

Mestrado em Engenharia Informática
Estágio
Relatório Final

Projeto EcoMobile desenvolvimento de uma aplicação iOS/Android para recolha e partilha de informação sensorial

Pedro Alexandre Almeida Freire Costa
pacosta@student.dei.uc.pt

Orientadores:

Orientador DEI: Prof. António Jorge da Costa Granjal

Co-Orientador: Prof. Bruno Miguel Brás Cabral

1 de Julho de 2014



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

Agradecimentos

Em primeiro lugar queria agradecer aos meus pais pelo suporte e investimento que fizeram em mim, sem o qual não seria possível chegar ao ponto da minha vida em que me encontro agora. Uma palavra também de agradecimento à minha irmã por sempre acreditar em mim e me encorajar.

Agradeço ao meus orientadores, Doutor Jorge Granjal e Doutor Bruno Cabral, por me darem a oportunidade de contribuir para este projeto bastante enriquecedor, mostrando-se sempre disponíveis em todos momentos no decorrer dos trabalhos.

Agradeço aos meus colegas de trabalho, Rui Chicória e David Miranda, por todo o apoio e conselhos que me deram ao longo do projeto, tendo o seu contributo ajudado a alcançar os bons resultados de todo o projeto.

Por último agradeço a todos os meus amigos, que me tem acompanhado sempre no meu percurso e que continuaram a ocupar o seu espaço no futuro: Amílcar, Diogo, Jorge, Luís e Nuno.

Resumo

No mundo do transporte automóvel cada vez mais se dá atenção à eficiência energética e segurança dos condutores. A elevada subida dos preços dos combustíveis e altas taxas de sinistralidade são uma constante preocupação para a população em geral.

Os dispositivos móveis encontram-se numa fase de massificação, destacando-se os populares *smartphones* e *tablets*. A sua adoção tem-se tornado cada vez maior por parte dos utilizadores, tornando-os objetos que automatizam e facilitam algumas das tarefas diárias dos utilizadores. Estes dispositivos incluem *hardware* que permite saber informação referente à posição e movimento do mesmo de forma simplificada recorrendo ao GPS e sensores de movimento.

O projecto EcoMobile foi criado com o objetivo de ajudar os condutores a tornarem a sua condução mais económica e segura, recorrendo a *smartphones*. O presente trabalho descreve a implementação de uma aplicação para dispositivos móveis que utilizando informação sensorial (acelerómetro, giroscópio e magnetómetro), regista os trajetos e estilo de condução do utilizador, permitindo que estes dados sejam visualizados e registados pelo utilizador, ao mesmo tempo que é feita uma análise à sua forma de condução, eficiência energética e nível de segurança, apresentando *feedback* em tempo real, permitindo ao utilizador tomar decisões mais conscientes sobre os aspetos a mudar na sua condução.

Palavras chave: aplicações móveis, condução, eficiência energética, segurança automóvel, GPS, sensores de movimento, *feedback* em tempo real.

Conteúdo

1	Introdução	1
1.1	Enquadramento e motivação	1
1.1.1	Projeto Ecomobile	2
1.2	Objetivos	3
1.3	Estrutura do documento	4
2	Metodologia e Planeamento	5
2.1	Metodologia de Desenvolvimento	5
2.2	Gestão do projeto	8
2.3	Planeamento	11
2.3.1	Primeiro semestre	11
2.3.2	Segundo semestre	13
2.3.3	Desvios	13
2.4	Análise de Riscos	16
3	Estado da arte	19
3.1	Sistemas Operativos Móveis	19
3.1.1	iOS	20
3.1.2	Android	21
3.1.3	Plataformas híbridas	22
3.2	Integração com redes sociais	23
3.3	Soluções existentes	24
3.3.1	My Car Tracks	24
3.3.2	Automatic	25
3.3.3	Drismo	26
3.3.4	My Tracks	27
3.3.5	TomTom Webfleet Mobile	27
3.3.6	Comparativo	28
3.3.7	Sumário	29

4	Análise de Requisitos	31
4.1	Identificação de Requisitos	31
4.1.1	Atores	31
4.1.2	Prototipagem	32
4.1.3	Requisitos funcionais	33
4.1.4	Requisitos não-funcionais	37
5	Especificação da Arquitetura	41
5.1	Descrição Geral da Arquitetura	41
5.2	Decisões Arquiteturais	43
5.3	Desenho da solução	43
5.3.1	Estrutura da aplicação móvel	43
5.3.2	Módulos	45
6	Trabalho realizado	49
6.1	Protótipo Phonegap	49
6.2	Aplicação iOS Nativa	50
6.2.1	Bibliotecas externas usadas	50
6.2.2	Tratamento dados sensoriais e de localização	51
6.2.3	Deteção de eventos	55
6.2.4	Cálculo de índices	58
6.2.5	Resultados da aplicação iOS	64
7	Testes	69
7.1	Testes de aceitação	69
7.2	Testes de software	74
7.2.1	Static analyzer	74
7.2.2	Testes tráfego de dados	74
7.2.3	Testes de internacionalização	75
7.3	Testes em ambiente real	75
8	Reflexões e trabalho futuro	77
8.1	Conclusão	77
8.2	Trabalho futuro	78
A	Análise de requisitos	83
A.1	Requisitos funcionais	83
A.1.1	Requisitos relacionados com contas de utilizador	83
A.1.2	Requisitos relacionados com registo de viagens	84
A.1.3	Requisitos relacionados com viagens	86
A.1.4	Requisitos relacionados com veículos	88

<i>CONTEÚDO</i>	ix
A.1.5 Requisitos sociais	89
A.1.6 Requisitos relacionados com grupos	90
A.1.7 Requisitos relacionados com configurações	90
B Prototipagem de baixo nível protótipo	93
C Prototipagem de baixo nível iOS nativo	95
D Resultados da implementação do protótipo	101
E Artefactos da metodologia ágil	105
F Modelo de dados da aplicação móvel	111
G Resumo informação sensorial e de localização	115

Lista de acrônimos

2D	<i>Duas dimensões</i>
A-GPS	<i>Assisted Global Positioning System</i>
API	<i>Application Programming Interface</i>
CPU	<i>Central Processing Unit</i>
CSS	<i>Cascading Style Sheets</i>
FCTUC	Faculdade de Ciências e Tecnologia da Universidade de Coimbra
GLONASS	<i>Global Navigation Satellite System</i>
GPU	<i>Graphics Processing Unit</i>
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
IPN	Instituto Pedro Nunes
JSON	<i>JavaScript Object Notation</i>
MEI	Mestrado de Engenharia Informática
MVC	<i>Model-view-controller</i>
OBD	<i>On-board diagnostics</i>
RAM	<i>Random-access memory</i>
REST	<i>Representational State Transfer</i>
SDK	<i>Software Development Kit</i>
UC	Universidade de Coimbra
UI	<i>User Interface</i>

Lista de Tabelas

3.1	Comparação soluções equivalentes	28
3.2	Comparação custos das soluções equivalentes	28
4.1	Requisitos relacionados com contas de utilizador	34
4.2	Requisitos relacionados com registo de viagens	34
4.3	Requisitos relacionados com viagens	35
4.4	Requisitos relacionados com veículos	35
4.5	Requisitos sociais	36
4.6	Requisitos relacionados com grupos	36
4.7	Requisitos relacionados com configurações	37
4.8	Requisitos de sistema	38
4.9	Requisitos de segurança	38
7.1	Especificações dos dispositivos de teste.	69
7.2	Testes relacionados com contas de utilizador	70
7.3	Testes relacionados com registo de viagens	70
7.4	Testes relacionados com viagens	71
7.5	Testes relacionados com veículos	72
7.6	Testes sociais	72
7.7	Testes relacionados com grupos	73
7.8	Testes relacionados com configurações	73
7.9	Testes compressão para viagem de 60 minutos	74
F.1	Descrição das entidades do modelo de dados	113

Lista de Figuras

2.1	Representação da metodologia usada. Retirada de[1]	6
2.2	Elementos da equipa com contributo no presente estágio	7
2.3	Imagem da ferramenta Jira.(Vista com o <i>backlog</i>)	8
2.4	Imagem da ferramenta Confluence.(Organização dos <i>mockups</i>)	9
2.5	Imagem da ferramenta Jitlab. (Lista de <i>commits</i>)	10
3.1	Sistemas operativos móveis <i>market share</i> em 2014[2]	20
3.2	Arquitectura <i>Model-View-Controller</i> aplicação iOS	21
3.3	Arquitectura de uma aplicação Phonegap	23
3.4	Imagem da aplicação MyCarTracks	25
3.5	Imagem da aplicação Automatic	26
3.6	Imagem da aplicação Drismo	26
3.7	Imagem da aplicação My Tracks	27
3.8	Imagem da aplicação TomTom WebFleet Mobile	28
4.1	Prototipagem de baixo nível para o Phonegap	32
4.2	Prototipagem baixo nível para a aplicação nativa iOS	33
5.1	Arquitectura geral do projecto Ecomobile	42
5.2	Arquitectura aplicação móvel	44
6.1	Eixo direção do movimento	54
6.2	Evento de curva brusca	55
6.3	Gráfico índice de segurança	59
6.4	Exemplo gráfico índice de segurança	61
6.5	Gráfico índice de economia	62
6.6	Exemplo gráfico índice de economia	64
6.7	Ecrã <i>Login</i>	65
6.8	Ecrã Registo	65
6.9	Ecrã Lista de Viagens	66
6.10	Ecrã lista de veículos	66
6.11	Ecrã social	67

6.12	Ecrã definições	67
6.13	Ecrã registo de viagem com mapa	68
6.14	Ecrã fim de viagem	68
D.1	Ecrã <i>Login</i> protótipo	101
D.2	Ecrã registo protótipo	101
D.3	Ecrã Inicial protótipo	102
D.4	Ecrã definições protótipo	102
D.5	Ecrã Lista de viagens protótipo	102
D.6	Ecrã lista de veículos protótipo	102
D.7	Ecrã viagem com o gráfico protótipo	103
D.8	Ecrã galeria de imagens protótipo	103
D.9	Ecrã detalhes veículo protótipo	103
D.10	Ecrã adicionar veículo	103
D.11	Ecrã registo de viagem com mapa protótipo	104
D.12	Ecrã registo de viagem com gráfico protótipo	104
D.13	Ecrã fim de viagem protótipo	104
F.1	Modelo de dados da aplicação móvel	112

Capítulo 1

Introdução

O presente estágio encontra-se inserido no âmbito da disciplina de Dissertação/Estágio do Mestrado em Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra, sendo dividido em dois semestres. Este relatório contempla todo o trabalho desenvolvido ao longo dos desses semestres, onde foram definidos o planeamento, requisitos e arquitetura, procedendo-se depois à implementação e testes finais do sistema. O trabalho realizado decorreu na empresa Sentilant¹, situada no incubadora do Instituto Pedro Nunes, IPN.

Neste capítulo introdutório são apresentadas as motivações e enquadramento que levaram ao desenvolvimento deste projeto, de forma a contextualizar o trabalho realizado.

1.1 Enquadramento e motivação

A crescente massificação e utilização dos *smartphones* veio revolucionar o quotidiano. Tem-se verificado um enorme desenvolvimento nesta área tecnológica, que está cada vez mais acessível ao cidadão comum. O estilo de condução é um dos fatores preponderantes na taxa de consumo de uma viatura automóvel, assim como na segurança na condução. Uma boa prática da condução pode levar a ganhos significativos no consumo e segurança dos condutores. A adoção de hábitos de condução com vista ao aumento da poupança e segurança são fatores de peso a ter em conta pelo condutor particular mas também pelas organizações que possuem frotas automóveis.

¹<http://www.sentilant.com/>

Surge com a massificação dos dispositivos móveis e sua evolução, a possibilidade de combinar a sua utilização para registar o estilo de condução ao longo das diversas viagens dos utilizadores, registando métricas que permitem extrapolar o quão eficiente e segura é a sua condução.

A empresa Sentilant pretende tirar partido das excelentes capacidades de *hardware* presentes nos *smartphones* atuais para recolher toda a informação necessária para poder caracterizar o estilo de condução dos utilizadores. Com o acesso aos sensores de movimentos e GPS do *smartphone*, consegue-se obter informação que permite determinar os fatores que mais influenciam a condução, tais como as acelerações, travagens, curvas acentuadas, subidas ou descidas, assim como registar a velocidade, altitude e trajeto da viagem. Com o registo de toda esta informação é possível estabelecer um estilo de condução, caracterizado pelos fatores principais, economia e segurança.

1.1.1 Projeto Ecomobile

O projeto EcoMobile pretende informar os condutores sobre o seu estilo de condução, usando para isso indicadores de segurança e economia, com o objetivo destes adotarem comportamentos de condução mais eficientes energeticamente, ao mesmo tempo melhorarem a sua segurança. Cumpre estes objetivos através da monitorização da condução dos utilizadores, recolhendo informação sensorial e de localização usando *smartphones*. Toda a informação recolhida será usada para obter indicadores de segurança e economia e também na caracterização do estilo de condução do utilizador, funcionando assim como suporte à tomada de decisão dos utilizadores sobre os aspetos a melhorar na sua condução.

Este projeto venceu dois prémios de inovação tecnológica, o concurso PT Galp Innovation Challenge em 2011² e o concurso Arrisca C em 2012³, tendo dado origem à empresa Sentilant.

O projeto é constituído na sua globalidade pela aplicação móvel, uma plataforma *backend* em ambiente *Cloud* e um *front-end web*⁴. A aplicação móvel é responsável pela monitorização, recolha e tratamento da informação sensorial e de localização de forma a obter métricas sobre as viagens dos

²http://www.telecom.pt/InternetResource/PTSite/PT/Canais/Media/DestaquesHP/Destaques_2011/PTGalp_+InnovationChallenge_vencedores.htm

³http://www.uc.pt/gats/docs/resumo_premios_arrisca_c_2012

⁴<http://www.drivian.com/>

utilizados, enviando esta informação para o *backend*. É constituída também por uma componente social, que permite a partilha das atividades dos utilizadores em redes sociais. Uma vez centralizada a informação esta pode ser visualizada e analisada no *front-end web*, onde está disponível diversa informação sobre as viagens extraída através da aplicação de mecanismos de *Business Intelligence*(BI) sobre os dados recolhidos pela aplicação móvel.

As tecnologias desenvolvidas no projeto EcoMobile motivam a criação de produtos nas vertentes de *business-to-consumer*(B2C) e *business-to-business*(B2C). No âmbito da vertente B2C o projeto terá todas as facetas em foram descritas cima. No caso do B2B pretende-se expandir o produto para o ambiente empresarial, concretamente para frotas, neste caso para além de todas as funcionalidades da vertente B2C, exceto a componente social, pretende-se desenvolver componentes de gestão de condutores, veículos, tarefas e também ferramentas de BI, com diversas análises com vista a ajudar as organizações na definição das suas estratégias de negócio.

1.2 Objetivos

O principal objetivo deste estágio passa por desenvolver uma aplicação móvel, através da qual os utilizadores poderão registar as suas viagens e veículos, com vista a posterior consulta e análise de forma a melhorarem a sua segurança e economia. Toda a informação recolhida será enviada para um *backend* onde a mesma ficará registada. Mais concretamente, os objetivos de maior relevância para o estágio são:

- Registo e visualização dos trajetos das viagens dos utilizadores.
- Recolha e tratamento de dados dos sensores de movimento.
- Recolha e tratamento de dados do GPS.
- Integração social, para partilha de experiências com os amigos.
- Comunicação com o *backend* que irá manter o registo de todos os dados.
- *Live feedback* aos utilizadores durante uma viagem sobre os eventos que surgem, travagens, acelerações e curvas bruscas.
- Elaboração e cálculo dos índices de segurança e economia.
- Registo de viagens em tempo real, permitindo o acompanhamento das mesmas por outros utilizadores.

- Funcionamento do registo de viagem sem necessidade de ligação de dados.

Em forma de conclusão, o objetivo principal deste estágio consiste na implementação de uma aplicação que com base nos dados sensoriais e de localização recolhidos durante as viagens, ajudar os condutores a otimizarem o seu estilo de condução durante de forma a conseguirem aumentar a sua segurança e economia.

Também incluídos do âmbito de estágio surgem outros objetivos relacionados com a Engenharia de Software. Pretende-se adquirir experiência profissional em ambiente empresarial, integração numa equipa de trabalho, familiarização com processos de engenharia, tais como metodologias de desenvolvimento.

1.3 Estrutura do documento

De modo a facilitar a leitura e compreensão deste relatório optou-se pela divisão do mesmo em sete capítulos distintos.

A presente secção tem como objetivo familiarizar o leitor com o contexto do trabalho efetuado, sendo também definidos os objetivos que se pretendem atingir com o mesmo.

No capítulo 2, Metodologia e Planeamento, é apresentada a metodologia de trabalho, bem como a cronologia de trabalho. Inclui um comentário com a justificação e o impacto que os desvios tiveram no projeto.

No capítulo 3, Estado da Arte, são analisadas e comparadas diversas aplicações concorrentes a este projeto. É também apresentada uma análise às tecnologias escolhidas para o desenvolvimento.

No capítulo 4, Análise de Requisitos, são apresentados os requisitos para o projeto a desenvolver e o processo de levantamento dos mesmos.

No capítulo 5, Especificação da Arquitetura, a arquitetura geral e específica da aplicação.

No capítulo. 6, Trabalho Realizado, descreve todo o trabalho realizado com vista a concretização da solução proposta no capítulo anterior.

O capítulo 7, Testes, apresenta os testes efetuados para validar as componentes desenvolvidas.

O capítulo 8, Reflexões e Trabalho Futuro, apresenta as conclusões retiradas pelo autor assim como o trabalho futuro identificado pelo mesmo.

Capítulo 2

Metodologia e Planeamento

Neste capítulo apresenta-se a metodologia de desenvolvimento de software aplicada e o planeamento do estágio.

2.1 Metodologia de Desenvolvimento

Durante a fase de implementação da aplicação para este estágio, foi utilizada uma metodologia de desenvolvimento de software ágil baseada em *Scrum* de modo a permitir um melhor acompanhamento e avaliação do decurso do mesmo. Esta metodologia permite o desenvolvimento de software de uma forma iterativa, curta e incremental. Com a presença de uma lista de tarefas correspondentes às necessidades do sistema, *Product Backlog*, que é atualizada de forma regular, desta lista são retiradas as *user stories* que irão constar em cada iteração de desenvolvimento, *Sprint*, de acordo com as prioridades definidas. O processo utilizado inclui ainda reuniões diárias, *Daily Scrum Meeting*, para discutir o que foi, está e vai ser feito em cada dia, assim como também uma reunião semanal de acompanhamento e uma reunião final onde se fazia o balanço de cada *Sprint*[3]. Os artefactos da metodologia ágil produzidos podem ser consultados no Anexo E.

A metodologia usada com sucesso ao longo do estágio é caracterizada de seguida nas suas componentes chave.

- **Sprint** – um *sprint* consiste num período de tempo, uma a duas semanas, onde a equipa está empenhada e concentrada no desenvolvimento de um conjunto de funcionalidades bem definido. No final do *sprint* é esperada a total concretização das funcionalidades definidas resultando num incremento funcional do produto final a atingir

- **Product backlog** – representa o conjunto de requisitos a atingir no âmbito do projeto podendo existir uma definição de prioridades. Este conjunto de requisitos apresenta-se iterativo e incremental, em linha com a metodologia ágil adotada sendo definido pelo *Product Owner* do projeto.
- **Sprint backlogs** – no início de cada *sprint*, é feito um levantamento de *user stories* do *product backlog* a serem realizados. Para cada item é realizada uma análise e estimativa de duração e complexidade permitindo assim especificar o conjunto de itens a constituir no presente *sprint* transitando os restantes para os seguintes.
- **Sprint plan meetings** – esta reunião visa analisar o *sprint* decorrido, se tudo correu como inicialmente planeado assim como analisar o resultado. Feita esta análise, é então especificado o *sprint backlog* do próximo *sprint*.
- **Daily meetings** – diariamente é realizada uma reunião com uma duração muito curta, entre 5-10 minutos, onde é feito um levantamento do que foi realizado até ao momento, obstáculos encontrados e se existem impedimentos no decurso do *sprint* tentando nesta eventualidade encontrar uma resolução desde logo. O *sprint backlog* é também atualizado permitindo ver o seu estado e respetiva evolução.

Na Figura E é possível visualizar o processo da metodologia usada ficando evidenciadas as componentes base que a suportam.

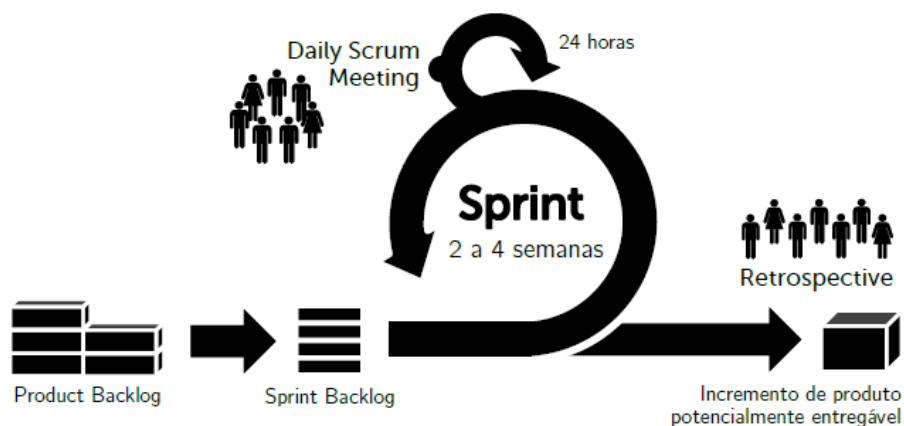


Figura 2.1: Representação da metodologia usada. Retirada de[1]

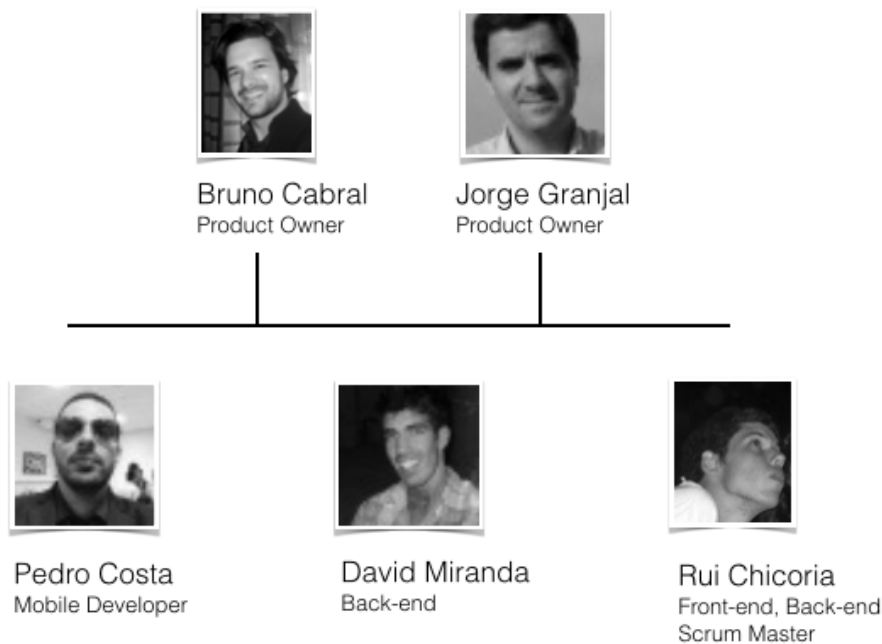


Figura 2.2: Elementos da equipa com contributo no presente estágio

Na Figura 2.2 é possível visualizar os elementos que constituem a equipa de trabalho no projeto e os respetivos papéis desempenhados.

Product Owner Representa o cliente e possui uma visão sobre aquilo que ele deseja, tendo assim a responsabilidade de transmitir essa visão para toda a equipa inerente ao *Scrum*. Este prioriza também as tarefas existentes no *Product Backlog*.

Mobile Developer Responsável pelo desenvolvimento da aplicação móvel do projeto, que representa os trabalhos inerente a este estágio.

Front-end Responsável pelo desenvolvimento da componente de *front-end* do projeto.

Back-end Responsável pelo desenvolvimento da componente de *back-end* do projeto.

Scrum Master Responsável por garantir que a equipa de desenvolvimento cumpre as práticas inerentes ao *Scrum*, tendo também um papel importante na ajuda aos elementos da equipa de desenvolvimento por forma a que o trabalho seja feito da melhor forma possível.

2.2 Gestão do projeto

Nesta secção serão apresentadas e descritas as ferramentas que foram usadas como apoio ao desenvolvimento e metodologia aplicada no projeto de estágio. Todas as ferramentas utilizadas foram configuradas nos servidores da Sentilant.

Jira

A ferramenta Jira[4], desenvolvida para Atlassian [5], que foi usada na gestão do processo de desenvolvimento ágil. Esta ferramenta permite gerir todas as componentes, já descritas na secção anterior do processo de desenvolvimento seguido, criação do *Product Backlog*, *Sprints*, acompanhamento da evolução de cada Sprint, visualização de métricas sobre o estado e velocidade de desenvolvimento do projeto. Na Figura 2.3, encontram-se algumas imagens ilustrativas da aplicação desta ferramenta.

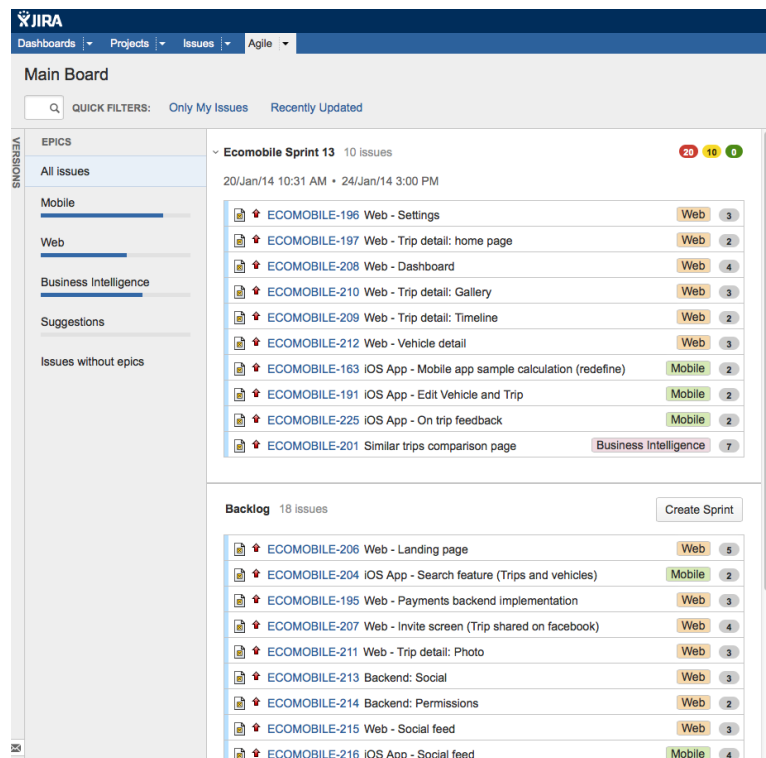


Figura 2.3: Imagem da ferramenta Jira.(Vista com o *backlog*)

Confluence

A ferramenta Confluence[6], consiste numa ferramenta para construção de Wikis corporativas, que permite criar e partilhar páginas, documentos e conteúdo entre os colaboradores de uma organização. Foi usado na gestão do projeto, todos os documentos e ficheiros associados ao projeto, desde estudos efetuados, criação de *mockups*, apresentações. Na Figura 2.4 está ilustrada as funcionalidades descritas em cima.

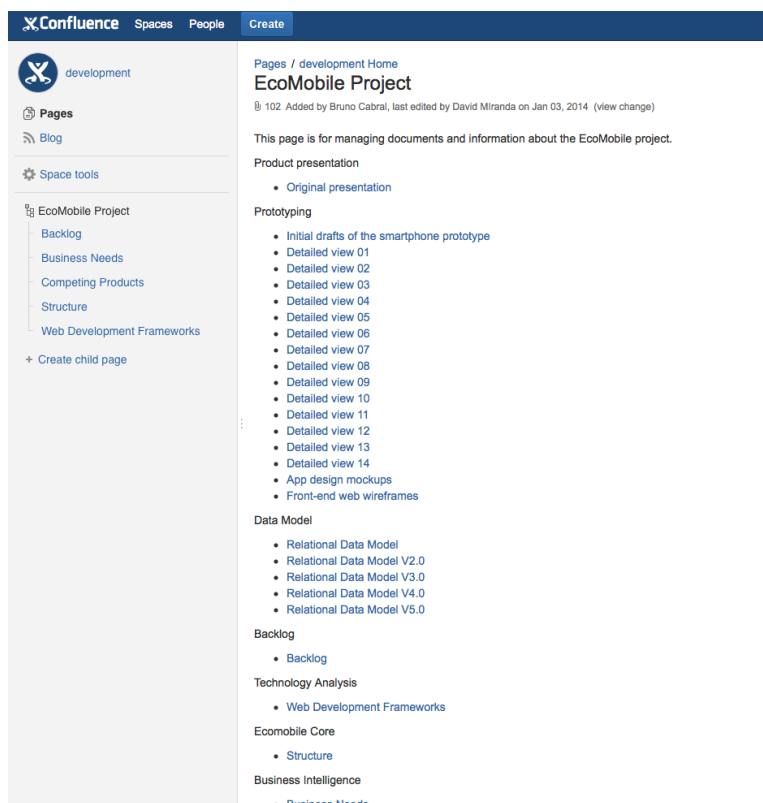


Figura 2.4: Imagem da ferramenta Confluence.(Organização dos *mockups*)

Gitlab

Atualmente, utilizar uma ferramenta de controlo de versões é essencial para desenvolver qualquer projeto de software. Com uma ferramenta deste género pode-se gerir as diferentes versões dos ficheiros em desenvolvimento, podendo o utilizador controlar o histórico dos ficheiros do código produzidos. Neste projeto a ferramenta adotada neste contexto foi o Gitlab[7] como plataforma de controlo de versões do código.

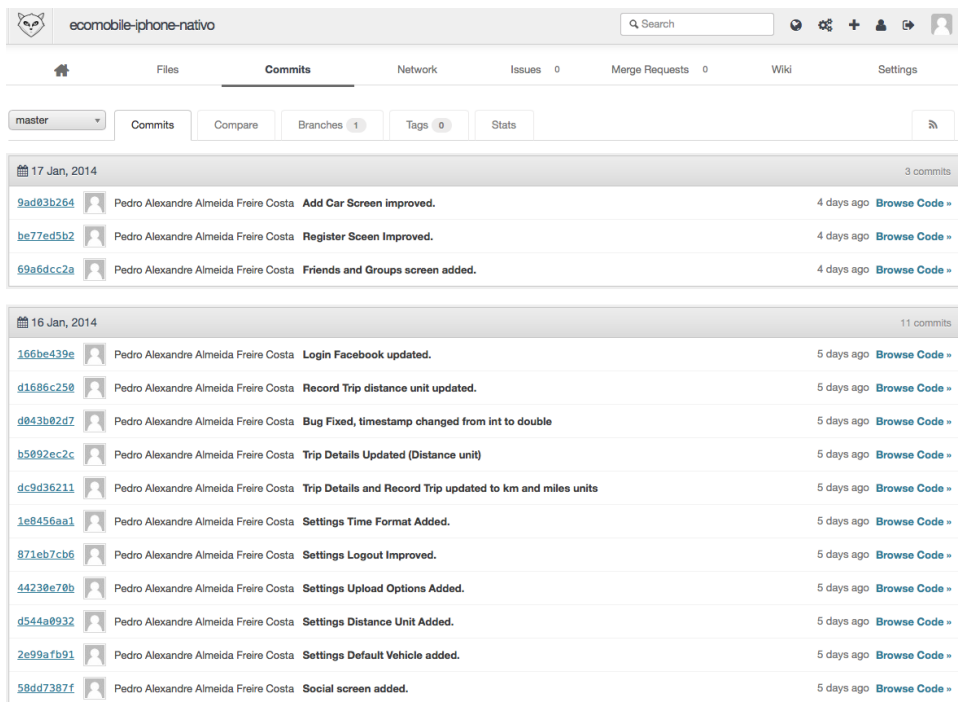


Figura 2.5: Imagem da ferramenta Jitlab. (Lista de *commits*)

Para estruturação e controlo de versões do código desenvolvido foi utilizado o Git¹, sendo este de uma natureza distribuída. O repositório Git não é mais que uma simples base de dados que contém toda a informação necessária para reter e gerir as revisões e o histórico de um projeto[8]. Para se perceber melhor os conceitos deste sistema de controlo de versões, é preciso perceber os seguintes conceitos:

- **Commits**: quando um utilizador faz um *commit* no Git, este guarda um objeto que contém um ponteiro para o que foi colocado na área de seleção, o autor e os metadados da mensagem e ainda zero ou mais ponteiros para o/os *commits* que são pais deste *commit*.

- **Branch**: os *branches* permitem ao programador definir e controlar as versões do seu código de forma fácil. Quando se muda de um *branch* para outro, altera-se o contexto (ou a versão) do código.

- **Merge**: Depois do programador trabalhar isoladamente no seu *branch*, poderá querer incorporar o código no *branch* principal e para isso terá que realizar um *merge*. Poderá acontecer que existam conflitos e desta forma é necessário que seja o próprio programador a resolve-los manualmente.

¹<http://git-scm.com>

Depois de entendidos os conceitos referidos, irá falar-se um pouco sobre o ambiente usado no estágio. Assim, foram usados três tipos de *branches*:

master: contém o código de cada versão da aplicação.

develop: para armazenar o código que está estável, que contém o código após cada ciclo/iteração de desenvolvimento.

funcionalidade: esta *branch* contém todas as funcionalidades que estão a ser implementadas, variando o seu nome em cada funcionalidade que está a ser implementada.

Os três tipos de ramos apresentados estão guardados na rede interna da empresa Sentilant. Os *commits* foram feitos regularmente (no mínimo 2 a 3 vezes por dia) e o *merge* não foi utilizado uma vez que não existiam problemas de alterações feitas por outros programadores, já que foi apenas o estagiário produziu o código da aplicação.

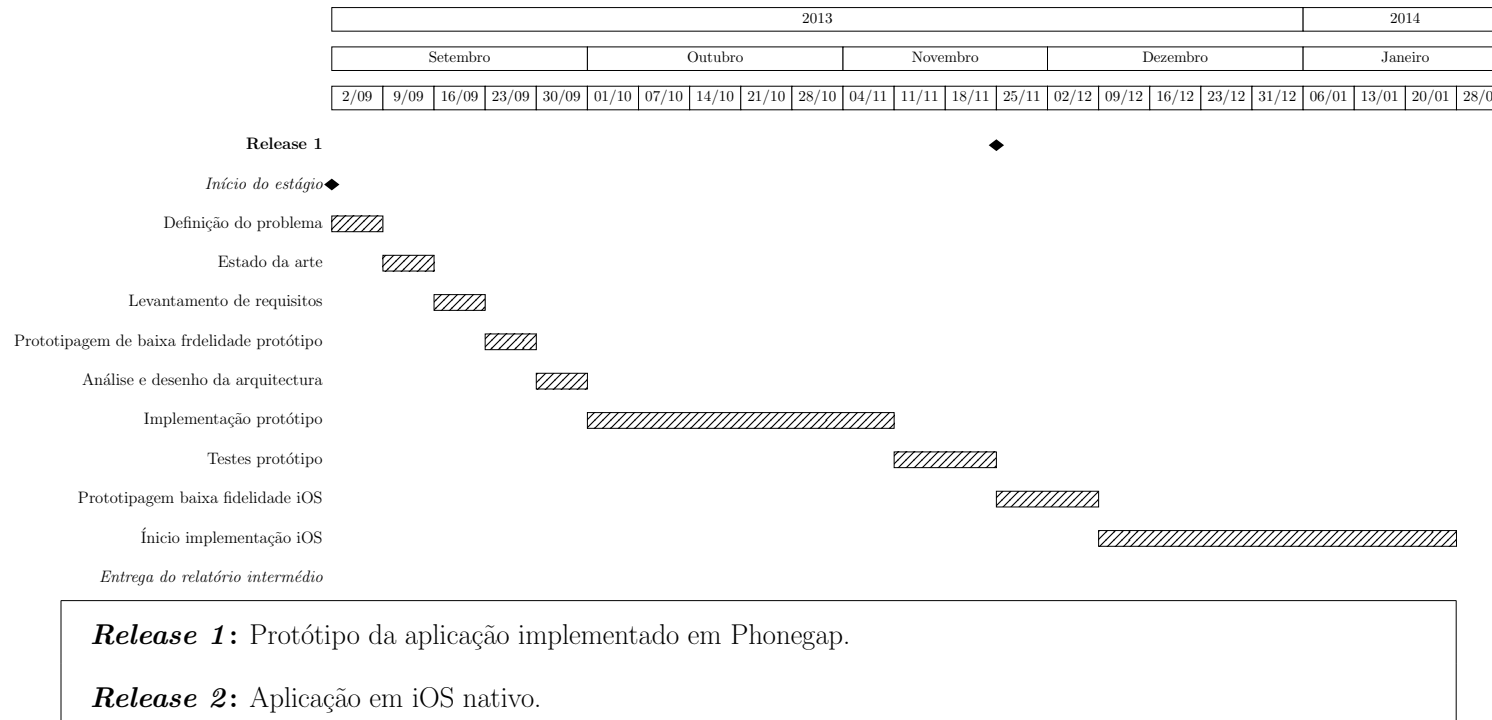
2.3 Planeamento

O planeamento do estágio foi organizado numa fase inicial de definição do problema e estudo de mercado, ficando o período de desenvolvimento dividido em três *releases* - versões estáveis do produto que englobam parte ou a totalidade das funcionalidades.

2.3.1 Primeiro semestre

O trabalho desenvolvido no primeiro semestre consistiu na análise do estado da arte, análise e definição de requisitos, prototipagem, definição da arquitetura, desenvolvimento do protótipo Phonegap e início da implementação nativa da aplicação para o sistema operativo iOS.

Planeamento do 1º Semestre



2.3.2 Segundo semestre

O segundo semestre foi caracterizado pela continuação do desenvolvimento da aplicação iOS, e melhoria de algumas tarefas já desenvolvidos, principalmente o tratamento da informação sensorial recolhida. Foi também elaborado o relatório final de estágio.

2.3.3 Desvios

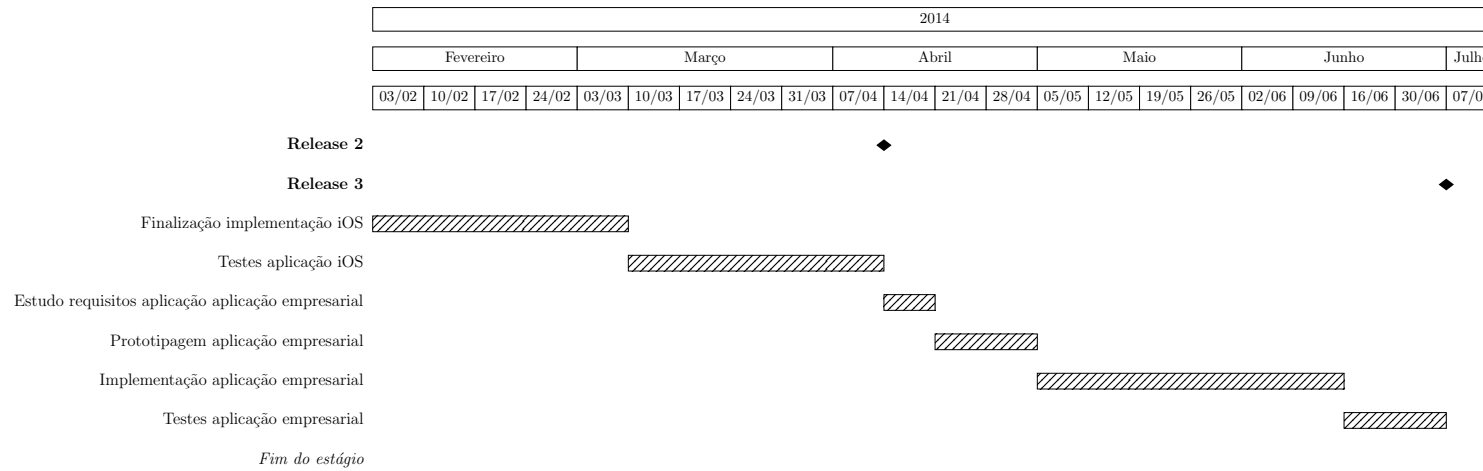
Qualquer projeto de software encontra-se suscetível a desvios relativamente ao que foi definido na sua concessão e especificação devido a alterações de prioridades, especificações incorretas entre outros fatores.

Durante o desenvolvimento houve algumas alterações relativamente ao inicialmente planeado sobretudo no que diz respeito ao tratamento dos dados sensoriais recolhidos pelo dispositivo móvel. Esta tarefa revelou-se bastante mais complexa do que aquilo que foi previsto inicialmente. A base da aplicação desenvolvida neste projeto concentra-se no correto tratamento dos dados sensoriais de forma a obter métricas e indicadores fidedignos, pelo que opção tomada pela empresa foi de prolongar esta tarefa até os resultados serem os satisfatórios.

Ao mesmo tempo que decorreu a implementação da solução móvel do projeto, outros elementos da equipa trabalhavam no *front-end* e *back-end*, onde também ocorreram atrasos. Estes atrasos levaram à adição de algumas novas funcionalidades na aplicação, conseqüentemente novos requisitos, alterando assim também o plano de trabalhos para o segundo semestre.

As alterações ao planeamento descritas em cima levaram à alteração dos planos de trabalho para o segundo semestre, ficando adiada a implementação para Android da aplicação empresarial.

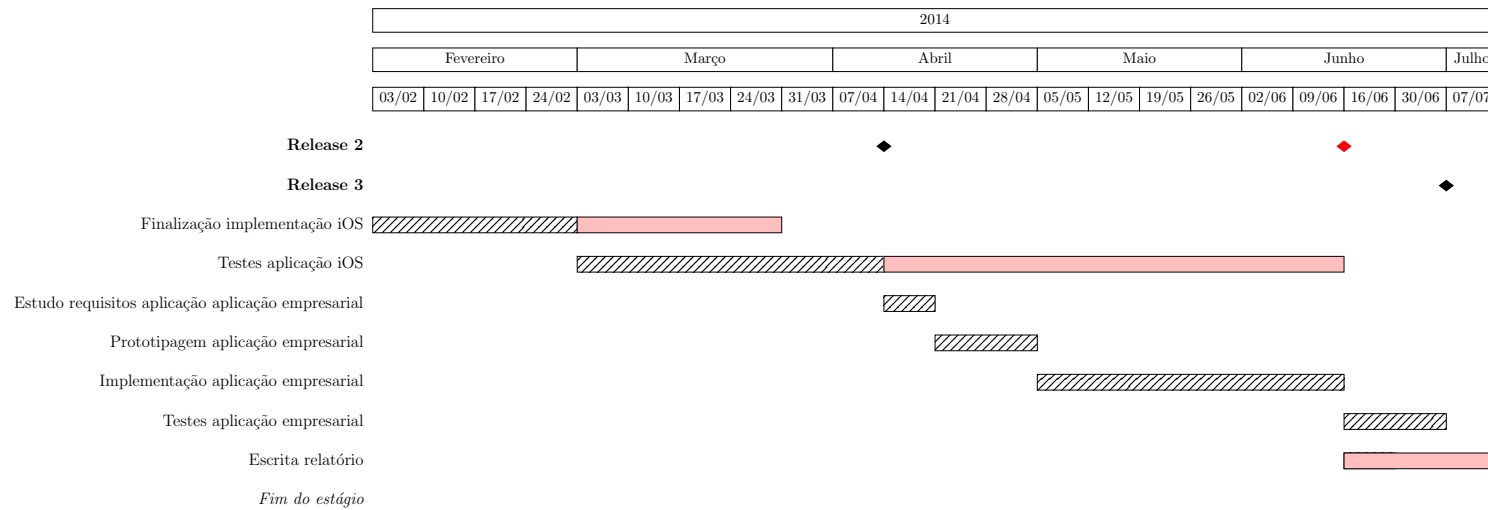
Planeamento do 2º Semestre



Release 2: Aplicação em iOS nativo.

Release 3: Aplicação em Android nativo para ambiente empresarial.

Execução do planeamento do 2o Semestre



Release 2: Aplicação em iOS nativo.

Release 3: Aplicação em Android nativo para ambiente empresarial.

2.4 Análise de Riscos

Nesta secção são identificados os riscos que poderão provocar um efeito negativo nos objetivos a que a aplicação a desenvolver se propõe a cumprir. Foi realizado um levantamento dos riscos previstos e da sua probabilidade de ocorrência. Juntamente com cada risco identificado é apresentada uma proposta de mitigação, uma forma de evitar ou reduzir os danos causados.

Risco 01	Tipo Técnico	Impacto Crítico	Probabilidade Média
Descrição Não aceitação da aplicação nas lojas de aplicações do iOS e Android.			
Estratégia de mitigação Seguir as orientações descritas nos guias de implementação de aplicações para estes sistemas, disponibilizados pela <i>Apple</i> e <i>Google</i>			
A ocorrência deste risco não se verificou, uma vez que a aplicação ainda não foi submetida para aprovação nas lojas de aplicações.			

Risco 02	Tipo Recurso	Impacto Crítico	Probabilidade Média
Descrição Limitações físicas dos dispositivos móveis, poder de processamento, memória disponível, fonte de alimentação limitada.			
Estratégia de mitigação Aplicação das ferramentas de <i>profiling</i> de uso de recursos			
A ocorrência deste risco não se verificou.			

Risco 03	Tipo Técnico	Impacto Crítico	Probabilidade Média
Descrição A curva de aprendizagem com as tecnologias utilizadas durante o projeto pode ser elevada levando a atrasos e incumprimento dos objetivos.			
Estratégia de mitigação Constante estudo das tecnologias usadas e em caso de dificuldades mudar antecipadamente de tecnologias			
A ocorrência deste risco não se verificou.			

Risco 04	Tipo Projeto	Impacto Crítico	Probabilidade Alta
Descrição Alterações de prioridade dos requisitos devido a demonstrações ou aparecimento de novos requisitos.			
Estratégia de mitigação Monitorização e acompanhamento constante de modo a garantir uma sincronização das necessidades da empresa e os objetivos do estágio.			
A ocorrência deste risco verificou-se, concretamente nos requisitos relacionados com o tratamento, cálculo de índices e deteção de eventos, tendo o seu impacto levado a atrasos nos planos do segundo semestre, como descrito em 2.3.3.			

Risco 04	Tipo Projeto	Impacto Crítico	Probabilidade Alta
Descrição Algoritmos de tratamento e cálculo de índices e eventos incorretos.			
Estratégia de mitigação Utilização de testes em ambiente real de forma a validar os algoritmos.			
A ocorrência deste risco verificou-se na fase de testes, em que foi detetado que o comportamento destes algoritmos não era o esperado. O seu impacto levado a atrasos nos planos do segundo semestre, como descrito em 2.3.3.			

Capítulo 3

Estado da arte

O levantamento do estado da arte é uma etapa de elevada importância no desenvolvimento de um projeto por permitir avaliar a sua viabilidade através da análise de soluções existentes, identificando os pontos fortes e fracos que estes possuem, assim como analisar o nível de saturação do mercado no sector em questão. É com base nesta análise que é possível determinar o nível de risco do projeto e respetiva probabilidade de sucesso.

Serve o presente capítulo para descrever as soluções presentes no mercado nas quais este projeto se enquadra. Será primeiramente apresentadas as tecnologias disponíveis que são necessárias para a criação da aplicação. Posteriormente será apresentado um estudo com as soluções mais semelhantes atualmente disponíveis no mercado, de forma a garantir que a aplicação a desenvolver será original. No final deste capítulo é apresentada uma comparação global entre todas as soluções estudadas.

3.1 Sistemas Operativos Móveis

Neste capítulo irão ser apresentadas as tecnologias móveis que servem como base para o desenvolvimento da aplicação pretendida.

Nesta secção são apresentadas as principais plataformas móveis presentes no mercado e que irão ser utilizadas na realização deste estágio.

Um dos objetivos principais quando se cria uma aplicação móvel é que esta alcance o maior número de pessoas possível. Na análise dos dados presentes na Figura 3.1, os sistemas operativos iOS e Android dominam claramente o mercado, deixando muito para trás os seus concorrentes.

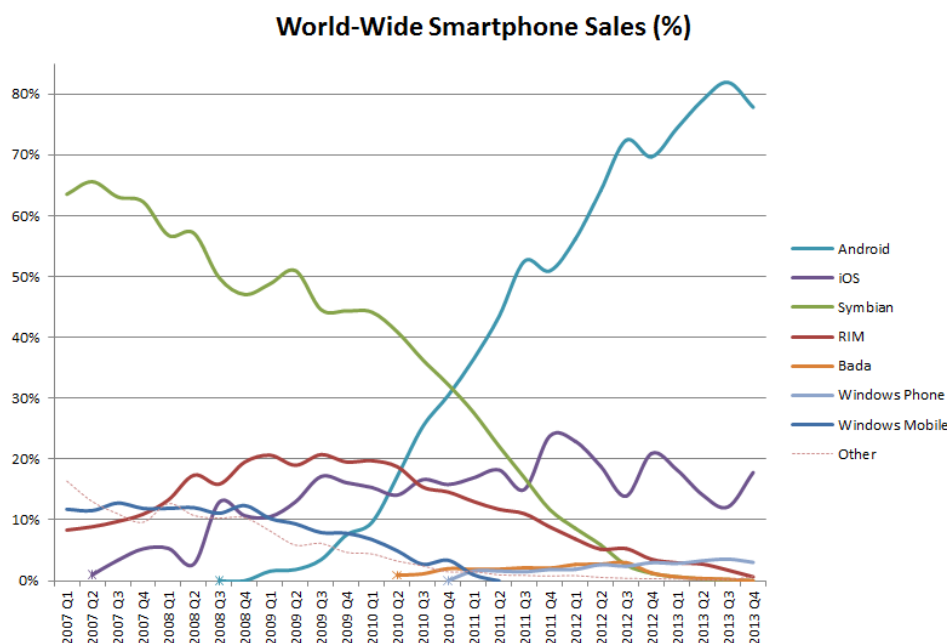


Figura 3.1: Sistemas operativos m3veis *market share* em 2014[2]

3.1.1 iOS

A plataforma iOS 3 o sistema operativo m3vel criado pela Apple em 2007, juntamente com o iPhone, tendo sido posteriormente adotado em outros lan3amentos da marca, como o iPad e iPod Touch. Teve uma escalada de popularidade impressionante com o advento dos *smartphones* nos anos recentes e com a cria33o de uma loja virtual, a App Store, que comercializa aplica33es para a plataforma. Para o desenvolvimento de uma aplica33o que seja executada no sistema iOS foi necess3ria a defini33o de alguns requisitos ao seu desenvolvimento, nomeadamente a utiliza33o do sistema operativo Mac OS X pois apenas neste 3 poss3vel executar o SDK disponibilizado pela Apple. Foi deste modo utilizado o software Xcode fornecido pela Apple que disponibiliza todas as ferramentas necess3rias ao desenvolvimento. A linguagem utilizada 3 obrigatoriamente Objective C uma vez que 3 a 3nica op33o existente. Relativamente 3 integra33o com mapas, 3 disponibilizada uma API de mapas nativos pelo SDK.

Este sistema caracteriza-se pela sua simplicidade e facilidade de utiliza33o, estabilidade e robustez, pois o seu *Kernel* 3 baseado no sistema operativo UNIX. O facto do seu *hardware* ser limitado aos dispositivos fabricados pela *Apple* apresenta-se como outro ponto forte, uma vez que o *software* 3 produzido tendo sempre em conta um *hardware* espec3fico.

Todas as aplicações iOS usam um modelo de arquitetura denominado ***Model-View-Controller***. Este modelo assenta no conceito de isolar a camada de lógica procedimental da camada de interface gráfica, ou seja, todo o código lógico que trata e modela dados versus a interação com o utilizador. A Figura 3.2 ilustra esta arquitetura.

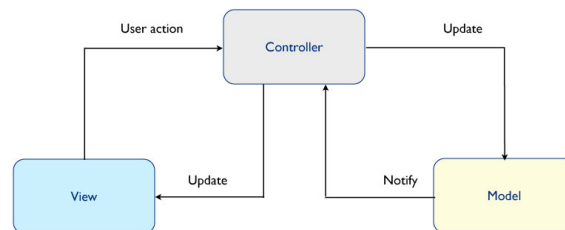


Figura 3.2: Arquitectura *Model-View-Controller* aplicação iOS

O *Controller* é responsável por contactar a interface gráfica, recolher toda a informação pertinente desta como toques, scrolls, swipes, e depois contactar o modelo de lógica (*model*) para este trabalhar os dados da tarefa pedida. Quando esses dados estiverem prontos, o controlador tem duas opções de ação, ou trata de atualizar a vista atual (*view*) com a nova informação ou cria um novo controlador de raiz, tendo este a responsabilidade de criar uma vista para si e tratá-la. Usando este modelo, a camada de lógica nunca interage com a camada gráfica. Este conceito permite uma maior estabilidade em *runtime* mas também uma maior liberdade de *debugging*, pois cada secção está separada e não interage diretamente com as outras, evitando acessos indevidos a dados[9].

3.1.2 Android

O sistema operativo Android, pertencente à *Google*, é constituído por um *kernel* baseado no sistema operativo UNIX, lançado em 2008. Sendo um sistema operativo com um *kernel* baseado em UNIX, foi desenvolvido para ser estável, seguro e robusto, permitindo que as suas aplicações sejam executadas num ambiente que não tem acesso aos recursos do sistema, a não ser com autorização do utilizador do dispositivo.

A *Google* disponibiliza um SDK de desenvolvimento bastante completo, com um emulador do sistema operativo compatível com Mac OS X, Linux e Windows. O desenvolvimento para Android é na sua maioria realizado na linguagem Java, sendo interpretado no sistema operativo móvel por uma máquina virtual designada de *Dalvik*.

A criação de uma loja virtual de aplicações, *Play Store*, permite aos utilizadores a distribuição das suas aplicações depois de aprovação, sendo gratuita a publicação de distribuição das mesmas, o que torna este sistema escolhido por muitos utilizadores.

No mercado de dispositivos móveis, a maioria das empresas adotou o Android como o sistema operativo dos seus dispositivos, o que tem levado a um crescimento de mercado bastante elevado por parte do Android.

A implementação nativa para o sistema operativo Android estava planeada para ocorrer no segundo semestre, sendo previsto realizar um estudo sobre as ferramentas a usar para a sua implementação, sabendo à partida que pode-se usar o mesmo modelo MVC que foi usado na aplicação nativa iOS, levando a crer que a sua implementação seja facilitada.

3.1.3 Plataformas híbridas

Phonogap

O Phonogap é uma *framework* de desenvolvimento móvel, desenvolvida pela Adobe Systems, que permite aos programadores criarem aplicações para dispositivos móveis usando JavaScript, HTML5 e CSS3, em vez das linguagens de programação específicas de cada dispositivo. O resultado consiste numa aplicação híbrida, visto que não é completamente nativa, sendo que toda a parte de renderização do *layout* é feita recorrendo a *webviews*, que representam um *browser* ser a sua UI.

Na Figura 3.3 pode-se ver em mais detalhe a arquitetura de uma aplicação Phonogap.

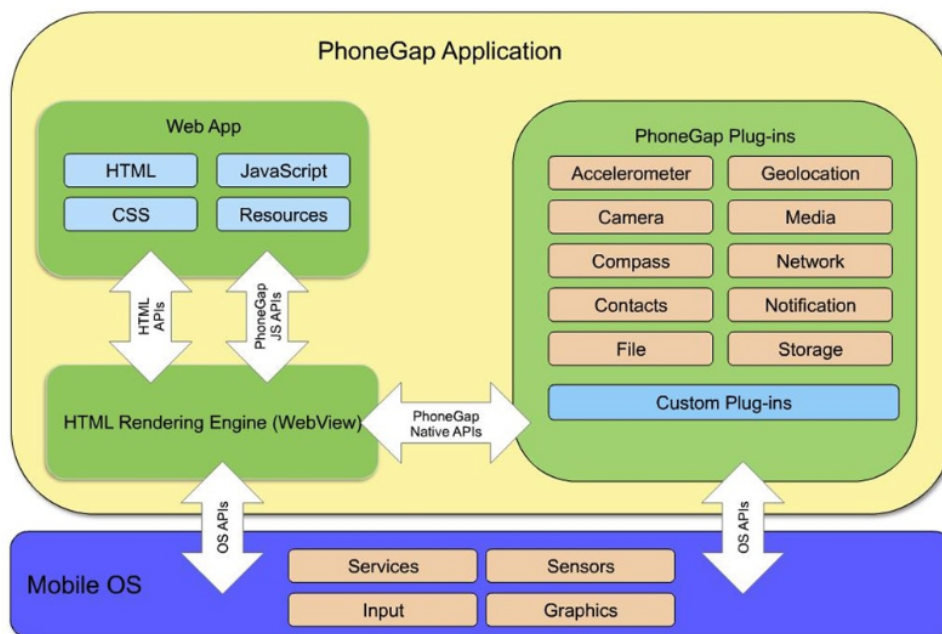


Figura 3.3: Arquitetura de uma aplicação Phonegap

Os sistemas operativos móveis para os quais se pretende desenvolver a aplicação, Android e iOS, são suportados pelo Phonegap, assim como os recursos que a aplicação necessita para o seu funcionamento, GPS, sensores de movimento, acesso à câmara, rede e armazenamento.

3.2 Integração com redes sociais

As redes sociais estão cada vez mais presentes no dia-a-dia das pessoas, tornando-se uma técnica chave na sociedade moderna, permitindo ao utilizador publicar de forma rápida diversa informação sobre os seus hábitos, atividades e gostos.

Neste tópico foi analisada a ferramenta para o rede social Facebook, disponibilizada pela mesma, que permite a integração dos sistemas operativos móveis abordados neste capítulo com o Facebook, denominada Facebook Graph API¹. Trata-se de uma API com base em HTTP, que permite efetuar consulta de dados, postar novas histórias ou efetuar qualquer outra operação disponibilizada pela mesma.

O Facebook Graph API apresenta uma visão simples e consistente do grafo social do Facebook, representando uniforme mente os objetos no grafo

¹<https://developers.facebook.com/docs/graph-api>

(pessoas, fotos, eventos e páginas) e as conexões entre eles (relacionamentos entre amigos, conteúdo partilhado, marcadores de fotos). Cada objeto no grafo tem uma identificação única. As respostas das solicitações são devolvidas no formato JSON. É aqui importante referir que a informação do grafo apenas pode ser consultada em perfis públicos, caso o perfil do utilizador esteja definido como privado, não é possível aceder à informação pretendida.

3.3 Soluções existentes

Neste capítulo é apresentada a análise a algumas aplicações existentes no mercado.

Antes de iniciar a análise dos requisitos da aplicação, foram estudadas as soluções atualmente disponíveis no mercado concorrentes à aplicação que foi desenvolvida durante o estágio. São apresentadas as aplicações consideradas mais relevantes bem como uma breve descrição dos objetivos funcionalidades que cada uma delas oferece.

Esta análise revelou-se muito importante pois permitiu identificar as empresas concorrentes e analisar os pontos fortes e os pontos fracos da concorrência. Após a análise individual das aplicações será apresentada uma tabela comparativa entre estas, bem como as conclusões retiradas após esta análise.

3.3.1 My Car Tracks

O MyCarTracks[10] é uma aplicação para o sistema operativo móvel Android, desenvolvida pela empresa Slash Idea s.r.o². Esta aplicação tira partido do GPS do *smartphone* para a motorização de veículos, fazendo o registo das viagens, associando a informação referente à velocidade, distância percorrida e duração. Toda a informação recolhida é enviada para um servidor, disponibilizando uma consulta posterior da informação das viagens quer na aplicação móvel quer num *front-end* web. No *front-end* é possível localizar veículos em tempo real, exportar a informação sobre a forma de relatórios. O MyCarTracks não faz uso dos sensores de movimentos, não detectando comportamento bruscos na condução, não dá *feedback* nem regista a informação referentes à economia e segurança dos condutores.

²<http://www.slashidea.com>

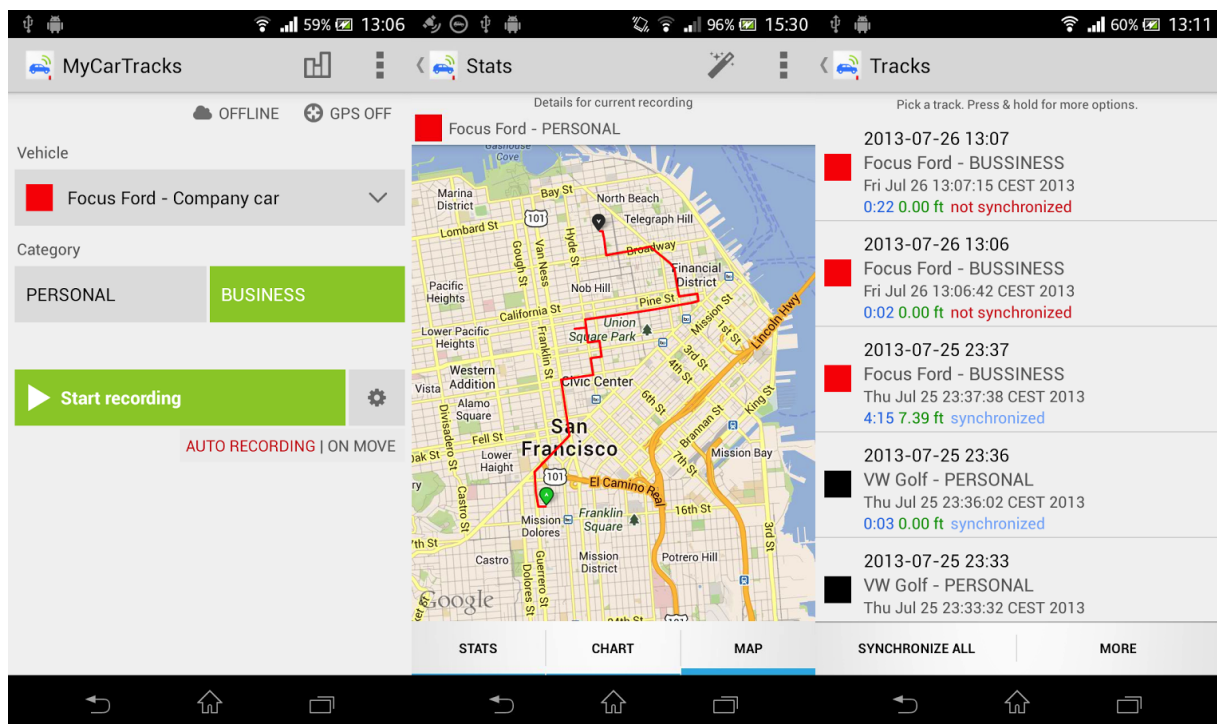


Figura 3.4: Imagem da aplicação MyCarTracks

3.3.2 Automatic

O Automatic[11] é uma aplicação para os sistemas operativos móveis iOS e Android, desenvolvida pela empresa Automatic Labs, Inc. Esta aplicação requer a utilização de um componente de hardware auxiliar, que se liga ao veículo, que regista toda a informação relevante para a segurança e economia da condução que é possível obter pela própria viatura, OBD2. Esta aplicação durante o registo das viagens, avisa o utilizador, através de alertas visuais e sonoros, dos eventos que estão a condicionar a sua economia e segurança. Uma vez que esta aplicação, através do componente instalado no veículo, lê a informação do mesmo, permite obter a informação sobre o consumo durante e ao longo de uma viagem, assim como o seu custo.

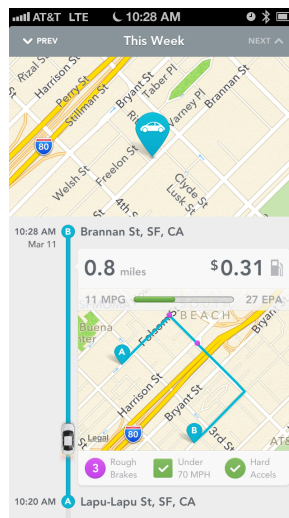


Figura 3.5: Imagem da aplicação Automatic

3.3.3 Drismo

O Drismo[12] é uma aplicação de código aberto para o sistema operativo móvel Android, que usa o GPS e os sensores de movimento para analisar a qualidade de condução de uma viagem, com o objectivo do utilizador perceber se está com um estilo de condução bom ou mau, de forma a poder melhorá-lo. Durante uma viagem é disponibilizado *feedback* sobre os eventos que influenciam a qualidade de condução do utilizador.

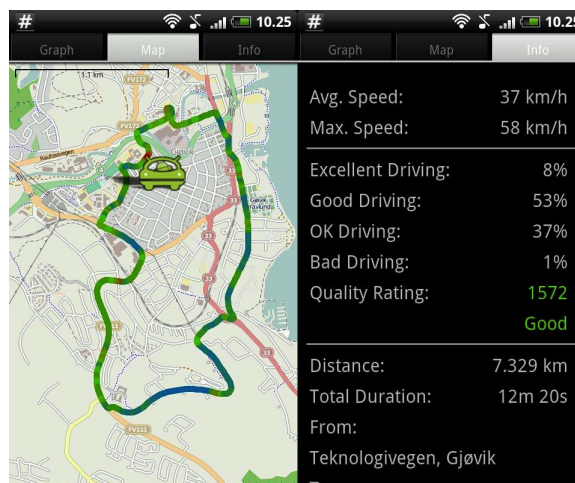


Figura 3.6: Imagem da aplicação Drismo

3.3.4 My Tracks

O My Tracks[13] é uma aplicação para o sistema operativo móvel Android, de código aberto desenvolvida pela Google. Esta aplicação faz uso do GPS para recolher a informação de altitude, velocidade, distância e localização. A informação recolhida fica guardada no dispositivo sendo possível a sua posterior consulta e também a exportação para análise externa. Na componente social permite a partilha das atividades registadas nas redes sociais Google Plus, Facebook e Twitter.

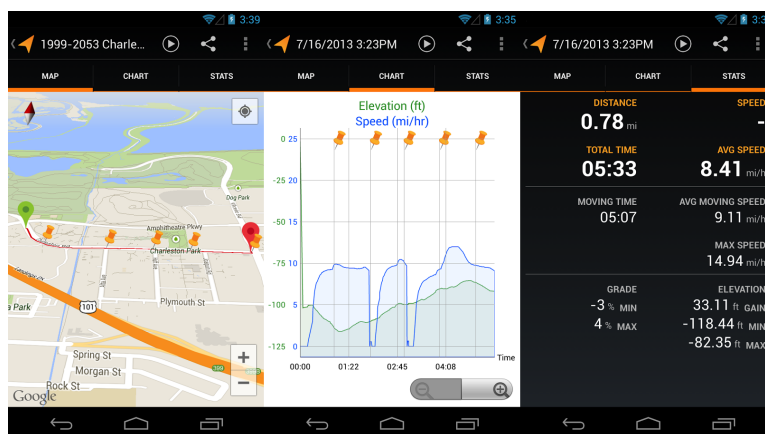


Figura 3.7: Imagem da aplicação My Tracks

3.3.5 TomTom Webfleet Mobile

O TomTom Webfleet Mobile[14] é uma aplicação para os sistemas aperitivos móveis iOS e Android, desenvolvida pela empresa TomTom International BV³. Esta aplicação não faz o registo de viagens, apenas permite a visualização da informação recolhida pelo hardware do TomTom Webfleet. A informação recolhida passa pela localização dos veículos em tempo-real, quilometragem, velocidade, duração das viagens, recomendação de rotas e criação de relatórios. O TomTom Webfleet está primordialmente focado na recolha de informação essencial para o monitorização e gestão de frotas.

³<http://www.tomtom.com>



Figura 3.8: Imagem da aplicação TomTom WebFleet Mobile

3.3.6 Comparativo

Nesta secção é apresentada a Tabela 3.1 com as diversas funcionalidades de cada uma das aplicações descritas anteriormente de forma a serem mais explicitas as diferenças entre si.

Tabela 3.1: Comparação soluções equivalentes

	MyCarTracks	Automatic	Drismo	MyTracks	TomTom Webfleet Mobile
Sistemas Operativos	Android	Android e iOS	Android	Android	Android iOS
Hardware Externo	-	Adaptador OBD2	-	-	Equipamento localização, monitorização, navegação.
Utiliza Sensores Movimento	Não	Não	Sim	Não	Não
Qualidade Condução	Não	Sim	Sim	Não	Não
Live feedback	Não	Sim	Sim	Não	Não
Front-end Web	Não	Sim	Não	Não	Sim
Partilha Social	Não	Não	Sim	Sim	Não

Um dos factores mais importantes na escolha da adoção de uma solução por parte dos utilizadores e empresa é o seu impacto monetário, daí ser de grande importância a análise dos custos de cada uma das aplicações estudadas. Está presente na Tabela 3.2 um resumo sobre os custos de cada solução, apenas para evidenciar as que realmente apresentam custos associados.

Tabela 3.2: Comparação custos das soluções equivalentes

	MyCarTracks	Automatic	Drismo	MyTracks	TomTom Webfleet Mobile
Aplicação	Grátis	Grátis	Grátis	Grátis	8.99
Hardware Externo	Não aplicável	73	Não aplicável	Não aplicável	9.90a 59.40
Custo anual	Gratuito ou até 130	Não aplicável	Não aplicável	Não aplicável	249a 379

3.3.7 Sumário

No estudo de mercado efetuado verificou-se a dificuldade em encontrar soluções móveis que ofereçam as funcionalidades pretendidas. Existem diversas soluções que vão de encontro à aplicação que se pretende desenvolver, mas geralmente nenhuma é composta por uma aplicação móvel, mas sim por *hardware* específico e um *front-end* web. Das aplicações móveis encontradas, a sua análise detalhada revelou que os produtos atualmente existentes no mercado, não faz uso dos sensores de movimento como forma de análise da qualidade de condução, limitando-se a registar a informação proveniente do GPS. Das soluções mais completas analisadas, o Automatic e o TomTom Webfleet Mobile, verifica-se a necessidade da instalação de *hardware* adicional nas viaturas, tornando estas soluções bastante dispendiosas para os utilizadores individuais, assim como para as pequenas empresas. No contexto social apenas o Drismo e o MyTracks permitem a partilha de viagens em redes sociais.

O produto a desenvolver destaca-se dos restantes por não requerer a instalação de nenhum *hardware* adicional, e por não se limitar à monitorização da viagens, mas também à sua análise em tempo real, tirando partido da informação dos sensores de movimento, fornecendo *live feedback* sobre os comportamentos detectados durante as viagens. O contexto social que se pretende atribuir à aplicação, com a partilha de informação em redes sociais destacasse também em relação à restante oferta existente no mercado.

Em conclusão pode-se dizer que a aplicação desenvolvida recolhe mais informação que as existentes no mercado. Não se limita a registar e disponibilizar a telemetria, mas também a detectar comportamentos em tempo real, geração de índices de economia e segurança e deteção de eventos, sem a necessidade de *hardware* adicional e por um custo bastante mais reduzido.

Capítulo 4

Análise de Requisitos

O processo de especificação de requisitos apresenta-se como uma das etapas cruciais no desenvolvimento de um projeto de software, permitindo identificar quais os objetivos pretendidos e a forma como devem ser atingidos, possibilitando não só definir uma linha orientadora para os desenvolvimentos assim como uma forma de avaliar objetivamente o sucesso e concretização do projeto.

Na elaboração do conjunto de requisitos, foi feito o levantamento de um conjunto de necessidades de negócio, com a finalidade de recolher vários indicadores-chave, com base no estudo de mercado realizado no capítulo anterior, como forma de conhecer as funcionalidades que as soluções já existentes oferecem.

Este capítulo serviu para elaborar os requisitos para a aplicação móvel. Com a elaboração deste documento possui-se uma visão mais clara das funcionalidades a implementar para a aplicação. Para isso recorreu-se a elaboração de *user stories*, requisitos funcionais apoiados por casos de uso, assim como a requisitos não funcionais.

4.1 Identificação de Requisitos

4.1.1 Atores

Os atores são os tipos de utilizadores que interagem diretamente com a aplicação, neste caso irão existir dois atores: *Utilizador com Registo* e *Utilizador sem Registo*. O ator *Utilizador com Registo* é aquele que já tem um registo na aplicação, tendo acesso a todas as funcionalidades que esta dispõe. O ator *Utilizador sem Registo*, como o nome indica, não se encontra registado

na aplicação, pelo que a única funcionalidade disponível para este é a de registo e autenticação na aplicação.

4.1.2 Prototipagem

A identificação dos requisitos foi efetuado recorrendo também à técnica de prototipagem, permitindo ter uma ideia visual sobre aquilo que se pretende obter, facilitando assim a especificação dos requisitos. Numa fase inicial de desenvolvimento dos protótipos, foram primeiramente tomadas decisões ligadas ao design e usabilidade da aplicação com toda a equipa de desenvolvimento, recorrendo a sessões de *brainstorming*. O desenvolvimento deste trabalho permitiu, através de prototipagem de baixa fidelidade, elaborar uma primeira versão da aplicação baseado em requisitos ainda pouco definidos, servindo assim como forma de especificar concretamente qual o conjunto de requisitos desejados. A totalidade dos resultados da prototipagem realizada na identificação dos requisitos encontra-se nos Anexos B e C.

Nas Figuras 4.1 e 4.2 encontram-se alguns exemplos dos protótipos de baixa fidelidade criados durante a fase de prototipagem.

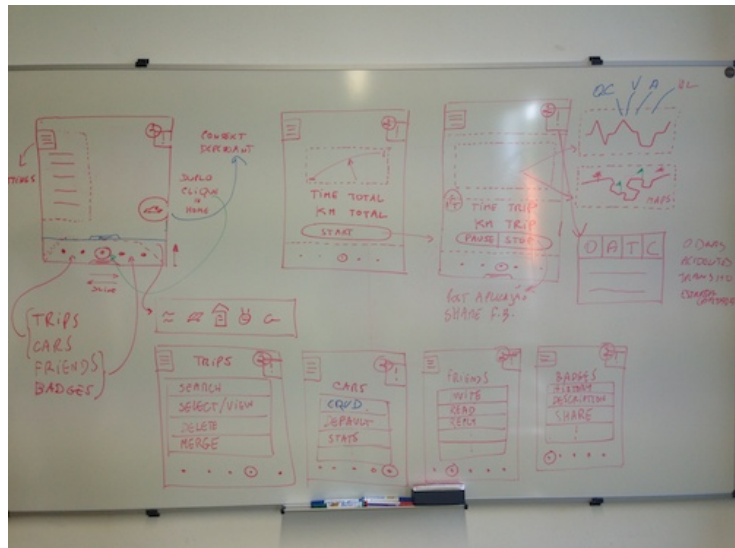


Figura 4.1: Prototipagem de baixo nível para o Phonegap

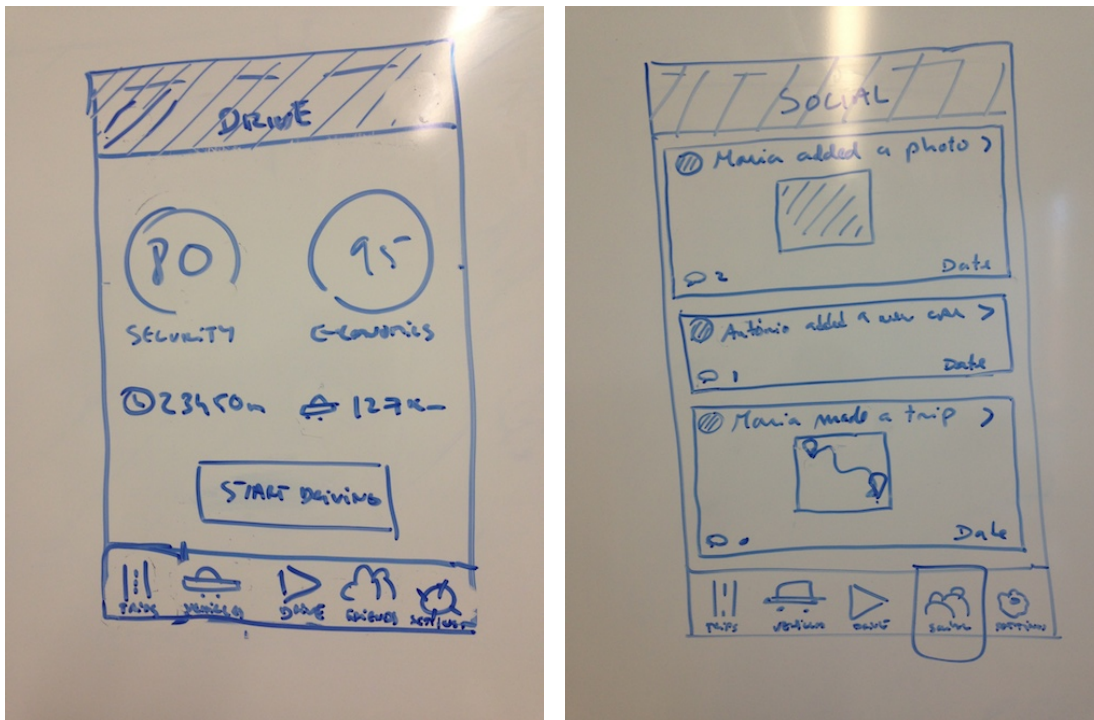


Figura 4.2: Prototipagem baixo nível para a aplicação nativa iOS

4.1.3 Requisitos funcionais

As funcionalidades do projeto foram especificadas através de *user stories*, que consistem numa breve descrição de uma determinada funcionalidade na perspetiva de um ator do sistema, para qual esta é relevante, evitando assim o foco em funcionalidades com um ganho reduzido para os utilizadores do produto final.

As *User Stories* foram escritas segundo o seguinte formato:

Enquanto <ator do sistema> (Quem?)
Quero <ação a executar> (O quê?)
De forma a <proposta de valor> (Porquê?)

A definição da prioridade dos requisitos é um aspeto bastante importante a definir, tendo para isso sido utilizadas as palavras-chave, definidas no método MoSCoW[15][16], "*Must*", "*Should*", "*Could*" e "*Won't*", como forma de criar uma priorização sobre as funcionalidades do sistema a desenvolver.

Nesta secção poderá ver-se a listagem dos requisitos e respetiva prioridade, sendo que a descrição dos mesmos encontra-se no Anexo A.

Tabela 4.1: Requisitos relacionados com contas de utilizador

ID	Resumo	Prioridade
ACCOUNT-01	Autenticar na aplicação com as suas credenciais	MUST
ACCOUNT-02	Como utilizador sem registo, poder criar uma conta	MUST
ACCOUNT-03	Autenticação com o Facebook	MUST
ACCOUNT-04	Fazer <i>logout</i> da aplicação	MUST

Tabela 4.2: Requisitos relacionados com registo de viagens

ID	Resumo	Prioridade
TRIP-01	Começar registo viagem	MUST
TRIP-02	Receber <i>live feedback</i> durante a viagem	MUST
TRIP-03	Pausa/resumo da viagem	MUST
TRIP-04	Parar registo viagem	MUST
TRIP-05	Visualizar informação de economia e segurança durante a viagem	MUST
TRIP-06	Visualizar informação de velocidade, altitude, duração e distância durante a viagem	MUST

ID	Resumo	Prioridade
TRIP-07	Partilhar o estado atual da viagem	MUST
TRIP-08	Adicionar imagens à viagem	MUST
TRIP-09	Adicionar vídeos à viagem	COULD

Tabela 4.3: Requisitos relacionados com viagens

ID	Resumo	Prioridade
TRIPS-01	Ver lista de viagens	MUST
TRIPS-02	Pesquisar viagens	MUST
TRIPS-03	Ver detalhes viagem	MUST
TRIPS-04	Editar informação da viagem	MUST
TRIPS-05	Apagar viagem	MUST
TRIPS-06	Agrupar viagens	COULD
TRIPS-07	Adicionar/remover imagens às viagens	MUST
TRIPS-08	Adicionar/remover vídeos às viagens	COULD

Tabela 4.4: Requisitos relacionados com veículos

ID	Resumo	Prioridade
CARS-01	Ver lista de veículos	MUST

ID	Resumo	Prioridade
CARS-02	Pesquisar veículos	MUST
CARS-03	Ver detalhes veículo	MUST
CARS-04	Editar informação do veículo	MUST
CARS-05	Apagar veículo	MUST
CARS-06	Adicionar/remover imagem ao veículo	MUST
CARS-07	Escolher como veículo <i>default</i>	MUST

Tabela 4.5: Requisitos sociais

ID	Resumo	Prioridade
SOCIAL-01	Ver lista atividades dos amigos	MUST
SOCIAL-02	Adicionar/remover amigos	MUST
SOCIAL-03	Comentar atividades dos amigos	MUST
SOCIAL-04	Ver perfil de um utilizador	MUST

Tabela 4.6: Requisitos relacionados com grupos

ID	Resumo	Prioridade
GROUPS-01	Criar grupos	SHOULD
GROUPS-02	Adicionar utilizadores a grupos	SHOULD

ID	Resumo	Prioridade
GROUPS-03	Apagar grupos	SHOULD

Tabela 4.7: Requisitos relacionados com configurações

ID	Resumo	Prioridade
SETTINGS-01	Definir veículo <i>default</i>	MUST
SETTINGS-02	Definir unidade de distância	MUST
SETTINGS-03	Definir formato horário	SHOULD
SETTINGS-04	Definir configurações de partilha	MUST
SETTINGS-05	Ver os <i>Terms and Conditions</i> da aplicação	MUST
SETTINGS-06	Classificar a aplicação	COULD

4.1.4 Requisitos não-funcionais

A seguinte secção apresenta o conjunto de requisitos não-funcionais necessários à aplicação. Consideram-se requisitos não-funcionais, todos os que se referem a qualidades ou atributos do sistema, contrastando com os requisitos funcionais que especificam comportamentos específicos do mesmo. A especificação dos requisitos não-funcionais foi feita sem qualquer restrição na forma, tendo estes sido descritos livremente.

Tabela 4.8: Requisitos de sistema

ID	Resumo	Prioridade
SYSTEM-01	Garantir recuperação do sistema em caso de falha	MUST
SYSTEM-02	A aplicação deverá funcionar no sistema operativo iOS, versão superior a 7.0 inclusive	MUST
SYSTEM-05	Obter a morado de início e fim de viagem.	MUST
SYSTEM-06	Garantir a gestão da energia consumida, minimizando o seu impacto	SHOULD
SYSTEM-07	Minimizar tráfego de dados	MUST
SYSTEM-08	Garantir a integridade de toda a informação gerada	MUST
SYSTEM-08	Garantir o aspeto nativo da aplicação	MUST
SYSTEM-08	Garantir o funcionamento da aplicação quando não há dados móveis	MUST

Tabela 4.9: Requisitos de segurança

ID	Resumo	Prioridade
SEC-01	Transmissão de dados através de ligações seguras	MUST
SEC-02	Obrigatoriedade de autenticação	MUST

ID	Resumo	Prioridade
SEC-03	Não permitir ver informação privada de outros utilizadores	MUST

Capítulo 5

Especificação da Arquitetura

Neste capítulo será apresentada uma especificação da arquitetura do sistema que constitui a aplicação e os seus cenários de validação. A especificação basear-se-á na lista de requisitos apresentados anteriormente. A definição da arquitetura aparece numa fase de transição entre o levantamento de requisitos do sistema e a implementação do mesmo.

A definição do desenho de software é uma etapa de extrema importância para o posterior processo de desenvolvimento, uma vez que uma boa definição da arquitetura permite o aumento da flexibilidade e portabilidade do software a desenvolver.

5.1 Descrição Geral da Arquitetura

A constituição da solução completa a desenvolver não consiste unicamente na aplicação móvel, como descrito anteriormente, engloba igualmente o *back-end* e o *front-end*. A Figura 5.1 representa a visão geral da plataforma, na qual constam todos os módulos e o respetivo fluxo de informação.

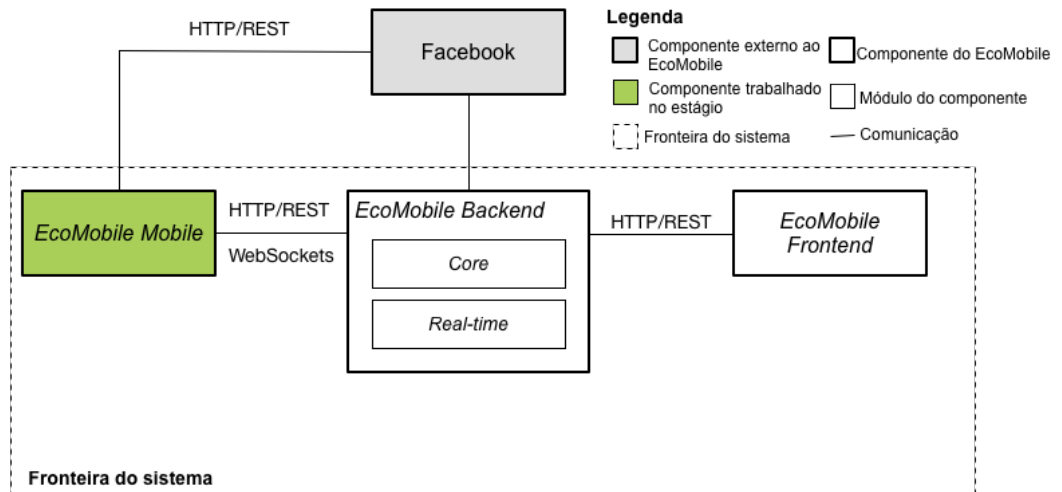


Figura 5.1: Arquitetura geral do projecto Ecomobile

Os trabalhos deste estágio englobam a parte do desenvolvimento da aplicação cliente. Todas as operações de envio e pedido de dados ao *back-end* será feita através da API REST, enquanto que toda a comunicação em tempo real e o envio de notificações pelo *back-end* para a aplicação é feita através de WebSockets¹.

EcoMobile Mobile Este componente é o que foi desenvolvido no decurso deste estágio.

EcoMobile Backend Este componente é responsável pela parte operacional e de BI de todo o sistema. O módulo *Core* é responsável pelas funcionalidades de gestão de utilizadores, veículos, viagens e rede social, enquanto que o *Real-time* gere toda a comunicação em tempo real, registo de viagens em modo *live* e notificações.

EcoMobile Frontend Este componente tem como principal função implementar a lógica aplicacional do lado do cliente.

¹<http://www.websocket.org>

Facebook É um componente externo ao sistema do EcoMobile responsável pela integração e partilha de conteúdos na rede social. Neste estágio é utilizado este componente pela aplicação móvel da partilha de conteúdos.

5.2 Decisões Arquiteturais

No início da definição da arquitetura do sistema a desenvolver e com o objetivo da criação de um protótipo funcional, como prova de conceito das funcionalidades a que a aplicação pretende disponibilizar, foram estudadas *frameworks* de desenvolvimento móvel, que utilizaram tecnologias *web*, resultando na criação de aplicações com suporte para sistemas operativos móveis distintos. Este protótipo foi implementado recorrendo ao Phonegap.

Na escolha dos sistemas operativos móveis para os quais se pretende disponibilizar a aplicação, foram escolhidos os que atualmente mais se destacam no mercado e cuja utilização tende a aumentar, recaindo a escolha nos sistemas Android e iOS.

Destes dois optou-se por iniciar o desenvolvimento nativo para a plataforma iOS, o sistema operativo Android foi deixado para desenvolvimento posterior. Ambos os sistemas operativos escolhidos disponibilizam os recursos necessários para a aplicação a desenvolver.

5.3 Desenho da solução

A presente secção apresenta a solução do sistema a desenvolver, concretamente a aplicação móvel.

5.3.1 Estrutura da aplicação móvel

Como já referido anteriormente, a aplicação foi desenvolvida para o sistema operativo móvel iOS e posteriormente Android. Decidiu-se que será desenvolvida para o a versão 7 do iOS e posteriores, a escolha desta versão prende-se com o facto de englobar cerca de 89% dos dispositivos iOS atuais².

²<https://developer.apple.com/support/appstore/>

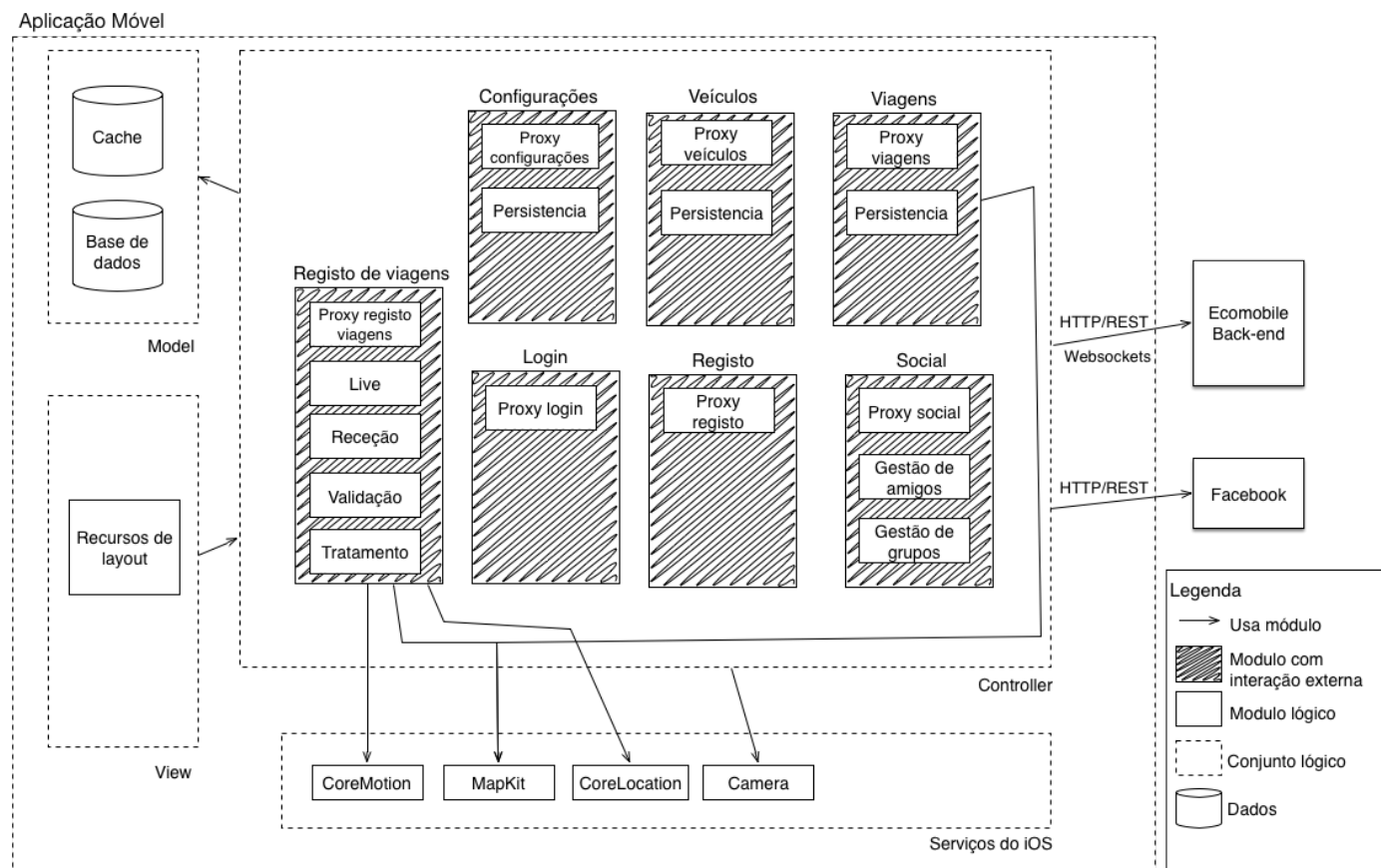


Figura 5.2: Arquitetura aplicação móvel

Na Figura 5.2 pode observar-se a vista estática da arquitetura da aplicação móvel, enquadrada no modelo MVC, em que são indicados os principais módulos da aplicação e os serviços do iOS e também externos que são utilizados, esta visualização é simples de entender e permite uma explicação detalhada do desenho da aplicação.

Existe uma primeira camada que se refere aos dados que a aplicação utiliza(*model*), que representa todos os dados persistentes da aplicação. É composta pela base de dados utilizada pela aplicação e também a informação que fica em *Cache*, neste caso maioritariamente imagens. A base de dados implementada recorre ao CoreData³ fornecido pelo SDK em que é utilizado o modelo de dados presente no Anexo F.

O sistema de *cache* gere as imagens, viagens, lista de viagens e *feed* social que são de utilização recorrente pela aplicação, de forma a evitar pedidos desnecessários ao *back-end* e também permitir que esta informação seja apresentada sempre que a aplicação não tem rede de dados.

A camada lógica(*Controller*), tal como o nome indica, é responsável por toda a lógica aplicacional e é aqui que são apresentados os módulos principais da aplicação. Mais à frente serão explicados em pormenor em que consiste cada um destes módulos.

Por último, a camada responsável pela apresentação da interface ao utilizador(*view*), que está representada de uma forma simplista uma vez que esta apenas capta as interações do utilizador com a *interface* refletindo posteriormente as alterações que estas ações provocam na mesma.

A comunicação com os serviços externos, *back-end* do EcoMobile e Facebook, é feita através de uma *API REST*. A comunicação em tempo real e notificações é conseguida através do recurso a *websockets*, utilizando para uma uma implementação em Objective-c do Socket.io⁴.

5.3.2 Módulos

Como já havia sido referido anteriormente, a arquitetura da aplicação passou pelo sua divisão em módulos, Figura 5.2, pelo que cada um realizará uma

³https://developer.apple.com/library/mac/documentation/cocoa/reference/CoreData_ObjC/_index.html

⁴<http://socket.io>

função específica descrita nos requisitos. De seguida serão explicados cada um dos módulos desenvolvidos.

Login

Este módulo permite a autenticação de um utilizador já existente na aplicação. Este módulo garante também que o utilizador permanece com a sua sessão iniciada na aplicação até que o mesmo faça *logout*, evitando assim que seja necessário introduzir as suas credenciais de cada vez que inicia a aplicação.

Registo

Este módulo permite o registo de um novo utilizador na aplicação. No registo são pedidas todas as informações necessárias sobre o utilizador, *e-mail*, nome e *password*, garantindo que apenas existe um utilizador registado com o mesmo *e-mail*. Depois de o registo ser efetuado com sucesso o utilizador é notificado do mesmo através da receção de um *e-mail* na caixa de correio eletrónico fornecida, no qual está presente um *link* para efetuar a validação do registo. Só depois desta confirmação é que a conta fica ativa de forma a se conseguir autenticar na aplicação.

Veículos

Este módulo é responsável por toda a gestão relacionada com veículos, registo, remoção e edição de veículos na aplicação. Permite igualmente definir qual é o veículo que vai ser utilizado por defeito nas viagens.

Viagens

Depois de registadas viagens na aplicação diversas operações são possíveis de realizar sobre as mesmas. Este módulo é responsável por toda a gestão relacionada com viagens, que começa pela visualização e edição da sua informação, gestão de fotos da viagem até a eliminação da viagem. Este módulo é também responsável pelas interações sociais sobre cada viagem, comentários as viagem e suas fotos e partilha da viagem no Facebook.

Configurações

Este módulo engloba as configurações específicas do utilizador, sendo responsável pela gestão/alteração das mesmas.

Social

Este módulo é responsável por toda a interação social da aplicação, passando pela partilha e visualização de todas as atividades sociais, gestão de grupos e amigos.

Registo Viagem

A base da aplicação construída assenta na registo de viagens dos condutores, sendo este o módulo responsável por esse registo. Dentro deste módulo encontra-se presente toda a lógica para o registo de viagem em tempo real e para a receção, validação e tratamento dos dados sensoriais e de localização.

A caracterização do estilo de condução e o cálculo dos índices de segurança e economia são efetuadas por este módulo ao longo de todo o processo de registo de uma viagem. Estas métricas são obtidas através de conjugação de vários fatores que são analisados com base na informação sensorial e de localização recolhida, sendo estes os que se seguem.

- Velocidade instantânea.
- Aceleração.
- Desaceleração.
- Aceleração lateral.
- Descidas.
- Subidas.
- Duração da viagem.
- Velocidade em relação com as caracterização da estrada.

Todos os fatores em cima mencionados tem um impacto positivo ou negativo na segurança e economia da condução dos utilizadores, pelo que são estes que são usados na elaboração das formulas usadas para calcular a segurança economia da condução, e também identificar os eventos prejudiciais à condução, acelerações bruscas, travagens bruscas e curvas agressivas.

Capítulo 6

Trabalho realizado

Neste capítulo é apresentado o trabalho efetuado ao longo do estágio. São apresentados detalhes sobre o código desenvolvido, assim como os principais problemas encontrados durante a fase de desenvolvimento e respetivas soluções adotadas quando possível.

6.1 Protótipo Phonegap

Nesta secção refere-se à demonstração dos resultados obtidos com a implementação do protótipo em Phonegap. Os protótipos de baixa fidelidade que conduziram ao resultado final do protótipo, assim como os ecrãs que representam os resultados obtidos encontram-se no Anexo B e C respetivamente.

Como descrito anteriormente, este protótipo deve como objetivo verificar se e possível alcançar aquilo que a aplicação pretende através da informação sensorial e de localização, para os sistemas operativos Android e iOS. No fim da implementação do protótipo verificou-se que este era compatível com os dois sistemas operativos em causa e que era capaz de recolher a informação sensorial e de localização pretendida.

Mesmo tendo-se conseguido chegar a resultados satisfatórios com o Phonegap, ao longo da sua implementação foram encontrados vários obstáculos. O facto de usar de o iOS e o Android usarem implementações diferentes de *webviews* levou a que a aplicação não tive-se comportamentos e aspeto diferente nestes dois sistemas operativos, sendo difícil uniformizar a aplicação entre sistemas operativos diferentes. O desempenho do protótipo não se mostrou o mais satisfatório, com transições entre ecrãs lentas e falta de fluidez, fazendo transparecer que não era uma aplicação nativa. No que toca aos

serviços que precisamos foi preciso implementar *plugins* para aceder a alguns serviços, como o caso do CoreMotion.

Em conclusão, o protótipo deu para provar o conceito da aplicação para os diversos sistemas operativos, mas não representa uma solução definitiva, muito por causa dos problemas identificados em cima.

6.2 Aplicação iOS Nativa

São apresentados nesta secção todo o processo de desenvolvimento da aplicação nativa iOS.

6.2.1 Bibliotecas externas usadas

Para a implementação de algumas componentes da aplicação, recorreu-se a utilização de algumas bibliotecas externas. Foi utilizado o CocoaPods¹ como gestor de dependências do projeto. Segue uma descrição de cada um dos recursos externos utilizados.

CorePlot²: O CorePlot é uma ferramenta de criação de gráficos para o OS X e iOS. Proporciona a visualização 2D de dados, é altamente customizável e capaz de desenhar vários tipos de gráficos. Encontrasse completamente integrado com as tecnologias Apple Core Animation, Core Data, and Cocoa Bindings. Esta biblioteca é usada no desenho de todos os gráficos presentes na aplicação.

TMCache³: A TMCache é *key/value store* desenhada para a persistência de objetos temporários que são dispendiosos de reproduzir, tais como *downloads* de informação. Esta biblioteca é usada na gestão de todos os objetos que ficam em *cache* na aplicação.

FXBlurView⁴: O FXBlurView é uma sub-classe do UIView na qual está presente o efeito de *blur* no seu fundo. É usada nos *popups* de alerta de eventos que ocorrem durante o registo de viagens.

¹<http://cocoapods.org>

²<https://github.com/core-plot/core-plot>

³<https://github.com/tumblr/TMCache>

⁴<https://github.com/nicklockwood/FXBlurView>

TSMessages⁵: Esta biblioteca permite mostrar pequenas notificações no topo de visor sem perturbar o utilizador. A TSMessages são usados em todo o tipo de notificações presentes na aplicação..

Socket.IO⁶: Esta biblioteca representa uma *Interface* de comunicação entre o Objective C e o Socket.IO. É usada na comunicação em tempo real existente na aplicação.

CocoaLumberjack⁷: O CocoaLumberjack é uma ferramenta de *logging* para o OSX e o iOS. Esta biblioteca foi utilizada na fase de testes dos algoritmos da aplicação, na criação de ficheiros com os dados recolhidos pela aplicação para posterior análise.

6.2.2 Tratamento dados sensoriais e de localização

Um dos componentes de maior importância no desenvolvimento desta aplicação consiste na recolha dos dados sensoriais e de localização por parte dos dispositivos. É necessário garantir que os dados obtidos por estes se enquadram nos parâmetros aceitáveis pela aplicação. Este módulo funciona de acordo com fluxo de atividades fixo, constituído pelas seguintes fases.

- Receção da informação sensorial e de localização.
- Validação da informação.
- Tratamento da informação.

Receção

A fase de receção dos dados consiste na ativação dos serviços do iOS que fornecem a informação sensorial e de localização, sendo eles respetivamente o CoreMotion⁸ e CoreLocation⁹. Ao iniciar estes serviços a aplicação passa a ser notificada de todas as alterações produzidas na informação que estes

⁵<https://github.com/toursprung/TSMessages>

⁶<https://github.com/pkyeck/socket.IO-objc>

⁷<https://github.com/CocoaLumberjack/CocoaLumberjack>

⁸https://developer.apple.com/library/ios/documentation/coremotion/reference/cmmotionmanager_class/Reference/Reference.html

⁹https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation_Framework/_index.html

monitorizam.

O CoreMotion é responsável por recepção da informação sensorial, ao qual se define a frequência de amostragem pretendida e também o eixo a ser usado como referência. Para chegar ao valor da frequência de amostragem teve-se em consideração fatores como o consumo de bateria e tipo de eventos a detetar. Como se pretende maximizar a duração da bateria e o tipo de eventos a detetar caracterizam-se por serem contínuos. De acordo com a experimentação efetuada este intervalo ficou definido como 0.25 segundos, sendo o melhor compromisso encontrado entre a fiabilidade da monitorização e a otimização do consumo da bateria. O eixo de referência que ficou definido para a recolha de dados sensoriais foi o *True north*¹⁰(direção tomada à superfície da Terra que aponta para o Polo Norte geográfico). Utilizou-se este eixo como referência, uma vez que é o mesmo que é utilizado pelo GPS, uniformizando desta forma os dados recolhidos entre ambos.

No caso de Corelocation, que recolhe toda a informação de localização, a única opção que temos aquando da sua inicialização é definir um valor de referência a ser usado pelo sistema operativo para decidir quando envia uma nova notificação de localização, isto é, quando a posição variar aproximadamente o valor o referência o sistema gera uma nova notificação. Não sendo o objetivo da aplicação focar-se na navegação altamente precisa, optou-se por um valor melhor representa o compromisso entre fiabilidade da informação o otimização do consumo da bateria. De acordo com a experimentação efetuada e com a experiência adquirida ao longo de toda a fase de teste, este valor ficou definido como sendo 10 metros.

No Anexo G encontram-se descrito todos os sensores que são ativados quando se ligam estes serviços, valores que produzem e o seu significado.

Validação

Um dos primeiros obstáculos encontrados na fase de recepção da informação sensorial e de localização, foi o ruído que pode vir com a mesma assim como a falta de precisão. Posto este facto todos os dados recebidos passam então por uma fase de validação onde se tenta despistar os valores incorretos e uniformizar a informação. Na caso da informação sensorial surgem em destaques os seguintes factos que influenciam negativamente os dados.

¹⁰http://en.wikipedia.org/wiki/True_north

- Valores residuais.
- Aparecimento de valores discrepantes, picos.
- Inconsistência dos valores.
- Movimentação do dispositivo.

De forma a minimizar o impacto destes fatores, decidiu-se pela aplicação de um filtro de passa-baixa¹¹ sobre os valores provenientes dos sensores, de forma a uniformizar os mesmos. Um filtro deste tipo tem como objetivo atenuar o efeito dos valores de grande amplitude em comparação com a amostragem normal.

Na validação dos valores provenientes do GPS, foi definido que apenas são consideradas localizações com precisão menor que 40 metros, todos os outros valores são descartados. Passando este primeiro filtro sobre a informação de localização, a mesma pode ser utilizada para atualizar a UI com a informação referente à altitude, velocidade e distância percorrida.

No caso da informação de localização que é utilizada na deteção dos eventos e cálculo dos índices, o filtro aplicado é mais restritivo, só aceitando valores com precisão menor que 15 metros. A filtragem da informação dos dados para esta fase tem como objetivo considerar a informação que tem garantias altas de ser correta, evitando assim efetuar cálculos sobre índices e eventos com valores incorretos, evitando desta forma produzir índices inconsistentes e falsos positivos/negativos na deteção de eventos.

Tratamento

Depois da validação da informação sensorial e de localização, já nos encontramos em condições de obter os índices de segurança e economia e detetar também os eventos pretendidos, travagens, acelerações e curvas bruscas. O primeiro passo no tratamento dos dados é obter a aceleração usando como referência a direção do movimento, nas suas componentes X,Y,Z, Figura 6.1, em que o Y representa a direção do movimento.

¹¹http://en.wikipedia.org/wiki/Low-pass_filter

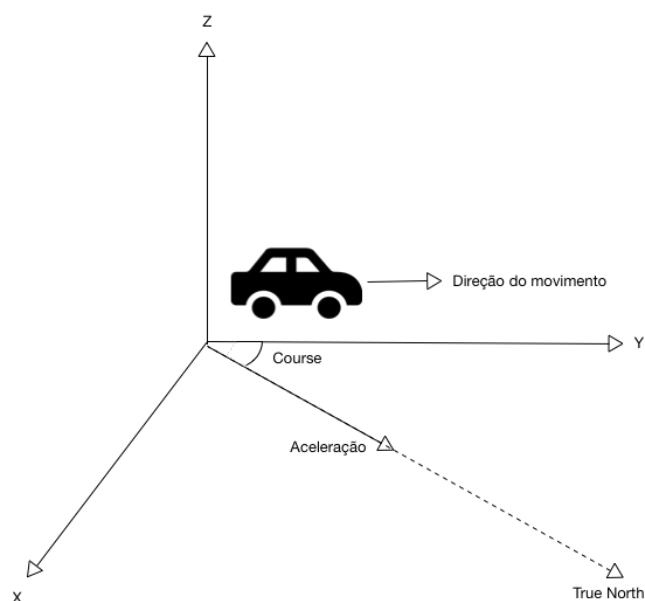


Figura 6.1: Eixo direção do movimento

Uma vez que temos a aceleração do dispositivo e o Course proveniente do GPS a utilizar o mesmo eixo de referência, *True North*, conseguimos obter a aceleração em conformidade com a direção do movimento.

Neste ponto a aplicação já tem todos os dados que precisa devidamente validados e convenientemente tratados, pelo que podemos passar para detecção de eventos e o cálculo dos índices. Na detecção de eventos e cálculo dos índices são sempre considerados valores referentes ao histórico dos últimos 3 segundos. Utilizamos este período histórico uma vez que existe um atraso nos valores do GPS em comparação com os valores produzidos pelos sensores de movimento. Enquanto que estes produzem os valores que acontecem no instante, o GPS apresenta um *delay* que pode chegar a vários segundos.

A aplicação desenvolvida não requer que o dispositivo móvel se mantenha estático para o seu correto funcionamento. Não incluindo esta imposição fica sujeita ao ruído introduzido nos sensores quando o dispositivo se movimenta,

sem ser por ação do movimento do veículo. O giroscópio permite saber todas as rotações efetuadas sobre o eixo de referência usado. Olhando para o histórico destas rotações, são eliminados todos os valores produzidos pelos sensores de movimento que apresentem rotação no eixo de referência. Aplicando esta estratégia torna-se praticamente residual o ruído introduzido pelas movimentações indesejadas do dispositivo.

6.2.3 Detecção de eventos

A detecção de eventos tem um papel importante para a aplicação, uma vez que estes são apresentados ao utilizador, em tempo real, durante as viagens com informação visual e sonora, Figura 6.2.

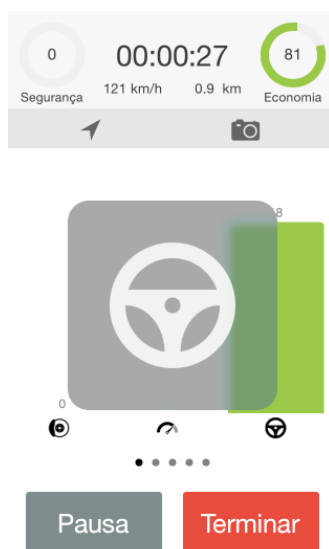


Figura 6.2: Evento de curva brusca

A todo o tipo de eventos são aplicados alguns critérios gerais. Não são considerados eventos se a velocidade for menor que 15km/h, com isto evitamos produzir falsos positivos quando o condutor se encontra em manobras ou a acabar de arrancar/parar.

Como forma de evitar que a aplicação produza falsos positivos/negativos nos eventos, é usada uma dupla verificação para validar os eventos. São sempre consideradas as acelerações indicadas pelos sensores de movimento do dispositivo ao mesmo tempo que se confirma com a informação de GPS. O GPS não indica a aceleração do veículo de forma direta, mas é possível obter a mesma a partir da informação fornecida pelo mesmo. A partir das

sucessivas velocidade e intervalos de tempo entre elas consegue-se extrapolar o valor da aceleração do movimento, através das equações do movimento.

A partir de toda a experiência adquirida ao longo do estágio, primordialmente na fase de testes, chegou-se ao seguinte algoritmo para a detecção de eventos dos vários eventos pretendidos.

Algorithm 1 Algoritmo de avaliação de eventos

GPSAccel = Aceleração obtida pelo GPS
Ax = Valor máximo da aceleração lateral
AxThreshold = Valor mínimo de aceleração lateral que origina evento
Ay = Valor máximo de aceleração na direção do movimento
AyAccel = Aceleração mínima direção movimento origina aceleração brusca
AyBrake = Aceleração mínima direção movimento origina travagem
MaxCourse = Valor máximo de variação do *course*
SpeedThreshold = Valor mínimo de velocidade para produzir eventos
CornerThreshold = Valor mínimo de variação do *course* para se considera curva

```

if (Instant Speed > SpeedThreshold ) then

    if (MaxCourse > CornerThreshold) then
        if (Ax > AxThreshold && Ax > Ay) then
            return Evento de curva brusca
        end if
    end if

    if (MaxCourse < CornerThreshold) then
        if (Ay > AyAccel && Ay > Ax && GPSAccel > AyAccel) then
            return Evento de aceleração brusca
        end if

        if (Ay < AyBrake && Ay > Ax && GPSAccel > AyBrake) then
            return Evento de travagem brusca
        end if
    end if
end if

```

Os valores presentes no algoritmo, limites máximos e mínimos, foram obtidos ao longo da fase de testes e da aplicação, sendo os que traduzem os

melhores resultados obtidos nessa mesma fase. Na fase de testes foi ontem se detetaram todos os problemas já enunciados ao longo deste capítulo, sendo o algoritmo apresentado o que melhor retrata a condução dos utilizadores. A definição de bastantes parâmetros no algoritmo permite ajustar a sensibilidade de mesmo de forma rápida, sem ser preciso mudar a sua lógica.

Segue nas próximas secções uma explicação pormenorizada sobre o deteção de cada evento em particular.

Curvas bruscas

As curvas bruscas são caracterizadas pela mudança de direção do movimento a velocidades excessivas. Na deteção destas acelerações são considerados os seguintes fatores:

- Valor da componente na aceleração lateral a direção do movimento.
- Variação da direção.
- Velocidade.

Analisando o algoritmo apresentado, na deteção das curvas bruscas verifica-se se o *course*, dado pelo GPS, variou o suficiente para se considerar que estamos numa curva. Se a condição anterior for verdade, analisa-se se a aceleração lateral é maior que a aceleração na direção do movimento e se é maior também que o limite estabelecido para validar uma curva. Se ambas as condições forem verdadeiras considera-se que houve uma curva brusca.

Acelerações bruscas

As acelerações bruscas são caracterizadas pelo excessivo aumento de velocidade num curto intervalo de tempo. Na deteção destas acelerações são considerados os seguintes fatores:

- Valor da componente na aceleração sobre a direção do movimento.
- Variação da velocidade.
- Variação da direção.

Analisando o algoritmo apresentado, na deteção das acelerações bruscas, em primeiro lugar, verifica-se que a direção do movimento não variou o suficiente para ser considerado uma curva, de seguida verifica-se se há aceleração positiva na direção do movimento que seja maior que um limite estabelecido, o mesmo é feito para a variação da velocidade dada pelo GPS, que também tem que ultrapassar um limite previamente estabelecido. Se ambas as condições forem verdadeiras considera-se que houve uma aceleração brusca.

Foram encontrados alguns problemas na deteção das acelerações bruscas que dificultaram bastante a correta interpretação dos mesmos. O principal obstáculo foi a o facto de a velocidades elevadas o sensor de movimento do dispositivo não consegue detetar acelerações relevantes. Para ultrapassar esta limitação para este tipo de eventos em particular, deu-se maior relevância aos dados provenientes do GPS em comparação com os dados sensoriais.

Travagens bruscas

As travagens bruscas são caracterizadas pela repentina perda de velocidade num curto intervalo de tempo. Na deteção destas travagens são considerados os seguintes fatores:

- Valor da componente na aceleração sobre a direção do movimento.
- Variação da velocidade.
- Variação da direção.
- Velocidade.

O funcionamento do algoritmo na deteção de travagens é idêntico ao aplicado nas acelerações bruscas, mas neste caso são utilizados verificados valores negativos de aceleração na direção do movimento. Ao contrario das acelerações bruscas, nas travagens não existe o problema de terem pouco impacto nos sensores de movimento, produzem acelerações relevantes, pelo que o peso que é dado ao GPS e ao sensor de movimento é igual.

6.2.4 Cálculo de índices

O cálculo dos índices de segurança e economia caracterizam uma viagem pelo que a elaboração de índices que traduzam com rigor o estilo de condução dos utilizados é uma das tarefas principais deste modulo. Com este objetivo em mente chegaram-se as seguintes conclusões sobre estes mesmos índices.

Índice de segurança

De todos os fatores que afetam negativamente a condução identificam-se os seguintes que influencia diretamente a segurança dos condutores.

- Velocidade instantânea.
- Aceleração.
- Aceleração lateral.
- Descidas.
- Duração da viagem.
- Velocidade em relação com a caracterização da estrada.

Na definição inicial do cálculo da segurança começou-se por encontrar uma curva que traduzi-se em conformidade a segurança em relação com a velocidade, de forma a depois usar os diferentes fatores identificados em cima, para penalizar ainda mais a segurança consoante e velocidade. Foi utilizada a curva presente na Figura 6.3.

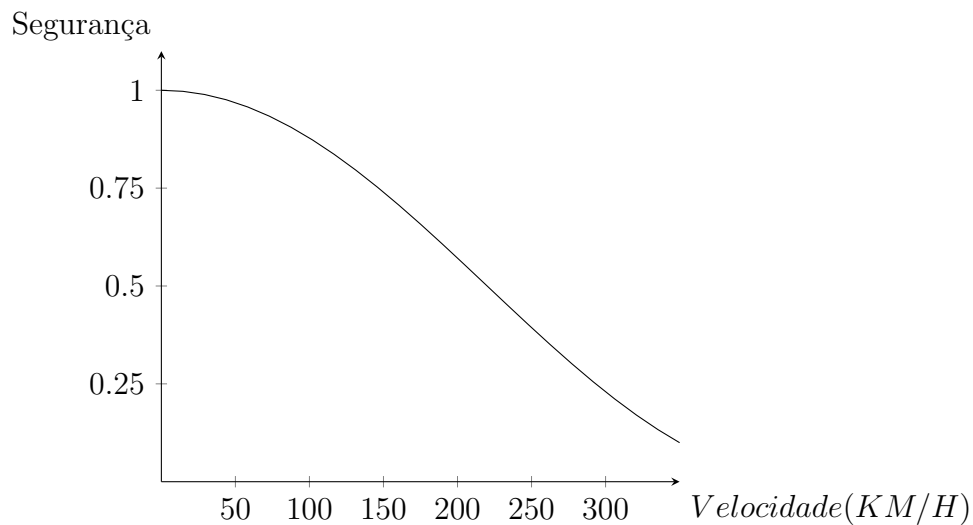


Figura 6.3: Gráfico índice de segurança

A curva apresentada em cima é representada pela seguinte fórmula.

$$IS = 0.5 * \sin\left(\frac{Velocidade * Inclinação}{140 * Curvas} + \frac{\pi}{2}\right) + (0.5 - Tempo - AccelX - AccelY)$$

Todas as variáveis presentes na fórmula tem como objetivo penalizar a segurança, fazendo acentuar o declive da curva.

Velocidade

Representa a velocidade instantânea obtida através do GPS.

Inclinação

Na inclinação, são consideradas apenas descidas, decidimos não penalizar nem beneficiar a segurança durante as subidas. Durante as descidas quando maior for o seu declive, maior o impacto negativo na segurança. O valor da inclinação é calculado a partir das sucessivas diferenças de altitude produzidas pelo GPS, durante o intervalo considerado.

Tempo

Quando maior a duração de uma viagem maior o desgaste e cansaço evidenciado pelos condutores. Considerando este facto, também a duração das viagens implica acentuar o declive da curva de segurança. Considera-se que todo o tempo, a partir das 2 horas de viagem, influencia negativamente a segurança da viagem.

AccelX

De forma a penalizar a segurança quando o condutor efetua curvas, considera-se que quanto maior for a aceleração lateral, maior o impacto negativo desta na segurança. O mesmo acontece para as acelerações na direção do movimento.

AccelY

De forma a penalizar a segurança quando o condutor acelerações ou travagem bruscas, considera-se que quanto maior for a sua aceleração na direção do movimento, maior o impacto negativo desta na segurança.

Curvas

Um dos principais obstáculos no cálculo da segurança, foi produzir valores que fizessem sentido independentemente do local onde o condutor se encontre. Exemplificando, conduzir a 90km/h em ambiente citadino ou à mesma velocidade mas numa auto-estrada, a segurança não pode ser igual. Para ultrapassar este obstáculo, a aplicação faz uma caracterização da estrada em que se encontra. Esta caracterização é feita contanto para cada nova posição de GPS, o número de curvas realizadas no último minuto. Considerou-se uma curva sempre que o *course*, varia mais de 15° entre duas posições de GPS consecutivas.

Na Figura 6.4 pode-se verificar o efeito produzido no gráfico de índice de segurança quando o fator curvas é elevado e está presente aceleração lateral.

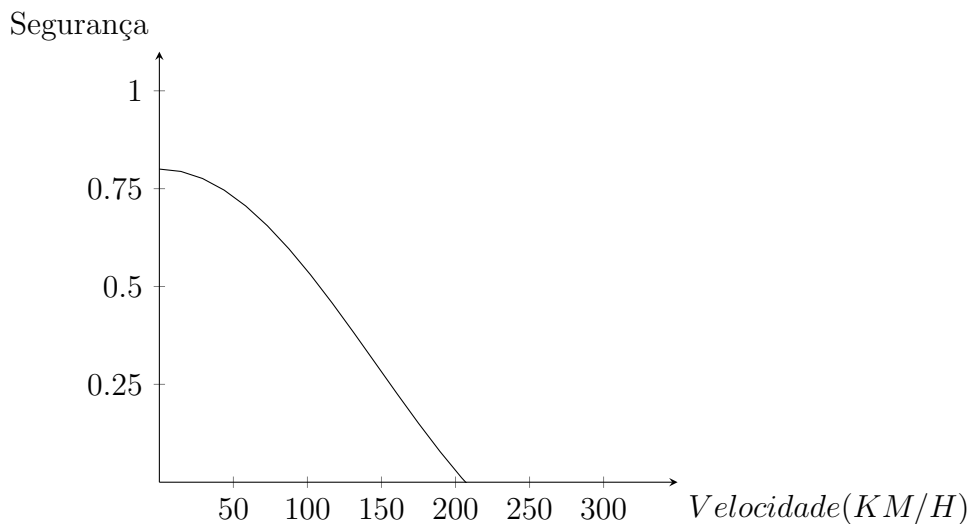


Figura 6.4: Exemplo gráfico índice de segurança

Índice de economia

De todos os fatores que afetam negativamente a condução identificam-se os seguintes que influencia diretamente a economia dos condutores.

- Velocidade instantânea.
- Aceleração.
- Desaceleração.

- Descidas.
- Subidas.

Em conformidade com aquilo que se fez para o índice de segurança, para a economia começou-se também por encontrar uma curva que traduzi-se em conformidade a economia em relação com a velocidade, de forma a depois usar os diferentes fatores identificados em cima, para a penalizar ainda mais consoante e velocidade. Foi utilizada a curva presente na Figura 6.5, baseada nos resultados do estudo presente em [17].

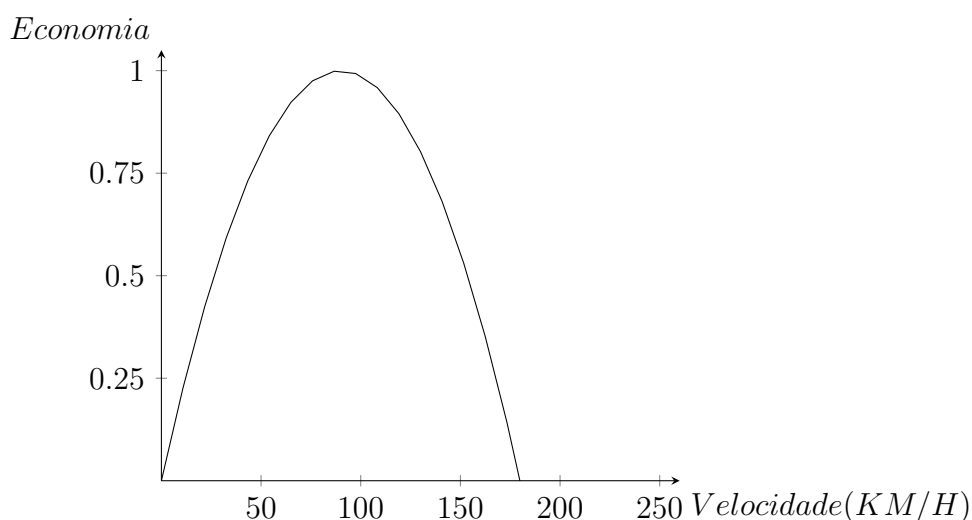


Figura 6.5: Gráfico índice de economia

A curva apresentada em cima é representada pela seguinte fórmula.

$$IE = \frac{-(Velocidade - Eco^2) + Eco^2}{((Eco + Eco * (Inclinao + AccelY))^2)}$$

Velocidade

Representa a velocidade instantânea obtida através do GPS.

Eco

Como indicado em [18], a velocidade ideal de consumo para a generalidade dos veículos automóveis ronda os 90km/h, sendo este valor o utilizado na fórmula para representar o máximo de economia. Ao contrário do que é indicado pelo senso comum, andar devagar não significa poupar combustível,

pelo contrario, andar muito devagar provoca elevados gastos de combustível. A curva apresenta em cima tenta contemplar da melhor forma estes fatores.

Inclinação

Na fórmula as descidas e subidas, inclinação, tem um impacto contrario na economia. As descidas contribuem para o aumento da economia enquanto que as subidas penalizam drasticamente a economia. Não é utilizado o mesmo peso para ambas, sendo que as decidas tem um impacto positivo na economia menor que o impacto negativo das subidas.

AccelY

Igualmente ao considerado para as subidas/descidas, o mesmo principio e aplicado nas perdas/ganhos de velocidade. O ganho de velocidade tem um impacto fortemente negativo na economia, enquanto que as perdas de velocidade favorecem a economia mas com um impacto mais reduzido. Pretende-se simular aqui as perdas de velocidade devido ao condutor aliviar a sua aceleração(carregar menos no acelerador), mas não completamente(retirar o pé do acelerador.)

Na Figura 6.6 pode-se verificar o efeito produzido no gráfico de índice de economia quando a viagem se encontra numa subida e a aceleração na direção do movimento é elevada.

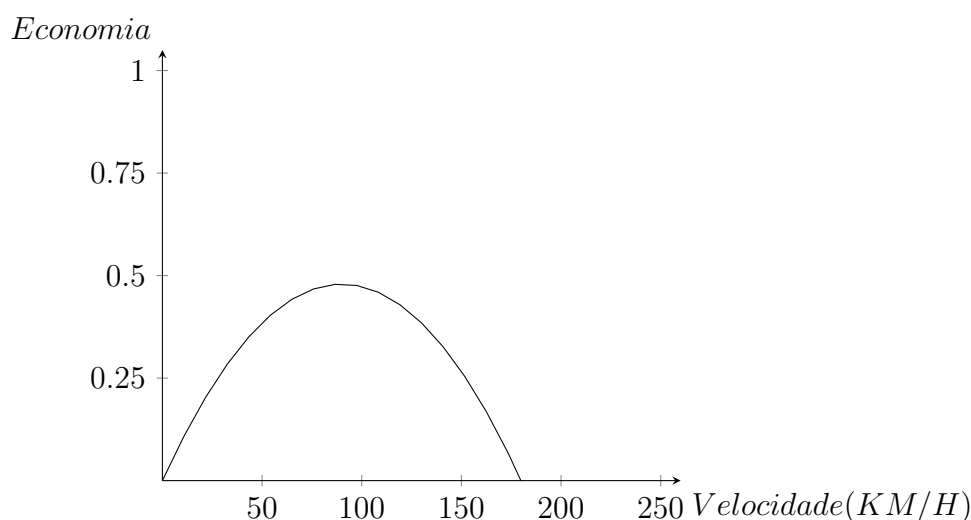


Figura 6.6: Exemplo gráfico índice de economia

6.2.5 Resultados da aplicação iOS

Nesta secção alguns dos ecrãs produzidos para a aplicação nativa iOS, seguidas de explicação para cada um destes, e detalhes da sua implementação. Os protótipos de baixa fidelidade que conduziu a este resultado final encontram-se no Anexo C.

Login

Este módulo permite a autenticação de um utilizador já existente na aplicação, garantindo que o utilizador permanece com a sua sessão iniciada na aplicação até que o mesmo faça *logout*, evitando assim que seja necessário introduzir as suas credenciais de cada vez que inicia a aplicação. A Figura 6.7 ilustra o ecrã de *Login*, mostrando a informação que é necessária introduzir pelo utilizador para executar esta ação. Quando a autenticação é realizada com sucesso, o servidor envia para a aplicação um *token* e um *refresh token* que identificam o utilizador, provenientes do mecanismo de autenticação OAUTH2.0¹² utilizado no *back-end*. De forma a manter o utilizador autenticado até que este faça *Logout* da aplicação, os *tokens* são guardados pela aplicação recorrendo ao `NSUserDefaults`¹³ disponibilizado pelo SDK do iOS.

¹²<http://tools.ietf.org/html/rfc6750>

¹³https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/nsuserdefaults_class/Reference/Reference.html

Registo

Este módulo permite o registo de um novo utilizador na aplicação. No registo são pedidas todas as informações necessárias sobre o utilizador, *e-mail*, nome e *password*, garantindo que apenas existe um utilizador registado com o mesmo *e-mail*, Figura 6.8. Depois de o registo ser efetuado com sucesso o utilizador será notificado do mesmo através da receção de um *e-mail* na caixa de correio eletrónico fornecida, no qual esta presente um *link* para efetuar a validação do registo, só depois desta confirmação é que a conta fica ativa de forma a se conseguir autenticar na aplicação.



Figura 6.7: Ecrã *Login*

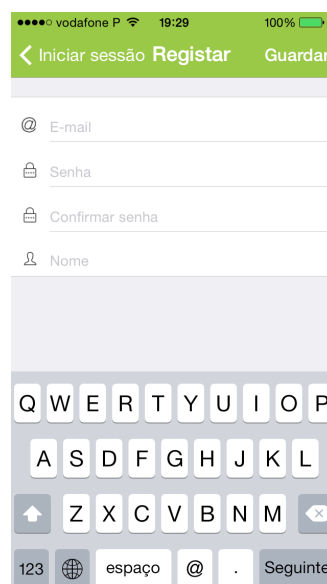


Figura 6.8: Ecrã *Registo*

Veículos

Na Figura 6.10 está presente a lista de veículos registados pelo utilizador, sendo este módulo responsável pela gestão relacionada com veículos, registo, remoção e edição de veículos na aplicação. Permite igualmente definir qual é o veículo que vai ser utilizado por defeito nas viagens.

Viagens

Na Figura 6.9 está presente a lista de viagens organizada por data, agrupando as viagens por dia, isto é as viagens que foram realizadas no mesmo dia pertencem ao mesmo grupo.

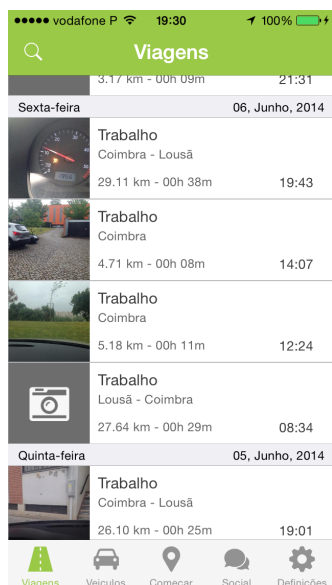


Figura 6.9: Ecrã Lista de Viagens

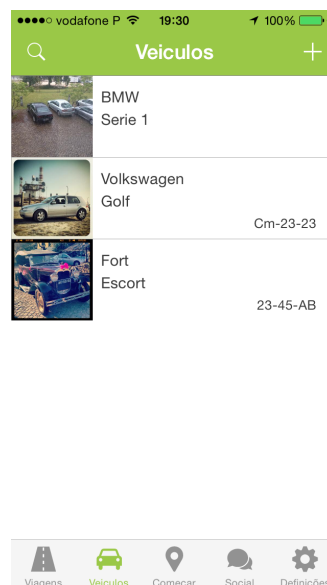


Figura 6.10: Ecrã lista de veículos

Configurações

Na Figura 6.12 encontram-se as diversas definições que o utilizador pode configurar agrupadas por grupos. Neste ecrã o utilizador pode alterar o seu perfil, configurar as opções específicas das viagens, alterar as definições de partilha e rede. A opção de *Logout* encontra-se igualmente disponível neste ecrã da aplicação. Este módulo engloba as configurações específicas do utilizador, sendo responsável pela gestão/alteração das mesmas. As configurações do utilizador ficam igualmente guardadas no `NSUserDefaults`.

Social

Este módulo é responsável por toda a interação social da aplicação, passando pela partilha e visualização de todas as atividades sociais, gestão de grupos e amigos. Na Figura 6.11 está contida toda a informação relacionada com a componente social e com grupos que a aplicação contém.



Figura 6.11: Ecrã social



Figura 6.12: Ecrã definições

Registo Viagem

A Figura 6.13 mostra a informação disponibilizada ao utilizador durante o registo de uma viagem. É apresentado um mapa que segue o percurso da viagem deixando marcado o seu trajeto, foi utilizado para este efeito os mapas disponibilizados pelo SDK, o Mapkit¹⁴. São apresentados também gráficos com a evolução da velocidade e altitude do condutor, podendo o utilizador a qualquer momento trocar entre os gráficos e o mapa, conforme aquilo que desejar visualizar. Como biblioteca para construção dos gráficos foi usado o CorePlot.

A Figura 6.14 mostra alguma da informação que é necessária introduzir no fim do registo de uma viagem para o sucesso da mesma e conseqüentemente envio para o *back-end*. Alguma da informação já vem automaticamente preenchida, como o veículo que é o escolhido por defeito pelo utilizador. Se existir uma ligação à Internet as localizações de início e fim da viagem também se encontram automaticamente preenchidas, recorrendo ao GeoCoder¹⁵ disponibilizado pelo SDK do iOS.

¹⁴https://developer.apple.com/library/ios/documentation/MapKit/Reference/MapKit_Framework_Reference/_index.html

¹⁵<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/LocationAwarenessPG/UsingGeocoders/UsingGeocoders.html>

O registo de uma viagem é a ação que mais dados na aplicação, sendo estes enviados para o *back-end*. De forma a minimizar o consumo de dados pela aplicação, estes dados são comprimidos antes de serem enviados para o *back-end*, recorrendo neste caso ao GZIP.

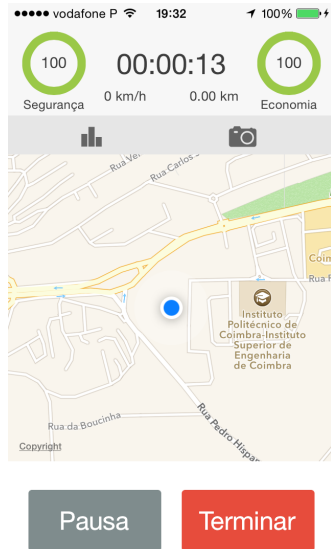


Figura 6.13: Ecrã registo de viagem com mapa



Figura 6.14: Ecrã fim de viagem

Capítulo 7

Testes

Este capítulo apresenta o plano de testes seguido tendo em vista a validação dos resultados obtidos para a aplicação móvel.

Para a realização dos testes ao trabalho desenvolvido foi necessários dispositivos Android e iOS. Foram utilizados um dispositivo iOS e dois Android fornecidos pela empresa Sentilant. As características de cada um podem ser visualizadas na Tabela 7.1.

Tabela 7.1: Especificações dos dispositivos de teste.

Model	iPhone 5 Model A1429	Sony Xperia E Dual	Samsung Galaxy S3 Mini
CPU	1.3 GHz dual core	1 GHz Cortex-A5	1 GHz dual-core Cortex-A9
GPU	PowerVR SGX543MP3	Adreno 200	Mali-400
Display	1,136x640 pixels, 4 inches	320x480 pixels, 3.5 inches	480x800 pixels, 4 inches
Memory	1 GB LPDDR2-1066 RAM	512 MB RAM	1 GB RAM
Motion Sensors	Gyroscope, Accelerometer	Accelerometer	Accelerometer, Gyroscope
GPS	GPS, GLONASS	GPS, A-GPS	GPS, A-GPS, GLONASS
Battery Capacity	1,440 mAh	1530 mAh	1500 mAh
System Version	iOS 7.04	Android OS, v4.0.4	Android OS, v4.1

Ao longo dos desenvolvimentos o projeto foi sucessivamente avaliado no seio da empresa com demos frequentes, tendo permitido uma recolha de *feedback* que possibilitou um continuo melhoramento do produto.

7.1 Testes de aceitação

Estes testes de software foram sucessivamente conduzidos ao longo dos *sprints* de desenvolvimento de modo a avaliar e validar as respetivas componentes a serem desenvolvidas, tendo-se revelado de elevada importância pelo estabelecimento de um processo que permitiu continuamente garantir o cumprimento

dos objetivos traçados com a qualidade exigida ao produto. Dessa forma, serão feitas sessões de demonstração à equipa de desenvolvimento após cada iteração, sendo validado com o *product owner*.

Tabela 7.2: Testes relacionados com contas de utilizador

Requisito	Descrição do teste	Validação
ACCOUNT-01	Autenticar na aplicação com as suas credenciais	✓
ACCOUNT-02	Como utilizador sem registo criar uma conta	✓
ACCOUNT-03	Autenticação com o Facebook	✓
ACCOUNT-04	Fazer <i>logout</i> da aplicação	✓

Tabela 7.3: Testes relacionados com registo de viagens

Requisito	Descrição do teste	Validação
TRIP-01	Começar registo viagem	✓
TRIP-02	Receber <i>live feedback</i> durante a viagem	✓
TRIP-03	Pausa/resumo da viagem	✓
TRIP-04	Parar registo viagem	✓
TRIP-05	Visualizar informação de economia e segurança durante a viagem	✓
TRIP-06	Visualizar informação de velocidade, altitude, duração e distância durante a viagem	✓

Requisito	Descrição do teste	Validação
TRIP-07	Partilhar o estado atual da viagem	✗ ¹
TRIP-08	Adicionar imagens à viagem	✓
TRIP-09	Adicionar vídeos à viagem	✗ ¹

Tabela 7.4: Testes relacionados com viagens

Requisito	Descrição do teste	Validação
TRIPS-01	Ver lista de viagens	✓
TRIPS-02	Pesquisar viagens	✓
TRIPS-03	Ver detalhes viagem	✓
TRIPS-04	Editar informação da viagem	✓
TRIPS-05	Apagar viagem	✓
TRIPS-06	Agrupar viagens	✗ ¹
TRIPS-07	Adicionar/remover imagens às viagens	✓
TRIPS-08	Adicionar/remover vídeos às viagens	✗ ¹

¹Requisito de baixa prioridade durante o decorrer do estágio.

Requisito	Descrição do teste	Validação
-----------	--------------------	-----------

Tabela 7.5: Testes relacionados com veículos

Requisito	Descrição do teste	Validação
CARS-01	Ver lista de veículos	✓
CARS-02	Pesquisar veículos	✓
CARS-03	Ver detalhes veículo	✓
CARS-04	Editar informação do veículo	✓
CARS-05	Apagar veículo	✓
CARS-06	Adicionar/remover imagem ao veículo	✓
CARS-07	Escolher como veículo <i>default</i>	✓

Tabela 7.6: Testes sociais

Requisito	Descrição do teste	Validação
SOCIAL-01	Ver lista atividades dos amigos	✓
SOCIAL-02	Adicionar/remover amigos	✓
SOCIAL-03	Comentar atividades dos amigos	✓
SOCIAL-04	Ver perfil de um utilizador	✓

Tabela 7.7: Testes relacionados com grupos

Requisito	Descrição do teste	Validação
GROUPS-01	Criar grupos	✓
GROUPS-02	Adicionar utilizadores a grupos	✓
GROUPS-03	Apagar grupos	✓

Tabela 7.8: Testes relacionados com configurações

Requisito	Descrição do teste	Validação
SETTINGS-01	Definir veículo por defeito	✓
SETTINGS-03	Definir unidade de distância	✓
SETTINGS-04	Definir formato horário	✓
SETTINGS-05	Definir configurações de partilha	✓
SETTINGS-06	Ver os <i>Terms and Conditions</i> da aplicação	✓
SETTINGS-07	Classificar a aplicação	✗ ¹

7.2 Testes de software

7.2.1 Static analyzer

As aplicações iOS são compiladas recorrendo a um compilador, Apple LLVM Compiler², que garante desde logo uma primeira validação de eventuais erros existentes no código através de uma análise sintática e semântica. No entanto, tendo em conta a função dos compiladores ser a geração de um binário num curto espaço de tempo esta componente de análise do código não se apresenta como profunda e exaustiva, existindo ferramentas especializadas apenas nessa análise, o denominado *static analyzer*. Durante os desenvolvimentos foi feito recurso ao *static analyzer* fornecido na ferramenta de desenvolvimento Xcode, Clang Static Analyzer³, que procede a uma análise detalhada de código tendo em consideração aspetos como a desalocação de objetos em falta que conduzem a *memory-leaks*, o cumprimento das *guidelines* de código especificadas pela Apple, variáveis inicializadas e não utilizadas entre outros aspetos. Esta análise é importante sendo feita periodicamente de modo a identificar possíveis problemas nos desenvolvimentos podendo ser prontamente resolvidos contribuindo ativamente para a garantia de qualidade do software.

7.2.2 Testes tráfego de dados

Este teste teve como objetivo a avaliação do peso da transferência de dados, nos dados móveis da aplicação. Neste teste também ficou perceptível que a alteração efetuada à arquitetura da aplicação traduziu-se em melhorias no consumo dos dados, muito devido ao suporte da compressão de dados através do GZIP, como evidenciado na Tabela 7.9.

Tabela 7.9: Testes compressão para viagem de 60 minutos

Método de compressão	Tamanho dos dados	Taxa compressão
Nenhum	980KB	0%

²<https://developer.apple.com/library/iOS/documentation/CompilerTools/Conceptual/LLVMCompilerOverview/index.html>

³<http://clang-analyzer.llvm.org>

Método de com- pressão	Tamanho dos dados	Taxa compres- são
LZW77	520KB	0.47%
GZIP	133KB	0.86%

Nos dados presentes em cada viagem toda a informação sensorial e de localização originalmente recolhida e também todos os novos dados obtidos através do tratamento dessa informação, índices de segurança e economia e eventos detetados. Se a viagem incluir fotografias as mesmas vão juntamente com esta informação. Neste caso de teste, como o que se pretende avaliação são os mecanismos de compressão, não se incluíram fotografias, uma vez que estas já se encontram num formato comprimido.

7.2.3 Testes de internacionalização

Tendo em conta que o público alvo do produto é global houve a necessidade e a preocupação de garantir a internacionalização do conteúdo textual da aplicação. Assim foi usado o mecanismo nativo do iOS, o qual usa ficheiros com um conjunto de itens chave-valor para cada uma das línguas suportadas, sendo utilizado o ficheiro correspondente à língua do dispositivo ou, em caso de falta de suporte, a língua padrão definida, neste caso será o inglês.

Ao longo dos desenvolvimentos foram feitos sucessivos testes para garantir a internacionalização da aplicação, tornado assim o processo de suporte a uma nova língua simples por requerer apenas a criação do respetivo ficheiro com o conjunto de chaves definidas e a respetiva tradução para a nova língua.

7.3 Testes em ambiente real

Os testes em ambiente real são fundamentais para a validação da aplicação. Estes testes consistem na utilização da aplicação num cenário real, registo de viagens reais com comportamentos definidos de forma a avaliar se a aplicação deteta os eventos que a que se propõe dar resposta. Estes testes ocorreram quer no fim da criação do protótipo como durante o desenvolvimento da aplicação nativa iOS, no decorrer da afinação do cálculo dos índices de segurança e economia e na deteção de eventos. Estes testes foram levados a cabo pelos

elementos da equipa do projeto.

No decorrer dos testes em ambiente real foram considerados cenários prévios de teste, de forma a prever todas as incidências sobre estes, e também permitir o controlo sobre os mesmos. Durante os testes a aplicação estava preparada para gerar ficheiros de *log* com toda a informação necessária para posterior análise.

Os testes em ambiente real foram de utilidade extrema para os trabalhos realizados durante o estágio, visto terem permitido detetar problemas no tratamento da informação sensorial, concretamente na obtenção da aceleração na direção do movimento. Estes testes permitiram também verificar se os índices estavam a obter os resultados desejados, assim como verificar a correta deteção dos eventos.

Capítulo 8

Reflexões e trabalho futuro

Este capítulo expõe as conclusões do presente estágio, o trabalho futuro a realizar no âmbito deste projeto e ainda algumas reflexões sobre o trabalho futuro.

8.1 Conclusão

O mercado dos dispositivos móveis tem vindo a tornar-se uma tecnologia dominante no mundo atual, sendo cada vez mais utilizados no dia a dia para as mais diversas tarefas. Com este trabalho tentou-se assim desenvolver uma aplicação inovadora, diferente das ofertas de mercado existentes, que permitisse não só algo divertido para o utilizador mas também útil. Foi assim desenvolvido o trabalho descrito durante este relatório, baseado numa arquitetura que permitisse futuramente a sua expansão a outros sistemas operativos móveis e à adição de novas funcionalidades. No desenrolar dos trabalhos o autor tentou sempre ao máximo utilizar as últimas versões das tecnologias utilizadas no projeto, garantindo assim uma maior compatibilidade com os sistemas atuais e futuros.

No presente documento é possível ver de forma clara o processo de construção de raiz da aplicação. Este iniciou-se com a pesquisa por ferramentas existentes, não apenas para garantir que se estava a realizar algo inovador mas também para obter ideias sobre as tendências do mercado atual de forma a enriquecer a solução pretendida. De seguida foram definidas quais as metas a atingir com a mesma. Após o desenho da solução, passou-se à implementação de um protótipo como prova de conceito dos objetivos que a aplicação pretende alcançar, seguindo-se a implementação da aplicação nativa iOS de acordo com o planeamento efetuado e as tecnologias estudadas. Antes de fi-

nalizar o documento foram realizados os testes à aplicação com os requisitos idealizados. Esta é uma tarefa que se reveste de toda a importância, isto porque é a forma do autor provar que conseguiu atingir os objetivos a que se propôs: a construção de uma aplicação capaz de registrar e fazer uma análise sobre o estilo de condução dos utilizadores através de informação sensorial e de localização, que funcionasse em diversos sistemas operativos diferentes.

Em termos de experiência o estágio forneceu um contributo muito especial ao estagiário uma vez que permitiu conhecer a realidade de uma equipa de desenvolvimento e a aplicação na prática do conceitos de Engenharia de *Software*. O autor ganhou também experiência em diversas áreas, destacando o desenvolvimento na linguagem Objective-c, javascript, CSS e para a plataforma iOS, sendo esta última considerada a mais importante para o autor pois o elevado crescimento deste sistema operativo torna-o um alvo bastante apelativo para um desenvolvimento de aplicações futuras. Certamente que a mais valia de conhecimento adquirido foi todo o processo de estudo, desenvolvimento e implementação de todo o projeto o que permitiu ao autor adquirir a experiência e conhecimento necessários para a continuidade, não só deste projeto, mas também de futuros projetos.

8.2 Trabalho futuro

Do ponto de vista do autor o desenvolvimento futuro para um trabalho deste tipo passa pelo lançamento de um produto para a Apple Store, a loja de aplicações iOS. Após o lançamento do produto no mercado de aplicações será possível então verificar o sucesso do mesmo, passando o futuro pelo desenvolvimento para outras plataformas, Android, com o posterior lançamento na loja de aplicações do mesmo. Ao mesmo tempo estes lançamentos permitem a correção de bugs encontrados pelos utilizadores e também proporcionam um *feedback* dos mesmos de modo a saber quais os problemas ou sugestões ao produto lançado.

A inclusão de novas funcionalidades é outro investimento futuro numa aplicação. De modo a despertar a curiosidade dos utilizadores seria interessante adicionar novas funcionalidades como a ligação com o carro através de OBD2 ou *Bluetooth*, registo automático de viagens quando o veículo entra em movimento são algumas das funcionalidades já pensadas para incluir futuramente na aplicação. O trabalho futuro passa igualmente pelo desenvolvimento de uma versão da aplicação para o ambiente empresarial, com vista a integrar com a componente B2B do projeto EcoMobile.

As tecnologias estão em constante avanço, tornando o desenvolvimento deste tipo de aplicações um trabalho em constante mutação, adaptando-se às circunstâncias que vão surgindo, como por exemplo, novas versões do sistema operativo que tornam funcionalidades utilizadas atualmente obsoletas.

Bibliografia

- [1] Rui Chicória. Real-time architecture and workflow for the One.Stop.Transport platform.
- [2] Wikipedia. Mobile operating system — wikipedia, the free encyclopedia, 2014. Acedido em 18 de Julho de 2014.
- [3] Roger S. Pressman. Software Engineering - A Practitioner's Approach.
- [4] Atlassian. Jira. <https://www.atlassian.com/software/jira>. Acedido em 20 de Janeiro de 2014.
- [5] Atlassian. Atlassian. <https://www.atlassian.com>. Acedido em 20 de Janeiro de 2014.
- [6] Atlassian. Confluence. <https://www.atlassian.com/software/confluence>. Acedido em 20 de Janeiro de 2014.
- [7] GitLab.org. Gitlab. <http://gitlab.org>. Acedido em 20 de Janeiro de 2014.
- [8] S Chacon. Pro git, 1 ed. books for professionals by professionals. Apress, San Francisco, California, 2009. Acedido em 20 de Janeiro de 2014.
- [9] Apple. Model-view-controller. <https://developer.apple.com/library/mac/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>. Acedido em 18 de Julho de 2014.
- [10] Slash Idea s.r.o. Mycartracks. <http://www.mycartracks.com/>. Acedido em 20 de Janeiro de 2014.
- [11] Inc. Automatic Labs. Automatic. <http://www.automatic.com/>. Acedido em 20 de Janeiro de 2014.
- [12] DriSMo. Drismo. <https://play.google.com/store/apps/details?id=com.drismo>. Acedido em 20 de Janeiro de 2014.

- [13] Google. Mytracks. <http://www.google.com/mobile/mytracks/>. Acedido em 20 de Janeiro de 2014.
- [14] TomTom International BV. Tomtom webfleet mobile. https://business.tomtom.com/pt_pt/products/webfleet-mobile/highlights/. Acedido em 20 de Janeiro de 2014.
- [15] *A Guide to the Business Analysis Body of Knowledge (BABOK Guide)*. International Institute of Business Analysis.
- [16] Eduardo Miranda. Time boxing planning: Buffered moscow rules. <http://www.stickyminds.com/article/time-boxing-planning-buffered-moscow-rules>. Acedido em 20 de Janeiro de 2014.
- [17] Stacy C. Davis, Susan W. Diegel, and C.Robert G. Boundy Clarisse. STransportation Energy Data Book.
- [18] IMTT. Manual de eco-condução. http://www.imtt.pt/sites/IMTT/Portugues/Condutores/Ecoconducao/Documents/Manual_EcoConducao.pdf. Acedido em 11 de Janeiro de 2014.
- [19] Mobile Statistics. Mobile statistics. <http://www.mobilestatistics.com/mobile-statistics/>. Acedido em 20 de Janeiro de 2014.

Apêndice A

Análise de requisitos

Para descrever a prioridade dos requisitos decidiu-se utilizar as palavras-chave definidas no método MoSCoW[15][16], "Must", "Should", "Could" e "Won't".

A.1 Requisitos funcionais

As funcionalidades do projeto foram especificadas através de *user stories*, que consistem numa breve descrição de uma determinada funcionalidade na perspetiva de um ator do sistema, para qual esta é relevante, evitando assim o foco em funcionalidades com um ganho reduzido para os utilizadores do produto final.

As *User Stories* foram escritas segundo o seguinte formato:

Enquanto <ator do sistema> (Quem?)

Quero <ação a executar> (O quê?)

De forma a <proposta de valor> (Porquê?)

A.1.1 Requisitos relacionados com contas de utilizador

User Story ACCOUNT-01	Prioridade MUST
Enquanto utilizador registado	
Quero autenticar-me na aplicação com as minhas credenciais	
De forma a poder utilizar usufruir das funcionalidades da mesma	

User Story ACCOUNT-02	Prioridade MUST
Enquanto utilizador sem registo	
Quero redirecionar para a formulário de adicionar conta na aplicação	
De forma a poder criar uma conta na aplicação	

User Story ACCOUNT-03	Prioridade MUST
Enquanto utilizador registado	
Quero autenticar-me na aplicação com o facebook	
De forma a poder utilizar usufruir das funcionalidades da mesma	

User Story ACCOUNT-04	Prioridade MUST
Enquanto utilizador registado	
Quero fazer <i>logout</i> da aplicação	
De forma a terminar a minha secção na mesma	

A.1.2 Requisitos relacionados com registo de viagens

User Story TRIP-01	Prioridade MUST
Enquanto utilizador registado	
Quero começar o registo de uma viagem carregando no botão para esse efeito presente no ecrã inicial	
De forma a poder registar a informação sobre viagens	

User Story TRIP-02	Prioridade MUST
Enquanto utilizador registado	
Quero receber <i>live feedback</i> da viagem que estou a registar sobre travagens, acelerações e curvas bruscas	
De forma a poder verificar em tempo real os eventos que condicionam a minha economia e segurança	

User Story TRIP-03	Prioridade MUST
Enquanto utilizador registado	
Quero ter durante o registo de uma viagem um botão de pausa e resumo da viagem	
De forma a poder pausar e retomar o registo da informação da viagem quando entender	

User Story TRIP-04	Prioridade MUST
Enquanto utilizador registado	
Quero ter durante o registo de uma viagem um botão para parar a viagem	
De forma a poder parar o registo de informação da viagem	

User Story TRIP-05	Prioridade MUST
Enquanto utilizador registado	
Quero ter durante a viagem gráficos indicativos sobre a minha segurança e economia	
De forma a poder acompanhar estes índices ao longo da viagem	

User Story TRIP-06	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero ter durante o registo de uma viagem a informação sobre velocidade, altitude, duração, distância percorrida e mapa com trajecto</p> <p>De forma a poder acompanhar estes indicadores ao longo da viagem</p>	
User Story TRIP-07	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero partilhar na rede social o estado atual da viagem a ser registada, os seus indicadores e índices</p> <p>De forma a os meus amigos poderem visualizar esta informação</p>	
User Story TRIP-08	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero usar a câmara do dispositivo</p> <p>De forma a poder adicionar fotografias à viagem</p>	

User Story TRIP-09	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero usar a câmara do dispositivo</p> <p>De forma a poder adicionar vídeos à viagem</p>	

A.1.3 Requisitos relacionados com viagens

User Story TRIPS-01	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero ter uma lista com as viagens registadas</p> <p>De forma a poder visualizar as minhas viagens registadas</p>	

User Story TRIPS-02	Prioridade MUST
Enquanto utilizador registado Quero pesquisar nas minhas viagens registadas por localização De forma a poder encontrar facilmente as viagens pretendidas	
User Story TRIPS-03	Prioridade MUST
Enquanto utilizador registado Quero seleccionar uma viagem da minha lista de viagens registadas De forma a poder ver os detalhes sobre ela	
User Story TRIPS-04	Prioridade MUST
Enquanto utilizador registado Quero editar a informação referente a uma viagem registada De forma a poder alterar o seu inicio, fim, veículo, passageiros	
User Story TRIPS-05	Prioridade MUST
Enquanto utilizador registado Quero ter a opção de apagar uma viagem registada De forma a apagar toda a sua informação	
User Story TRIPS-06	Prioridade COULD
Enquanto utilizador registado Quero agrupar viagens registadas De forma a poder juntar viagens que passam a representar uma só viagem	
User Story TRIPS-07	Prioridade MUST
Enquanto utilizador registado Quero editar a galeria de imagens da viagem De forma a poder adicionar ou remover imagens a uma viagem	

User Story TRIPS-08	Prioridade COULD
Enquanto utilizador registado	
Quero editar a galeria de vídeos da viagem	
De forma a poder adicionar ou remover vídeos a uma viagem	

A.1.4 Requisitos relacionados com veículos

User Story CARS-01	Prioridade MUST
Enquanto utilizador registado	
Quero ter uma lista com as veículos registados	
De forma a poder visualizar os meus veículos registados	

User Story TRIPS-02	Prioridade MUST
Enquanto utilizador registado	
Quero pesquisar nos meus veículos registados por marca e modelo	
De forma a poder encontrar facilmente os veículos pretendidos	

User Story TRIPS-03	Prioridade MUST
Enquanto utilizador registado	
Quero selecionar um veículo da minha lista de veículos registados	
De forma a poder ver os detalhes sobre ele	

User Story TRIPS-04	Prioridade MUST
Enquanto utilizador registado	
Quero editar a informação referente a um veículo registado	
De forma a poder alterar o seu ano, tipo	

User Story TRIPS-05	Prioridade MUST
Enquanto utilizador registado	
Quero ter a opção de apagar uma veículo registado	
De forma a apagar toda a sua informação	

User Story TRIPS-06	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero editar a galeria de imagens do veículo</p> <p>De forma a poder adicionar ou remover imagens a um veículo</p>	
User Story TRIPS-07	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero marcar um veículo como <i>default</i></p> <p>De forma a ser o veículo associado por defeito no registo de viagens</p>	

A.1.5 Requisitos sociais

User Story SOCIAL-01	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero ter a lista de atividades dos meus amigos</p> <p>De forma a poder ver a informação sobre todas as atividades dos meus amigos</p>	
User Story SOCIAL-02	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero adicionar ou remover amigos da minha lista de amigos</p> <p>De forma a controlar quem é que são os meus amigos</p>	
User Story SOCIAL-03	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero comentar as atividades dos meus amigos</p> <p>De forma a estes puderem ver os comentários</p>	
User Story SOCIAL-04	Prioridade MUST
<p>Enquanto utilizador registado</p> <p>Quero ver o perfil de um utilizador</p> <p>De forma a poder ver as suas informações</p>	

A.1.6 Requisitos relacionados com grupos

User Story GROUPS-01	Prioridade SHOULD
Enquanto utilizador registado	
Quero ter a opção para criar grupos	
De forma a poder ter grupos restritos	

User Story GROUPS-02	Prioridade SHOULD
Enquanto utilizador registado	
Quero adicionar amigos a um grupo	
De forma a poder seleccionar os amigos que quero nos meus grupos restritos	

User Story GROUPS-03	Prioridade SHOULD
Enquanto utilizador registado	
Quero apagar um grupo que tenha criado	
De forma a remover o seu registo	

A.1.7 Requisitos relacionados com configurações

User Story SETTINGS-01	Prioridade MUST
Enquanto utilizador registado	
Quero ter a opção para seleccionar qual vai ser o meu veículo <i>default</i> de entre os meus veículos	
De forma a ser a veículo associado por defeito no registo de viagens	

User Story SETTINGS-02	Prioridade MUST
Enquanto utilizador registado	
Quero definir a unidade de distância usada, milhas ou quilómetros	
De forma a ser a unidade usada como referencia por toda a aplicação	

User Story SETTINGS-03	Prioridade SHOULD
Enquanto utilizador registado	
Quero definir o formato horário usado	
De forma a ser o formato horário usado como referencia por toda a aplicação	

User Story SETTINGS-04	Prioridade MUST
Enquanto utilizador registado	
Quero definir as opções de partilha de informação com os amigos, como mapas, imagens e viagens	
De forma a controlar a informação que os amigos podem ver	

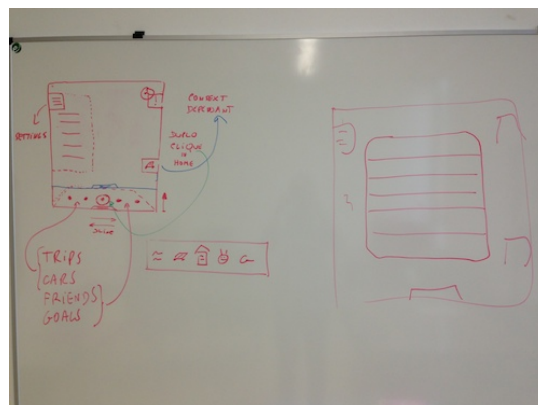
User Story SETTINGS-05	Prioridade MUST
Enquanto utilizador registado	
Quero ver o <i>Legal Agreement</i> da aplicação	
De forma a saber a informação contida no mesmo	

User Story SETTINGS-06	Prioridade COULD
Enquanto utilizador registado	
Quero ter a opção para classificar a aplicação	
De forma a poder atribuir uma avaliação à aplicação	

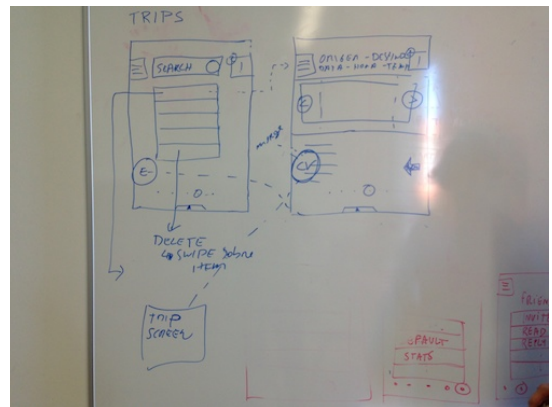
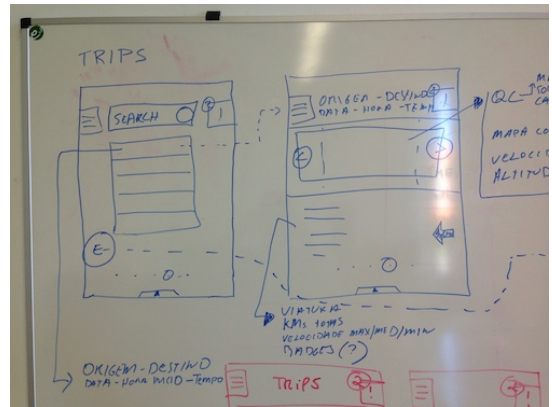
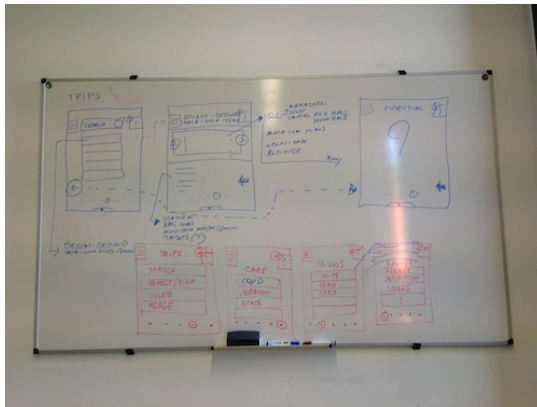
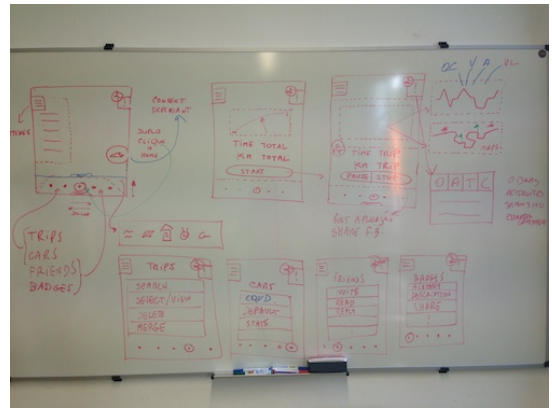
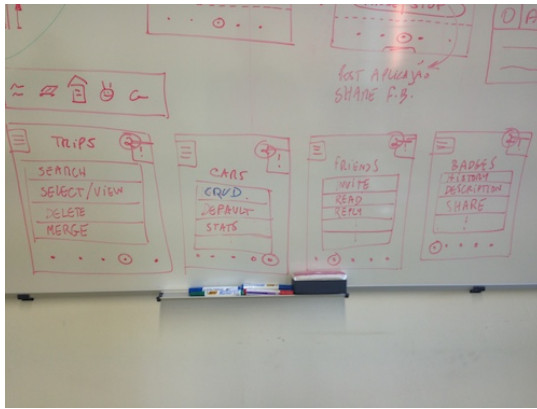
Apêndice B

Prototipagem de baixo nível protótipo

Tal como referido na Secção 6.1, neste Apêndice são apresentados os *mockups* criação da prototipagem de baixo nível do protótipo da aplicação.



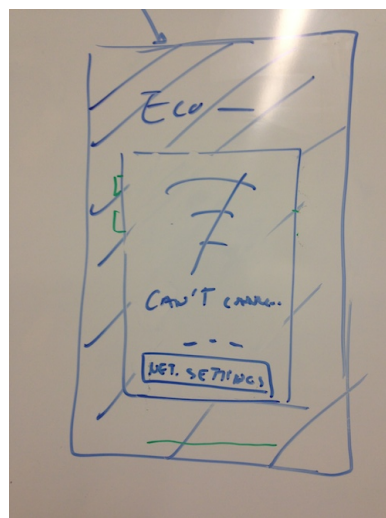
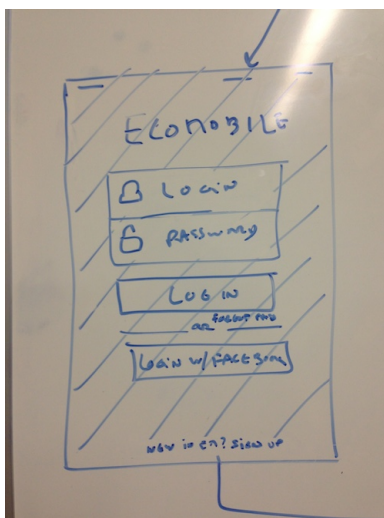
94 APÊNDICE B. PROTOTIPAGEM DE BAIXO NÍVEL PROTÓTIPO

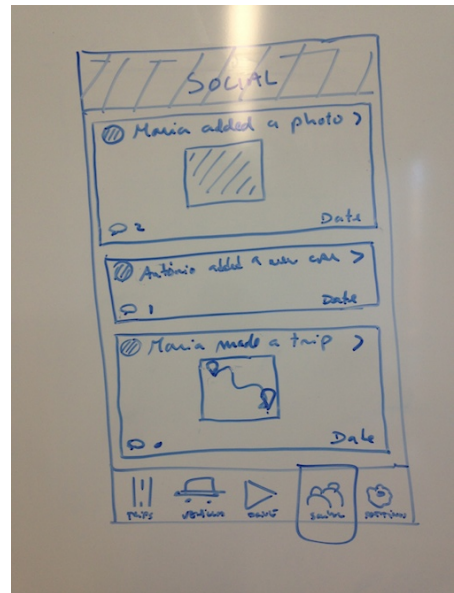
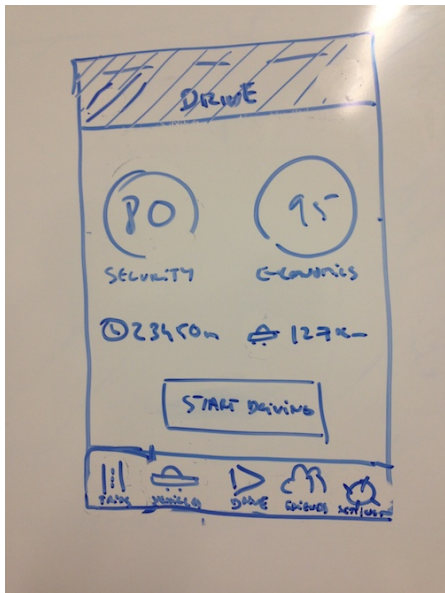
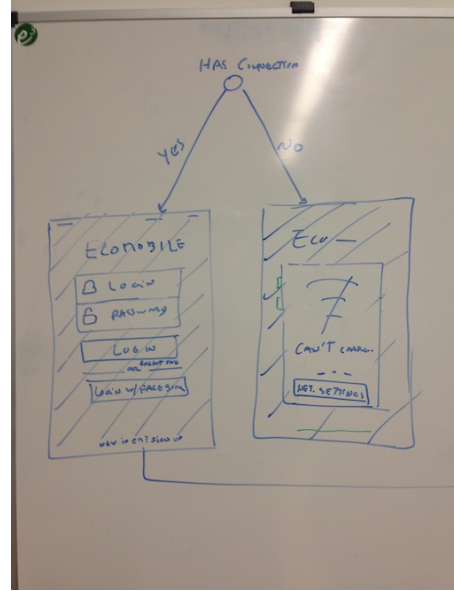
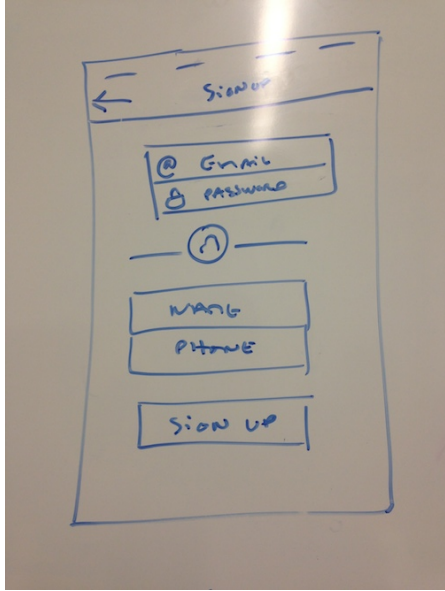


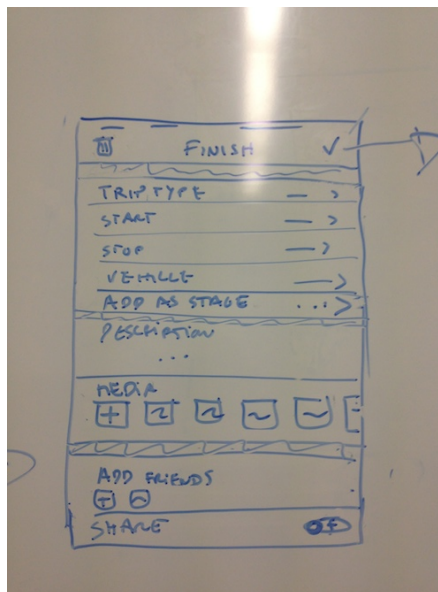
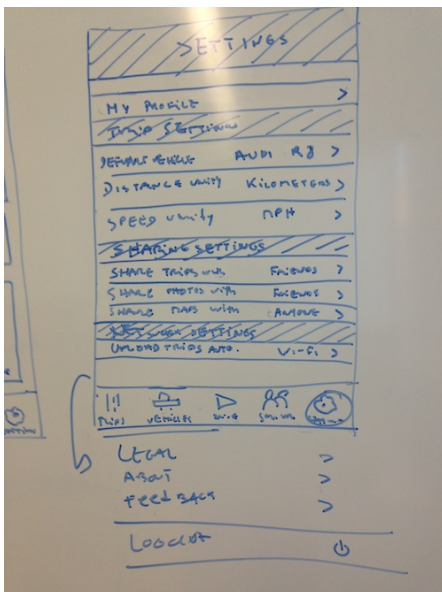
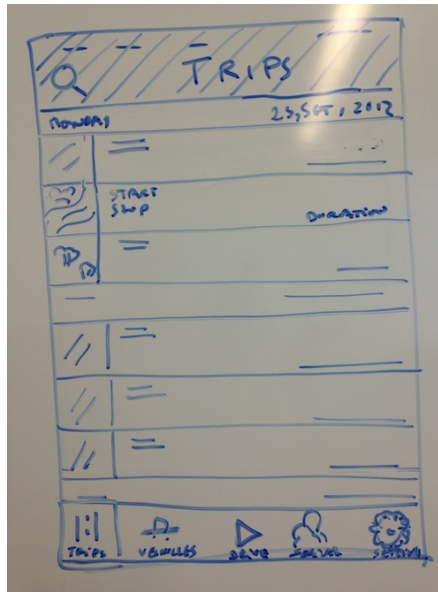
Apêndice C

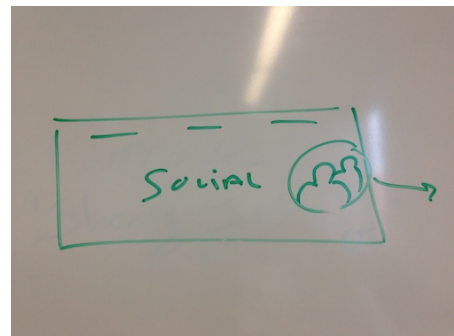
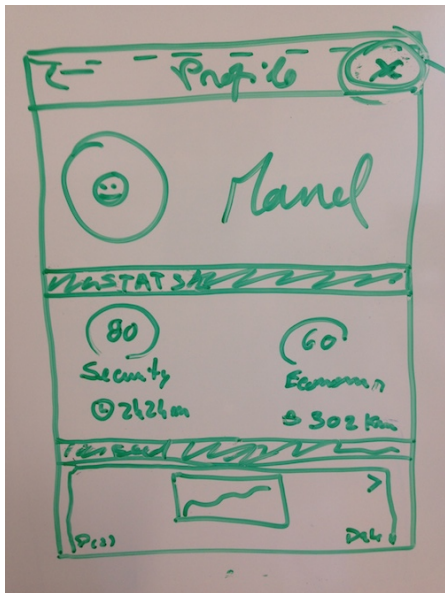
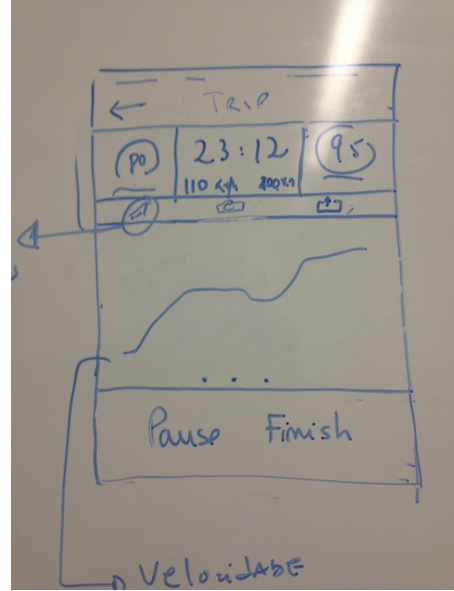
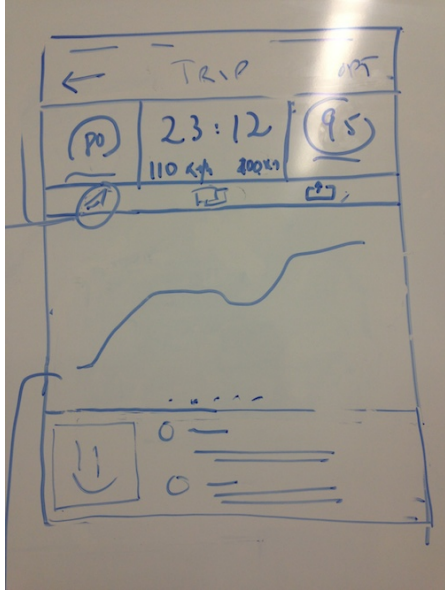
Prototipagem de baixo nível iOS nativo

Tal como referido na Secção 6.2, neste Apêndice são apresentados os *mockups* criação da prototipagem de baixo nível da aplicação nativa para o sistema operativo móvel iOS.









Apêndice D

Resultados da implementação do protótipo

Neste Apêndice são apresentados os ecrãs do protótipo da aplicação.

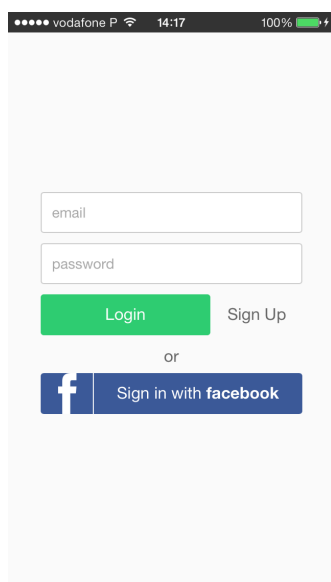


Figura D.1: Ecrã *Login* protótipo

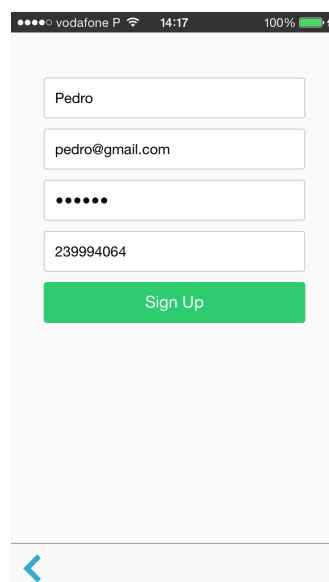


Figura D.2: Ecrã registo protótipo

102 APÊNDICE D. RESULTADOS DA IMPLEMENTAÇÃO DO PROTÓTIPO

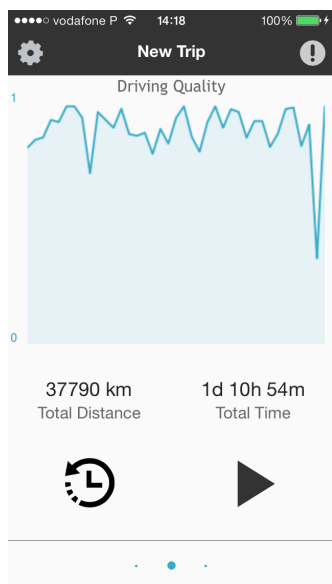


Figura D.3: Ecrã Inicial protótipo

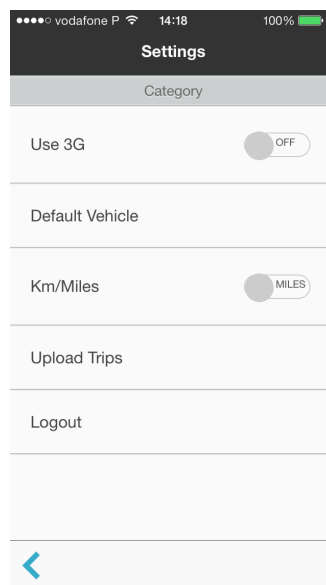


Figura D.4: Ecrã definições protótipo



Figura D.5: Ecrã Lista de viagens protótipo

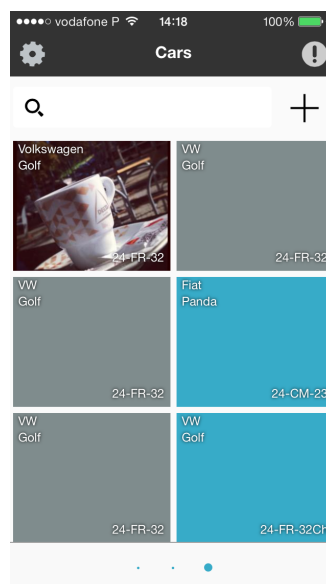


Figura D.6: Ecrã lista de veículos protótipo

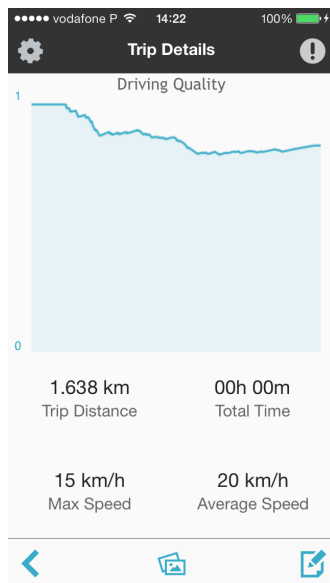


Figura D.7: Ecrã viagem com o gráfico protótipo

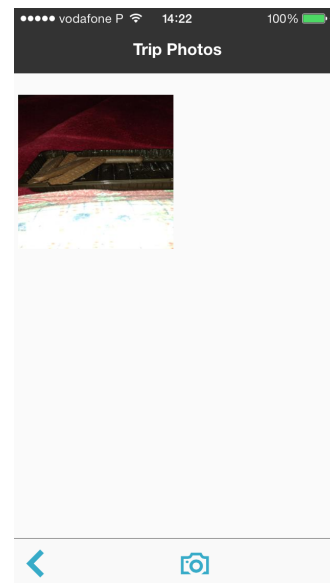


Figura D.8: Ecrã galeria de imagens protótipo

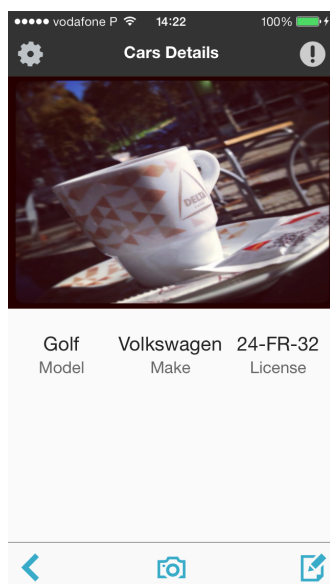


Figura D.9: Ecrã detalhes veículo protótipo

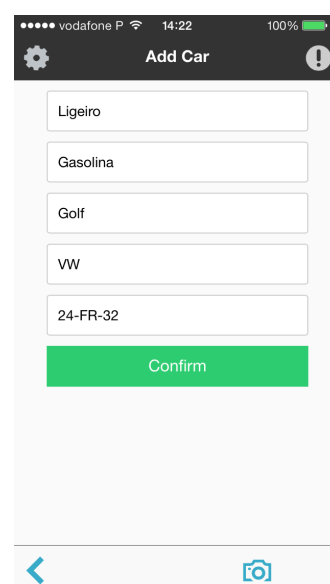


Figura D.10: Ecrã adicionar veículo

104 APÊNDICE D. RESULTADOS DA IMPLEMENTAÇÃO DO PROTÓTIPO

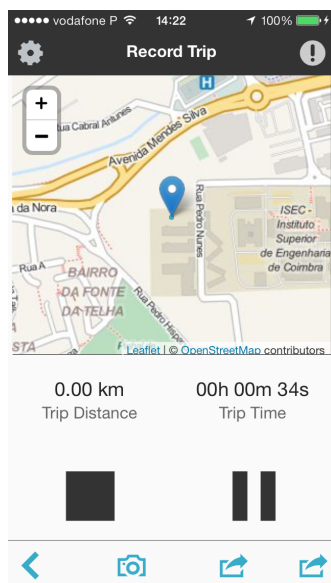


Figura D.11: Ecrã registo de viagem com mapa protótipo

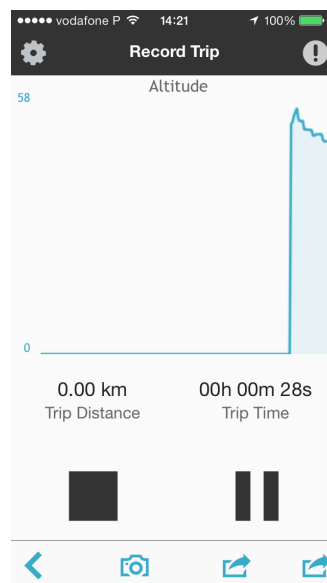


Figura D.12: Ecrã registo de viagem com gráfico protótipo

Figura D.13: Ecrã fim de viagem protótipo

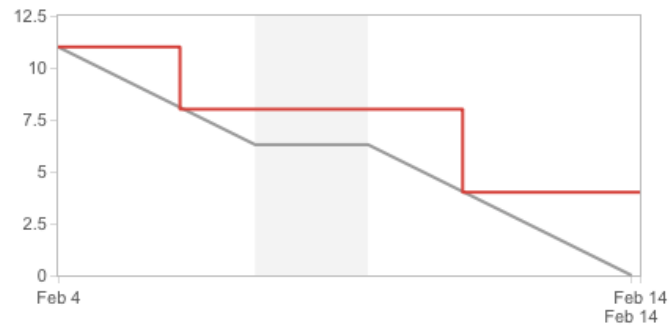
Apêndice E

Artefactos da metodologia ágil

Como referido na secção 2.1 aqui são apresentados os artefactos, relativos à metodologia ágil, produzidos durante a elaboração deste estágio. Aqui serão apresentados os *Sprint Reports* associados a cada uma das *sprints*.

Closed Sprint 04/Feb/14 12:15 PM - 14/Feb/14 7:31 PM

[View Ecomobile Sprint 14 in Issue Navigator](#)

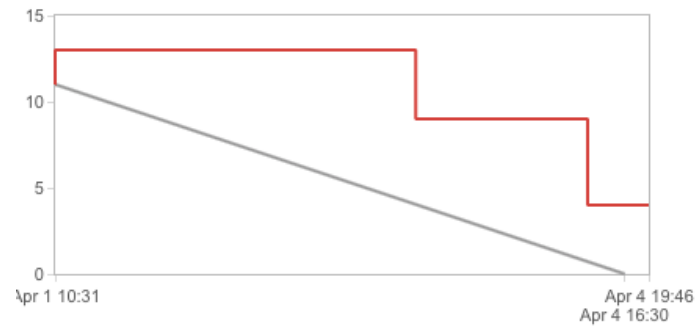


Completed Issues						View in Issue Navigator
Key	Summary	Issue Type	Priority	Status	Story Points (7)	
ECOMOBILE-163	iOS App - Mobile app sample calculation (redefine)	Story	Major	Resolved	3	
ECOMOBILE-239	iOS App - Debug and corrections	Story	Major	Resolved	4	

Issues Not Completed						View in Issue Navigator
Key	Summary	Issue Type	Priority	Status	Story Points (4)	
ECOMOBILE-225	iOS App - On trip feedback	Story	Major	In Progress	2	
ECOMOBILE-240	iOS App - Safety value	Story	Major	Open	2	

Closed Sprint 01/Apr/14 10:31 AM - 04/Apr/14 7:46 PM

[View Ecomobile Sprint 20 in Issue Navigator](#)



Completed Issues

[View in Issue Navigator](#)

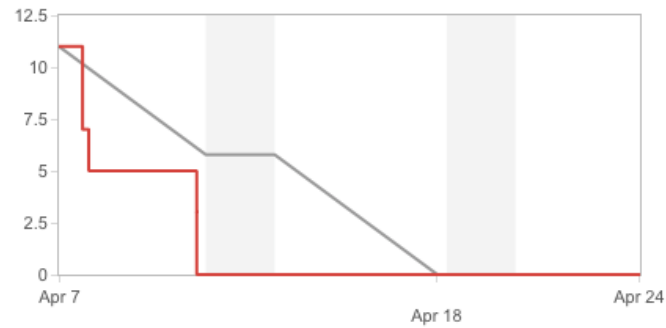
Key	Summary	Issue Type	Priority	Status	Story Points (9)
ECOMOBILE-289	iOS App - Get reliable economy values with altitude variation	💡 Story	🔴 Major	🟢 Resolved	4
ECOMOBILE-294	iOS App - Get reliable cornering events	💡 Story	🔴 Major	🟢 Resolved	5

Issues Not Completed

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (2)
ECOMOBILE-288	iOS App - Get reliable security values	💡 Story	🔴 Major	🟠 In Progress	2

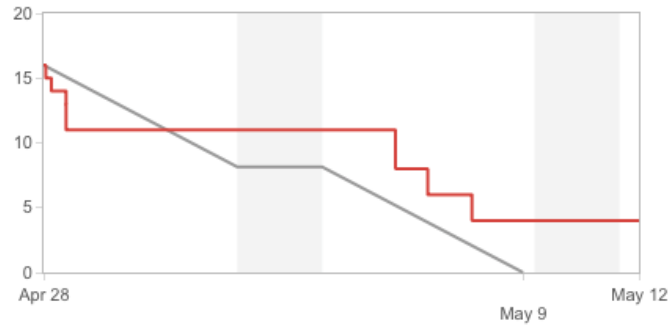
Closed Sprint 07/Apr/14 5:30 PM - 24/Apr/14 2:52 PM

[View Ecomobile Sprint 21 in Issue Navigator](#)**Completed Issues**[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (11)
ECOMOBILE-288	iOS App - Get reliable security values	Story	Major	Resolved	4
ECOMOBILE-291	iOS App - Show events	Story	Major	Resolved	2
ECOMOBILE-295	iOS App - Get cumulative economy values including acceleration events	Story	Major	Resolved	2
ECOMOBILE-296	iOS App - Get cumulative safety values including acceleration, cornering and braking events	Story	Major	Resolved	3

Closed Sprint 28/Apr/14 10:31 AM - 12/May/14 11:01 AM

[View Ecomobile Sprint 22 in Issue Navigator](#)



Completed Issues						View in Issue Navigator
Key	Summary	Issue Type	Priority	Status	Story Points (12)	
ECOMOBILE-216	iOS App - Social feed	Story	Major	Resolved	2	
ECOMOBILE-220	iOS App - Add group	Story	Major	Resolved	1	
ECOMOBILE-224	iOS App: Social profile	Story	Major	Resolved	1	
ECOMOBILE-277	iOS App - Realtime	Story	Major	Resolved	3	
ECOMOBILE-283	iOS App - Add friend	Story	Major	Resolved	1	
ECOMOBILE-284	iOS App - Add friend to group	Story	Major	Resolved	2	
ECOMOBILE-298	iOS App - Offline info	Story	Major	Resolved	2	

Apêndice F

Modelo de dados da aplicação móvel

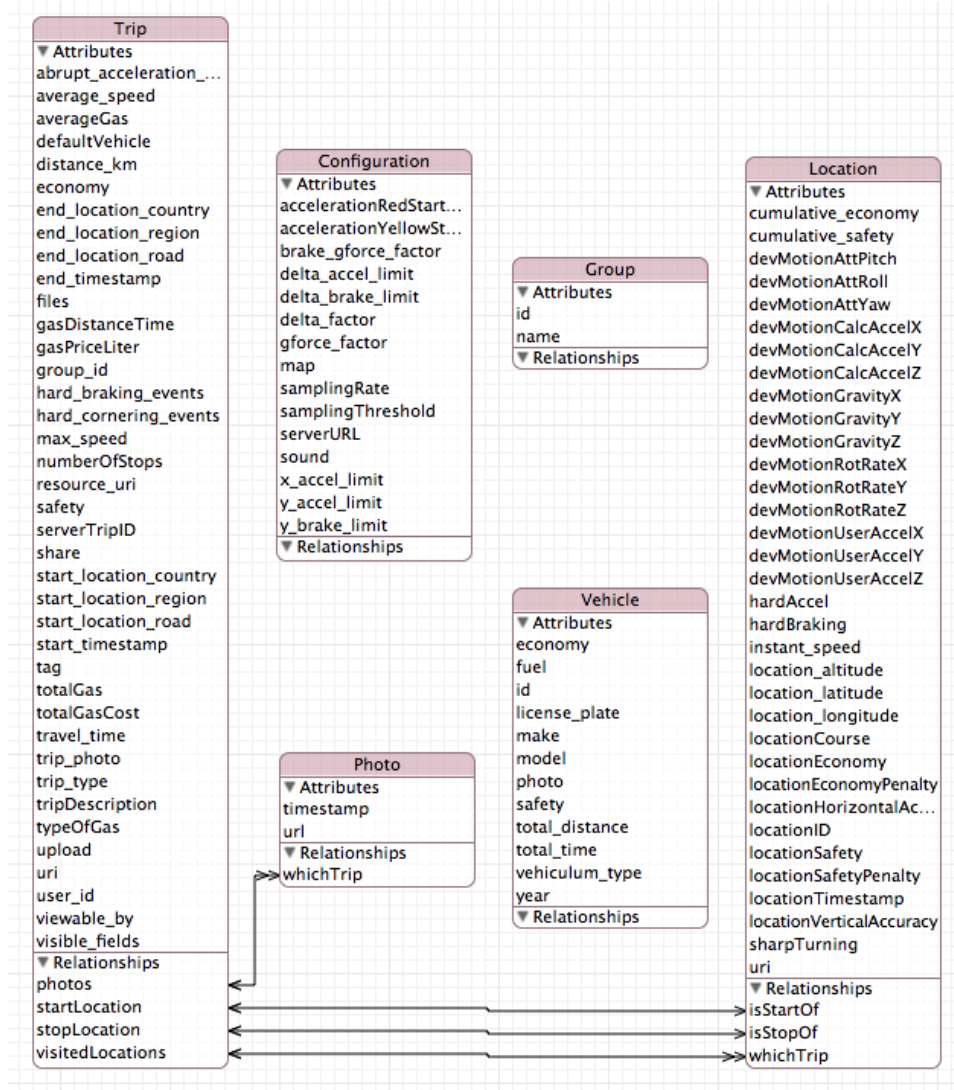


Figura F.1: Modelo de dados da aplicação móvel

Tabela F.1: Descrição das entidades do modelo de dados

Entidade	Descrição
Trip	Representa uma viagem registrada pela aplicação.
Vehicle	Representa um veículo registrado na aplicação.
Configuration	Representa as configurações definidas pela utilizador e também parâmetros internos da aplicação.
Photo	Representa a localização de uma fotografia tirada pela aplicação.
Group	Representa um grupo registrado pela aplicação.
Location	Representa todos os valores associados a cada localização de uma viagem.

Apêndice G

Resumo informação sensorial e de localização

GPS and Motion Data

Description of all readings from motion sensors and GPS performed by the Flitt application, as well as the operations and calculations performed on these data. Since two different operating systems are used, will be made a distinction between the iOS and Android systems.

GPS

	Android	iOS
Latitude	WGS 84	WGS 84
Longitude	WGS 84	WGS 84
Altitude	Meters	Meters
Horizontal Accuracy	Meters	Meters
Vertical Accuracy	-	Meters
Timestamp	Milliseconds since January 1, 1970	NSDate
Speed	M/s	M/s
Course/Bearing	Degrees	Degrees

iOS

- **Latitude:** The latitude in degrees. Positive values indicate latitudes north of the equator. Negative values indicate latitudes south of the equator.
- **Longitude:** The longitude in degrees. Measurements are relative to the zero meridian, with positive values extending east of the meridian and negative values extending west of the meridian.
- **Altitude:** Positive values indicate altitudes above sea level. Negative values indicate altitudes below sea level.
- **Horizontal Accuracy:** The location's latitude and longitude identify the center of the circle, and this value indicates the radius of that circle. A negative value indicates that the location's latitude and longitude are invalid.
- **Vertical Accuracy:** The value in the `altitude` property could be plus or minus the value indicated by this property. A negative value indicates that the altitude value is invalid.
- **Timestamp:** The time at which this location was determined.
- **Speed:** This value reflects the instantaneous speed of the device in the direction of its current heading. A negative value indicates an invalid speed. Because the actual speed can change many times between the delivery of subsequent location events, you should use this property for informational purposes only.
- **Course:** Course values are measured in degrees starting at due north and continuing clockwise around the compass. Thus, north is 0 degrees, east is 90 degrees, south is 180 degrees, and so on. Course values may not be available on all devices. A negative value indicates that the direction is invalid.

Android

- **Latitude:** Latitude, in degrees.
- **Longitude:** Longitude, in degrees.
- **Altitude:** Altitude if available, in meters above sea level. If this location does not have an altitude then 0.0 is returned.

- **Horizontal Accuracy:** We define accuracy as the radius of 68% confidence. In other words, if you draw a circle centered at this location's latitude and longitude, and with a radius equal to the accuracy, then there is a 68% probability that the true location is inside the circle. In statistical terms, it is assumed that location errors are random with a normal distribution, so the 68% confidence circle represents one standard deviation. Note that in practice, location errors do not always follow such a simple distribution.
- **Vertical Accuracy:** Unavailable
- **Timestamp:** Return the UTC time of this fix, in milliseconds since January 1, 1970. Note that the UTC time on a device is not monotonic: it can jump forwards or backwards unpredictably. So always use `getElapsedRealtimeNanos()` when calculating time deltas. On the other hand, `getTime()` is useful for presenting a human readable time to the user, or for carefully comparing location fixes across reboot or across devices. All locations generated by the `LocationManager` are guaranteed to have a valid UTC time, however remember that the system time may have changed since the location was generated.
- **Speed:** Speed if it is available, in meters/second over ground. If this location does not have a speed then 0.0 is returned.
- **Bearing:** Bearing is the horizontal direction of travel of this device, and is not related to the device orientation. It is guaranteed to be in the range (0.0, 360.0] if the device has a bearing. If this location does not have a bearing then 0.0 is returned.

Motion Sensors

iOS

A G is a unit of gravitation force equal to that exerted by the earth's gravitational field (9.81 m s²).

CoreMotion

The Core Motion framework lets your application receive motion data from device hardware and process that data. The framework supports accessing both raw and processed accelerometer data using block-based interfaces. For devices with a built-in gyroscope, you can retrieve the raw gyro data as well as processed data reflecting the attitude and rotation rates of the device. You can use both the accelerometer and gyro-based data for games or other apps that use motion as input or as a way to enhance the overall user experience.

CMDeviceMotion

An instance of `CMDeviceMotion` encapsulates measurements of the attitude, rotation rate, and acceleration of a device.

An application receives or samples `CMDeviceMotion` objects at regular intervals after calling the `startDeviceMotionUpdatesUsingReferenceFrame:toQueue:withHandler:` method, the `startDeviceMotionUpdatesToQueue:withHandler:` method, the `startDeviceMotionUpdatesUsingReferenceFrame:` method, or the `startDeviceMotionUpdates` method of the `CMMotionManager` class.

The accelerometer measures the sum of two acceleration vectors: gravity and user acceleration. User acceleration is the acceleration that the user imparts to the device. Because Core Motion is able to track a device's attitude using both the gyroscope and the accelerometer, it can differentiate between gravity and user acceleration. A `CMDeviceMotion` object provides both measurements in the `gravity` and `userAcceleration` properties.

attitude

- **roll:** The roll of the device, in radians. A roll is a rotation around a longitudinal axis that passes through the device from its top to bottom.
- **pitch:** The pitch of the device, in radians. A pitch is a rotation around a lateral axis that passes through the device from side to side.
- **yaw:** The yaw of the device, in radians. A yaw is a rotation around an axis that runs vertically through the device. It is perpendicular to the body of the device, with its origin at the center of gravity and directed toward the bottom of the device.

rotationRate

Contains data specifying the device's rate of rotation around three axes. The value of this property contains a measurement of gyroscope data whose bias has been removed by Core Motion algorithms.

- **x**: The X-axis rotation rate in radians per second. The sign follows the right hand rule: If the right hand is wrapped around the X axis such that the tip of the thumb points toward positive X, a positive rotation is one toward the tips of the other four fingers.
- **y**: The Y-axis rotation rate in radians per second. The sign follows the right hand rule: If the right hand is wrapped around the Y axis such that the tip of the thumb points toward positive Y, a positive rotation is one toward the tips of the other four fingers.
- **z**: The Z-axis rotation rate in radians per second. The sign follows the right hand rule: If the right hand is wrapped around the Z axis such that the tip of the thumb points toward positive Z, a positive rotation is one toward the tips of the other four fingers.

gravity

The gravity acceleration vector expressed in the device's reference frame.

- X-axis acceleration in G's (gravitational force).
- Y-axis acceleration in G's (gravitational force).
- Z-axis acceleration in G's (gravitational force).

userAcceleration

The acceleration that the user is giving to the device.

- X-axis acceleration in G's (gravitational force).
- Y-axis acceleration in G's (gravitational force).
- Z-axis acceleration in G's (gravitational force).

magneticField

Returns the magnetic field vector with respect to the device. The `CMCalibratedMagneticField` returned by this property gives you the total magnetic field in the device's vicinity without device bias. Unlike the `magneticField` property of the `CMMagnetometer` class, these values reflect the earth's magnetic field plus surrounding fields, minus device bias.

- **field**: A structure containing 3-axis calibrated magnetic field data. See the description of the `CMMagneticField` structure.
- **accuracy**: An enum-constant value that indicates the accuracy of the magnetic field estimate.

CMAccelerometerData

An instance of the `CMAccelerometerData` class represents an accelerometer event. It is a measurement of acceleration along the three spatial axes at a moment of time.

An application accesses `CMAccelerometerData` objects through the block handler specified as the last parameter of the `startAccelerometerUpdatesToQueue:withHandler:` method and through the `accelerometerData` property, both declared by the `CMMotionManager` class. The superclass of `CMAccelerometerData`, `CMLogItem`, defines a `timestamp` property that records when the acceleration measurement was taken.

acceleration

The gravity acceleration vector expressed in the device's reference frame.

- X-axis acceleration in G's (gravitational force).
- Y-axis acceleration in G's (gravitational force).
- Z-axis acceleration in G's (gravitational force).

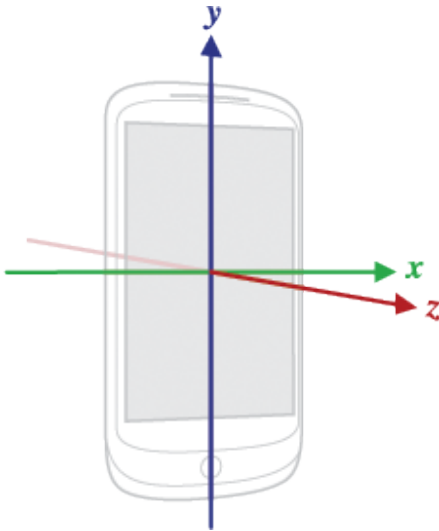
Android

Sensor Coordinate System

In general, the sensor framework uses a standard 3-axis coordinate system to express data values. For most sensors, the coordinate system is defined relative to the device's screen when the device is held in its default orientation (see figure 1). When a device is held in its default

orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values. This coordinate system is used by the following sensors:

- Acceleration sensor
- Gravity sensor
- Gyroscope
- Linear acceleration sensor
- Geomagnetic field sensor



The most important point to understand about this coordinate system is that the axes are not swapped when the device's screen orientation changes—that is, the sensor's coordinate system never changes as the device moves. This behavior is the same as the behavior of the OpenGL coordinate system.

Another point to understand is that your application must not assume that a device's natural (default) orientation is portrait. The natural orientation for many tablet devices is landscape. And the sensor coordinate system is always based on the natural orientation of a device.

Finally, if your application matches sensor data to the on-screen display, you need to use the `getRotation()` method to determine screen rotation, and then use the `remapCoordinateSystem()` method to map sensor coordinates to screen coordinates. You need to do this even if your manifest specifies portrait-only display.

Acceleration sensor

Acceleration sensor measures the acceleration applied to the device, including the force of gravity.

- X-axis acceleration in G's (gravitational force).
- Y-axis acceleration in G's (gravitational force).
- Z-axis acceleration in G's (gravitational force).

Gyroscope

Gyroscope measures the rate or rotation in rad/s around a device's x, y, and z axis.

Linear acceleration sensor

Linear acceleration sensor provides you with a three-dimensional vector representing acceleration along each device axis, excluding gravity.

Geomagnetic field sensor

Geomagnetic field sensor lets you monitor changes in the earth's magnetic field.

Samples

Acceleration Calculation

Acceleration calculation formula based on the data from the motion sensors.

Algorithm 1 Acceleration values calculation algorithm

Vector A = Vector acceleration

Vector G = Vector gravity

Vector Z = Vector invert gravity

bool hasDirection = false

if (Gyroscope) **then**

A = UserAcceleration

G = Gravity

Z = invertdirection(G)

else

 Vector RefGravity = Vector with reference gravity

 Vector tmp = Acceleration

if (RefGravity == null) **then**

if (norm(tmp) == GRAVITY) **then**

RefGravity = tmp

end if

if (RefGravity == null) **then return**

end if

end if

A = RefGravity - Acceleration

Z = invertdirection(RefGravity)

end if

Vector vectorCL = Vector current location

Vector vectorLL = Vector last location

Vector vectorCLtoLL = Vector current location to last location

if (norm(vectorCLtoLL) >0) **then**

hasDirection = true

end if

if (hasDirection) **then**

 Vector AX = X axis acceleration

 Vector AY = Y axis acceleration

 Vector AZ = Z axis acceleration

end if

	Description	Units
timestamp	The time at which this location was determined	Milliseconds since January 1, 1970
instant_speed	Speed from GPS	M/s
location_altitude	Altitude from GPS	Meters
location_latitude	Latitude from GPS	WGS 84
location_longitude	Longitude from GPS	WGS 84
motion_att_pitch		
motion_att_roll		
motion_att_yaw		
motion_calc_accel_x	X axis acceleration value excluding the gravity and pointing to the direction of the movement.	M/s ²
motion_calc_accel_y	Y axis acceleration value excluding the gravity and pointing to the direction of the movement.	M/s ²
motion_calc_accel_z	Z axis acceleration value excluding the gravity and pointing to the direction of the movement.	M/s ²
motion_gravity_x		M/s ²
motion_gravity_y		M/s ²
motion_gravity_z		M/s ²
motion_rot_rate_x		Rad/s
motion_rot_rate_y		Rad/s
motion_rot_rate_z		Rad/s
motion_user_accel_x	Raw x axis acceleration without gravity.	M/s ²
motion_user_accel_y	Raw y axis acceleration without gravity.	M/s ²
motion_user_accel_z	Raw z axis acceleration without gravity.	M/s ²
location_course		Degrees
location_horizontal_accuracy		Meters
location_vertical_accuracy		Meters
drive_quality_green	Number of green flags	
drive_quality_red	Number of red flags	
drive_quality_yellow	Number of yellow flags	
yellow_threshold		M/s ²
red_threshold		M/s ²
sampling_rate		Milliseconds

acc_threshold		M/s ²
action		

Driving indices

Safety Index

$$IS = 0.5 * \sin\left(\frac{Velocidade * Inclinao}{140 * Estrada} + \frac{\pi}{2}\right) + (0.5 - Tempo - AccelX - AccelY)$$

Economy Index

$$IE = \frac{-(Velocidade - Eco^2) + Eco^2}{((Eco + Eco * (Inclinao + AccelY))^2)}$$

Events Detection

With the values of the acceleration of the vehicle we can detect the most common driving events that result in bad safety and economy.

Algorithm 1 Algoritmo de avaliação de eventos

GPSAccel = Aceleração obtida pelo GPS

Ax = Valor máximo da aceleração lateral

AxThreshold = Valor mínimo de aceleração lateral que origina evento

Ay = Valor máximo de aceleração na direção do movimento

AyAccel = Aceleração mínima direção movimento origina aceleração brusca

AyBrake = Aceleração mínima direção movimento origina travagem

MaxCourse = Valor máximo de variação do *course*

SpeedThreshold = Valor mínimo de velocidade para produzir eventos

CornerThreshold = Valor mínimo de variação do *course* para se considera curva

if (Instant Speed > SpeedThreshold) **then**

if (MaxCourse > CornerThreshold) **then**

if (Ax > AxThreshold && Ax > Ay) **then**

return Evento de curva brusca

end if

end if

if (MaxCourse < CornerThreshold) **then**

if (Ay > AyAccel && Ay > Ax && GPSAccel > AyAccel) **then**

return Evento de aceleração brusca

end if

if (Ay < AyBrake && Ay > Ax && GPSAccel > AyBrake) **then**

return Evento de travagem brusca

end if

end if

end if
