Setembro de 2015 e Junho de 2016.

### 1.1. Contexto

Por forma a melhor compreender o âmbito do estágio é necessário explorar duas vertentes atuais do mundo das comunicações, a primeira é a competição entre aplicações OTT (*Over-The-Top*) e os operadores de telecomunicações, onde está inserido o produto base sobre o qual o estágio foi aplicado, a aplicação RCS+. Por outro lado, é necessário entender a questão atual da comunicação via *smartphone* na condução, bem como os problemas que daí resultam e que se pretendem minimizar com a solução final deste trabalho.

#### 1.1.1. Mundo das Telecomunicações

Hoje em dia existem cerca de 2,6 biliões de utilizadores de *smartphones* em todo o mundo, e as previsões indicam que até 2020 esse número aumente para cerca de 6,1 biliões. É assim esperado que 70% da população do mundo utilize *smartphone* em 2020 [1].

A popularização destes dispositivos trouxe novas formas de comunicação sobre IP através das aplicações *Over-The-Top*, como *Skype* [2], *WhatsApp* [3], *Viber* [4] entre outras, que utilizam as infraestruturas dos operadores de telecomunicação (rede de dados) para fornecer aos seus utilizadores acesso rápido e maioritariamente gratuito a ferramentas de comunicação.

Os operadores de telecomunicações entendem estas aplicações como um enorme risco para as suas receitas. De acordo com os analistas, a indústria de telecomunicações deverá perder, entre 2012 e 2018, cerca de 386 biliões de dólares para aplicações OTT [5].

O combate a esta tendência de perdas exige que os operadores de telecomunicações inovem, para isso a *GSM Association* [6], da qual a WIT Software faz parte desde 2011, criou o programa *Rich Communication Services* (RCS) que permite aos operadores enriquecer os seus serviços de comunicação para além das tradicionais mensagens SMS e chamadas de voz, com serviços de mensagens e voz sobre IP, mensagens em grupo, transferência de ficheiros, localização ou presença.

A Figura 1 traduz a adoção atual da especificação RCS por parte dos operadores. Nela identifica-se uma forte difusão pelo mercado Europeu e Americano.

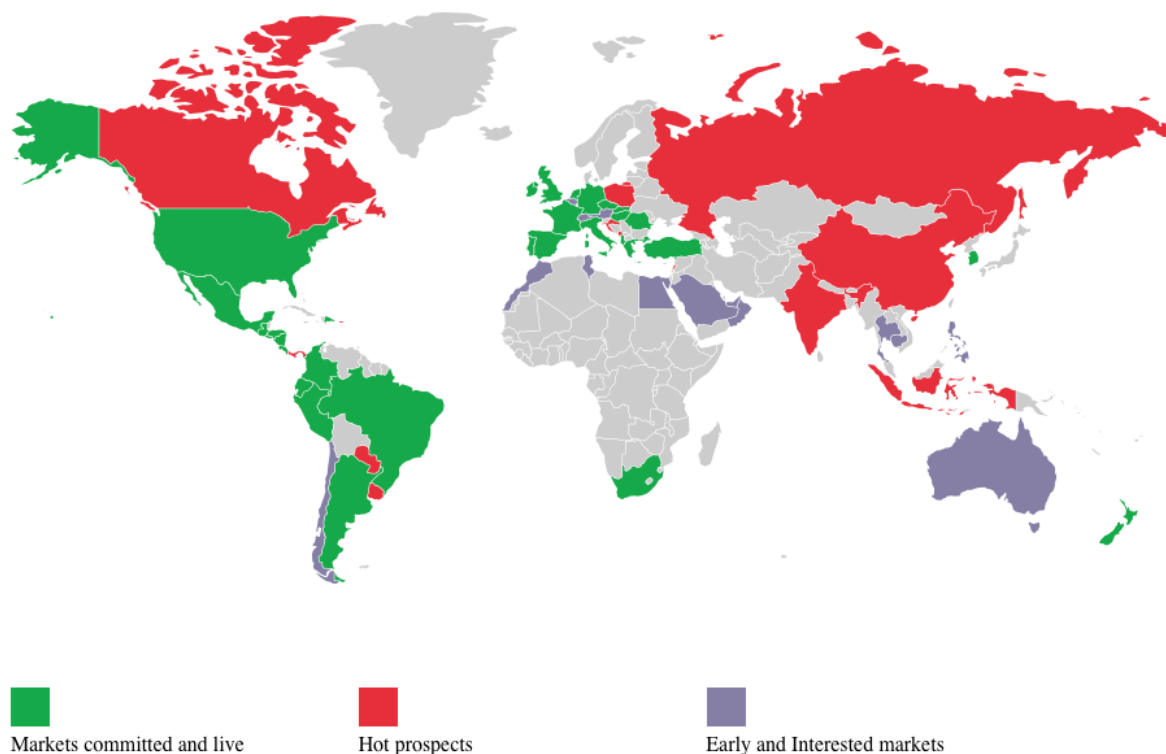


Figura 1 - Adoção RCS [7]

A aplicação RCS+ da WIT Software implementa a especificação RCS criada pela GSMA e faz parte das soluções criadas para os operadores combaterem as OTT pelo que é importante aumentar o seu valor com funcionalidades inovadoras. Espera-se que o resultado deste estágio se traduza num forte contributo para o valor da aplicação, com a adição de um novo módulo numa área pouco explorada pelos operadores e desenvolvedores de aplicações OTT.

### 1.1.2. Comunicação em condução

O aumento do número de *smartphones* origina um acréscimo do número de pessoas que o utiliza enquanto conduz. As estatísticas são alarmantes e indicam que é necessário abordar o problema de uma forma diferente da atual.

De acordo com o *National Safety Council* [8] ocorrem cerca de 1.600.000 acidentes por ano devido ao envio de mensagens enquanto se conduz. Para o *Harvard Center for Risk Analysis Studies* [9] acontecem 330.000 lesões por ano e cerca de 11 mortes todos os dias. A comunicação por *smartphone* durante a condução representa 25% do total de acidentes na estrada [10].

Os números são preocupantes e as campanhas de prevenção não têm conseguido reduzir suficientemente este problema, pelo que a solução poderá passar, não por cortar completamente a comunicação, mas por criar uma forma de esta ser feita em segurança.



A WIT Software, sensível a este problema, considerou ser importante investir no seu produto RCS+, criando uma solução para a condução segura dentro da sua aplicação de mensagens, aplicação esta inexistente em plataformas RCS concorrentes.

## 1.2. Objetivos

Para a WIT Software o objetivo deste estágio passa por explorar o segmento da comunicação na condução, de forma a acrescentar valor ao seu produto RCS+ através da adição de funcionalidades que permitem a utilização da aplicação em condução. Fazendo também um investimento na aprendizagem do estagiário e no seu desenvolvimento profissional.

Por sua vez, para o estagiário esta é uma oportunidade de pôr em prática e consolidar os ensinamentos que foi adquirindo ao longo da licenciatura e mestrado em engenharia informática, assim como ganhar prática no mundo profissional, integrando uma empresa experiente e de sucesso na sua área.

## 1.3. Empresa

A WIT Software foi fundada em 2001 como *spin-off* da Universidade de Coimbra. Atualmente a empresa conta com escritórios em Coimbra onde decorre este estágio, Porto, Lisboa, Leiria, Dusseldorf, Reading e Silicon Valley, tendo mais de 250 colaboradores.

A empresa desenvolve soluções de *software* na área das telecomunicações com foco na especificação RCS e redes IMS (*IP Multimedia Subsystem*), para dispositivos móveis predominantemente para os sistemas operativos Android, iOS, Windows Phone e para a área da televisão por cabo e operadores IPTV (*Internet Protocol Television*).

## 1.4. Planeamento

Um planeamento mais detalhado do estágio pode ser consultado no **Capítulo 3 – Abordagem e Apêndice A – Abordagem**, sendo abaixo apresentado um sumário do plano.

### Primeiro Semestre

O plano do primeiro semestre, realizado entre Setembro de 2015 e Janeiro de 2016, incluiu as funcionalidades de gestão de mensagens e chamadas no módulo de condução, bem como a perceção de condução por parte dos utilizadores da aplicação RCS+. Estando planeadas, foram concluídas as seguintes funcionalidades:

Módulo de condução:

- Receção de SMS e mensagens de *chat*;
- Leitura de mensagens através de mecanismo de *text-to-speech*;
- Navegação pelas mensagens recebidas;
- Resposta a mensagens através da clíp de áudio;
- Resposta a mensagem através de chamada de voz;
- Detecção de mecanismos de mãos-livres.
- Detecção e bloqueio de notificações.

#### Perceção de condução:

- Conhecimento se um contacto está a conduzir;
- Envio de mensagens a utilizadores em condução;
- Início de chamada com utilizador em condução, caso exista um sistema mãos-livres;
- Adiantamento de envio de mensagem até terminar condução;
- Receção de notificação após o utilizador terminar condução;
- Receção de notificação quando um contacto favorito inicia o modo de condução.

### Segundo Semestre

No segundo semestre, que decorreu entre Fevereiro e Junho de 2016 foram planeadas: a implementação de um mecanismo de jogos (*gamification*), a deteção de condução automática através de sensores do *smartphone* ou ligações *bluetooth* e diversas funcionalidades para aumentar a qualidade do protótipo. Assim, as tarefas planeadas e realizadas foram:

#### Funcionalidades Gerais:

- Modo de leitura de mensagens sequencial;
- Menu de início de chamadas a partir dos menus de condução;
- Acesso rápido a número de emergência;
- Receção e audição de mensagens de áudio;
- Possibilidade de envio de mensagens de áudio a condutores;
- Indicador de bateria no menu de condução;
- *Floating Button* para acesso rápido aos menus de condução;
- Apresentação de notificações recebidas em condução;

#### *Gamification:*

- Implementação de um sistema de distribuição de pontos;
- Gestão da comunicação para partilha de pontos entre utilizadores;
- Menu de apresentação de pontuações.

#### Deteção de Condução:

- Gestão e configuração de dispositivos *bluetooth* para início de condução;
- Estudo de mecanismos de reconhecimento de atividade;
- Algoritmo de deteção de condução com base em reconhecimento de atividade.

## 1.5. Estrutura do Documento

O relatório principal está dividido em sete capítulos. A Introdução serve o intuito de contextualizar o estágio e dar uma visão geral do resultado final esperado. Os restantes capítulos são:

- **2. Estado da Arte**  
Análise de aplicações e outros sistemas, cujo propósito principal é, tal como o protótipo deste estágio, a segurança na condução com foco na comunicação por *smartphone*. Nesta análise estudaram-se as funcionalidades principais, bem como aspetos positivos e negativos. Para além do estudo de aplicações, foi explorada a aplicação RCS+ da WIT Software, onde o módulo a ser criado está inserido. Foram também estudados diferentes mecanismos de *text-to-speech* disponíveis para a plataforma *Android*, assim como mecanismos de reconhecimento de atividade. Por fim são apresentados conceitos e estruturas *Android* relevantes para o entendimento da arquitetura da solução.
- **3. Abordagem**  
Secção onde se descreve a metodologia utilizada durante o decorrer do estágio e o modo como esta foi aplicada no desenvolvimento da solução.
- **4. Arquitetura e Desenvolvimento**  
Explicação da organização arquitetural do *WIT Mobile Communicator*, da aplicação RCS+ e do módulo desenvolvido. Sendo a arquitetura do protótipo classificado como confidencial pela empresa a informação mais relevante e detalhada será colocada no Apêndice B – Arquitetura e Desenvolvimento, pelo que se recomenda a sua leitura caso se pretenda obter conhecimento mais específico da arquitetura final. Em cada funcionalidade é também abordada a forma como foi desenvolvida indicando problemas encontrados, se foi caso disso.
- **5. Análise de Condução**  
Neste capítulo é explicado o estudo efetuado para obter parâmetros ideais a utilizar no algoritmo de deteção de condução, assim como a comparação entre os mecanismos de reconhecimento de atividade disponíveis para *Android*.
- **6. Testes**  
Apresentação dos testes funcionais realizados sobre o protótipo e dos testes realizados sobre o algoritmo de deteção de condução.
- **7. Conclusões**  
Este último capítulo apresenta uma reflexão final do estagiário em relação ao trabalho desenvolvido, as contribuições dadas ao longo de todo o estágio e as possíveis expansões que o trabalho poderá ainda ser alvo no futuro.

De forma a completar alguns capítulos, foram criados quatro apêndices contendo informação adicional ou confidencial que não pode constar do relatório principal. A sua leitura é altamente recomendada.

- **Apêndice A – Abordagem**  
Análise de requisitos que o protótipo deve implementar, incluindo atributos como complexidade e prioridade para cada funcionalidade. Revisão dos *sprints* efetuados durante o estágio e apresentação do *Minimum Viable Prototype*.
- **Apêndice B – Arquitetura e Desenvolvimento**  
Informação detalhada e confidencial sobre a arquitetura do protótipo realizado.
- **Apêndice C – Testes Funcionais**  
Apresentação do conjunto de testes funcionais realizados para as diferentes funcionalidades do protótipo.
- **Apêndice D – Funcionalidades**  
Apresentação de todas as funcionalidades implementadas no protótipo final.

## Capítulo 2

### Estado da Arte

O conteúdo desta secção assenta na (i) exploração das tecnologias a utilizar durante o decorrer do estágio e (ii) no estudo dos competidores com o intuito de perceber quais os requisitos mais relevantes e onde se pode inovar na implementação da solução.

Na **secção 2.1 – Competidores** apresentam-se as aplicações cujo objetivo é solucionar o mesmo problema que o protótipo a executar no estágio, isto é, a utilização do *smartphone* durante a condução de forma segura. Foram recolhidas informações de dezoito aplicações *Android* e outros quatro sistemas diferentes.

A **secção 2.2 – Aplicação RCS+** serve para estudar a aplicação na qual se inseriu o trabalho desenvolvido neste estágio.

A **secção 2.3 – Mecanismos Text-To-Speech** incide em três diferentes mecanismos de *text-to-speech* existentes para a plataforma *Android*, tendo como objetivo a escolha do mais adequado a utilizar na solução final.

A **secção 2.4 – Mecanismos de Reconhecimento de Atividade** mostra os dois principais *softwares* de reconhecimento disponíveis para *Android*.

Por fim, na **secção 2.5 – Conceitos Android** são apresentadas as componentes da plataforma *Android* mais relevantes para o estágio.

#### 2.1. Competidores

O estudo e análise dos competidores é uma parte importante da criação de *software*, permitindo perceber as lacunas dos sistemas já existentes e construir uma solução com elementos inovadores. Para o presente estágio, os competidores são as aplicações ou sistemas que permitem a utilização do *smartphone* para comunicação de forma segura, em condução.

##### 2.1.1. Competidores Diretos

Esta secção fornece informações acerca das aplicações de empresas ou programadores independentes que oferecem mecanismos de condução segura para a nossa plataforma alvo, *Android*. Estes são os competidores mais relevantes para o estágio, dado que oferecem os mesmos ou serviços comparáveis.

##### AT&T Drivemode [11]



**Programador:** AT & T Services, Inc.  
**Play Store Rating:** 3.4 (5,029 ratings)  
**Última Atualização:** 2 de Novembro de 2015  
**Número de Downloads:** 10,000,000 – 50,000,000  
**Preço:** Gratuito

Figura 2 - AT&T Logótipo

AT&T Inc é uma empresa multinacional de telecomunicações, com sede em *Whitecrater Tower, Texas*. A empresa é um dos maiores fornecedores de telefones móveis e fixos nos Estados Unidos. Desde Outubro de 2015 é considerada a 27ª maior empresa do mundo [12], de acordo com a *Forbes*.

A aplicação *Android* da empresa, *AT&T Drivemode* tem o objetivo de ajudar a reduzir a distração dos seus utilizadores durante a condução. Para o garantir, a abordagem da AT&T, passa pela remoção da maior parte das interações com o telefone enquanto este está em modo condução.

### Funcionalidades Principais:

- Detecção automática de condução através de velocidade;
- Resposta automática a SMS e MMS com mensagens predefinidas;
- Silêncio automático de alertas recebidos;
- Controlo parental: Alerta outros utilizadores (pais) quando funcionalidades de segurança são desativadas;
- Atalho para acesso a navegação e música.

### Car Home Ultra [13]



Figura 3 - Car Home Ultra  
Logótipo

**Programador:** The SpinningHead

**Play Store Rating:** 4.2 (10,494 ratings)

**Última Atualização:** 24 de Dezembro de 2015

**Número de Downloads:** 1,000,000 – 5,000,000

**Preço:** Versão experimental 30 dias, com desbloqueio de funcionalidades por 2,98€.

Criada pela empresa TheSpinningHead, llc esta aplicação pretende fornecer mecanismos de segurança, enquanto os utilizadores têm acesso a funcionalidades essenciais do telemóvel. Apresenta uma interface semelhante a um *dashboard* de um carro onde o utilizador pode interagir com navegação e música, assim como fazer chamadas e aceder a informação (meteorologia, velocidade e altitude).

### Funcionalidades Principais

- Personalização de atalhos para outras aplicações do *smartphone*, acesso rápido a números ou navegação para localizações específicas;
- Controlos de *media* para gerir música ou *podcasts*;
- Resposta automática a SMS com mensagem predefinidas;
- Alertas de voz, através de *text-to-speech*, de eventos tais como excesso de velocidade.

## Drive Safe.ly Free SMS Reader [14]



**Programador:** iSpeech.org  
**Play Store Rating:** 3.9 (8,111 ratings)  
**Última Atualização:** 16 de Julho de 2014  
**Número de Downloads:** 1,000,000 – 5,000,000  
**Preço:** Versão gratuita e versão PRO 7,40€.

Figura 4 - Drive Safe.ly Logótipo

iSpeech é uma empresa com sede em *Newar, Nova Jérsia* que fornece *software* de alta qualidade na área de texto para voz e reconhecimento de voz para texto. De acordo com a empresa, a sua aplicação DriveSafe.ly foi já instalada mais de 35.000.000 de vezes nas plataformas *Android* e *Blackberry*.

A interface da aplicação é minimalista permitindo aos seus utilizadores a configuração de algumas definições de *text-to-speech* e áudio.

### Funcionalidades Principais:

- Leitura de mensagens de texto e *email* através de mecanismos de *text-to-speech*.
- Comandos de voz para funcionalidades como repetição de mensagens;
- Definições pormenorizadas de *text-to-speech* como número de palavras, tipo de voz ou velocidade de leitura.

## Safe Driving Text Machine [15]



**Programador:** Warez My Software  
**Play Store Rating:** 4.1 (906 ratings)  
**Última Atualização:** 14 de Maio de 2015  
**Número de Downloads:** 100,000 – 500,000  
**Preço:** Gratuito.

Figura 5 - Safe Driving Text Machine Logótipo

Apesar do programador possuir bastantes aplicações na *Google Play Store* não existe muita informação acerca da empresa ou pessoa que desenvolve esta aplicação.

Safe Driving Text Machine foca-se na deteção de movimentos do utilizador (a caminhar ou conduzir) para enviar mensagens automáticas de resposta conforme o nível de movimento que o utilizador escolher.

## Funcionalidades Principais

- Resposta automática a SMS com mensagem predefinidas, de acordo com o nível de movimento selecionado pelo utilizador;
- *Widget* para acesso rápido à aplicação.

## Drivemode – overlay driving UI [16]



**Programador:** Drivemode, Inc  
**Play Store Rating:** 4.2 (4,704 ratings)  
**Última Atualização:** 8 de Janeiro de 2016  
**Número de Downloads:** 100,000 – 500,000  
**Preço:** Gratuito.

Figura 6 - Drivemode  
Logótipo

A empresa Drivemode, Inc foi fundada em 2014 por empresários da zipcar, Tesla Motors e outros especialistas na indústria [17]. O único produto da empresa é a aplicação Drivemode que pretende criar uma forma segura e fácil para as pessoas utilizarem o telemóvel enquanto conduzem.

As três características principais são: (i) *Driving Hub*, interface simples controlável através de *swipes* para acesso a navegação e música; (ii) Chamadas e mensagens seguras através da audição de mensagens recebidas e resposta a partir de uma lista predefinida de textos e (iii) *Interface no-look* de forma a simplificar a necessidade de visualizar o telefone enquanto se o utiliza.

## Funcionalidades Principais

- Integração com aplicações de música e navegação;
- Detecção de condução através de ligação *bluetooth*;
- Detecção de condução através de sensores;
- Mecanismo de *text-to-speech* para audição de mensagens recebidas;
- Possibilidade de ignorar, responder e ouvir automaticamente mensagens;
- Lista de contactos favoritos para acesso rápido.



## text-STAR [18]



**Programador:** Cinqpoint, LLC  
**Play Store Rating:** 3.3 (33 ratings)  
**Última Atualização:** 13 de Janeiro de 2014  
**Número de Downloads:** 1,000 – 5,000  
**Preço:** Gratuito.

Figura 7 - text-Star Logótipo

A aplicação text-STAR é o único produto da empresa Cinqpoint, LLC cujo objetivo é premiar quem não utiliza o *smartphone* durante a condução. Para tal foi criado um algoritmo de pontuação, cuja patente está pendente, que dá pontos aos seus utilizadores que, mais tarde, podem trocar por prémios nos operadores.

### Funcionalidades Principais

- Detecção automática de condução através de velocidade;
- Resposta automática a SMS e MMS com mensagem predefinida;
- Calendarização para iniciar a aplicação em determinada hora;
- Pontuação de condução e obtenção de prémios através de operadores.

## SafeDrive rewards you [19]



**Programador:** XL Team  
**Play Store Rating:** 4.3 (862 ratings)  
**Última Atualização:** 22 de Outubro de 2015  
**Número de Downloads:** 10,000 – 50,000  
**Preço:** Gratuito.

Figura 8 - SafeDrive rewards you

XL Team é uma empresa localizada em Bucareste na Roménia proprietária da aplicação SafeDrive que aborda a segurança na estrada na perspectiva de jogo. Enquanto o utilizador conduz e não utiliza o telemóvel para enviar mensagens ganha pontos que podem ser gastos em lojas de parceiros.

### Funcionalidades Principais

- Premiar pontos por não utilizar o telemóvel;
- Possibilidade de comprar bens através de pontos ganhos na aplicação.

## TextDrive – No Texting While Driving [20]



**Programador:** SmallTalk  
**Play Store Rating:** 4.5 (273 ratings)  
**Última Atualização:** 18 de Dezembro de 2015  
**Número de Downloads:** 10,000 – 50,000  
**Preço:** Gratuito.

Figura 9 - TextDrive Logótipo

Esta aplicação é bastante simples e tem poucas funcionalidades. Permite ao utilizador definir uma mensagem para ser enviada como resposta quando está em condução. O utilizador tem também a possibilidade de ouvir a mensagem através de *text-to-speech*.

### Funcionalidades Principais

- Resposta automática a mensagens com texto predefinido;
- Leitura de mensagens através de *text-to-speech*;
- Audição repetida da última mensagem recebida.

## Sprint Drive First [21]



**Programador:** Safely  
**Play Store Rating:** 4.0 (1,220 ratings)  
**Última Atualização:** 15 de Outubro de 2015  
**Número de Downloads:** 100,000 – 500,000  
**Preço:** Gratuito.

Figura 10 - Sprint Drive First Logótipo

Safely é uma empresa localizada em *Emeryville*, Califórnia que se especializa em aplicações para segurança pessoal direcionadas para famílias. A aplicação faz parte de um conjunto de aplicações cujo objetivo é manter as famílias ligadas e seguras.

### Funcionalidades Principais

- Detecção automática do modo de condução;
- Atalhos diretos para aplicações predefinidas como navegação ou música;
- Acesso rápido ao número de emergência;
- Lista de favoritos: três pessoas que podem efetuar chamadas quando está em condução;
- Resposta automática a mensagens com mensagem predefinida.
- Controlo parental: Apagar ou desligar a aplicação apenas é permitido através de PIN.

## Drive Mode [22]



**Programador:** Vincent Gagnon  
**Play Store Rating:** 4.0 (25 ratings)  
**Última Atualização:** 26 de Julho de 2015  
**Número de Downloads:** 1,000 – 5,000  
**Preço:** Gratuito.

Figura 11 - Drive Mode  
Logótipo

Do programador particular Vincent Gagnon a aplicação Drive Mode é bastante simples e apresenta poucas opções ao utilizador, focando-se apenas no bloqueio de SMS e chamadas.

### Funcionalidades Principais

- Resposta automática a SMS com mensagem predefinida;
- Resposta automática a chamada de voz com mensagem predefinida;
- Silêncio automático de som e vibrações durante a condução;
- Ao sair de modo de condução o utilizador recebe informação sobre o que perdeu.

## Car dashdroid [23]



**Programador:** nez droid  
**Play Store Rating:** 4.4 (4,552 ratings)  
**Última Atualização:** 14 de Dezembro de 2015  
**Número de Downloads:** 100,000- 500,000  
**Preço:** Gratuito. Versão Premium 3,00€.

Figura 12 - Car Dashdroid  
Logótipo

Nez droid é um programador Android que criou uma das aplicações visualmente mais agradáveis entre as estudadas para este estágio. Ao contrário da maioria, esta aplicação não bloqueia a generalidade das funcionalidades mas tenta apresentá-las de forma mais segura ao utilizador. É possível configurar atalhos para outras aplicações ou contactos, aceder a música e navegação e avançar através de comandos de voz. A aplicação também nos informa da velocidade em tempo útil, meteorologia, data entre outras informações.

### Funcionalidades Principais

- Rápido acesso a navegação, música e contactos;
- Atalhos personalizados;
- Leitura de mensagens através de *text-to-speech*;
- Resposta automática a SMS e MMS com mensagens predefinidas.

- Temas para personalização dos ecrãs;

### Safely Go [24]



**Programador:** Safely  
**Play Store Rating:** 4.0 (862 ratings)  
**Última Atualização:** 3 de Maio de 2013  
**Número de downloads:** 50,000 – 100,000  
**Preço:** Gratuita.

Figura 13 - Safely Go Logótipo

Da mesma empresa criadora da aplicação Sprint Drive First a aplicação Safely Go permite apenas receber chamadas e mensagens de três contactos previamente definidos, assim como aceder a três aplicações do *smartphone*, como música ou navegação.

### Funcionalidades Principais

- Resposta automática a mensagens com texto predefinido;
- Lista de favoritos: Apenas três contactos podem fazer chamadas e enviar mensagens;
- Atalho para três aplicações do *smartphone*;
- Acesso rápido a números de emergência.

### DriveSafe Mode [25]



**Programador:** Drive Safe Mode  
**Play Store Rating:** 3.4 (24 ratings)  
**Último Atualização:** 13 de Agosto de 2015  
**Número de Downloads:** 1,000 – 5,000  
**Preço:** Gratuita.

Figura 14 - DriveSafe Mode Logótipo

A aplicação Drive Safe Mode foi criada pela empresa Text Safe Teens, LLC e a abordagem da aplicação é um pouco intrusiva, bloqueando todas as funcionalidades do telemóvel enquanto o utilizador está a conduzir.

### Funcionalidades Principais

- Bloqueia o telefone enquanto o utilizador está a conduzir;
- Bloqueia a utilização do teclado;
- Controlo Parental: Podem definir-se números de emergência para contacto;
- Controlo Parental: Recebe relatórios e notificações, caso seja desativado.

## Drive Alive Lite [26]



Figura 15 - Drive Alive Lite  
Logótipo

**Programador:** Drive Alive, LLC  
**Play Store Rating:** 3.9 (116 ratings)  
**Última Atualização:** 1 de Fevereiro de 2013  
**Número de downloads:** 5,000 – 10,000  
**Preço:** Versão Gratuita. Versão paga sem anúncios 3.65€.

Criada pela Drive Alive, LLC, a aplicação pretende premiar os utilizadores que não utilizam o *smartphone* enquanto conduzem. Antes de iniciar a viagem o utilizador deve ligar-se na aplicação e ter o GPS ligado, no final da viagem um relatório com os dados da viagem é gerado. Ações que vão contra a segurança na condução implicam a perda de crédito.

### Principais funcionalidades

- Premiar utilizador com crédito que pode ser transformado em dinheiro ou descontos;
- Incentivo à não utilização do telemóvel enquanto se conduz através de crédito.

### Principais problemas:

- Não existe bloqueio de qualquer funcionalidade, o que não resolve o problema de mensagens e chamadas durante a condução.

## Ultimate Car Dock [27]



Figura 16 - Ultimate Car  
Dock Logótipo

**Programador:** Apps on Toast  
**Play Store Rating:** 3.9 (1,913 ratings)  
**Última Atualização:** 11 de Janeiro de 2016  
**Número de Downloads:** 100,000 – 500,000  
**Preço:** Versão experimental de 7 dias. Versão completa 3.99€.

Criada pela equipa Apps on Toast sediada em Espanha esta é uma das aplicações mais evoluídas para utilização de *smartphone* na condução tendo ganho o prémio *The Best Technological Developments for the Connected Car*, na edição de 2015 da *Barcelona Motor Show*.

Os focos principais da aplicação são o *Speech-To-Text* e os comandos de voz que permitem ao utilizador interagir com a aplicação sem contacto físico com o *smartphone*.

### Funcionalidades Principais:

- Leitura de mensagens SMS, aplicações OTT e emails via *text-to-speech*;
- Detecção de condução através de ligação *bluetooth* ao dispositivo;
- Leitor de música na aplicação;

- Vários ecrãs com atalhos para aplicações;
- Enorme coleção de comandos de voz:
  - Iniciar chamadas e mensagens;
  - Controlar leitor de música;
  - Iniciar outras aplicações;
  - Gerir mensagens e chamadas recebidas.

### LifeSaver – Distracted Driving [28]



**Programador:** Life Apps LLC  
**Play Store Rating:** 3.9 (62 ratings)  
**Última Atualização:** 24 de Setembro de 2015  
**Número de Downloads:** 1,000 – 5,000  
**Preço:** Versão Gratuita. Com compras na aplicação.

Figura 17 - LifeSaver Logótipo

A empresa Life Apps, LLC oferece mais do que apenas uma aplicação Android. O seu foco é o controlo parental e para além da aplicação existe um portal *online* dedicado aos pais onde se podem visualizar informações como localização ou número de viagens assim como número de desbloqueios (para quando o utilizador é passageiro e não condutor).

#### Principais Funcionalidades:

- Detecção automática quando o utilizador está a conduzir;
- Pontuações e prémios pela utilização da aplicação em condução;
- Desbloqueio para emergência ou caso seja passageiro;
- Controlo Parental: Notificações aquando da alteração de definições ou quando uma viagem termina.

### Dashdroid – Safe Driving App [29]



**Programador:** DashDroid  
**Play Store Rating:** 3.8 (111 ratings)  
**Última Atualização:** 9 de Setembro de 2014  
**Número de Downloads:** 5,000 – 10,000  
**Preço:** Gratuita.

Figura 18 - Dashdroid Logótipo

À semelhança de outras aplicações que simulam o *dashboard* de um carro, esta aplicação pretende ser um menu de acesso a funcionalidades do telemóvel de forma mais segura. A aplicação deteta se o utilizador está a conduzir através de GPS ou *bluetooth*, sendo a verificação de condução feita no intervalo de tempo que o utilizador decidir.

### Principais Funcionalidades:

- Atalhos para aplicações do telefone;
- Definição da velocidade mínima para ativar modo de condução;
- Detecção de condução por velocidade ou ligação *bluetooth*;
- Bloqueio de chamadas e mensagens;
- Resposta automática a SMS e chamadas com mensagens predefinida;
- Desativação do código de bloqueio do *smartphone*;
- Personalização da imagem de fundo do ecrã inicial.

### Automate – Car dashboard [30]



**Programador:** BitSpice  
**Play Store Rating:** 4.5 (693 ratings)  
**Última Atualização:** 28 de Dezembro de 2015  
**Número de Downloads:** 50,000 – 100,000  
**Preço:** Gratuito. Versão premium 3.59€.

Figura 19 - Automate Logótipo

Fundada em 2014, a BitSpice é uma empresa sediada em Toronto no Canadá que desenvolve aplicações móveis [31]. O objetivo da aplicação AutoMate é transformar o *smartphone* num *dashboard* inteligente. O ecrã inicial apresenta uma lista de cartões com informações de calendário, velocidade, temperatura, contactos favoritos, entre outros, que podem ser configurados pelo utilizador através de *swipes* horizontais.

### Principais Funcionalidades:

- Atalhos para outras aplicações do *smartphone*, assim como definições rápidas;
- Controlo de aplicações de media;
- Comandos de voz para navegação e controlo de música;
- Envio e resposta a SMS através de comandos de voz;
- Início de chamadas a partir da aplicação;
- Notificações do telemóvel escondidas em condução.

### 2.1.2. Competidores Indiretos

Esta secção fornece informações acerca das aplicações de empresas ou programadores independentes que oferecem mecanismos de condução segura para outras plataformas além do Android.

#### Windows Phone Driving Mode [32]



Figura 20 - Windows Phone Driving Mode Menu

Para utilizar o *Driving Mode* é necessário a configuração prévia de um *kit* de mãos-livres ou *headset* bluetooth. Sempre que o *smartphone* se ligar ao dispositivo configurado o modo de condução é ativado.

Durante o modo de condução todas as notificações são desligadas e chamadas de emergência ou mensagens podem não ser recebidas.

O utilizador pode configurar de modo a optar por enviar as chamadas recebidas diretamente para o *voicemail* e ignorá-las. Tem também a possibilidade de configurar uma mensagem para ser automaticamente enviada como resposta a SMS e chamadas.

#### Android Auto [33]



Figura 21 - Android Auto Logótipo

Em 2014 várias empresas tecnológicas e líderes da indústria automóvel juntaram-se para criar o grupo que intitularam *Open Automotive Alliance* [34] com o objetivo de aumentar o valor dos seus produtos através da introdução de tecnologia na indústria automóvel, assim como desenvolver soluções para melhoria da segurança na condução.

A forma de alcançar esse objetivo passou pela introdução da plataforma Android nos *dashboards* dos carros.

O sistema Android Auto permite ao utilizador interagir com o ecrã do carro tendo acesso a funcionalidades de navegação através do *Google Maps*, interagir com aplicações de mensagens e chamadas por comandos de voz com recurso ao *Google Now*, e aplicações de *media* como música e *podcasts*.

A plataforma destaca-se pelo controlo por voz, permitindo ao utilizador o mínimo de distração com o ecrã.



### Cell Control [35]



Figura 22 - Cell Control Logótipo

Cell Control é um sistema de segurança na estrada criado pela empresa obdEdge, LLC com sede em Louisiana. A empresa desenvolveu um dispositivo, *DriveID*, que deve ser colocado no carro e logo que este entra em movimento o *DriveID* ligado à aplicação Cell Control por *bluetooth*, bloqueia mensagens, chamadas e acesso a aplicações.

No portal *online* o utilizador consegue configurar o que pode ou não ser acedido quando está em condução, assim como verificar relatórios e monitorizar a sua utilização.

### Apple Carplay [36]



Figura 23 - Apple Carplay Logótipo

À semelhança do Android Auto, a Apple Carplay é a solução da Apple para a utilização do *smartphone* durante a condução.

Uma vez ligado ao *dashboard* do carro o utilizador pode controlar aplicações através de comandos de voz.

Pode ainda aceder a navegação através do *Apple Maps*, receber e enviar mensagens através de *text-to-speech* e *speech-to-text*, bem como controlar *media* tal como música.

O Carplay ao contrário do Android Auto possui algumas limitações, como a falta de suporte a comandos de voz para aplicações de terceiros e falta de suporte a envio de mensagens para além da aplicação *iMessage*.

### 2.1.3. Análise Comparativa

A tabela 1 apresenta as aplicações estudadas que se destacaram pelo número de utilizadores ou pela quantidade de funcionalidades que exibem. As funcionalidades escolhidas para comparação foram as mais populares de entre estas aplicações e as mais interessantes no âmbito do estágio.

|   | AT&T Drivemode | Car Home Ultra | Drive Safe.ly | Drivemode | Car Dashdroid | Dashdroid | Automate | Ultimate Car Dock |
|---|----------------|----------------|---------------|-----------|---------------|-----------|----------|-------------------|
| Aplicação Gratuita e disponível globalmente                           | ×              | ×              | ✓             | ✓         | ✓             | ✓         | ✓        | ✓                 |
| Mecanismo de <i>text-to-speech</i> para ouvir mensagens recebidas     | ×              | -              | ✓             | ✓         | ✓             | ×         | ✓        | ✓                 |
| Permitir receção de chamadas (com ou sem sistema mãos-livres)         | ✓              | ✓              | ✓             | ✓         | ✓             | ✓         | ✓        | ✓                 |
| Resposta automática a SMS com mensagem predefinida                    | ✓              | ✓              | ✓             | ✓         | ✓             | ✓         | ✓        | ×                 |
| Deteção automática de condução através sensores (exemplo: velocidade) | ✓              | -              | ×             | -         | ×             | ✓         | ×        | ×                 |
| Deteção automática de condução através de ligação <i>bluetooth</i>    | -              | ✓              | ✓             | ✓         | ✓             | ✓         | ✓        | -                 |
| Comandos de voz (por Google Voice Search ou aplicação)                | ×              | ✓              | ×             | ×         | ✓             | ×         | ✓        | ✓                 |
| Acesso rápida a lista de contactos favoritos                          | ✓              | ✓              | ×             | ✓         | ✓             | ×         | ✓        | ✓                 |
| Controlo de música e navegação na aplicação                           | ×              | ✓              | ×             | ✓         | ✓             | ×         | ✓        | ✓                 |
| <i>Gamification</i> (algum tipo de pontuação e prémios)               | ×              | ×              | ×             | ×         | ×             | ×         | ×        | ×                 |
| Controlo Parental (Alertas ou Blocos)                                 | ✓              | ×              | ×             | ✓         | ×             | ×         | ×        | ×                 |

Tabela 1 - Comparação de Funcionalidades entre Competidores Diretos

Ao analisar a tabela 1 percebe-se que não existem muitas funcionalidades que sejam transversais em todas as aplicações. Excluindo as funcionalidades de receber chamadas, resposta automática a mensagens e deteção automática de condução via ligação *Bluetooth*, não existe consenso entre as principais aplicações, naquelas que devem ser as restantes funcionalidades de uma aplicação deste género. Ainda assim, existem algumas funcionalidades populares, como leitura através de *text-to-*

*speech* e acesso rápido a lista de contactos favoritos, às quais deve ser dado prioridade uma vez que aparecem com frequência.

## 2.2. WIT RCS Suite

A aplicação *Android RCS+* faz parte do conjunto de aplicações e servidores RCS da WIT Software representado na figura 24 [37].

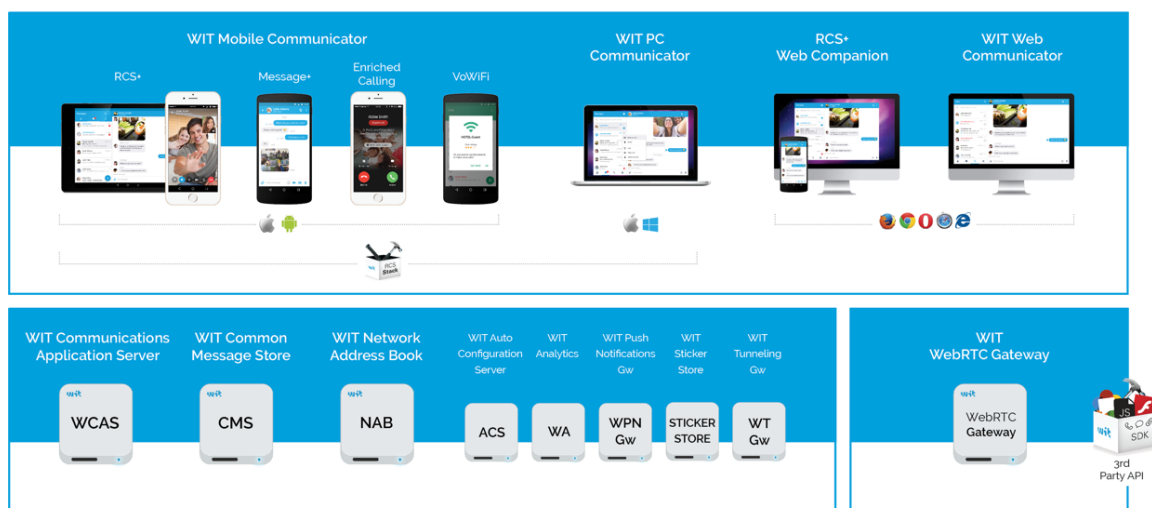


Figura 24 - WIT RCS Suite

Desta *suite* fazem parte os seguintes componentes:

### WIT Communications Application Server

O *Application Server* implementa a especificação RCS 5.1 e está a par com a evolução da especificação RCS. Este pode ser instalado numa infraestrutura IMS ou num ambiente pré-IMS utilizando um *Session Border Controller* (SBC).

### WIT PC Communicator/Web Communicator

O WIT PC Communicator e WIT Web Communicator, este último através do protocolo *WebRTC*, fazem parte das soluções para computador.

### RCS Stack

A biblioteca *RCS Stack* permite a terceiros e operadores acelerar a inclusão de funcionalidades de *Rich Communication* nas suas aplicações. Está disponível para *Android*, *iOS*, *Windows* e *Mac OS X*.

## Aplicações WIT RCS 5.1 [38]

O conjunto de aplicações móveis da qual faz parte a aplicação em que este estágio se inseriu possui as seguintes funcionalidades:

### Descoberta e Ativação

- Mecanismo de autoconfiguração por Rede Móvel e Wi-Fi;
- UI/UX controlado pelo Operador;
- *Provisioning* via *Push*.

### Gestão de contactos e *Capabilities*

- Descoberta de contactos e *service capabilities* utilizando *SIP Options*;
- Integração com a lista de contactos do dispositivo;
- *Blacklist* local (bloqueio de contactos).

### Localização geográfica via *Push*

- Partilha de localização durante chamada ou *chat*;
- Localização atual. Localização no Mapa. Minhas localizações;
- Mostrar no mapa.

### Chamadas IP

- Chamadas de voz HD (IR.92) e vídeo (IR.94);
- Partilha de ficheiro em chamada: Imagens, Vídeo, Localização, Contactos entre outros;
- *Upgrade* e *Downgrade* de chamadas (voz para vídeo e vídeo para voz);
- Mensagens de vídeo e voz;
- *Codecs* de Áudio - AMR-NB, AMR-WB, *Skype Silk*, *Speex*, G711 (PCMA e PCMU), GSM, G723, G729, G721, G722, iLBC entre outros);
- *Codecs* de vídeo.

### Mensagens

- *Chat* 1 para 1 com *Integrated Messaging*;
- SMS *Interworking*;
- *Chat* de grupo;
- Partilha de conteúdo: Imagens, Localização, Vídeo, Contactos;
- *Stickers*. *Emoticons* e *Emojis*;
- Guardar mensagem e enviar mais tarde (*Store and Forward*);
- Notificações de Envio.

### Transferência de Ficheiros

- Transferências de Ficheiros (MSRP e HTTP)
- Transferências de Ficheiros em *chats* de grupo;
- Guardar e enviar mais tarde (*Store and Forward*) ficheiros.

### Funcionalidades Extra

- *Chat Stickers*;
- Filtros e desenho de fotos;
- Personalização (Temas, *Wallpapers*, Tamanho de Letra);
- *Push Notifications*;

- Integração com rede social (*Facebook*);
- Publicidade *Mobile*;
- Compras *In-App* de Conteúdo *Premium* e Jogos;
- Troca de *Capabilities* utilizando Presença;
- Suporte a dispositivos secundários;
- *Network Address Book*;
- *Central Message Store*.

## 2.3. Mecanismos Text-To-Speech

Tendo em conta a necessidade do protótipo oferecer a possibilidade de ouvir as mensagens de texto recebidas, é importante perceber as soluções de *text-to-speech* existentes atualmente para a plataforma Android. Tal conhecimento permite optar por utilizar a mais adequada à implementação que se deseja.

### 2.3.1. IVONA Text-to-Speech HQ [39]



Figura 25 - IVONA TTS  
Logótipo

IVONA é uma empresa da Amazon focada em criar tecnologia *Text-To-Speech* de alta qualidade [40]. A sua aplicação, de acordo com as *reviews*, é o mecanismo que oferece as vozes mais realistas e precisas. Utiliza a tecnologia *BrightVoice* para gerar sons mais naturais tornando a leitura de grandes quantidades de texto mais confortável [41]. Para instalar o motor é necessário instalar a aplicação principal e depois individualmente instalar cada pacote de voz. Este motor não tem suporte para Português.

### 2.3.2. SVOX [42]



Figura 26 - SVOX TTS  
Logótipo

O motor SVOX é um dos que possui mais idiomas, incluindo linguagem Portuguesa na versão masculina e feminina e múltiplas vozes em Inglês.

Estatisticamente, utilizadores do mecanismo SVOX ouvem o dobro dos artigos, em comparação com utilizadores IVONA, e até dez vezes mais que os utilizadores de outros motores [43].

### 2.3.3. Google Text-To-Speech [44]



Apesar de ter uma qualidade de som inferior aos motores IVONA e SVOX, o Google *Text-To-Speech* tem a vantagem de vir instalado de origem na maior parte dos telefones Android recentes, evitando assim a necessidade de requisitar a instalação de motores extra. Apesar de ter uma quantidade considerável de idiomas, o Português apenas está disponível na versão do Brasil.

Figura 27 - Google TTS  
Logótipo

## 2.4. Mecanismos de Reconhecimento de Atividade

Para criar a funcionalidade de deteção de condução é necessário utilizar *software* que consiga transformar a informação dos sensores do *smartphone* em estimativas sobre qual a atividade que o utilizador do dispositivo está a efetuar. Para a plataforma *Android* existem dois principais mecanismos, o *Google Activity Recognition* e o *PathSense Activity Recognition*.

### 2.4.1. Google Activity Recognition API

O *Google Activity Recognition* lançado em 2013 está presente na maioria dos *smartphones* Android através do Google Play Services [45]. Quando implementado esta API acorda periodicamente o dispositivo e lê pequenos períodos de dados dos sensores. Através da análise desses dados via algoritmos de *Machine learning*, a API calcula qual a atividade mais provável que o dispositivo está a realizar nesse momento. Atualmente as atividades suportadas são: em veículo, em bicicleta, a pé, a correr, parado, a abanar ou andar.

### 2.4.2. PathSense Activity Recognition



A *startup* PathSense com sede em *Carlsbad*, Califórnia [46] especializa-se em soluções de localização para Android e iOS. Um dos seus produtos é o *Activity Recognition*, uma API grátis que pode ser incluída nas aplicações Android e que, à semelhança do *Google Activity Recognition*, utiliza dados recolhidos pelos sensores do *smartphone* para calcular a atividade que o dispositivo está a realizar.

Figura 28 - PathSense  
Logótipo

## 2.5. Conceitos Android [47]

As aplicações Android são escritas na linguagem de programação *Java*. O *Android* SDK compila o código e gera um APK (*Android Package*) que inclui todos os conteúdos da aplicação e pode ser instalado num dispositivo com sistema operativo (SO) *Android*. Logo que uma aplicação é instalada num dispositivo, esta vive isolada das restantes aplicações, podendo apenas aceder a dados de outras através de permissão ou pedido, que deve ser aceite pela aplicação requisitada.

Seguem-se as principais componentes de uma aplicação *Android* utilizadas durante a realização do estágio:

### Activity

Uma *Activity* é um tipo de classe em Android responsável pela criação e gestão de um ecrã. Na *activity* são inseridos elementos de *UI* e são geridas as interações com o utilizador.

### Fragment

Um *Fragment* representa o comportamento de uma porção da interface do utilizador numa *Activity*.

### Service

Um *Service* é um componente que não tem interface para o utilizador e corre em *background* para executar operações de longa duração.

### Broadcast Receivers

Um *Broadcast Receiver* é um componente que responde a eventos enviados para todo o sistema. Muitos eventos são originados pelo próprio sistema, como o evento de ecrã desligado, ou bateria fraca.

### App Widgets

Um *App Widget* é uma miniatura das *views* de uma aplicação que pode ser colocada em outras aplicações, como o ecrã inicial. É possível definir um intervalo de tempo para o *Widget* receber atualizações e modificar a sua interface.

### Intents

Um *Intent* é uma descrição abstrata de uma operação a ser realizada. Pode ser utilizado para iniciar uma *Activity*, para enviar um evento a um *Broadcast Receivers*, ou enviar um evento para comunicar com um *Service* em *background*.

### ViewPager (UI)

*ViewPager* é um gestor de *layout* que permite a implementação de funcionalidade de navegação por diferentes ecrãs, representados geralmente por *Fragments*.

### RecyclerView (UI)

Um *RecyclerView* é uma componente utilizada para mostrar grandes quantidades de dados de forma eficiente, em formato de lista.

### Adapter

Um *Adapter* é um objeto que serve de ligação entre uma *view* e os dados que irão ser apresentados na *view*. O *Adapter* fornece o acesso aos dados.

### **Bottom Sheet**

As *Bottom Sheets* são componentes de UI que permitem a criação de um menu flutuante acima da vista principal.

### **Dialog**

Os *dialogs* são componentes representados por pequenas janelas que pedem ao utilizador a confirmação de uma ação.



## Capítulo 3

### Abordagem

Neste capítulo dá-se a conhecer a metodologia de desenvolvimento utilizada no decorrer do estágio e a forma como esta foi adotada no projeto, assim como os riscos encontrados para o mesmo.

Na **secção 3.1 Scrum vs Waterfall** explica-se a razão pela qual foi preferida a metodologia ágil *Scrum* em relação a uma metodologia sequencial como o *Waterfall*.

A **secção 3.2 Metodologia** serve para aclarar a forma como a metodologia *Scrum* foi utilizada no decorrer do estágio, bem como apresentar os conceitos teóricos da mesma.

A **secção 3.3 Riscos** apresenta os riscos do projeto e a forma de os mitigar, caso surgissem.

O **Apêndice A – Abordagem** complementa este capítulo com informações detalhadas sobre a forma como as *User Stories* foram definidas, o *Product Backlog* completo e os diferentes *Sprints* realizados, pelo que é recomendada a sua leitura.

#### 3.1. Scrum vs Waterfall

Apesar de existir um conjunto de requisitos base, o protótipo que se implementou no estágio não foi cem por cento definido, mas sim construído à medida que os *stakeholders* (colaboradores da WIT interessados na solução) foram dando *feedback*. Por esta razão uma metodologia como o *Waterfall* foi, à partida, descartada.

Entre as metodologias ágeis existentes, o *Scrum* é o mais utilizado por ser o mais simples e de fácil adoção e adaptação [48]. Seguem algumas das razões para a escolha do *Scrum* para o estágio:

- Melhor adaptação à mudança de requisitos de acordo com o *feedback* dado pelos interessados no produto;
- Possibilidade de obter um protótipo demonstrável rapidamente e, progressivamente, incrementar funcionalidades;
- Metodologia já utilizada em projetos da empresa, quer pelo tutor, quer pelo orientador, o que facilitou a aprendizagem por parte do estagiário.

### 3.2. Metodologia Scrum

A figura 29 apresenta uma visão geral do processo de Scrum. Este processo inicia-se com a construção do **Product Backlog (3.2.4)** que representa o conjunto de funcionalidades a desenvolver para o produto. As funcionalidades são representadas através de **User Stories (3.2.2)**.

Após a criação do *Product Backlog* a equipa define os requisitos que uma determinada funcionalidade deve conter para ser dada como terminada. Ao documento produzido dá-se o nome de **Definition of Done (3.2.3)**. É acordada a duração de cada **Sprint (3.2.5)** que normalmente fica entre duas a quatro semanas.

Antes de iniciar cada *Sprint*, realiza-se a **Reunião de Planeamento de Sprint (3.2.6)** onde são escolhidas um conjunto de *User Stories* do *Product Backlog* que irão constituir o *Sprint Backlog* a ser desenvolvido durante esse *Sprint*. A quantidade de funcionalidades deve ser decidida de acordo com a velocidade da equipa em *Sprints* anteriores.

Durante o *Sprint* a equipa reúne diariamente na **Reunião Diária (3.2.7)** com o propósito de apresentar as funcionalidades em que cada membro está a trabalhar e apresentar igualmente os problemas encontrados, se for caso disso. Esta reunião reveste-se de grande importância na medida em que todos os elementos da equipa tomam conhecimento do que os seus colegas estão a trabalhar e desenvolvem mecanismos de ação colaborativa no ultrapassar de obstáculos.

Terminado o *Sprint* deve existir uma **Sprint Review Meeting** onde é apresentado o que foi concluído no *Sprint* e uma **Sprint Retrospective** cujo foco é discutir o que está a decorrer, bem ou mal, no processo de *Scrum*.

No final de cada *Sprint* é também esperado um produto final demonstrável e com as funcionalidades do *Sprint* implementados e testadas.

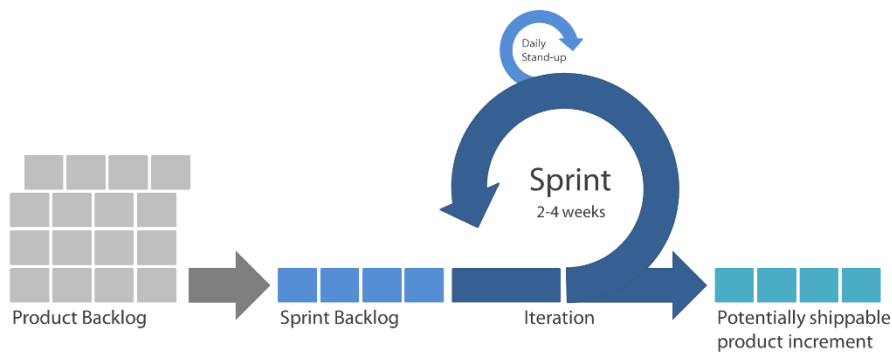


Figura 29 - Processo de Desenvolvimento Scrum [49]

### 3.2.1. Cargos

Existem três cargos essenciais no desenvolvimento *Scrum*: o *Product Owner*, o *Scrum Master* e a equipa, cujas responsabilidades são descritas de seguida [50].

O *Product Owner* é o responsável pela ligação entre a equipa de desenvolvimento e as restantes partes envolvidas. Este representa os interesses dos *stakeholders* e portanto deve gerir os requisitos funcionais (*product backlog*) e comunicar com a equipa de forma a garantir que esta tenha a visão correta do que deverá ser a solução final. O papel de *Product Owner* ficou a cargo do orientador da WIT, **Frederico Lopes**, tendo o estagiário colaborado na construção do *Product Backlog*.

O *Scrum Master* é um elo de ligação entre a equipa e o *Product Owner*. A sua função está intrinsecamente ligada à eliminação de obstáculos que impeçam a equipa de alcançar os seus objetivos. A existência deste elemento permite à equipa manter-se criativa e produtiva, assegurando o sucesso visível ao *Product Owner*. O papel de ***Scrum Master*** é efetuado pelo tutor **João Lopes**.

A equipa deve organizar-se de forma autónoma e decidir como completar o trabalho. Num projeto de *software*, a mesma é composta por uma mistura de engenheiros de *software*, arquitetos, programadores, analistas, *testers* ou *designers*. Em cada *sprint* a equipa é responsável por determinar a quantidade de trabalho a completar de acordo com a velocidade registada em *sprints* anteriores. Dado que o **estagiário** é o único responsável por produzir a solução, incluindo desenvolvimento e testes, este representa a **Equipa**.

### 3.2.2. User Stories - Requisitos

Uma *User Story* representa um requisito funcional na metodologia *Scrum*. A construção de uma *User Story* passa pela definição de quem vai usufruir da funcionalidade, o que é necessário implementar e a razão para a existência dessa funcionalidade. Um exemplo de uma *user story* criada para o estágio é:

*Como condutor quero ouvir uma mensagem para evitar a sua leitura enquanto conduzo.*

Uma *User Story* é caracterizada por vários atributos. No **apêndice A – Abordagem** são detalhados os significados de cada característica como: Tipo; Categoria; Ator; Pontos; Prioridade e Estado.

### 3.2.3. Definition of Done

O *Definition of Done* (DoD) representa todos os requisitos que uma *User Story* deve cumprir para ser considerada terminada pelo *Product Owner*. Para além das *User Stories*, os *Sprints* têm o seu próprio DoD, que representa aquilo que a equipa deve completar no final de cada *Sprint*. Foram definidos para este projeto os seguintes DoD:

#### Definition of Done de User Story

##### Implementação

- Implementação do código completa;
- Código integrado com o projeto base (a nível de *software*).

### Testes

- Testes de aceitação.

### Desenvolvimento

- Código submetido no SVN com o ID da plataforma *redmine* na descrição.

### Documentação

- Código comentado, quando necessário.

### Definition of Done de Sprint

- Documento com lista de todas as funcionalidades atualizado;
- Existência de ficheiro de instalação (*apk*) com funcionalidades do *Sprint*;
- Aplicação pronta a ser demonstrada;
- Número da versão atualizado;
- Versão final submetida no SVN com o ID da plataforma *redmine* na descrição.

### 3.2.4. Product Backlog

O *Product Backlog* representa o conjunto de todas as *user stories* do projeto. Este é gerido pelo *Product Owner* através da introdução de novas *user stories* ou priorização de funcionalidades e pela equipa através da atualização de informações como previsão de complexidade, tempo gasto ou progresso.

Uma vez que se faz uso de metodologia ágil o *product backlog* está em constante alteração. Um fragmento do *product backlog* pode ser visualizado na tabela 2. O *Product Backlog* completo do estágio, pode ser consultado no **Apêndice A – Abordagem**.

| #   | Categoria     | User Story   | Pontos | Estado |
|-----|---------------|--|--------|--------|
| ... | ...           | ...  | ...    | ...    |
| 15  | Car Mode View | As a driver I want to be able to reply to a received message with a voice call                         | 3.0    | Closed |
| 16  | Car Mode View | As a driver I don't want to receive any phone notification while in car mode to avoid any distraction, | 5.0    | Closed |
| 17  | Car Mode View | As a driver I want to be able to reply to a received messaged with a audio message                     | 8.0    | Closed |
| ... | ...           | ...  | ...    | ...    |

Tabela 2 - Sub-conjunto do Product Backlog

### 3.2.5. Sprints

O *Scrum Master* em conjunto com a equipa decide o período de tempo dos *sprints*, normalmente entre uma e quatro semanas. É neste período que a equipa desenvolve as *User Stories* definidas para esse *Sprint* (*Sprint Backlog*).

Para informação mais detalhada sobre os *sprints* realizados durante o decorrer do estágio recomenda-se a leitura do **Apêndice A – Abordagem**.

Durante o primeiro semestre foram realizados seis *Sprints* de desenvolvimento de duas semanas de duração cada e um *sprint* inicial de planeamento de uma semana. No segundo semestre foram realizados dez *Sprints*.

#### **Sprint #0**

**Data:** 20/10/2015 – 26/10/2015

Durante o *sprint #0* foi finalizado o estudo do estado da arte e tecnologias, assim como a definição e estimação do *Product Backlog*. Definição do *Minimum Viable Product* para a aplicação ser considerada para o *Mobile World Congress* e definição do *Definition of Done*.

#### **Sprint #1**

**Data:** 27/10/15 – 10/11/2015

Este *Sprint* marcou o início do desenvolvimento do módulo, o objetivo principal foi a familiarização com a arquitetura do RCS+, com a *framework* Java do Android e com as boas práticas do desenvolvimento Android. As funcionalidades concluídas foram a criação do *layout* do menu quando a aplicação está em modo de condução, a possibilidade de entrar nesse menu selecionando uma opção nas definições e a possibilidade dos utilizadores da aplicação saberem quando os seus contactos estão a conduzir, através da colocação de um ícone, quer na lista de mensagens, quer no menu de *chat*.

#### **Sprint #2**

**Data:** 11/11/2015 – 24/11/2015

Durante o decorrer do *Sprint #1* houve a necessidade de redefinir a interface gráfica planeada, tendo sido acrescentadas *User Stories* para essas alterações, que foram efetuadas no *sprint #2*. Além disso foi implementado o menu de mensagens por ler e a possibilidade do utilizador fazer *swipe* horizontal para navegar pelas mensagens.

#### **Sprint #3**

**Data:** 25/11/2015 – 8/12/2015

No *sprint #3* o foco do desenvolvimento foi no mecanismo de *text-to-speech* para ouvir as mensagens recebidas e na deteção de *hardware* que permitisse efetuar chamadas de forma segura. Foi necessário explorar a API de *text-to-speech* do Android e a API de deteção de dispositivos *bluetooth* e *headphones* com microfone para realizar estas funcionalidades. Por fim aplicou-se uma forma de receber e apagar as notificações do *smartphone* enquanto este está em modo de condução, para evitar, ao máximo, distrações do utilizador.

#### **Sprint #4**

**Data:** 9/12/15 – 22/12/2015

O foco do *sprint* #4 foi desenvolver a funcionalidade de resposta a uma mensagem através da gravação de um clip de áudio. Sendo esta uma funcionalidade complexa foi estimado cerca de meio *sprint* para a concluir. Foi também implementada a opção de enviar mensagem a um utilizador em condução bem como o menu que permite ao utilizador escolher entre várias opções de contacto. A última *User Story* da tabela 6 não se apresenta concluída uma vez que não cumpria o *Definition of Done*, pelo que foi planeada novamente para o *Sprint* #5.

#### **Sprint #5**

**Data:** 23/12/2015 – 5/01/2016

O *Sprint* #5 para além de ter servido para terminar a *User Story* de notificação que ficou pendente no *Sprint* #4, foi utilizado para corrigir um problema detetado com uma *User Story* dada como terminada. Para além destas correções, foi criada a opção de calendarização do envio de mensagem para quando o utilizador sai do modo de condução e o início de uma chamada de voz, caso o utilizador em condução tenha um sistema de mãos-livres.

#### **Sprint #6**

**Data:** 6/01/2016 – 19/01/2016

Cerca de metade do *sprint* #6 foi dedicado exclusivamente à escrita do relatório de estágio intermédio. O restante *sprint* focou-se na correção de alguns *bugs* detetados ou pequenas melhorias a *User Stories* já concluídas. A última *User Story* apresentada na figura 6 corresponde à maior funcionalidade criada: geração de notificação em contactos favoritos quando um utilizador entra em modo de condução. Esta funcionalidade não ficou concluída passando para o novo *sprint*.

#### **Sprint #7**

**Data:** 20/01/2016 – 02/02/2016

O *sprint* #7 foi dedicado à criação de funcionalidade de jogos, adicionando a possibilidade de obter pontos estando nos menus de condução. Foi igualmente implementada a troca de pontos entre utilizadores e gravação de pontuações em base de dados local. Para além de tudo isto, procedeu-se a pequenas alterações como a alteração da onda de gravação de áudio e a correção de problemas identificados nos *avatars* apresentados em notificações.

#### **Sprint #8**

**Data:** 03/02/2016 – 12/02/2016

O *Sprint* #8 foi reduzido em número de dias de desenvolvimento em comparação com os restantes *sprints* de duas semanas. Este *sprint* serviu para fazer estudos sobre API de reconhecimento de atividade e outras aplicações como o *zendrive*. Uma vez que faltavam pormenores, algumas *user stories* foram movidas para o *sprint* seguinte.

#### **Sprint #9**

**Data:** 15/02/2016 – 01/03/2016

No *sprint* #9 o foco foi a estabilização da aplicação, corrigindo vários problemas identificados e integrando a aplicação na versão de demonstração do *Mobile World Congress*. Para além disso foi feita uma nova análise às aplicações estudadas, com o intuito de recolher novas funcionalidades a implementar. Desta análise resultaram algumas *User Stories* que foram acrescentadas ao *Product Backlog*.

### **Sprint #10**

**Data:** 02/03/2016 – 15/03/2016

O *sprint* #10 serviu essencialmente para corrigir pequenos problemas encontrados e adicionar a funcionalidade de início de chamadas em condução.

### **Sprint #11**

**Data:** 16/03/2016 – 29/03/2016

No *sprint* #11 implementou-se a deteção de condução através da conexão de dispositivos *bluetooth* e o envio de mensagens de áudio por parte dos utilizadores da aplicação RCS+, assim como a receção e audição das mesmas por parte dos condutores.

### **Sprint #12**

**Data:** 30/03/2016 – 12/04/2016

Neste *sprint* foi implementado o *Floating button* de acesso rápido à aplicação, a apresentação de notificações perdidas durante a condução, a possibilidade de remoção de dispositivos *Bluetooth* adicionados e o indicador de bateria nos menus de condução.

### **Sprint #13**

**Data:** 13/04/2016 – 26/04/2016

No *Sprint* #13 privilegiou-se a implementação do algoritmo de deteção de condução bem como o menu de escolha entre condutor e passageiro. Dada a complexidade das funcionalidades a estimação de sete pontos foi claramente subvalorizada, apresentando o *sprint* bastante mais complexidade. Nesta fase dedicou-se algum tempo à documentação do projeto.

### **Sprint #14**

**Data:** 27/04/2016 – 10/05/2016

Neste *sprint* deu-se ênfase à criação de uma nova aplicação Android cujo objetivo era recolher dados e através deles descobrir quais os parâmetros ideais a utilizar na deteção de condução, assim como a comparação entre mecanismos de reconhecimento de atividade para *Android*. Para além destas tarefas foram resolvidos vários problemas previamente detetados.

### **Sprint #15**

**Data:** 11/05/2016 – 24/05/16

O objetivo do *sprint* #15 foi melhorar a experiência das funcionalidades relativas à deteção de condução, através da análise dos *logs* recolhidos pela aplicação realizada no *sprint* anterior. Procedeu-se também à implementação de pequenas alterações como alertas sonoros após deteção e bloqueio do algoritmo enquanto o utilizador está em chamada. Neste *sprint* iniciou-se a escrita do relatório final.

### **Sprint #16**

**Data:** 25/05/2016 – 15/06/2016

O *sprint* #16 contemplou a conclusão do relatório, a criação de uma apresentação final do protótipo realizado e estabilização do protótipo através da correção de pequenos problemas.

### 3.2.6. Reunião de Planeamento de Sprint

A reunião de planeamento em *Scrum* serve para a equipa decidir quais as *user stories* a implementar no *sprint* seguinte. Esta escolha deve ser feita de acordo com a velocidade dos *sprints* anteriores. A velocidade representa os pontos concluídos por *Sprint*.

Durante o estágio, na reunião de planeamento, foram realizadas as *sprint review* e *sprint retrospective*.

Na dita reunião são efetuadas as seguintes tarefas:

- Equipa (estagiário) apresenta as funcionalidades concluídas no *Sprint*;
- *Product Owner* marca *User Stories* como concluídas caso cumpram o DoD;
- De acordo com a velocidade, define-se o *Sprint Backlog* para o novo *Sprint*;
- Discussão de problemas ou processos a serem alterados para aumentar a produtividade.

### 3.2.7. Reunião Diária

Durante o decorrer de cada *sprint* deverá existir uma reunião rápida, chamada *Daily Meeting*, que não deverá ultrapassar os quinze minutos onde cada membro da equipa deve responder a três perguntas:

- O que fizeste ontem?
- O que irás fazer hoje?
- Existe algum obstáculo atualmente?

Esta reunião serve para os membros da equipa conhecerem o estado do trabalho dos seus colegas e, caso exista algum obstáculo, este seja ultrapassado pelos membros mais aptos.

Durante o estágio esta reunião teve lugar regularmente entre o estagiário e o tutor, permitindo ao estagiário evoluir de forma consistente.

## 3.3. Análise de Riscos

É importante antecipar problemas que poderão afetar o projeto, de forma a prevenir o seu aparecimento ou a ter um plano para os ultrapassar com esforço mínimo. Para tal é necessário fazer uma análise de riscos. Nesta secção irão descrever-se os riscos encontrados para este projeto, assim como os planos de mitigação dos mesmos. Níveis de prioridade definidos são: baixo, moderado, alto e extremo.

### Risco 1 - Desenvolvimento numa plataforma sem experiência (*Android*)

Uma vez que o estagiário nunca tinha trabalhado em desenvolvimento *mobile*, *Android*, *iOS* ou *Windows Phone*, a sua experiência em aplicações móveis era reduzida o que poderia levá-lo a sentir dificuldade em adaptar-se à forma de programação.

Prioridade: Muito Alta

Plano de Mitigação

- Treino através de *tutorials*, curso *Android* da WIT Software, exemplos e livros sobre desenvolvimento *Android*;



- Estimativa de complexidade, acompanhado do tutor, de forma a evitar erros que comprometam o desenvolvimento dos *sprints*;
- Acompanhamento próximo do tutor, no início do desenvolvimento, de modo a tornar o processo de aprendizagem mais rápido.

### **Risco 2 – Adaptação à Arquitetura RCS+ já existente**

A solução a desenvolver para o estágio insere-se numa aplicação Android complexa com vários módulos e práticas de programação já definidas, pelo que a adaptação pode ser demorada.

Prioridade: Média

Plano de Mitigação

- Estudo de toda a documentação existente sobre a aplicação RCS+;
- Acompanhamento próximo do tutor no início do desenvolvimento para tornar o processo de adaptação mais rápido.

### **Risco 3 - Atualizações à *Framework* Android**

O desenvolvimento de Android está em constante atualização e cada atualização de versão traz várias alterações às APIs. Isto pode tornar algum código criado para o módulo obsoleto.

Prioridade: Baixa

Plano de Mitigação

- Em caso de atualização da API verificar se alguma implementação do módulo utiliza versões obsoletas do código e gerar *User Stories* para os resolver.

### **Risco 4 - Novas Especificações RCS**

Funcionalidades de condução não estão definidas em nenhuma especificação RCS neste momento, mas é possível que no futuro um módulo de condução seja introduzido na especificação, uma vez que é um tema ligado às comunicações móveis. Se isso suceder a solução deste estágio poderá não se enquadrar naquilo que é esperado pela GSMA.

Prioridade: Baixa

Plano de Mitigação

- A WIT Software faz parte da GSMA, isto significa que é uma das entidades que ajudam na definição de novas especificações RCS. É possível que haja a sugestão para que a nova especificação do módulo siga os moldes da solução deste estágio



## Capítulo 4

# Arquitetura e Desenvolvimento

Neste capítulo apresenta-se a arquitetura da solução desenvolvida e a forma como cada funcionalidade principal foi estudada, incluindo problemas encontrados e soluções para os mesmos.

A grande maioria da informação acerca da arquitetura do protótipo é considerada confidencial pela empresa, pelo que a informação mais detalhada foi remetida para o **Apêndice B – Arquitetura e Desenvolvimento**.

Dado que o módulo de condução desenvolvido se insere na aplicação RCS+ para *Android*, é importante proceder à leitura do capítulo 2.5 do estado da arte - Conceitos de *Android*, por forma a facilitar o entendimento das referências aos componentes específicos da plataforma *Android* que são abordados no decorrer deste capítulo.

Na secção **4.1 WIT Mobile Communicator** é apresentada a estrutura de rede onde se inserem as aplicações RCS. Na secção **4.2 Arquitetura Android RCS+** é abordada a estrutura da aplicação RCS+, especificamente os diferentes módulos que a compõem, como estão organizados e como comunicam com a rede através do WIT RCS *Stack*. Nos capítulos **4.3 Módulo de Condução** e **4.4. Funcionalidades do Módulo** apresenta-se a arquitetura e a forma como foram desenvolvidas as funcionalidades do projeto.

## 4.1. WIT Mobile Communicator

A figura 30 representa a aplicação RCS da WIT implementada numa rede IMS de operador de comunicação. As aplicações *mobile* e computador ligam-se através da *Internet* (*RCS Stack*) aos servidores da WIT, como o WCAS (*WIT Communications Server*), CMS (*Central Message Store*) ou NAB (*Network Address Book*) que estão instalados nas infraestruturas de rede IMS dos operadores.

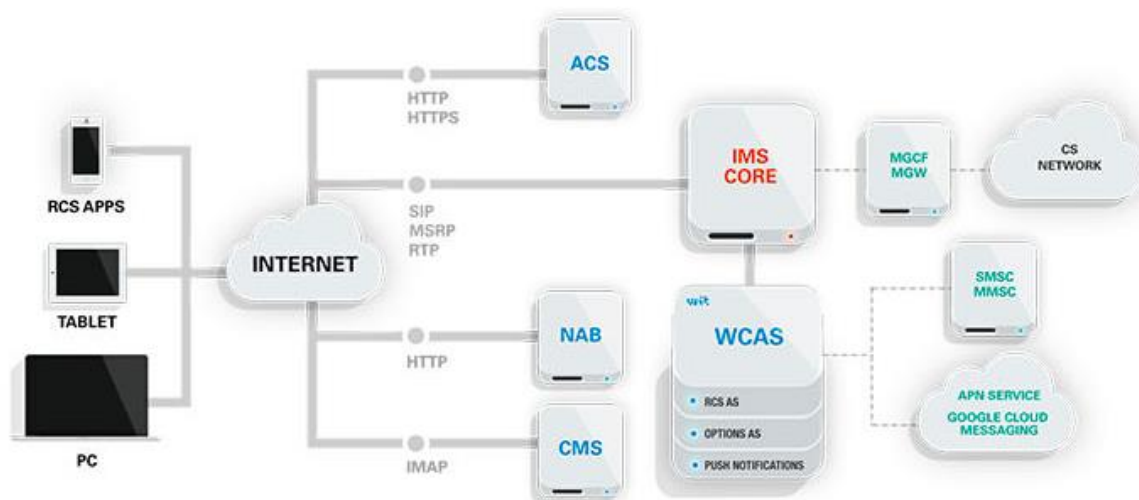


Figura 30 - WIT RCS na infraestrutura IMS

A ligação entre aplicações RCS e a rede IMS é alcançada através de um módulo de comunicação, proprietário da WIT Software, designado por *WIT RCS Stack*. As aplicações *mobile* e PC comunicam com o *RCS Stack*, ficando este com a responsabilidade de gerir toda a comunicação entre aplicações e rede.

O *RCS Stack*, cuja estrutura está representada na figura 31, é desenvolvido na linguagem de programação C++ por razões de portabilidade e performance. Para que a aplicação RCS+ para *Android*, utilizada neste estágio, possa comunicar com a biblioteca, existe um *wrapper* que faz o mapeamento entre o código Java e o código C++ do *RCS Stack* recorrendo ao *Java Native Interface* (JNI).

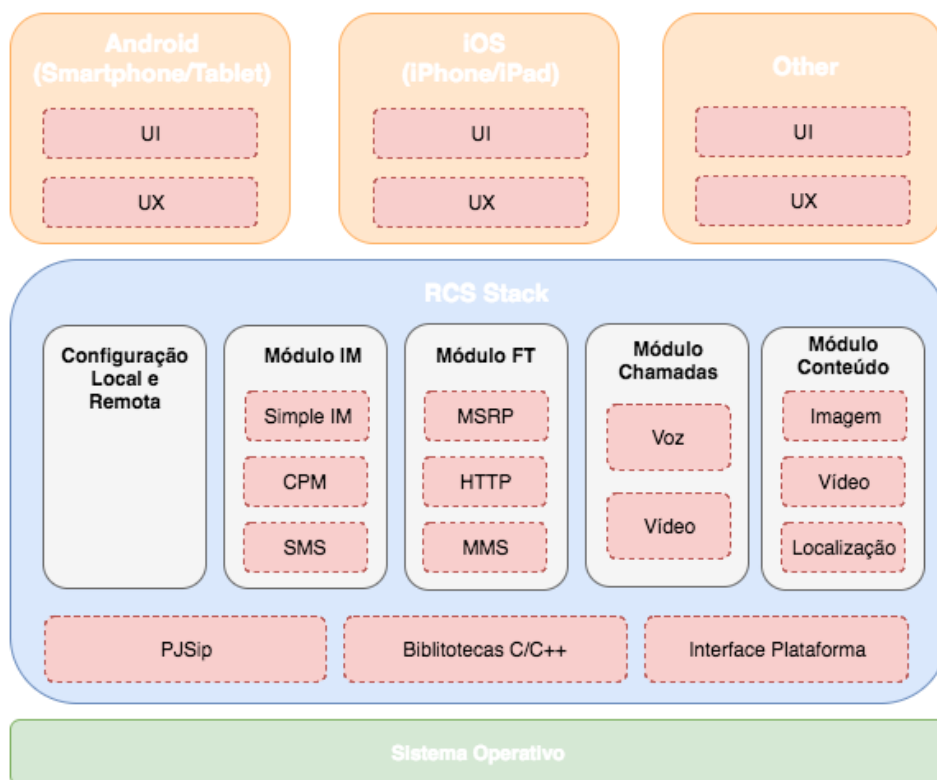


Figura 31 - Arquitetura alto nível do WTT RCS Stack

O RCS *Stack* apresenta cinco principais módulos, o módulo de configuração, de mensagens (IM), de transferência de ficheiros (FT), chamadas e conteúdo. Estes módulos comunicam através da interface específica da plataforma com as aplicações *Android*, *iOS*, entre outras, para fornecer dados de comunicação.

## 4.2. Arquitetura Android RCS+

A aplicação RCS+ representa a *UI/UX Android* do WIT Mobile Communicator. Esta aplicação está dividida em vários módulos, tendo sido acrescentado um novo, designado por *Car Mode*, na realização deste projeto. Foi também necessário alterar alguns módulos já existentes, de forma a completar funcionalidades fora dos menus de condução. O objetivo desta secção é explicar os principais módulos que compõem a aplicação e a forma como estes estão organizados.

### 4.2.1. Módulos

Os principais módulos da aplicação estão representados na figura 32, cada módulo é constituído por, pelo menos, um *Manager* que contém a lógica desse módulo. Pode também conter uma componente de *UI*, caso seja necessário apresentar visualmente informação ao utilizador. Para o estágio foi necessário alterar os módulos já existentes de *Notifications*, *Chat*, *Settings* e *Capabilities*.

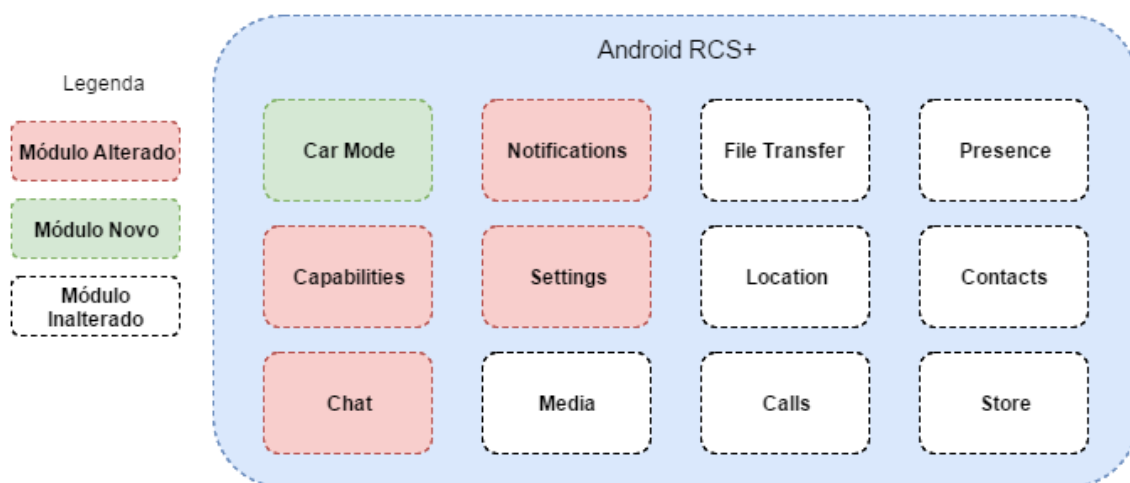


Figura 32 - Principais Módulos RCS+

Uma vez que o *Manager* de cada módulo é um *Singleton*, este pode ser acedido pelos outros módulos através da sua instância global, fazendo uso dos métodos específicos. As funcionalidades mais importantes para o estágio, que cada módulo disponibiliza, são:

**Calls** – Início de chamada de voz ou chamada sobre IP (*VoIP*).

**Capabilities** – Envio e pedido de *Capabilities* a utilizadores. *Capabilities* refere-se às funcionalidades que o dispositivo ou utilizador tem acesso, por exemplo, chamada sobre IP.

**Media** – Gestão da gravação de ficheiros de áudio, incluindo funções para iniciar, pausa ou parar a reprodução de um ficheiro.

**Contacts** – Acesso à lista de contactos do utilizador.

**FileTransfer** – Envio e receção de ficheiros de áudio.

**Notifications** – Geração de notificações *Android*.

#### 4.2.2. Estrutura dos Módulos

A figura 33 apresenta o diagrama de camadas dos módulos da aplicação RCS+. São constituídos por três camadas: apresentação, lógica e dados. A comunicação com o *RCS Stack* é geralmente executada pela camada de lógica de cada módulo, através dos *Managers* e *Controllers*, podendo ser também diretamente consultada pela camada de apresentação através da chamada de métodos síncronos.

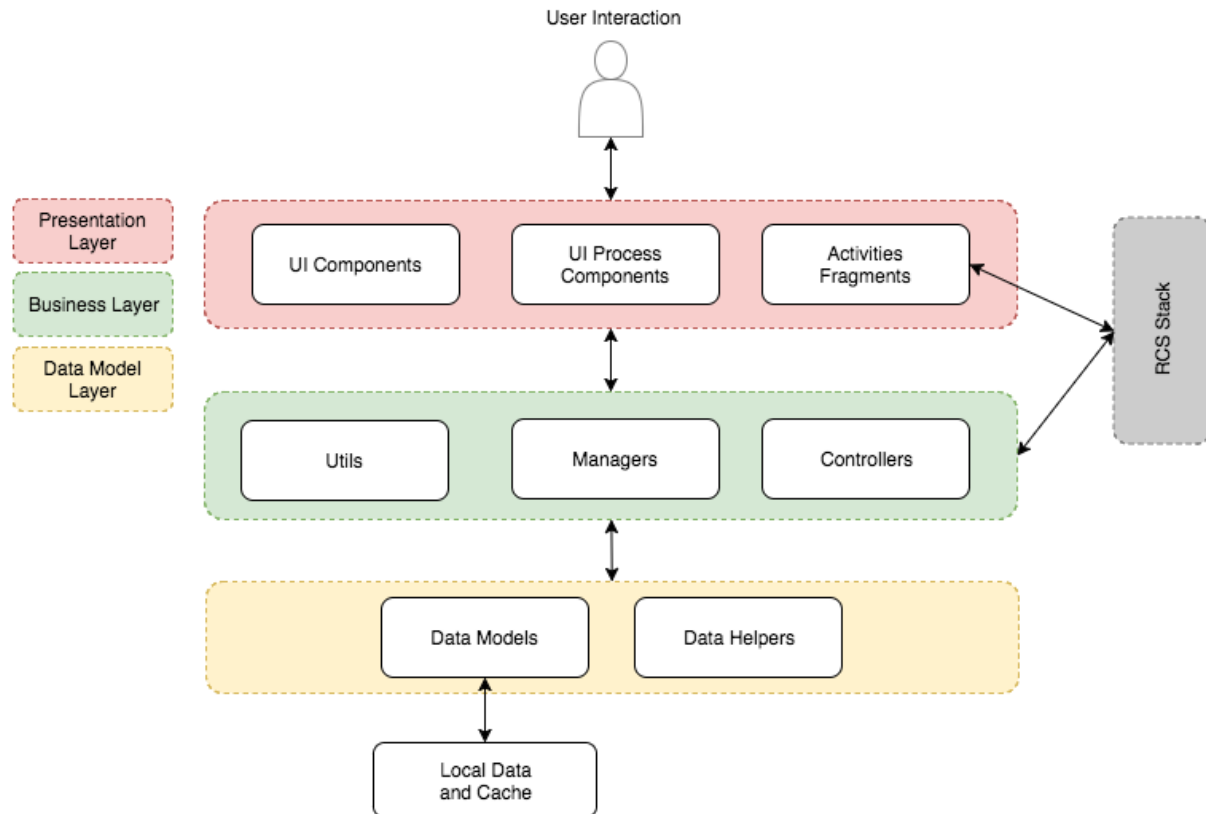


Figura 33 - Diagrama de Camadas Módulos

#### Camada de Apresentação

- Componentes *UI* - Conjunto de ficheiros XML – *layout*, estilos, *strings*, etc.
- *UI Process Components* - Corresponde às vistas que estendem as base de *Android*;
- *Activities/Fragments* - Componentes que fazem a gestão do código de apresentação, incluindo *layouts* ativos num determinado momento e ações que o utilizador pode executar.

## Camada de Negócio

- *Managers* – Todos os módulos têm um *manager* que é responsável pelas ações independentes da *UI*, a lógica do módulo. Geralmente subscrevem eventos do *RCS Stack* e tratam a informação recebida por essa camada, assim como gerem estados em *cache*.
- *Controllers* – Os *Controllers* implementam funcionalidades específicas como, por exemplo, transferência de ficheiros. Permitem que uma *Activity* os possa instanciar e, desta forma, podem interagir diretamente com a *UI*.
- *Utils* – Funções auxiliares do módulo.

## Camada de Dados

- *Local Data and Cache* – Corresponde às bases de dados locais da aplicação. Contém, entre outros, informação dos contactos do dispositivo e imagens dos contactos.

### 4.2.3. Comunicação com o RCS Stack

A comunicação entre os módulos da aplicação RCS+ (incluído o módulo de condução) e o *RCS Stack* pode ser efetuado de três formas diferentes:

#### Getters

Operações síncronas que podem ser executadas a partir de uma *activity/controller* diretamente.

#### Ações

Eventos assíncronos que podem demorar algum tempo a executar. Para que a aplicação não fique a aguardar a resposta e bloquee o utilizador, devem ser efetuados numa nova *thread*.

#### Eventos

Subscrições que notificam os *Managers* quando alguma alteração sucede no *RCS Stack*. Estes eventos são normalmente registados nos métodos *onResume()* e *onPause()* das *activities*, para que não haja a receção de eventos quando o utilizador não está a interagir com o ecrã afetado pelos mesmos.

A figura 34 representa alguns estados de uma *activity*. Os eventos *onResume()* e *onPause()* são os mais adequados para a subscrição de eventos, dado que é entre eles que o utilizador está a interagir com o ecrã controlado pela *activity* ativa.



Figura 34 – Ciclo de vida compacto de uma *Activity*



### 4.3. Módulo de Condução

O módulo de condução, criado para a realização do estágio, segue a mesma estrutura dos restantes módulos da aplicação, em conformidade com as boas práticas de *Android*.

As principais camadas e componentes criadas ou utilizadas pelo módulo são:

- Camada de apresentação com as componentes responsáveis pela apresentação da interface gráfica ao utilizador;
- Camada lógica que gere toda a interação entre a *UI*, o *RCS Stack* e a base de dados local;
- Camada de dados que persiste os dados no dispositivo: lista de utilizadores a ser notificados, dispositivos *bluetooth* guardados e pontuações.
- Camada *RCS Stack* de onde provêm os eventos, tais como chamadas, mensagens ou *capabilities* de um utilizador.

#### 4.3.1. Camada de Apresentação

A camada de apresentação é constituída por um conjunto de *Activities*, *Fragments*, *Adapters* e elementos de *UI* como *layouts XML* ou imagens cujo objetivo é a apresentação da interface ao utilizador. Desta camada fazem parte os componentes:

##### **Main Activity/Fragment**

*User Interface* do ecrã inicial do menu de condução.

##### **Messages Activity, Fragment e Adapter**

*User Interface* do ecrã de mensagens recebidas em condução.

##### **Reply Audio Activity/Fragment**

*User Interface* do ecrã onde se grava e envia uma mensagem de áudio em condução.

##### **Calls Adapter**

*User Interface* do menu de chamadas em condução.

##### **Notifications Activity/Adapter**

Menu, nas definições de *Car Mode*, onde se apresenta a lista de todos os contactos do utilizador. Pode um determinado contacto ser marcado para notificação ou removido.

##### **Floating Button**

Vista que permite ao utilizador aceder rapidamente aos menus de condução, quando está em condução mas fora da aplicação.

##### **Car Mode Widget**

*User Interface* do *widget* de acesso rápido ao *Car Mode*.

##### **Bluetooth Activity/Fragment**

Menu de gestão de dispositivos *bluetooth* para utilização em condução.

##### **Scores Activity/Adapter**

Menu de apresentação de pontuações de condução.

### **Start Options Activity**

Ecrã onde, após ser detetado que o dispositivo está em condução, o utilizador pode seleccionar se é condutor ou passageiro.

### **Missed Notifications Adapter**

Interface que apresenta as notificações de sistema ou outras aplicações perdidas enquanto se conduz.

## **4.3.2. Camada Lógica**

A camada lógica contém as componentes do módulo que fazem a gestão de dados e informam os elementos da camada de apresentação de alterações necessárias às suas vistas. É também nesta camada que se faz o tratamento de dados provenientes do *RCS Stack*.

### **Car Mode Manager**

*Singleton* responsável pela gestão do módulo. As suas principais funções são: Gestão dos eventos recebidos do *RCS Stack*; Registo de *capabilities*; Controlo de acesso à base de dados local; Gestão das mensagens recebidas e dos utilizadores em condução; Informação às componentes de *UI* de alterações necessárias.

### **Text To Speech Manager**

O *Text-To-Speech Manager* é o responsável pela transformação de mensagens de texto em áudio através da utilização da API de *text-to-speech* do *Android*.

### **Notifications Service**

*Service* de notificações que estende o *NotificationListenerService* nativo do *Android* para receber notificações de todo o sistema. Quando o utilizador está em modo de condução, as notificações recebidas são bloqueadas para evitar ao máximo a distração do condutor.

### **Driving Detection**

Componente responsável pela aplicação e gestão do algoritmo de deteção de condução.

### **Bluetooth Broadcast Receiver e Wired Headset Broadcast Receiver**

*Broadcast Receivers* que tratam da gestão de ligações dos dispositivos mãos-livres.

### **Utils**

Funções auxiliares do módulo.

## **4.3.3. Camada de Dados**

A camada de dados contém o **Database Manager** cuja responsabilidade é a gravação local das pontuações, da lista de dispositivos *bluetooth* e da lista de utilizadores a serem notificados quando se entra em condução. O motor de base de dados utilizado é o *SQLite*.

## 4.4. Funcionalidades do Módulo

Nesta secção apresenta-se informação de desenvolvimento das diferentes funcionalidades do protótipo, para obter mais detalhe recomenda-se a leitura do **Apêndice B - Arquitetura e Desenvolvimento**.

### 4.4.1 Gestão de mensagens e ficheiros de áudio

Uma das principais funcionalidades do ponto de vista do utilizador do módulo de condução é a possibilidade de receção e leitura de mensagens enquanto conduz. Para o fazer de forma segura foi necessário implementar uma interface que oferece o mínimo de interação possível entre o condutor e a aplicação, ainda assim mantendo a possibilidade de saber o conteúdo das mensagens.

O *Manager* é responsável pela gestão das mensagens, incluindo se são ou não apresentadas, se estão lidas ou quais as informações a apresentar. O *RCS Stack* trata da comunicação com o servidor, passando as informações de mensagens ou clips de áudio ao *Manager* do módulo. O *Manager*, ao receber informação de novas mensagens, alerta as componentes de *UI* à escuta através de *callbacks*, permitindo que estas possam atualizar as suas vistas em tempo real.

Para além da criação das funcionalidades no módulo de condução foi também necessário alterar os módulos de *File Transfer* e *Chat* da aplicação base com vista a evitar a criação de notificações quando o utilizador está em condução.

### 4.4.2 Text-To-Speech

Com o propósito de tornar possível a leitura de mensagens recebidas em alta voz, a aplicação faz uso do *text-to-speech* nativo do *Android*. No capítulo de estado da arte foram abordados diferentes motores de *text-to-speech* tendo sido escolhido o *Google Text-To-Speech* para o estágio por: (i) estar presente nativamente na maioria dos terminais *Android* recentes, evitando assim a necessidade de instalação de aplicações extra e (ii) apresentar boa qualidade de reprodução de voz.

Para que as funcionalidades de *text-to-speech* estejam acessíveis a toda a aplicação, foi criado um *wrapper* sobre a API nativa de *Android*. Os vários módulos da aplicação, onde se inclui o módulo de condução, registam eventos e fazem os pedidos de leitura ao *wrapper*, estando este responsável pela gestão da API nativa.

### 4.4.3 Detecção de Condução via Sensores

A deteção de condução é feita recorrendo a *software* de reconhecimento de atividade do *Android*, designado *Google Activity Recognition*. A escolha deste *software* é abordada em detalhe no capítulo **5. Análise de Condução**.

Este mecanismo recolhe dados dos sensores e fornece periodicamente informação acerca da atividade mais provável do dispositivo, assim como a confiança nessa atividade.

Após a recolha destes dados em condução, ficou claro que o início e saída do módulo de condução não poderia depender de uma única atividade reportada, já que ocorriam algumas variações de resultados.

Para evitar falsos positivos do *software* de reconhecimento são guardadas algumas atividades permitindo uma análise sobre um histórico de atividades e não apenas sobre uma atividade isolada.

Após receber uma atividade da API de reconhecimento de atividade, o algoritmo irá iterar sobre um pequeno número de atividades recebidas e verificar se os dados que apresentam indicam que o utilizador começou a conduzir ou parou de conduzir.

#### *Em condução*

Caso o estado do algoritmo seja a condução, vão ser analisadas as últimas atividades recebidas, sendo a quantidade definida por parâmetro.

Para o algoritmo considerar que o utilizador parou de conduzir, é necessário que todas as atividades que são analisadas sejam diferentes de *IN\_VEHICLE* ou alternativamente sejam *IN\_VEHICLE* com uma confiança baixa.

Caso tal suceda, o dispositivo irá automaticamente fechar os menus de condução se estiverem em exposição.

#### *Fora de condução*

Caso o estado atual do algoritmo seja fora de condução, a análise vai ser feita com o objetivo de verificar se o dispositivo entrou em condução, para isso são avaliados um conjunto de últimas atividades reportadas.

Caso todas as atividades analisadas sejam *IN\_VEHICLE* com um nível de confiança alto o algoritmo considera que o dispositivo começou a conduzir.

### 4.4.4 Detecção de Condução via Bluetooth

Uma das formas mais fáceis e convenientes para se iniciar o módulo de condução é a ligação de dispositivos *bluetooth*, presente num grande número de automóveis lançados nos últimos anos ou *headsets* adquiridos.

Ligar a aplicação em modo de condução, após emparelhar com dispositivos *bluetooth*, garante que o utilizador irá fazer a viagem completa em segurança, deixando de depender da deteção via sensores que pode não ser imediata.

De modo a realizar esta funcionalidade o primeiro desafio foi perceber se existe forma de distinguir os dispositivos *bluetooth* presentes nos carros, de outros dispositivos. Para tal recorreu-se à API de *Bluetooth* de *Android* tendo-se verificado a existência de 49 tipos diferentes de dispositivos que a API consegue detetar, incluindo a categoria *AUDIO\_VIDEO\_CAR\_AUDIO*. Após alguns testes em carros, descobriu-se que a API não é fiável, tendo detetado *bluetooth* originais de carro como *AUDIO\_VIDEO\_HANDSFREE*, entre outros.

Foi decidido criar uma lista de dispositivos *bluetooth* específico para a condução, podendo o utilizador selecionar quais os dispositivos que fazem o módulo de condução iniciar.

#### 4.4.5 Floating Button

O *Floating Button* é um botão que flutua no ecrã e permite ao utilizador voltar rapidamente aos menus de condução. Esta é uma funcionalidade importante para evitar distrações extra caso ocorram comportamentos não esperados, tais como uma aplicação forçar a sua vista enquanto se está em condução, ou o condutor clicar, sem intenção, no botão *Home* ou *Recents* do *Android*.

No âmbito do estágio, esta funcionalidade apresentou dificuldade acrescida não prevista, uma vez que o *Android* não nos permite saber o exato momento em que um ecrã da aplicação (*activity/fragment*) deixa de estar visível no ecrã para dar lugar a uma aplicação diferente. Foi necessário criar uma variável de controlo no **Manager**.

Sempre que ocorre uma mudança de *Activity* durante a condução é feita a marcação na variável de controlo. Assim, caso estejamos em condução e uma *activity* em condução seja escondida, sem a variável estar marcada, significa que o utilizador deixou de ter o menu de condução visível no ecrã.

#### 4.4.6 Perceção de Condução

Para além de oferecer meios para o condutor conseguir receber e enviar mensagens e chamadas, um dos objetivos principais do projeto é providenciar aos utilizadores, que não estão a conduzir, informação sobre o estado de condução dos seus contactos. Deste modo, um utilizador antes de contactar alguém a conduzir, vai ter a noção de que será perigoso para o recetor da mensagem a leitura da mesma.

A implementação desta funcionalidade exigiu o registo e envio de uma nova *capability* que indica se o utilizador está a conduzir. Com essa informação a aplicação pode adicionar símbolos que permitem identificar com facilidade os seus contactos em condução.

#### 4.4.7 Opções de envio a condutor

Uma das funcionalidades inovadoras em relação às aplicações concorrentes é a possibilidade de oferecer diferentes opções de comunicação segura aos utilizadores da aplicação que pretendem contactar um utilizador a conduzir.

Quando um utilizador escreve uma mensagem e seleciona o botão de enviar são-lhe apresentadas alternativas ao envio da mensagem, consciencializando, desta forma, o remetente para o perigo que poderá ser a leitura da mensagem por parte do condutor.

Para além da opção de enviar a mensagem normalmente, são oferecidas as seguintes alternativas ao utilizador:

- Ser notificado quando o condutor parar de conduzir;
- Enviar a mensagem quando o condutor parar de conduzir;
- Enviar uma mensagem de áudio em vez de uma mensagem de texto;
- Iniciar uma chamada de voz. Esta opção só está disponível se durante o pedido de *capabilities* efetuado no *Base Chat Fragment*, a *capability* `HANDS_FREE` estiver presente.

Caso o utilizador decida pela primeira opção, envio da mensagem, o procedimento da funcionalidade segue o mesmo fluxo de envio de mensagem que seguiria se o recetor da mensagem não estivesse a conduzir, sendo o *Chat Controller* responsável pelo envio da mensagem. Um dos requisitos da funcionalidade foi, independentemente da escolha do utilizador, que o teclado de escrita estivesse sempre visível enquanto as opções de envio estivessem igualmente visíveis. A abordagem inicial para a implementação desta funcionalidade foi a criação de um *custom dialog* a ocupar o ecrã completo, mas, por limitação da plataforma *Android*, mostrar um menu *fullscreen* obriga o teclado a desaparecer.

Foi então decidido utilizar o *fragment* descrito acima que deve fazer desaparecer as opções de envio sempre que o utilizador escreva no teclado. Para tal ser possível, alterou-se a componente *Chat Input Fragment* para alertar o *Manager* de condução sempre que o utilizador escreva no teclado, ficando este responsável por alertar os componentes que subscrevem o evento de escrita no teclado.

Desta forma, o *Chat Input Fragment* avisa o *Manager* quando o utilizador escreve no teclado. Este, por sua vez, envia um *callback* ao *Car Mode Options Fragment* que procede à alteração da *UI* de modo a esconder as opções de envio de mensagem.

#### 4.4.8. Resposta com mensagem de áudio

Em condução, a interação entre o dispositivo e o condutor deve ser mantida no mínimo possível. Escrever uma mensagem é uma tarefa impossível de fazer em segurança, todavia existem outras formas de comunicar enquanto se conduz, como por exemplo, chamada de voz com recurso a um dispositivo mãos-livres ou gravação de uma mensagem de voz.

Esta funcionalidade foi criada a pensar nos condutores que querem responder a uma mensagem recebida, de forma breve ou, caso não possuam um dispositivo mãos-livres, para efetuar uma chamada de voz.

A funcionalidade tem início quando o condutor seleciona a opção ‘*Reply*’ no menu de mensagens. Esta componente faz a chamada dos elementos que fazem a gestão do ecrã de gravação de áudio, o *Reply Audio Activity/Fragment*.

#### 4.4.9. Gestão de dispositivos mãos-livres

Ter um dispositivo mãos-livres conectado ao dispositivo, em condução, tem forte implicação na forma como o utilizador pode usar a aplicação. Um dispositivo mãos-livres significa que existe segurança e legalidade para comunicar enquanto se conduz, pelo que foi considerado de extrema importância a deteção destes dispositivos com o objetivo de apresentar funcionalidades extras ao utilizador.

No âmbito do projeto foram definidos como dispositivos mãos-livres todos os *headsets bluetooth* e auscultadores com microfone.

Quando o *Android* deteta a ligação ou desconexão de um dispositivo mãos-livres dispara um *Intent* no sistema que é recebido pelas aplicações à escuta. Os *broadcasts receivers* quando recebem um evento do sistema informam o *Manager* do módulo que controla o estado das ligações e informa as componentes de *UI* à escuta sobre as alterações.

#### 4.4.10. Serviço de Notificações

A receção de notificações enquanto se conduz, especialmente se for frequente, é prejudicial à condução, pois distrai o condutor recorrentemente. Estando a conduzir, as notificações são maioritariamente irrelevantes, já que o condutor não tem condições para responder adequadamente e em segurança ou interagir com a aplicação que gerou a notificação.

Para melhorar a segurança do condutor e evitar a distração com notificações, foi implementada a funcionalidade de bloqueio de notificações enquanto o módulo de condução está ativo, sendo posteriormente mostrado ao utilizador as notificações perdidas.

Caso uma aplicação ou o sistema envie uma notificação e o módulo de condução esteja ativo, as informações dessa notificação são guardadas numa estrutura do *Notifications Service*. Quando o condutor pára de conduzir, o Manager informa o *Notifications Service*, ficando a cargo desta componente informar o utilizador que tem notificações perdidas.

#### 4.4.11. Notificação de Início de Condução

Uma das áreas onde a aplicação RCS+ pode inovar, em relação aos concorrentes, é na interação entre contactos que estão a conduzir e os que não estão. A possibilidade de um utilizador saber que os seus contactos estão a conduzir permite a este ter consciência do perigo que é a comunicação e assim evitá-lo.

Uma das funcionalidades concebidas para esta interação foi a possibilidade de enviar uma notificação aos contactos quando se entra em condução. Uma vez que não é do interesse informar toda a lista de contactos, criou-se um menu nas definições de *car mode* onde se podem seleccionar os contactos a receber a notificação.

O *Manager* desempenha um papel fundamental nesta funcionalidade. Quando a aplicação inicia dá-se o registo de *capability NOTIFICATION* que fica registada para ser recebida de outros contactos. Caso seja recebida, significa que o contacto que enviou entrou em condução e deve disparar uma notificação de início de condução.

Do ponto de vista do utilizador que conduz, quando este inicia a condução, o *Manager* informa a *Capability API* de que deve registar a *NOTIFICATION* para envio. Após esse registo, deve percorrer a lista de utilizadores a notificar e requisitar troca de *capabilities* com todos os utilizadores.

#### 4.4.12. Pontuações (Gamification)

Uma das formas encontradas para incentivar a utilização do módulo de condução, e consequentemente, diminuir o número de utilizadores de *smartphone* a conduzir, foi adicionar uma componente de jogo.

O objetivo da funcionalidade é premiar os condutores que fazem uso das diferentes opções do módulo de condução. Através da comparação de resultados cria-se alguma rivalidade que promove o aumento do tempo de utilização do módulo.



O *Manager* do módulo é o ponto mais importante desta funcionalidade. Estabelece a conexão entre as ações do utilizador, o registo da *capability* para informar os restantes contactos da pontuação e a base de dados local onde se guardam os registos de pontos de todos os contactos.

Sempre que ocorre uma opção que deve ser premiada, por exemplo, a leitura de uma mensagem via *text-to-speech*, o *Manager* é informado dessa ação e executa duas ações:

- Registo da *capability SCORE* com a pontuação atualizada;
- Gravação na base de dados local da nova pontuação.

#### 4.4.13. Iniciar Chamada em Condução

Para um condutor com um dispositivo mãos-livres ligado, iniciar chamadas é uma funcionalidade muito importante, já que é uma das poucas interações seguras que pode fazer com o seu dispositivo. Assim, é importante garantir que o condutor tem uma forma fácil e rápida de iniciar uma chamada, com o mínimo de toques e interações com o ecrã.

A possibilidade de fazer chamadas foi colocada no menu inicial de condução, onde o utilizador, caso tenha um dispositivo mãos-livres ligado, pode fazer *swipe up* ou carregar no ícone colocado na zona inferior do ecrã. Ao fazê-lo é mostrada uma lista que contém os últimos contactos com os quais efetuou chamadas e os seus contactos favoritos. Carregar no *avatar* de um contacto irá estabelecer chamada de voz com o mesmo.

#### 4.4.14. Indicador de Bateria

A informação do nível de bateria é apresentada ao condutor no ecrã inicial da aplicação, gerido pela componente *Main Activity/Fragment*.

A realização desta funcionalidade exigiu a criação de um *Timertask* na *Main Activity* cujo objetivo é pedir periodicamente ao sistema, o nível de bateria do dispositivo. Este pedido é feito recorrendo ao *Intent.ACTION\_BATTERY\_CHANGED* do *Android*.

Dependendo do nível, 0 a 14%, 15% a 49%, 50% a 84% e 85% a 100% é apresentado um ícone diferente no indicador de bateria.



## Capítulo 5

### Análise de Condução

O algoritmo de deteção de condução é de extrema importância para o projeto, uma vez que, uma adequada implementação implicará um aumento de probabilidade das funcionalidades de condução serem utilizadas, consequência do início automático do modo de condução.

Para implementar um algoritmo de deteção de condução é necessário que o dispositivo consiga reconhecer quando está num automóvel. Para a plataforma *Android* existem dois principais mecanismos de reconhecimento de atividade, o *Google Activity Recognition* e o *PathSense Activity Recognition*.

De forma a descobrir qual o mecanismo que reporta valores mais fiáveis, e assim ser utilizado no protótipo, foi decidido fazer um estudo de comparação entre os dois mecanismos. Para tal, foi criada uma aplicação cujo fim é o recolher dados de ambos os mecanismos e posteriormente fazer a análise dos mesmos, verificando qual produz resultados mais fiáveis.

O objetivo deste capítulo é informar o leitor sobre os procedimentos efetuados para selecionar o mecanismo escolhido, incluindo o estudo concebido, o mecanismo de reconhecimento de atividade utilizado no protótipo e os valores utilizados no algoritmo de deteção.

Na secção **5.1. Recolha de Dados** aborda-se a aplicação criada para recolha de dados dos mecanismos de reconhecimento.

Na secção **5.2. Análise** apresentam-se os dados recolhidos e faz-se uma comparação entre os valores obtidos pelos sistemas de reconhecimento.

Na secção **5.3. Conclusão** discute-se as razões que levaram à escolha do mecanismo de reconhecimento de atividade da *Google* e os valores óptimos encontrados para o algoritmo de deteção.

## 5.1. Recolha de Dados

A recolha de dados foi realizada recorrendo a uma aplicação *Android*, independente do protótipo desenvolvido neste estágio, cujo objetivo foi exclusivamente obter dados dos dois *software* de reconhecimento. A aplicação foi desenvolvida no decorrer do segundo semestre.

Na figura 35 pode ser visto o menu principal da aplicação.

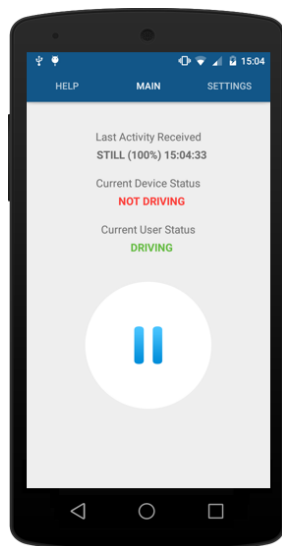


Figura 35 - Menu Principal da aplicação de recolha de dados

Após ser aberta pela primeira vez a aplicação começa a recolher dados sobre a atividade do dispositivo até ser desinstalada. Ambos os *software* de reconhecimento estão, simultaneamente, a recolher dados na aplicação, ficando guardada na base de dados local *SQLite*, informação sobre as atividades que recolhem.

O utilizador da aplicação foi instruído para assinalar na aplicação quando está a conduzir, através da seleção do botão *Play*, e quando o deixa de fazer, através da seleção do botão *Pause*. Desta forma os *logs* irão identificar quando o utilizador está realmente a conduzir, de forma a ser possível, posteriormente, fazer a comparação entre a atividade recolhida pelos mecanismos e a verdadeira atividade do utilizador.

## 5.2. Análise de Dados

A aplicação foi distribuída por cinco utilizadores, tendo sido recolhidos dados durante cinco dias. Um excerto de um ficheiro de dados recolhido pode ser visto na Tabela 3. Neste exemplo, observa-se que o *software* de reconhecimento detetou às 02:15:51 do dia 15 que o utilizador estava a andar (*ON\_FOOT*) com 54% de certeza. Na atividade seguinte, recebida 11 segundos depois a atividade reportada passou a abanar (*TILTING*) com 100% de certeza. Na última atividade o reconhecimento calculou que o utilizador está num veículo com 69% de certeza. A última coluna representa a atividade real de acordo com o que o utilizador definiu no momento em que os dados estavam a ser gravados. Entre a segunda e a terceira atividade da tabela o utilizador entrou em condução.

| Atividade  | Confiança | Data           | Conduzir |
|------------|-----------|----------------|----------|
| ON_FOOT    | 54        | 15/05 02:15:51 | NO       |
| TILTING    | 100       | 15/05 02:16:02 | NO       |
| IN_VEHICLE | 69        | 15/05 02:16:17 | YES      |
| ...        | ...       | ...            | ...      |

Tabela 3 - Excerto de um ficheiro de dados

As atividades possíveis de ser recolhidas dependem do mecanismo de reconhecimento, para o *Google Activity Recognition* é possível detetar: em veículo (*IN\_VEHICLE*), a pé (*ON\_FOOT*), de bicicleta (*ON\_BICYCLE*), a correr (*RUNNING*), parado (*STILL*), a abanar (*TILTING*), a andar (*WALKING*) ou caso não consiga calcular a atividade *UNKNOWN*. As atividades detetadas pelo *PathSense Activity Recognition* são: a andar (*WALKING*), a conduzir (*DRIVING*, *IN\_VEHICLE*), a segurar (*HOLDING*), a abanar (*SHAKING*), a pé (*ON\_FOOT*) ou parado (*STILL*).

A confiança será sempre um valor entre 0 e 100% em qualquer biblioteca.

Na Tabela 4 apresentam-se o total de dados recolhidos por cada utilizador. O cômputo geral representa a amostra total de dados estudada.

| Utilizador | Dispositivo      | Número de Atividades |
|------------|------------------|----------------------|
| A          | BQ Aquaris E5 HD | 8246                 |
| B          | One Plus X       | 6397                 |
| C          | Samsung S3       | 3708                 |
| D          | Samsung S5       | 41702                |
| E          | One Plus X       | 21425                |

Tabela 4 - Estatísticas de Dados recolhidos

Após a observação dos dados ficou claro que a deteção de ambos os mecanismos é bastante irregular, acontecendo com alguma frequência a deteção de atividade *IN\_VEHICLE* com alta probabilidade de confiança quando o dispositivo estava apenas a ser transportado na mão.

Tendo em conta as irregularidades dos dados reportados, foi necessário perceber que condições teriam de ser estabelecidas para obter certeza de que o dispositivo passou para o estado de condução ou saiu do mesmo. Por exemplo, três atividades *IN\_VEHICLE*, reportadas consecutivamente com alta confiança indicará sempre que o utilizador iniciou condução? Estes seriam valores interessantes a utilizar no protótipo?

Para descobrir os parâmetros ideais foi criado um programa que recebe como *input* um ficheiro com dados recolhidos de atividades e calcula o número de atividades *IN\_VEHICLE* consecutivas detetadas fora de condução. Na Tabela 5 pode visualizar-se um excerto do resultado da análise de um ficheiro de dados (*dados1.log*).

| Confiança  | 20% |   |   |   |   |   |   |
|------------|-----|---|---|---|---|---|---|
| Ficheiro   | 1   | 2 | 3 | 4 | 5 | 6 | 7 |
| dados1.log | 6   | 1 | 0 | 0 | 0 | 0 | 0 |

Tabela 5 - Exemplo de resultados do programa de análise de dados

A tabela deve ser lida da seguinte forma, caso sejam consideradas todas as atividades *IN\_VEHICLE* com confiança igual ou superior a 20% como o utilizador estando a conduzir, no ficheiro de dados *dados1.log* iriam ocorrer seis deteções incorretas de início de condução. No caso de serem consideradas duas atividades consecutivas então iria apenas ocorrer uma deteção incorreta.

### Google Activity Recognition

O mecanismo de deteção da *Google* reportou os dados apresentados nas tabelas 6 e 7, onde cada linha representa um ficheiro de dados diferente, sendo avaliados no total 21 ficheiros.

| 20% |    |    |   |   |   |   | 30% |    |    |   |   |   |   | 40% |    |    |   |   |   |   |
|-----|----|----|---|---|---|---|-----|----|----|---|---|---|---|-----|----|----|---|---|---|---|
| 1   | 2  | 3  | 4 | 5 | 6 | 7 | 1   | 2  | 3  | 4 | 5 | 6 | 7 | 1   | 2  | 3  | 4 | 5 | 6 | 7 |
| 6   | 1  | 0  | 0 | 0 | 0 | 0 | 6   | 1  | 0  | 0 | 0 | 0 | 0 | 3   | 1  | 0  | 0 | 0 | 0 | 0 |
| 6   | 3  | 1  | 1 | 0 | 0 | 0 | 6   | 3  | 1  | 1 | 0 | 0 | 0 | 4   | 1  | 1  | 0 | 0 | 0 | 0 |
| 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 |
| 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 |
| 7   | 3  | 2  | 1 | 0 | 0 | 0 | 7   | 3  | 2  | 1 | 0 | 0 | 0 | 4   | 1  | 0  | 0 | 0 | 0 | 0 |
| 7   | 3  | 1  | 1 | 0 | 0 | 0 | 7   | 3  | 1  | 1 | 0 | 0 | 0 | 4   | 2  | 1  | 1 | 0 | 0 | 0 |
| 87  | 25 | 10 | 5 | 3 | 2 | 0 | 83  | 24 | 10 | 5 | 3 | 2 | 0 | 53  | 14 | 5  | 3 | 1 | 1 | 0 |
| 3   | 1  | 0  | 0 | 0 | 0 | 0 | 3   | 1  | 0  | 0 | 0 | 0 | 0 | 2   | 0  | 0  | 0 | 0 | 0 | 0 |
| 7   | 2  | 1  | 1 | 1 | 0 | 0 | 7   | 2  | 1  | 1 | 1 | 0 | 0 | 5   | 2  | 1  | 1 | 0 | 0 | 0 |
| 2   | 1  | 0  | 0 | 0 | 0 | 0 | 2   | 1  | 0  | 0 | 0 | 0 | 0 | 1   | 0  | 0  | 0 | 0 | 0 | 0 |
| 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 |
| 15  | 6  | 4  | 2 | 2 | 2 | 0 | 15  | 6  | 4  | 2 | 2 | 2 | 0 | 14  | 5  | 3  | 2 | 2 | 1 | 0 |
| 42  | 14 | 5  | 3 | 2 | 2 | 0 | 41  | 14 | 5  | 3 | 2 | 2 | 0 | 31  | 12 | 3  | 2 | 0 | 0 | 0 |
| 63  | 16 | 6  | 3 | 0 | 0 | 0 | 59  | 16 | 6  | 3 | 0 | 0 | 0 | 41  | 10 | 4  | 1 | 0 | 0 | 0 |
| 140 | 38 | 15 | 6 | 3 | 3 | 0 | 135 | 38 | 14 | 6 | 3 | 3 | 0 | 101 | 30 | 10 | 6 | 3 | 3 | 0 |
| 87  | 25 | 8  | 5 | 2 | 1 | 0 | 85  | 24 | 8  | 5 | 2 | 1 | 0 | 59  | 16 | 6  | 5 | 1 | 1 | 0 |
| 2   | 0  | 0  | 0 | 0 | 0 | 0 | 2   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 |
| 18  | 6  | 3  | 2 | 2 | 1 | 0 | 18  | 6  | 3  | 2 | 2 | 1 | 0 | 14  | 5  | 3  | 2 | 2 | 1 | 0 |
| 87  | 27 | 11 | 6 | 5 | 2 | 0 | 86  | 27 | 10 | 6 | 5 | 2 | 0 | 65  | 19 | 8  | 3 | 2 | 2 | 0 |
| 76  | 21 | 8  | 4 | 3 | 0 | 0 | 74  | 21 | 8  | 4 | 3 | 0 | 0 | 45  | 13 | 4  | 1 | 1 | 0 | 0 |
| 1   | 0  | 0  | 0 | 0 | 0 | 0 | 1   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 |

Tabela 6 - Atividades recolhidas pela API da Google entre 20 e 40% de confiança

| 50% |    |   |   |   |   |   | 60% |   |   |   |   |   |   | 70% |   |   |   |   |   |   |
|-----|----|---|---|---|---|---|-----|---|---|---|---|---|---|-----|---|---|---|---|---|---|
| 1   | 2  | 3 | 4 | 5 | 6 | 7 | 1   | 2 | 3 | 4 | 5 | 6 | 7 | 1   | 2 | 3 | 4 | 5 | 6 | 7 |
| 1   | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 |
| 4   | 1  | 1 | 0 | 0 | 0 | 0 | 3   | 1 | 1 | 0 | 0 | 0 | 0 | 2   | 1 | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 |
| 4   | 1  | 0 | 0 | 0 | 0 | 0 | 2   | 1 | 0 | 0 | 0 | 0 | 0 | 1   | 0 | 0 | 0 | 0 | 0 | 0 |
| 3   | 1  | 0 | 0 | 0 | 0 | 0 | 3   | 1 | 0 | 0 | 0 | 0 | 0 | 3   | 1 | 0 | 0 | 0 | 0 | 0 |
| 32  | 7  | 1 | 1 | 0 | 0 | 0 | 19  | 3 | 1 | 1 | 0 | 0 | 0 | 9   | 1 | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 |
| 3   | 1  | 1 | 0 | 0 | 0 | 0 | 3   | 1 | 1 | 0 | 0 | 0 | 0 | 3   | 1 | 1 | 0 | 0 | 0 | 0 |
| 1   | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 |
| 13  | 5  | 2 | 2 | 1 | 0 | 0 | 9   | 3 | 1 | 1 | 1 | 0 | 0 | 7   | 2 | 1 | 0 | 0 | 0 | 0 |
| 23  | 6  | 1 | 1 | 0 | 0 | 0 | 9   | 1 | 0 | 0 | 0 | 0 | 0 | 2   | 0 | 0 | 0 | 0 | 0 | 0 |
| 21  | 5  | 1 | 0 | 0 | 0 | 0 | 11  | 2 | 1 | 0 | 0 | 0 | 0 | 4   | 0 | 0 | 0 | 0 | 0 | 0 |
| 55  | 11 | 5 | 1 | 0 | 0 | 0 | 31  | 5 | 2 | 1 | 0 | 0 | 0 | 7   | 0 | 0 | 0 | 0 | 0 | 0 |
| 33  | 7  | 4 | 1 | 0 | 0 | 0 | 20  | 5 | 2 | 1 | 0 | 0 | 0 | 12  | 3 | 1 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 |
| 9   | 3  | 2 | 1 | 1 | 0 | 0 | 5   | 1 | 1 | 0 | 0 | 0 | 0 | 3   | 1 | 0 | 0 | 0 | 0 | 0 |
| 35  | 7  | 4 | 2 | 0 | 0 | 0 | 22  | 4 | 3 | 1 | 0 | 0 | 0 | 9   | 2 | 0 | 0 | 0 | 0 | 0 |
| 31  | 8  | 4 | 1 | 1 | 0 | 0 | 21  | 5 | 1 | 0 | 0 | 0 | 0 | 7   | 1 | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 |

Tabela 7 - Atividades recolhidas pela API da Google entre 50 e 70% de confiança

Analisando as tabelas chega-se à conclusão que utilizando 20, 30 ou 40% de confiança teriam de ser avaliadas pelo menos 7 atividades consecutivas de forma a evitar a deteção incorreta de condução. Caso os valores de confiança subam para 50% ou 60% passa a ser necessário avaliar apenas 6 atividades. Considerando 70% de confiança o número de atividades necessárias baixa para 4.

### PathSense Activity Recognition

O mecanismo de deteção *PathSense* reportou os dados apresentados nas tabelas 8 e 9, onde cada linha representa um ficheiro de dados diferente, sendo avaliados 15 ficheiros.

| 20% |    |    |   |   |   |   | 30% |    |    |   |   |   |   | 40% |    |    |   |   |   |   |   |
|-----|----|----|---|---|---|---|-----|----|----|---|---|---|---|-----|----|----|---|---|---|---|---|
| 1   | 2  | 3  | 4 | 5 | 6 | 7 | 1   | 2  | 3  | 4 | 5 | 6 | 7 | 1   | 2  | 3  | 4 | 5 | 6 | 7 |   |
| 5   | 1  | 0  | 0 | 0 | 0 | 0 | 5   | 1  | 0  | 0 | 0 | 0 | 0 | 5   | 1  | 0  | 0 | 0 | 0 | 0 | 0 |
| 1   | 0  | 0  | 0 | 0 | 0 | 0 | 1   | 0  | 0  | 0 | 0 | 0 | 0 | 1   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 4   | 0  | 0  | 0 | 0 | 0 | 0 | 4   | 0  | 0  | 0 | 0 | 0 | 0 | 4   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 13  | 2  | 1  | 0 | 0 | 0 | 0 | 13  | 2  | 1  | 0 | 0 | 0 | 0 | 13  | 2  | 1  | 0 | 0 | 0 | 0 | 0 |
| 70  | 12 | 3  | 0 | 0 | 0 | 0 | 70  | 12 | 3  | 0 | 0 | 0 | 0 | 70  | 12 | 3  | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 20  | 0  | 0  | 0 | 0 | 0 | 0 | 20  | 0  | 0  | 0 | 0 | 0 | 0 | 20  | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 67  | 5  | 3  | 0 | 0 | 0 | 0 | 67  | 5  | 3  | 0 | 0 | 0 | 0 | 67  | 5  | 3  | 0 | 0 | 0 | 0 | 0 |
| 138 | 10 | 2  | 1 | 1 | 0 | 0 | 138 | 10 | 2  | 1 | 1 | 0 | 0 | 138 | 10 | 2  | 1 | 1 | 0 | 0 | 0 |
| 126 | 10 | 4  | 1 | 0 | 0 | 0 | 126 | 10 | 4  | 1 | 0 | 0 | 0 | 126 | 10 | 4  | 1 | 0 | 0 | 0 | 0 |
| 3   | 0  | 0  | 0 | 0 | 0 | 0 | 3   | 0  | 0  | 0 | 0 | 0 | 0 | 3   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 101 | 15 | 3  | 1 | 0 | 0 | 0 | 101 | 15 | 3  | 1 | 0 | 0 | 0 | 101 | 15 | 3  | 1 | 0 | 0 | 0 | 0 |
| 194 | 33 | 15 | 7 | 2 | 1 | 0 | 194 | 33 | 15 | 7 | 2 | 1 | 0 | 194 | 33 | 15 | 7 | 2 | 1 | 0 | 0 |
| 181 | 20 | 6  | 3 | 1 | 0 | 0 | 181 | 20 | 6  | 3 | 1 | 0 | 0 | 181 | 20 | 6  | 3 | 1 | 0 | 0 | 0 |
| 6   | 0  | 0  | 0 | 0 | 0 | 0 | 6   | 0  | 0  | 0 | 0 | 0 | 0 | 6   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |

Tabela 8 - Atividades recolhidas pela PathSense API entre 20 e 40% de confiança

| 50% |    |    |   |   |   |   | 60% |    |    |   |   |   |   | 70% |    |    |   |   |   |   |   |
|-----|----|----|---|---|---|---|-----|----|----|---|---|---|---|-----|----|----|---|---|---|---|---|
| 1   | 2  | 3  | 4 | 5 | 6 | 7 | 1   | 2  | 3  | 4 | 5 | 6 | 7 | 1   | 2  | 3  | 4 | 5 | 6 | 7 |   |
| 5   | 1  | 0  | 0 | 0 | 0 | 0 | 5   | 1  | 0  | 0 | 0 | 0 | 0 | 5   | 1  | 0  | 0 | 0 | 0 | 0 | 0 |
| 1   | 0  | 0  | 0 | 0 | 0 | 0 | 1   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 4   | 0  | 0  | 0 | 0 | 0 | 0 | 4   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 13  | 2  | 1  | 0 | 0 | 0 | 0 | 13  | 2  | 1  | 0 | 0 | 0 | 0 | 11  | 2  | 1  | 0 | 0 | 0 | 0 | 0 |
| 70  | 12 | 3  | 0 | 0 | 0 | 0 | 70  | 12 | 3  | 0 | 0 | 0 | 0 | 65  | 12 | 3  | 0 | 0 | 0 | 0 | 0 |
| 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 20  | 0  | 0  | 0 | 0 | 0 | 0 | 20  | 0  | 0  | 0 | 0 | 0 | 0 | 8   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 67  | 5  | 3  | 0 | 0 | 0 | 0 | 67  | 5  | 3  | 0 | 0 | 0 | 0 | 52  | 5  | 1  | 0 | 0 | 0 | 0 | 0 |
| 138 | 10 | 2  | 1 | 1 | 0 | 0 | 138 | 10 | 2  | 1 | 1 | 0 | 0 | 82  | 7  | 2  | 1 | 1 | 0 | 0 | 0 |
| 126 | 10 | 4  | 1 | 0 | 0 | 0 | 126 | 10 | 4  | 1 | 0 | 0 | 0 | 96  | 9  | 4  | 0 | 0 | 0 | 0 | 0 |
| 3   | 0  | 0  | 0 | 0 | 0 | 0 | 3   | 0  | 0  | 0 | 0 | 0 | 0 | 3   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |
| 101 | 15 | 3  | 1 | 0 | 0 | 0 | 101 | 15 | 3  | 1 | 0 | 0 | 0 | 82  | 14 | 3  | 1 | 0 | 0 | 0 | 0 |
| 194 | 33 | 15 | 7 | 2 | 0 | 0 | 194 | 33 | 15 | 7 | 2 | 0 | 0 | 162 | 31 | 15 | 7 | 2 | 0 | 0 | 0 |
| 181 | 20 | 6  | 3 | 1 | 0 | 0 | 181 | 20 | 6  | 3 | 1 | 0 | 0 | 131 | 19 | 6  | 3 | 1 | 0 | 0 | 0 |
| 6   | 0  | 0  | 0 | 0 | 0 | 0 | 6   | 0  | 0  | 0 | 0 | 0 | 0 | 5   | 0  | 0  | 0 | 0 | 0 | 0 | 0 |

Tabela 9 - Atividades recolhidas pela PathSense API entre 50 e 70% de confiança

O *PathSense Activity Recognition* apesar de estar ligado simultaneamente com o *Google Activity Recognition*, recolheu menos dados, tendo parado a recolha em certos momentos como evidencia o número inferior de ficheiros de dados. Há semelhança do mecanismo *Google* para níveis de confiança baixas (20, 30 e 40%) necessita de avaliar pelo menos 7 atividades, valor que baixa para 6 quando a confiança sobe para valores iguais ou superiores a 50%.

Tal como se procurou encontrar valores para evitar a deteção incorreta do início de condução, o mesmo estudo foi efetuado para o número de atividades e confiança necessários para evitar detetar erradamente que o utilizador deixou de conduzir.

### 5.3. Conclusão

O estudo realizado serviu essencialmente de fundamentação às respostas a duas questões, a saber: qual o melhor sistema de reconhecimento a utilizar no protótipo? e quais os valores de confiança e de número de atividades para o algoritmo de deteção ser fiável?.

#### **Google Activity Recognition vs PathSense Activity Recognition**

Tal como se pode observar pelos dados recolhidos, ambos os mecanismos de reconhecimento são similares nos valores que reportam, necessitando ambos de avaliar um número semelhante de atividades para obter certeza na deteção de condução. Uma vez que o *PathSense* se mostrou irregular ao não registar valores durante várias porções de tempo, a escolha pendeu para o *Google Activity Recognition* atendendo a esse problema.

#### **Parâmetros para o Algoritmo de Deteção**

Tendo em conta os valores obtidos, foram definidos os seguintes valores face aos resultados do programa de análise:

- Número de atividades para detetar início de condução: 4
- % de Confiança para início de condução: 70%
- Número de atividades para detetar saída de condução: 5
- % de Confiança para saída de condução: 50%

Concluindo, para o protótipo desenvolvido no estágio foi decidido utilizar o *Google Activity Recognition* com um algoritmo que avalia as últimas quatro atividades recebidas. Caso essas entradas sejam do tipo *IN\_VEHICLE* com uma percentagem de confiança superior a 70%, o algoritmo irá considerar que o dispositivo está num veículo em condução. Se durante a condução forem recebidos cinco atividades e todas elas forem diferentes de *IN\_VEHICLE* ou forem *IN\_VEHICLE* com percentagem de confiança inferior a 50%, então considera-se que o utilizador parou de conduzir.





## Capítulo 6

### Testes

Em qualquer projeto de *software* a validação da solução é um passo fundamental para o sucesso do mesmo. Uma aplicação sem erros melhora a experiência de utilização e mantém os utilizadores satisfeitos.

Para o protótipo realizado no estágio foi criado e executado um conjunto de testes funcionais cujo objetivo foi a validação de cada funcionalidade. Para além dos testes funcionais, houve um foco especial no algoritmo de deteção, tendo sido criado um teste unitário para confirmar o correto funcionamento, quer no momento de detetar o início de condução, quer na deteção de fim de condução.

#### 6.1. Testes Funcionais – Protótipo

Nos testes funcionais pretende-se validar o correto funcionamento de cada funcionalidade. Em cada teste funcional é criado um conjunto de comportamentos que devem ser executados assim como definido o resultado esperado. Quando o resultado final da execução desses passos no protótipo correspondeu ao resultado esperado considerou-se que o teste teve sucesso. Caso contrário, foi necessário proceder à correção dos problemas encontrados na funcionalidade. Os testes foram realizados à medida que as funcionalidades foram implementadas tendo sido executados no final novamente para garantir que a geração de novas funcionalidades não criou problemas em funcionalidades previamente testadas.

Foram criados 67 testes funcionais, a Tabela 10 apresenta os resultados obtidos no final do projeto, para mais detalhe acerca de cada teste deve ser consultado o **Apêndice C – Testes Funcionais**.

| ID | Teste   | Resultado |
|----|---|-----------|
| 1  | Início de condução através das definições                     | Sucesso   |
| 2  | Saída do menu de condução manualmente                         | Sucesso   |
| 3  | Contabilização de tempo de viagem                             | Sucesso   |
| 4  | Indicador de bateria – nível baixo                            | Sucesso   |
| 5  | Indicador de bateria – nível médio                            | Sucesso   |
| 6  | Indicador de bateria – nível alto                             | Sucesso   |
| 7  | Indicador de bateria – nível muito alto                       | Sucesso   |
| 8  | Alerta sonoro de receção de SMS                               | Sucesso   |
| 9  | Alerta sonoro de receção de mensagem de <i>chat</i>           | Sucesso   |
| 10 | Alerta visual da receção de SMS                               | Sucesso   |
| 11 | Alerta visual da receção de mensagem de <i>chat</i>           | Sucesso   |
| 12 | Entrada no menu de mensagens                                  | Sucesso   |
| 13 | Apresentação de dados no menu de mensagens                    | Sucesso   |
| 14 | Indicador de mensagens – 1 mensagem                           | Sucesso   |
| 15 | Indicador de mensagens – 2 ou mais mensagens                  | Sucesso   |
| 16 | Indicador de mensagens - navegação                            | Sucesso   |
| 17 | Retroceder de menu de mensagens                               | Sucesso   |
| 18 | Audição de mensagem de texto via <i>text-to-speech</i>        | Sucesso   |
| 19 | Audição de mensagem de áudio                                  | Sucesso   |
| 20 | Parar reprodução de leitura de mensagem <i>text-to-speech</i> | Sucesso   |

|    |   |         |
|----|---|---------|
| 21 | Parar reprodução de mensagem de áudio                                   | Sucesso |
| 22 | Opção de chamadas no menu de mensagens                                  | Sucesso |
| 23 | Opção de resposta com áudio no menu de mensagens                        | Sucesso |
| 24 | Iniciar gravação de áudio   | Sucesso |
| 25 | Parar gravação de áudio - manualmente                                   | Sucesso |
| 26 | Parar gravação de áudio – automaticamente                               | Sucesso |
| 27 | Atualização do gráfico e do tempo de gravação                           | Sucesso |
| 28 | Apresentação da onda do volume de gravação                              | Sucesso |
| 29 | Enviar mensagem de áudio gravada - manualmente                          | Sucesso |
| 30 | Enviar mensagem de áudio gravada - automaticamente                      | Sucesso |
| 31 | Voltar para o menu inicial de gravação                                  | Sucesso |
| 32 | Ouvir a mensagem de áudio gravada                                       | Sucesso |
| 33 | Apresentação do ícone de acesso de menu de chamadas                     | Sucesso |
| 34 | Prioridade do número de emergência                                      | Sucesso |
| 35 | Início de chamadas IP   | Sucesso |
| 36 | Início de chamada de voz tradicional                                    | Sucesso |
| 37 | Apresentação do Floating button   | Sucesso |
| 38 | Atualização do número de mensagens no <i>floating button</i>            | Sucesso |
| 39 | Acesso aos menus de condução via <i>floating button</i>                 | Sucesso |
| 40 | Movimentação do <i>floating button</i> pelo ecrã                        | Sucesso |
| 41 | Apresentação de notificação com notificações perdidas                   | Sucesso |
| 42 | Apresentação detalhada de notificações perdidas                         | Sucesso |
| 43 | Início de condução através do <i>widget</i>                             | Sucesso |
| 44 | Apresentação de todas as pontuações de contactos                        | Sucesso |
| 45 | Adição de pontos após envio de notificação de condução                  | Sucesso |
| 46 | Adição de pontos após receção de uma mensagem                           | Sucesso |
| 47 | Adição de pontos após receção de chamada                                | Sucesso |
| 48 | Adição de pontos após início de chamada                                 | Sucesso |
| 49 | Adição de pontos após audição de mensagem                               | Sucesso |
| 50 | Adição de pontos após envio de mensagem de áudio                        | Sucesso |
| 51 | Leitura sequencial de mensagens   | Sucesso |
| 52 | Informação de condução na janela de <i>chat</i>                         | Sucesso |
| 53 | Informação de condução na lista de <i>chats</i>                         | Sucesso |
| 54 | Envio de mensagem urgente a utilizador a conduzir                       | Sucesso |
| 55 | Escolha de notificação nas opções de envio de mensagem                  | Sucesso |
| 56 | Agendamento de envio de mensagem nas opções de envio                    | Sucesso |
| 57 | Gravação e envio de áudio a condutor                                    | Sucesso |
| 58 | Apresentação da opção e início de chamada com condutor                  | Sucesso |
| 59 | Botão de cancelar nas opções de envio a condutor                        | Sucesso |
| 60 | Apresentação do teclado nas opções de envio a condutor                  | Sucesso |
| 61 | Adicionar e enviar notificação nas <i>Car Mode Notifications</i>        | Sucesso |
| 62 | Remover do envio de notificações  | Sucesso |
| 63 | Receção de notificação de início de condução                            | Sucesso |
| 64 | Adicionar novo dispositivo <i>bluetooth</i> para abrir menu de condução | Sucesso |
| 65 | Remover dispositivo <i>bluetooth</i> da lista                           | Sucesso |
| 66 | Selecionar condutor após deteção de condução                            | Sucesso |
| 67 | Selecionar passageiro após deteção de condução                          | Sucesso |

Tabela 10 - Testes Funcionais do Protótipo

## 6.2. Teste Unitário - Algoritmo de Detecção

Dada a importância do algoritmo de deteção foi necessário certificar a sua correta implementação. Para tal, foi criado um teste unitário cujo objetivo é introduzir no algoritmo de deteção dados simulados da API de reconhecimento de atividade, e verificar se os resultados do algoritmo são os esperados.

Os dados de *input* utilizados para executar o teste foram recolhidos pela aplicação descrita no capítulo 5 – Análise de Condução. Todos os ficheiros foram tratados de forma a ocorrer uma entrada e saída real de condução.

| Ficheiro  | Início Condução | Deteção Início | Fim Condução | Deteção Fim |
|-----------|-----------------|----------------|--------------|-------------|
| log1.data | 00:01:18        | 00:02:12       | 00:20:38     | 00:21:11    |
| log2.data | 08:11:46        | 08:12:11       | 09:02:48     | 09:03:30    |
| log3.data | 08:59:34        | 08:59:44       | 09:21:44     | 09:22:08    |
| log4.data | 19:14:17        | 19:14:35       | 19:28:56     | 19:29:33    |

*Tabela 11 - Resultado teste unitário ao algoritmo de deteção*

Como se pode observar pelos resultados, o algoritmo de deteção conseguiu sempre num curto espaço de tempo confirmar a entrada ou saída do dispositivo do estado de condução.



## Capítulo 7

### Conclusões

O objetivo deste capítulo é apresentar uma reflexão final sobre o trabalho desenvolvido. Na primeira secção 7.1 Avaliação Final são apresentadas as opiniões finais do estagiário em relação ao projeto desenvolvido, na secção 7.2. Trabalho Futuro discute-se os passos que podem ser seguidos para expandir o protótipo criado.

#### 7.1. Avaliação Final

Tal como foi referenciado, na última década assistimos a uma rápida evolução no mundo das comunicações, evolução esta que determina a necessidade de dar resposta com valor e atempada a questões que se vão colocando. No caso concreto, refiro o aumento do número de acidentes na estrada, resultado da distração pelo uso do telemóvel.

O trabalho desenvolvido constitui-se como um contributo para a resolução do problema supracitado, tal como para a valorização da aplicação *Android RCS+*, uma vez que se acrescentou um conjunto de funcionalidades que possibilitam uma condução segura, mesmo em comunicação.

Uma incursão pela literatura permitiu a exploração das tecnologias a utilizar durante o decorrer do trabalho, tal como o estudo dos competidores, com o intuito de perceber quais os requisitos mais relevantes e onde inovar na implementação da solução pretendida.

O domínio da tecnologia existente, em parceria com a exploração da aplicação RCS+ da WIT Software, do estudo dos diferentes mecanismos de *text-to-speech* disponíveis para a plataforma *Android*, dos mecanismos de reconhecimento de atividade e dos conceitos e estruturas *Android* relevantes, impulsionaram a planificação e implementação do trabalho.

O plano do primeiro semestre, incluiu as funcionalidades de gestão de mensagens e chamadas no módulo de condução, bem como, a perceção de condução por parte dos utilizadores da aplicação RCS+. Face ao desenvolvimento satisfatório destas funcionalidades o protótipo foi apresentado no *Mobile World Congress* a um número restrito de interessados.

O segundo semestre contemplou a implementação de um mecanismo de jogos (*gamification*), a deteção de condução automática através de sensores do *smartphone* ou ligações *bluetooth* e diversas funcionalidades para aumentar a qualidade do protótipo, bem como a estabilização do protótipo. Foi também iniciado o processo de patentear a ideia, tendo o estagiário auxiliado o processo de documentação.

Desde o início do estágio a contribuição do estagiário para a solução foi predominante. A definição dos requisitos a implementar no protótipo ficou a cargo do estagiário com base na visão da empresa para o estágio. A participação quer na definição da *User Experience* quer na *User Interface* tornaram o processo de desenvolvimento bastante gratificante e frutífero em termos de conhecimento.

A sensação final é bastante gratificante a nível pessoal e profissional face ao vasto conjunto de conhecimentos em diferentes tecnologias e processos que certamente serão uma mais-valia no mundo profissional.

## 7.2. Trabalho Futuro

O protótipo desenvolvido encontra-se com todas as funcionalidades, consideradas importantes e prioritárias para a WIT Software, implementadas. O *backlog* do projeto contém algumas funcionalidades resultantes de pesquisa e *brainstorming* do estagiário e interessados no projeto. Estas funcionalidades embora não sendo prioritárias podem vir a ser realizadas com o objetivo de adicionar valor ao protótipo já existente. Exemplos dessas funcionalidades são a execução de ações através de comandos de voz ou integração com aplicações de localização.

Tal como apresentado no estado da arte o protótipo realizado durante o decorrer deste estágio encontra-se integrado num conjunto de aplicações móveis, *web* e para sistemas operativos *Windows* e *Os X*. Para que se possa tirar proveito do protótipo criado para *Android* seria importante as restantes plataformas serem capazes de identificar quando um dispositivo está em modo de condução de forma a simular o comportamento já existente no protótipo. Em termos de trabalho futuro este seria um ponto principal, a implementação das funcionalidades de perceção de condução em aplicações *web* e aplicações nativas dos sistemas operativos, assim como a criação de um protótipo semelhante ao desenvolvido no estágio para a plataforma *iOS*.

## Referências

- [1] Ingrid Lunden. Techcrunch. [Online]. <http://techcrunch.com/2015/06/02/6-1b-smartphone-users-globally-by-2020-overtaking-basic-fixed-phone-subscriptions/>
- [2] Página Web Skype. [Online]. <https://www.skype.com/>
- [3] (2015, Novembro) Página Web Whatsapp. [Online]. <https://www.whatsapp.com/>
- [4] (2015, Novembro) Página Web Viber. [Online]. <http://www.viber.com/pt/>
- [5] Erik Heinrich. Fortune. [Online]. <http://fortune.com/2014/06/23/telecom-companies-count-386-billion-in-lost-revenue-to-skype-whatsapp-others/>
- [6] (2015, Novembro) Página Web da GSMA. [Online]. <http://www.gsma.com>
- [7] GSMA. [Online]. <http://www.gsma.com/network2020/rcs/>
- [8] (Novembro, 2015) Página Web National Safety Council. [Online]. <http://www.nsc.org/pages/home.aspx>
- [9] (Novembro, 2015) Página Web Harvard Center for Risk Analysis. [Online]. <https://www.hsph.harvard.edu/hcra/>
- [10] [Online]. <http://www.textinganddrivingsafety.com/texting-and-driving-stats>
- [11] AT&T Drivemode na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.drivemode>
- [12] The World's Biggest Public Companies. [Online]. <http://www.forbes.com/forbes/welcome/#tab:overall>
- [13] Car Home Ultra na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=spinninghead.carhome&hl=en>
- [14] Drive Safe.ly na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.drivesafe.ly&hl=en>
- [15] Safe Driving Text Machine na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.client.DrivingSMSAnsweringMachine&hl=en>
- [16] Drivemode na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.drivemode.android&hl=en>
- [17] [Online]. <https://drivemode.com/#about>
- [18] text-STAR na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.cinqpoint.textstar&hl=en>
- [19] SafeDrive rewards you. [Online]. <https://play.google.com/store/apps/details?id=nl.xlteam.safedrive&hl=en>
- [20] TextDrive na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.smalltalkapps.textdrive&hl=en>
- [21] Sprint Drive First na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.sprint.drive.first.safely.driver.safety.stop.texting.driving&hl=en>
- [22] Drive Mode na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.vincent.drivemode&hl=en>
- [23] Car dashdroid na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.nezdroid.cardashdroid&hl=en>
- [24] Safely Go na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.safely.go.driver.safety.stop.texting.driving&hl=en>

- [25] DriveSafe Mode Logótipo. [Online]. <https://play.google.com/store/apps/details?id=com.ds.android&hl=en>
- [26] Drive Alive Lite na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.drivelive.demo&hl=en>
- [27] Ultimate Car Dock na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.appsontoast.ultimatecardock&hl=en>
- [28] LifeSaver na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.lifesaverapp&hl=en>
- [29] Dashdroid na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.app.dashdroid&hl=en>
- [30] Automate na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.bitspice.automate&hl=en>
- [31] BitSpice. [Online]. <http://www.bitspice.net/index.html>
- [32] [Online]. <http://www.windowsphone.com/en-us/how-to/wp8/basics/use-driving-mode>
- [33] Android Auto. [Online]. <https://www.android.com/auto/>
- [34] Open Automative Alliance. [Online]. <http://www.openautoalliance.net/#about>
- [35] Cell Control. [Online]. <https://www.cellcontrol.com/texting-and-distracted-driving-solutions-driver-safety>
- [36] Apple Carplay. [Online]. <http://www.apple.com/ios/carplay/>
- [37] WIT Software Website. [Online]. <https://www.wit-software.com/products/rcs-suite/>
- [38] WIT Software Website. [Online]. <https://www.wit-software.com/products/rcs-suite/#tabid2>
- [39] IVONA Text-To-Speech na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.ivona.tts&hl=en>
- [40] IVONA Website. [Online]. <https://www.ivona.com/us/about-us/company/>
- [41] Geekersmagazine. [Online]. <http://www.geekersmagazine.com/20130617/text-to-speech-android-apps/>
- [42] SVOX na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.svox.classic>
- [43] [Online]. <http://www.geoffsimons.com/2012/06/7-best-android-text-to-speech-engines.html>
- [44] Google Text-to-Speech na Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.google.android.tts&hl=en>
- [45] Google Play Store. [Online]. <https://play.google.com/store/apps/details?id=com.google.android.gms&hl=en>
- [46] TechRepublic. [Online]. <http://www.techrepublic.com/article/how-pathsense-cuts-battery-drain-from-location-services-by-up-to-90/>
- [47] Android Developers. [Online]. <http://developer.android.com/guide/components/fundamentals.html>
- [48] [Online]. [https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/mas\\_o\\_que\\_s\\_c3\\_a3o\\_essas\\_tais\\_de\\_metodologias\\_c3\\_a1geis?lang=en](https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/mas_o_que_s_c3_a3o_essas_tais_de_metodologias_c3_a1geis?lang=en)
- [49] Scrum: A new perspective on agile development. [Online]. <https://github.com/bounswe/bounswe2016group6/wiki/Scrum:-A-new-perspective-on-agile-development>



- [50] [Online]. <http://scrummethodology.com>]:
- [51] [Online]. <https://www.wit-software.com/products/rcs-suite/>
- [52] [Online]. <https://github.com/bounswe/bounswe2016group6/wiki/Scrum:-A-new-perspective-on-agile-development>