



Relatório de Estágio

Mestrado em Engenharia Informática - Computação Móvel

Mobile Fleet Management

João Miguel Henriques Domingos

Leiria, *setembro* de 2015



Relatório de Estágio

Mestrado em Engenharia Informática - Computação Móvel

Mobile Fleet Management

João Miguel Henriques Domingos

Estágio de Mestrado realizado sob a orientação do Doutor Vítor Manuel Basto Fernandes,
Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, *setembro* de 2015

À Minha Família

Agradecimentos

Ao longo da unidade curricular de Estágio do Mestrado em Engenharia Informática - Computação Móvel, tive o apoio de várias pessoas que contribuíram para a respetiva preparação, organização e redação deste documento.

Em primeiro lugar, gostaria de agradecer a todos aqueles que com o seu saber, colaboração e apoio crítico me ajudaram, direta ou indiretamente, no decorrer deste estágio.

Queria agradecer a orientação, colaboração e contribuição que o professor Vítor Basto Fernandes me disponibilizou, tanto durante a elaboração do presente documento, como do artigo científico redigido no âmbito do projeto Mobile Fleet Management.

Queria deixar uma palavra de agradecimento aos colaboradores da empresa Tecmic S.A. que colaboraram comigo ao longo deste período, quer a nível laboral, na resolução de problemas ou validação dos mesmos, quer a nível social na integração e bom convívio tanto no local de trabalho, como fora do mesmo.

Estou especialmente agradecido ao Pedro Marques, da Tecmic S.A. por acreditar em mim para desenvolver este projeto tão motivante e pelo apoio disponibilizado. Também um agradecimento muito especial ao David Mendes pela motivação, pelo incentivo, compreensão, encorajamento, paciência e valiosa orientação que se tornou essencial ao sucesso deste período de estágio.

Um agradecimento muito especial ao senhor Jorge Domingos e à senhora Maria da Conceição Henriques, meus pais, que contribuíram com um grande esforço para me proporcionar as melhores condições de modo a alcançar o sucesso desejado.

Por último, mas não menos importante, o meu muito obrigado à Cátia Sousa. Obrigado pelo apoio que nunca faltou, como pela compreensão e paciência que teve quando não podia estar presente devido à redação deste documento. Muito Obrigado!

Resumo

Este documento foi elaborado para dar a conhecer o trabalho realizado no âmbito do projeto Mobile Fleet Management. Este projeto foi proposto pela empresa Tecmic - Tecnologias de Microeletrónica, S.A. e insere-se na unidade curricular de Estágio do Mestrado em Engenharia Informática - Computação Móvel (MEI-CM).

O objetivo deste projeto consistiu no desenvolvimento de uma aplicação móvel para utilização de operadores de frota, apelidada de iZiTraN Mobile.

A aplicação permite aos operadores de frota efetuar, a qualquer hora e em qualquer lugar, uma gestão remota de todos os veículos da sua frota, tendo disponível para a tomada de decisões informação em tempo real. Os operadores de frota podem consultar informações referentes a veículos, contactar os seus condutores pela troca de mensagens, receber alertas dos mais variados incidentes que possam surgir e tratar dos mesmos, tudo em tempo real e num dispositivo móvel.

A aplicação iZiTraN Mobile tem uma forte componente tecnológica que recorre a várias tecnologias emergentes, tendo sido desenhada para ser modular, com camadas independentes e separadas de forma a facilitar futuras manutenções e atualizações de *software*. As opções de desenho e implementação seguidas permitiram reduzir de forma considerável o custo de desenvolvimento, o futuro custo de manutenção e de exploração deste produto da Tecmic S.A.. Este produto, posiciona-se dentro do portfólio de produtos da empresa como um dos produtos mais inovadores e com maior potencial de crescimento nesta área de negócio do setor da gestão de frotas de veículos.

Palavras-chave: Aplicação Móvel, Gestão de Frotas, Tempo Real, Mensagens, Alarmes.

Abstract

This document describes the work done under the Mobile Fleet Management project. This project was proposed by Tecmic Microelectronics Technologies S.A., as the internship component of the Master in Computer Engineering - Mobile Computing.

The objective of this project was to develop a mobile application for fleet operators, named iZiTraN Mobile.

The application allows fleet operators for the remote management of all vehicles at anytime, anywhere, with real time information support. Fleet operators can access information about vehicles, contact drivers by messaging, receive various incidents alerts that may arise and dealing with them, all in real time and on a mobile device mobile device operating in an anytime, anywhere paradigm.

The application iZiTraN Mobile adopts several emerging technology and technological trends, has been designed to be modular, following a layering approach, with separated and independent layers, in order to facilitate future software upgrades and maintenance. Design and implementation options have reduced considerably development, future maintenance and exploration costs of this software product, which is positioned among the most innovative and promising products in Tecmic's portfolio, in this business domain.

Key-Words: Mobile application, Fleet Management, Real Time, Messages, Alarms.

Índice de Figuras

Figura 1 - Visualização da frota no mapa [6].....	9
Figura 2 - Relatórios gerais da frota [6].....	10
Figura 3 - <i>Chat</i> de Mensagens (adaptação da Figura 1).....	11
Figura 4 - Gestor de Notificações.....	12
Figura 5 - Arquitetura iZiTraN Web [6].....	13
Figura 6 - Aplicação Telogis Coach apresentando o quadro de liderança [12] e o quadro individual de cada condutor [12].	15
Figura 7 - Aplicação Telogis Supervisor [14].....	16
Figura 8 - Aplicação Telogis WorkPlan [15].....	17
Figura 9 - Aplicação Fleetmatics Reveal apresentando mapa e alerta [18].....	18
Figura 10 - A Aplicação Fleetmatics Reveal apresentando o histórico diário e a distância percorrida [18].	19
Figura 11 - Aplicação Fleetmatics Reveal LogBook apresentando os tempos de condução e descanso dos condutores [20].	20
Figura 12 - Aplicação Scania Fleet Management apresentando o mapa e as mensagens [23].	21
Figura 13 - Fluxograma com o fluxo de desenvolvimento [37].....	26
Figura 14 - Cronograma do projeto Mobile Fleet Management.....	29
Figura 15 - Arquitetura do sistema informático iZiTraN antes do projeto.....	42
Figura 16 - Arquitetura do sistema informático iZiTraN com o Mobile Fleet Management integrado.....	43
Figura 17 - Diagrama do modelo de domínio.....	44
Figura 18 - Entidade Account.	45

Figura 19 - Entidade AccountInfo.....	45
Figura 20 - Entidade Fleet.....	46
Figura 21 - Entidades Vehicle, VehicleGroup, VehicleDailySnapshotEvents e VehicleSnapshotEvent.	46
Figura 22 - Entidade Location.....	47
Figura 23 - Entidade Travel.....	47
Figura 24 - Entidade TravelDailyRoutes e TravelRoute.....	47
Figura 25 - Entidade PanicScenario.....	48
Figura 26 - Entidade Alarm e AlarmHigh.....	48
Figura 27 - Entidade Conversation.....	49
Figura 28 - Entidade TextMessage, TextMessageSend e TextMessageReceive.....	49
Figura 29 - Enquadramento arquitetural do iZiTraN Mobile Services.....	50
Figura 30 - Arquitectura da aplicação servidora.....	51
Figura 31 - Fluxo de segurança utilizando filtro de autenticação.....	55
Figura 32 - Arquitectura organizada por <i>packages</i>	56
Figura 33 - Módulos da arquitetura iZiTraN Mobile.....	58
Figura 34 - Adição do módulo Logging à arquitetura iZiTraN Mobile.....	60
Figura 35 - Barra de Navegação com protótipo inicial (esquerda) e final (direita).....	61
Figura 36 - Ecrã Principal dos Veículos com protótipo inisial (esquerda) e final (direita).....	62
Figura 37 - Ecrã Tempo Real de um veículo com protótipo inicial (esquerda) e final (direita).....	63
Figura 38 - Ecrã Viagens de um veículo com protótipo inicial (esquerda) e final (direira).....	64
Figura 39 - Ecrã Alarmes de um veículo com protótipo inicial (esquerda) e final (direira).....	65
Figura 40 - Ecrã Mensagens de um veículo com protótipo inicial (esquerda) e final (direira).....	66
Figura 41 - Diretriz de ícones [125].....	67
Figura 42 - Diretriz de estilo de escrita [128].....	68
Figura 43 - Exemplo de mensagem com estilo de escrita.....	68
Figura 44 - Diretriz de espaçamento de componentes [129].....	69

Figura 45 - Exemplo de espaçamento entre componentes.	69
Figura 46 - Diretriz de navegação padrão [130].	70
Figura 47 - Exemplo da diretriz de navegação padrão utilizada na aplicação.	70
Figura 48 - Diretriz de navegação por painel [131].	71
Figura 49 - Exemplo da diretriz de navegação por painel utilizada na aplicação.	71
Figura 50 - Diretriz de notificação de confirmação [132].	72
Figura 51 - Exemplo de notificação de confirmação utilizada na aplicação.	72
Figura 52 - Teste de integração assíncrono para envio de mensagem e resultado do mesmo.	74
Figura 53 - Teste de carga assíncrono para pedido da lista de veículos e resultado do mesmo.	75
Figura 54 - Comparativo de melhoria de tempos médios obtidos pelos testes de carga.	76
Figura 55 - Teste de segurança para pedido da lista de veículos e resultado do mesmo.	77
Figura 56 - Gráfico de Gantt com as principais tarefas.	97
Figura 57 - Ecrã Login.	98
Figura 58 - Ecrã Painel Lateral.	98
Figura 59 - Painel Lateral com dados.	98
Figura 60 - Opções da Aplicação.	98
Figura 61 - Ecrã Preferências.	99
Figura 62 - Ecrã Sobre a Aplicação.	99
Figura 63 - Ecrã Terminar Sessão.	99
Figura 64 - Mensagem de Terminar Sessão.	99
Figura 65 - Ecrã Real Time Geral.	100
Figura 66 - Ecrã de Veículos.	100
Figura 67 - Ecrã Seguir.	100
Figura 68 - Ecrã Diário.	100
Figura 69 - Ecrã Viagens.	101
Figura 70 - Ecrã Alarmes.	101
Figura 71 - Opções sobre Alarme.	101

Figura 72 - Ecrã Eliminar Alarme.	101
Figura 73 - Ecrã Notificação de Novo Alarme.....	102
Figura 74 - <i>Chat</i> de Resposta a Alarme.	102
Figura 75 - <i>Chat</i> de Resposta a Alarme com teclado.....	102
Figura 76 - Ecrã Mensagens.....	102
Figura 77 - Ecrã Escolha Veículo para Nova Mensagem.....	103
Figura 78 - Opções Sobre Mensagem.....	103
Figura 79 - Ecrã Eliminar Mensagem.....	103
Figura 80 - Notificação de Nova Mensagem.	103
Figura 81 - <i>Chat</i> de Mensagens.	104
Figura 82 - Ecrã Login.	105
Figura 83 - Ecrã Painel Lateral.	105
Figura 84 - Ecrã Painel Lateral com dados.....	105
Figura 85 - Opções da Aplicação.	105
Figura 86 - Ecrã Preferências.....	106
Figura 87 - Ecrã Sobre a Aplicação.	106
Figura 88 - Ecrã Terminar Sessão.	106
Figura 89 - Mensagem de Terminar Sessão.	106
Figura 90 - Ecrã Tempo Real Geral.	107
Figura 91 - Ecrã Veículos.	107
Figura 92 - Ecrã Seguir.....	107
Figura 93 - Ecrã Diário.	107
Figura 94 - Ecrã Viagens.	108
Figura 95 - Ecrã Alarmes.....	108
Figura 96 - Opções sobre Alarme.	108
Figura 97 - Ecrã Eliminar Alarme.	108
Figura 98 - Notificação de Novo Alarme.	109

Figura 99 - <i>Chat</i> de Resposta a Alarme.	109
Figura 100 - <i>Chat</i> de Resposta a Alarme com teclado.	109
Figura 101 - Ecrã Mensagens.	109
Figura 102 - Ecrã Escolha Veículo para Nova Mensagem.	110
Figura 103 - Opções sobre Mensagem.....	110
Figura 104 - Ecrã Eliminar Mensagem.....	110
Figura 105 - Notificação de Nova Mensagem.	110
Figura 106 - <i>Chat</i> de Mensagens.....	111
Figura 107 - Ecrã Login.	112
Figura 108 - Ecrã Login com Aguarde.....	112
Figura 109 - Ecrã Principal.....	112
Figura 110 - Ecrã Painel Lateral com dados.	112
Figura 111 - Ecrã Definições.	113
Figura 112 - Ecrã Definições de Mapa.....	113
Figura 113 - Ecrã Definições de mensagens.....	113
Figura 114 - Ecrã Sobre a Aplicação.	113
Figura 115 - Ecrã Terminar Sessão.	114
Figura 116 - Ecrã Tempo Real Geral.	114
Figura 117 - Ecrã Frota de veículos.	114
Figura 118 - Ecrã Pesquisar.	114
Figura 119 - Ecrã Pesquisar com veículos.	115
Figura 120 - Ecrã Pesquisar por filtro.	115
Figura 121 - Ecrã Seguir Veículo.....	115
Figura 122 - Ecrã Diário de Veículo.	115
Figura 123 - Ecrã Viagens Geral.....	116
Figura 124 - Ecrã Viagens - Menu de Contexto.....	116
Figura 125 - Ecrã Viagens - Seleção de Dia.....	116

Figura 126 - Ecrã Viagem.....	116
Figura 127 - Ecrã Viagem - Seleção de Dia.	117
Figura 128 - Ecrã Viagem - Detalhe Global.....	117
Figura 129 - Ecrã Viagem - Detalhe de Percurso.	117
Figura 130 - Ecrã Alarmes.....	117
Figura 131 - Ecrã Alarmes - Menu de Contexto.	118
Figura 132 - Ecrã Alarmes do Veículo.....	118
Figura 133 - Ecrã Alarme - Detalhe do Alarme.	118
Figura 134 - Ecrã Mensagens.....	118
Figura 135 - Ecrã Mensagens - Menu de Contexto.	119
Figura 136 - Ecrã <i>Chat</i> de Mensagens.	119
Figura 137 - Testes de Usabilidade da primeira fase apresentando o número de erros em cada tarefa.	123
Figura 138 - Testes de Usabilidade da segunda fase apresentando o número de erros em cada tarefa.	123
Figura 139 - Protótipo inicial para os Tempos de Condução.....	134

Índice de Tabelas

Tabela 1 - Produtos da Tecmic S.A.....	3
Tabela 2 - Recursos humanos e perfil de cada recurso.....	33
Tabela 3 - Recursos materiais e sua descrição.....	33
Tabela 4 - Comunicação entre a equipa de trabalho.....	34
Tabela 5 - Lista de serviços disponibilizados pelo iZiTraN Mobile Service.....	52
Tabela 6 - Lista de serviços disponibilizados pelo iZiTraN Mobile RealTime.....	53
Tabela 7 - Informação recolhida pelo registo de canais utilizando o módulo Logging.....	74
Tabela 8 - Dados médios obtidos do pedido e resposta para obtenção da lista de veículos na aplicação.....	76
Tabela 9 - User Stories.....	94
Tabela 10 - Matriz de Wiegers.....	96
Tabela 11 - Testes Unitários apresentando o resultado obtido.....	121
Tabela 12 - Guião de Tarefas para os Testes de Usabilidade.....	122

Lista de Siglas

O presente documento faz uso frequente de siglas, cujo significado está tipicamente apresentado aquando da sua primeira utilização. Na lista seguinte são apresentadas, por ordem alfabética, as siglas presentes no documento e o seu real significado.

Sigla	Significado
API	Application Programming Interface
DAL	Data Access Layer
DP	Density-independent Pixel
EDA	Event-Driven Architecture
ESTG	Escola Superior de Tecnologia e Gestão de Leiria
GPRS	General Packet Radio Service
GPS	Global Positioning System
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IDC	International Data Corporation
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IoT	Internet of Things
IPL	Instituto Politécnico de Leiria
JSON	JavaScript Object Notation
MEI-CM	Mestrado em Engenharia Informática - Computação Móvel
MVC	Model-View-Controller
MVP	Model-View-Presenter
PDA	Personal Digital Assistant
REST	Representational State Transfer

RPC	Remote Procedure Call
SMS	Short Message Service
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SVN	Apache Subversion
TMS	Tile Map Service
US	User Story
W3C	World Wide Web Consortium
WCF	Windows Communication Foundation
WMS	Web Map Service
XML	eXtensible Markup Language
XP	eXtreme Programming

Índice

INTRODUÇÃO	1
1.1 ENTIDADE DE ACOLHIMENTO	2
1.2 ENQUADRAMENTO.....	3
1.3 MOTIVAÇÃO E OBJETIVOS	4
1.4 ESTRUTURA DO DOCUMENTO	4
ESTADO DA ARTE	7
2.1 IZITRAN WEB	7
2.1.1 FUNCIONALIDADES	8
2.1.1.1 TEMPO REAL	8
2.1.1.2 RELATÓRIOS.....	9
2.1.1.3 MENSAGENS	10
2.1.1.4 ALARMÍSTICAS	11
2.1.2 FUNCIONAMENTO.....	12
2.2 APLICAÇÕES MÓVEIS CONCORRENTES.....	13
2.2.1 TELOGIS.....	13
2.2.1.1 TELOGIS COACH	14
2.2.1.2 TELOGIS SUPERVISOR.....	15
2.2.1.3 TELOGIS WORKPLAN	16
2.2.2 FLEETMATICS.....	17
2.2.2.1 FLEETMATICS REVEAL.....	17
2.2.2.2 FLEETMATICS REVEAL LOGBOOK.....	19
2.2.3 SCANIA	20
2.2.3.1 SCANIA FLEET MANAGEMENT	21
2.2.4 OUTRAS APLICAÇÕES	22
2.2.5 COMPARAÇÃO ENTRE APLICAÇÕES	22
2.3 IZITRAN MOBILE	23
METODOLOGIA E GESTÃO DE PROJETO	25
3.1 METODOLOGIA.....	25
3.2 GESTÃO DE PROJETO	28

3.2.1	DESCRIÇÃO DO ÂMBITO	28
3.2.2	GESTÃO DE TEMPO	29
3.2.3	GESTÃO DE RISCO	31
3.2.4	GESTÃO DE RECURSOS.....	32
3.2.5	GESTÃO DA COMUNICAÇÃO	33
	DESENVOLVIMENTO.....	35
4.1	ENQUADRAMENTO TECNOLÓGICO	35
4.1.1	REPOSITÓRIOS	36
4.1.2	COMUNICAÇÃO	37
4.1.3	MAPAS	40
4.2	ENQUADRAMENTO ARQUITETURAL DO IZITRAN	41
4.3	MODELO DE DOMÍNIO	43
4.4	SERVIDOR	49
4.4.1	ARQUITETURA	50
4.4.2	IZITRAN MOBILE SERVICE	52
4.4.3	IZITRAN MOBILE REALTIME	53
4.4.4	SEGURANÇA	54
4.5	APLICAÇÃO MÓVEL	55
4.5.1	IZITRAN MOBILE	55
4.5.1.1	ARQUITETURA	55
4.5.1.1.1	SEPARAÇÃO DA LÓGICA DE APRESENTAÇÃO E DE NEGÓCIO	56
4.5.1.1.2	ARQUITETURA MODULAR	57
4.5.1.2	INTERFACE GRÁFICA	61
4.5.1.3	DIRETRIZES PARA DESENHO DE INTERFACES PARA A PLATAFORMA ANDROID.....	66
	TESTES.....	73
5.1	TESTES REALIZADOS	73
5.1.1	TESTES UNITÁRIOS.....	73
5.1.2	TESTE DE INTEGRAÇÃO	73
5.1.3	TESTE DE CARGA	74
5.1.4	TESTE DE SEGURANÇA	76
5.1.5	TESTES DE USABILIDADE	77
5.1.5.1	PRIMEIRA FASE.....	77
5.1.5.2	SEGUNDA FASE.....	78
5.2	DISCUSSÃO DE RESULTADOS.....	78
	CONCLUSÕES E TRABALHO FUTURO.....	79
6.1	CONCLUSÕES	79
6.2	TRABALHO FUTURO	80
	REFERÊNCIAS.....	81

APÊNDICES.....	91
APÊNDICE I - USER STORIES.....	92
APÊNDICE II - MATRIZ DE WIEGERS.....	95
APÊNDICE III - GRÁFICO DE GANTT.....	97
APÊNDICE IV - PROTÓTIPOS INICIAIS.....	98
APÊNDICE V - PROTÓTIPOS INTERMÉDIOS.....	105
APÊNDICE VI - ECRÃS FINAIS.....	112
APÊNDICE VII - TESTES UNITÁRIOS.....	120
APÊNDICE VIII - GUIÃO DE TAREFAS.....	122
APÊNDICE IX - TESTES DE USABILIDADE.....	123
APÊNDICE X - ARTIGO CIENTÍFICO.....	124
ANEXOS.....	133
ANEXO I - PROJETO TEMPOS DE CONDUÇÃO.....	134
ANEXO II - PROJETO AVISTAMENTOS.....	135
ANEXO III - PROJETO RENT-A-CAR.....	136
ANEXO IV - PROJETO CONSOLA GRÁFICA PARA TRANSPORTES.....	137

Introdução

Este documento foi elaborado no âmbito da unidade curricular de Estágio que pertence ao último ano do Mestrado em Engenharia Informática - Computação Móvel (MEI-CM), lecionado na Escola Superior de Tecnologia e Gestão de Leiria (ESTG), pertencente ao Instituto Politécnico de Leiria (IPL).

A entidade de acolhimento onde foi realizado o estágio foi a Tecmic - Tecnologias de Microeletrónica, S.A. [1]. Tendo o estágio decorrido num contexto empresarial, foi necessária a existência de um período de adaptação compreendido entre 15 de setembro a 14 de novembro, com a finalidade de integração com a empresa. Neste período não só foram ministradas várias ações de formação internas acerca dos vários produtos, como também se realizou a integração com a equipa de desenvolvimento, nas principais instalações da empresa, no Taguspark em Oeiras, Lisboa. Após este período, o estágio decorreu, entre 17 de novembro a 19 de junho, nas instalações secundárias da empresa, na Incubadora D. Dinis, em Leiria. Neste segundo período de estágio procedeu-se ao desenvolvimento do projeto Mobile Fleet Management.

O produto desenvolvido no âmbito deste projeto estende o produto iZiTraN Web para as plataformas móveis de forma a incluir um conjunto de funcionalidades importantes, adaptadas para a realidade dos dispositivos móveis, tendo como utilizador final os operadores de frota de veículos. O projeto divide-se em duas componentes técnicas: componente de servidor e componente cliente móvel. A componente de servidor consiste na criação de dois Web services (iZiTraN Mobile Service e iZiTraN Mobile RealTime) que disponibilizam um conjunto de serviços de acesso e controlo de dados relacionados com frotas de veículos. Estes dados são consumidos pela componente cliente móvel, uma aplicação móvel apelidada de iZiTraN Mobile. A aplicação permite visualizar, em tempo real (período de tempo compreendido entre 0 e 3 minutos, dependendo da prioridade da informação), a posição precisa de veículos num mapa e informação dos mesmos, como dados pormenorizados dos tempos de condução, quilómetros percorridos, consumos de combustível, visualização de

histórico de viagens do veículo, entre outros. O iZiTraN Mobile permite também aos operadores de frota contactar os seus condutores pela troca de mensagens, receber alertas dos mais variados incidentes que possam surgir e tratar dos mesmos. Todas estas funcionalidades se tornam possíveis devido à constante comunicação entre a aplicação e os Web services que disponibilizam dados enviados pelo equipamento Tecmic (caixa colocada no veículo para recolha de informação de estado, posição geográfica, etc.), presente nos veículos. Esta constante comunicação permite que tudo seja tratado em tempo real e num dispositivo móvel, cujas funcionalidades avançadas são imensas e a sua utilização tem a grande vantagem de poder ser acessível a partir de qualquer lugar.

Este documento surge como suporte teórico ao estágio realizado, tendo como objetivo dar a conhecer ao leitor os principais tópicos do projeto desenvolvido e as decisões tomadas.

1.1 Entidade de Acolhimento

A entidade de acolhimento do estágio foi a Tecmic - Tecnologias de Microeletrónica, S.A. [1], empresa multinacional portuguesa fundada em 1988, cuja especialização começou pelos equipamentos de *hardware* e comunicações. No entanto, nunca perdendo esta valência foi também adquirindo conhecimento na área do *software* através do desenvolvimento de aplicações *desktop*, assim como de aplicações móveis para consolas/terminais de navegação por Global Positioning System (GPS), passando finalmente para as aplicações *web*. A empresa conseguiu com esta estratégia um crescimento sustentado ao longo de mais de 25 anos de existência, contando em 2015 com cerca de quarenta colaboradores.

A empresa opera no setor das tecnologias da informação e serviços, sendo que a sua área principal de negócio é a gestão de frotas de veículos, que permite aos seus clientes aumentar a produtividade e eficiência das suas operações. Possui também soluções em áreas complementares de negócio, desde *hardware* e *software*, sistemas embarcados, gestão de logística, gestão de recursos remotos, gestão de controlo de entrada e saída, gestão de sistemas de monitorização remota e telemetria, gestão de emergências e proteção civil, passando pela área da saúde, entre outras [2]. A Tabela 1 enumera os principais produtos da empresa Tecmic S.A. [3].

Produto	Área de aplicação
XTraN	Gestão Profissional de Frotas
XTraN Passenger	Sistema de Suporte Operacional para Transporte de Passageiros
iZiTraN	Gestão de Frotas <i>Online</i>
iZiGPS	Localização de Veículos

4Forces	Sistema Integrado de Gestão de Emergências
Ecogest	Gestão de Ecopontos e Contentores
Siga	Gestão de Acessos
Simor	Monitorização Remota de Equipamentos

Tabela 1 - Produtos da Tecmic S.A..

A família de produtos disponibilizada, conta com soluções verticais para cada área de atividade específica, realizando neste momento a gestão de frotas de organizações líderes nos diferentes sectores: transportes de mercadorias e logística, transportes de passageiros, forças de segurança pública, proteção civil e emergência, segurança privada, assistência técnica, construção civil, recolha de resíduos, entre outros. Estes produtos já deram à empresa um vasto historial de prémios, entre os quais o prémio de Melhor Empresa Nacional, atribuído pela Tecparques - Associação Portuguesa de Parques de Ciência e Tecnologia na II Gala Anual do Empreendedorismo em Ciência e Tecnologia [4].

No que diz respeito a parceiros, a Tecmic S.A. conta com entidades como Vodafone, Motorola, Tele Atlas, Link, INOV, entre outros. A área de influência da Tecmic S.A. estende-se a vários países como Espanha, Suíça, Alemanha, Áustria, Irlanda, Marrocos, Angola e, mais recentemente, o Brasil. São vários os clientes dos produtos Tecmic S.A., de entre os quais se destacam empresas como Alliance Healthcare, Carris, Correios de Portugal (CTT), Infraestruturas de Portugal (IP), Portugal Telecom (PT), Luís Simões e Unilever. Para além de empresas destacam-se instituições como Assembleia da República, Automóvel Club de Portugal (ACP), Banco de Portugal, Exército Português, entre outras.

1.2 Enquadramento

O trabalho de estágio descrito no presente documento consiste genericamente no desenvolvimento de uma aplicação móvel para gestão de frotas. De um modo abrangente, foi pretendido a criação de um novo produto para maximizar o principal setor de negócio da empresa, a gestão de frotas de veículos. Tradicionalmente, uma frota ou conjunto de veículos é gerida por um operador de frota. As frotas são, normalmente, divididas em grupos, sendo cada um gerido por um ou vários operadores de frota, tornando-se estes os principais utilizadores da aplicação móvel.

O projeto desenvolvido procurou ir de encontro às necessidades da empresa e por conseguinte, ir de encontro às necessidades dos próprios clientes que insistiam numa aplicação móvel. Com este projeto, a Tecmic S.A. pode disponibilizar um novo produto vocacionado para dispositivos móveis que satisfaça as necessidades dos seus clientes e

cumpra determinados procedimentos exigidos por parte dos mesmos, que sentiam a necessidade de poderem acompanhar, em tempo real, todos os veículos da sua frota. Essa necessidade é satisfeita pelas vantagens que os dispositivos móveis oferecem, devido à sua utilização em qualquer hora e em qualquer lugar [5], podendo estes fazer ações automáticas sem interação do utilizador, tornando estes dispositivos imersos numa realidade humana.

1.3 Motivação e Objetivos

A motivação para este projeto nasceu naturalmente, do lado da empresa que assim passou a disponibilizar um produto inovador na forma como os seus clientes interagem com os veículos. O produto dá às empresas uma visão mais completa do que acontece no campo, concebendo assim conhecimento que pode ser usado posteriormente para melhorar os seus negócios, permitindo-lhes reduzir os custos operacionais e de capital, bem como o aumento de receita.

O aparecimento desta aplicação móvel na Tecmic S.A. surge num momento em que existe uma abertura de mercado para a utilização de aplicações nos dispositivos móveis, área com imenso potencial para ser explorado pela empresa. Do lado pessoal a motivação existe por se tratar de um projeto de raiz, onde foram aplicadas tecnologias inovadoras, como também pela área dos dispositivos móveis, área em que existe um particular interesse. Um outro aspeto motivacional e que exige um cuidado redobrado é a existência do desenvolvimento de uma aplicação que preenche necessidades reais dos clientes da empresa, como tal prevê-se que este projeto seja frequentemente utilizado, criando uma motivação extra.

No que respeita a objetivos, estes surgiram naturalmente ao longo do projeto, tendo-se iniciado por um período de adaptação à empresa, conhecendo-a gradualmente, bem como aos seus produtos e área de negócio. Após este período houve a necessidade de criar o planeamento do projeto, onde se estipulou tarefas e se definiu prazos para entregas que se tentou cumprir com o máximo rigor. Em relação ao desenvolvimento da aplicação móvel, o objetivo principal foi de a implementar de forma modular e interoperável, por forma a minimizar os custos futuros com atualizações da aplicação.

1.4 Estrutura do Documento

Esta secção descreve a organização do documento. Os restantes capítulos deste documento estão organizados do seguinte modo: No Capítulo 2 é descrito o produto iZiTraN Web cujas funcionalidades serviram como base para a aplicação móvel iZiTraN Mobile. É feita ainda

uma apresentação das aplicações concorrentes de gestão de frotas para dispositivos móveis, finalizando o capítulo com a explicação das funcionalidades da aplicação iZiTraN Mobile; No Capítulo 3 encontra-se descrita a metodologia utilizada para o desenvolvimento do projeto, apresentando os pontos fortes da mesma e as adaptações que foram feitas ao longo do seu uso. É ainda descrita a gestão do projeto nas suas várias dimensões, o âmbito, a gestão do tempo, a gestão de risco, a gestão de recursos e a gestão da comunicação; O Capítulo 4 descreve em pormenor o desenvolvimento realizado, começando por explicar as opções tecnológicas e a arquitetura do sistema informático existente, passando para o modelo de domínio e implementação das aplicações cliente e servidor; No Capítulo 5 encontram-se descritos os testes efetuados e são discutidos os respetivos resultados; Por fim, no Capítulo 6 são apresentadas as conclusões e são tecidas considerações a ter em conta em trabalho futuro.

Estado da Arte

Este capítulo visa dar a conhecer o produto iZiTraN Web, apresentando os seus benefícios para quem o utiliza (operador de frota) e as suas funcionalidades, tendo as mesmas sido extraídas para formarem a base da aplicação móvel iZiTraN Mobile. O capítulo faz ainda uma apresentação das aplicações concorrentes de gestão de frotas para dispositivos móveis e termina com a explicação da origem das funcionalidades adquiridas pela aplicação iZiTraN Mobile.

2.1 iZiTraN Web

O iZiTraN Web [6] é um sistema profissional *web* para gestão de frotas que permite o controlo e acompanhamento em tempo real de todos os veículos de uma frota. Fruto de uma parceria entre a Tecmic S.A. [1] e a Vodafone Portugal [7], é uma solução ideal para qualquer empresa, de qualquer dimensão, cuja atividade pressuponha a gestão e monitorização de frotas de veículos.

Este produto é muito simples de utilizar, a partir de qualquer *Web browser*, permitindo aceder a informação em tempo real sobre as frotas (v.g localização dos veículos), a custo fixo, bem como a elaboração e exportação de relatórios de planeamento, operação e históricos. No entanto, não está preparado para dispositivos móveis, devido aos ecrãs de pequenas dimensões destes dispositivos.

Através da utilização do iZiTraN Web as empresas beneficiam de um aumento da produtividade e eficiência da frota, porque disponibiliza aos clientes ferramentas de planeamento da atividade, otimização de rotas, notificações em tempo real sobre o estado do veículo e tomada de decisões em tempo real. Alguns dos benefícios deste produto incluem encurtar tempos de resposta enviando o veículo mais próximo para os locais onde são necessários, o cumprimento de horários com atuações em tempo real face a imprevistos, entre outros. Como resultado, estes benefícios traduzem-se na redução dos custos operacionais, tal

como a redução do consumo de combustível, redução de quilómetros percorridos, redução do tempo desperdiçado em locais de carga e descarga, redução do tempo de inatividade dos veículos e ainda a redução de custos com comunicações, uma vez que este produto simplifica as comunicações e substitui chamadas telefónicas (entre operador de frota e condutor) por informação obtida de forma automática.

Concluindo, este produto permite gerir toda a atividade de uma frota, através de um conjunto bastante completo de funcionalidades que são descritas de seguida.

2.1.1 Funcionalidades

As funcionalidades do iZiTraN Web são muitas e estão em constante adição, sendo englobadas em quatro módulos principais: Tempo Real, Relatórios, Mensagens e Alarmísticas. Com o Tempo Real é possível visualizar a localização de todos os veículos em mapa, existindo também a possibilidade da visualização dos indicadores de cada veículo, mostrando os quilómetros percorridos, velocidade instantânea, litros no depósito, entre outros indicadores. Através dos Relatórios é possível obter um conjunto de informações detalhadas que permitem a auditoria e *reporting* de toda a atividade desenvolvida pelos veículos. Com as Mensagens é permitida a troca de mensagens entre operadores de frota e condutores, bem como acesso ao histórico dessas interações. Já com as Alarmísticas existe a possibilidade da definição de alguns eventos (parametrizáveis) que podem ser desencadeados quando se verificam situações que interessa monitorizar de forma instantânea, bem como as formas como devem ser tratados esses eventos. Os módulos descritos são apresentados com mais detalhe nas próximas secções do presente documento.

2.1.1.1 Tempo Real

O iZiTraN Web dispõe de um vasto conjunto de funcionalidades e indicadores em tempo real sobre os veículos, como a sua localização, velocidade, quilómetros percorridos, estado atual (desligado, parado, em trânsito, em movimento), entre outros. Estes podem apresentar informações mais detalhadas através da visualização pela forma tabular, mas também pela visualização em mapa.

A Tecmic S.A. tem uma ampla cobertura de cartografia proprietária que permite apresentar a informação recolhida em tempo real através da visualização da localização dos veículos no mapa permitindo uma melhor orientação da localização do veículo. Com os mapas o operador de frota tem um detalhe mais preciso da localização do veículo, podendo visualizar o percurso

atual e todos os percursos realizados pelos veículos, ajudando à identificação dos locais de início e fim de atividade de toda a frota. Com a cartografia usada no iZiTraN Web também existe a possibilidade de marcação de balizas (área geométrica definida pelo operador de frota) no mapa, como por exemplo, marcação de clientes, armazéns, zonas a evitar, entre outros. A Figura 1 apresenta a visualização no mapa de toda a frota presente na área geográfica selecionada e ainda, os indicadores atuais do veículo selecionado (45-CA).



Figura 1 - Visualização da frota no mapa [6].

Na Figura 1 é possível observar o mapa ao centro com a identificação de todos os veículos presentes na área selecionada. Na parte inferior do mapa são apresentados indicadores do veículo selecionado (45-CA), sendo que na esquerda o operador de frota tem sempre uma listagem de todos os veículos da frota, organizada por grupos para que facilmente o utilizador possa seleccionar o veículo que deseja visualizar.

2.1.1.2 Relatórios

O iZiTraN Web tem um vasto conjunto de relatórios que permitem a auditoria e *reporting* de toda a atividade desenvolvida pelos veículos, podendo estes ser úteis para controlo de manutenção, de gastos, como o controlo de abastecimentos de combustível, entre outros.

Os relatórios podem ser apresentados por frota, com indicadores gerais do estado dos

veículos, quilómetros percorridos, por dia, por semana, entre outros. Podem ainda ser apresentados por condutor, para uma mais fácil análise da produtividade dos condutores, mas também podem ser apresentados por data e hora, por viagem com identificação de todos os percursos realizados, por veículo, por dia, entre outros, tendo um resumo dos parâmetros principais de cada viagem (tempo condução, distância percorrida, velocidade média, tempo parado). Também existem relatórios específicos por veículo, com informações data/hora de entrada, saída e duração em balizas.

Como se constata existe a possibilidade da visualização de diferentes relatórios, com as mais variadas informações relevantes para corresponder às especificidades da gestão da frota. Na Figura 2 é apresentado um dos modos de consulta de relatórios presentes no iZiTraN Web.

Relatórios

Configurações

Visitas a Locais - Viaturas: 01-12-2013 00:00 - 31-12-2013 23:59

Descrição	Código	Nome	Duração	Visitas
Viatura Estafeta	4009	Estafeta		
Totais do Mês Dezembro 2013	4009	Estafeta	32:02:07	19
Totais da Viatura Estafeta	4009	Estafeta	32:02:07	19
Viatura PJS	4010	PJS		
Totais do Mês Dezembro 2013	4010	PJS	193:26:35	78
Totais da Viatura PJS	4010	PJS	193:26:35	78

Ver 1 - 30 de 30

Descrição	Código	Nome	Local	Início	Local Partida	Fim	Duração
Visita Estafeta	4009	Estafeta	Delegação Belas	05-12-2013 08:14:53	Praceta Dona Estefânia Casal da Barota	05-12-2013 08:14:58	00:00:05
Visita Estafeta	4009	Estafeta	Tecmic	02-12-2013 10:30:08	Rua Professor Doutor Cavaco Silva	02-12-2013 10:30:14	00:00:06
Visita Estafeta	4009	Estafeta	Tecmic	02-12-2013 11:04:21	Rua Professor Doutor Cavaco Silva	04-12-2013 17:34:18	54:29:57

Figura 2 - Relatórios gerais da frota [6].

De notar, que estes relatórios são ainda exportáveis para vários formatos: HTML, Word, Excel, PDF e CSV permitindo que o cliente possa fazer uma alargada análise de auditoria e *reporting*.

2.1.1.3 Mensagens

No iZiTraN Web também existe uma importante vertente de comunicação, com troca de mensagens de texto entre os operadores de frota autenticados na plataforma *web* e os condutores presentes nos veículos. Esta comunicação é baseada num “*chat online*” para comunicação de início e fim de viagem, estado de veículo, entre outras. As mensagens

enviadas podem ser predefinidas, para uma resposta automática de forma a facilitar e aumentar a rapidez da comunicação de mensagens. A Figura 3 apresenta a destacado, no canto inferior direito, o *chat* de mensagens que se encontra integrado com o Tempo Real.

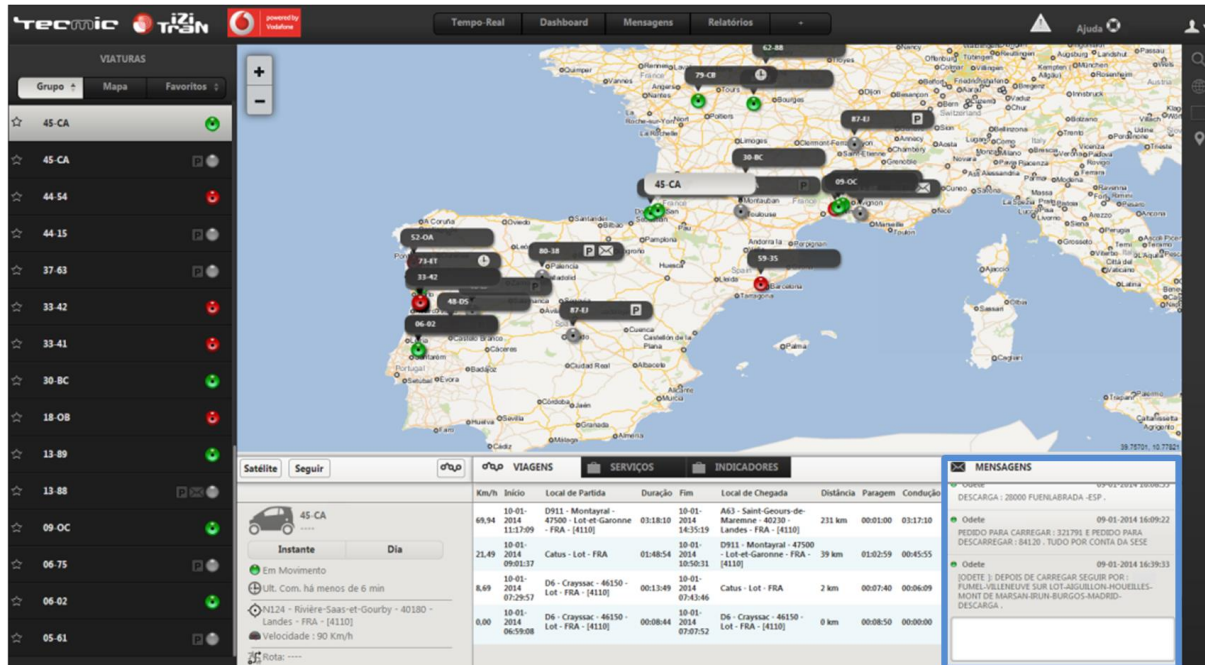


Figura 3 - *Chat* de Mensagens (adaptação da Figura 1).

É destacado na Figura 3 o *chat* de mensagens utilizado pelos operadores de frota para agilizar as comunicações com os condutores. A plataforma iZiTraN Web também disponibiliza um histórico de mensagens para que o operador de frota possa ter acesso a todas as mensagens trocadas com os condutores dos veículos, sendo que as mesmas podem ser filtradas através de uma conjunção de filtros como grupo de veículos, uma data específica, entre outros.

2.1.1.4 Alarmísticas

O iZiTraN Web possui um conjunto de alarmísticas que estão inseridas em diferentes categorias, consoante o seu nível de perigosidade. Quando o alarme é desencadeado, o operador de frota é notificado de forma diferenciada para que a resposta seja desencadeada com maior ou menor urgência. Estes alarmes podem ser configurados, no Gestor de Notificações, para serem recebidos por correio eletrónico ou por Short Message Service (SMS), para que o operador de frota possa agir/tratar, em tempo real, o alarme desencadeado. Geralmente os alarmes estão relacionados com incumprimentos, por exemplo, alarmes de excesso de velocidade, de travagens bruscas, de tempo parado, de entrada e saída de balizas

definidas como sendo zonas com um elevado grau de criminalidade, mas também podem ser alarmes de cumprimento, por exemplo, de chegada ao destino. A Figura 4 apresenta o Gestor de Notificações onde poderão ser visualizados todos os alarmes desencadeados.

Tempo-Real Dashboard Mensagens Relatórios + Ajuda jdomingos

Gestão de Notificações

Listagem das Notificações actualmente configuradas. Para configurar uma nova notificação carregue em 'Adicionar Notificação'.

Adicionar Notificação

Nome do Evento	Nome do Veículo	Nome Baliza	Condutor	Grupo	Editar
Abertura de Portas ON	----	----	----	----	
Temperatura de Galera ON	----	----	----	----	
Temperatura de Galera OFF	----	----	----	----	
Excesso de velocidade	----	----	----	----	
Excesso de velocidade	----	----	----	----	
Excesso de velocidade	----	----	----	----	
Paragem prolongada	----	----	----	----	
Paragem prolongada	----	----	----	----	
Paragem prolongada	----	----	----	----	
Recepção de Mensagem de Texto	----	----	----	----	
Recepção de Mensagem de Texto	----	----	----	----	
Entrada em Baliza	----	Tecmic H	----	----	
Entrada em Baliza	----	----	----	----	
Entrada em Baliza	----	Venta a Vapor	----	----	
Entrada em Baliza	----	Sede	----	----	
Início de actividade	----	----	----	----	
Início de actividade	----	----	----	----	
Fim de actividade	----	----	----	----	
Fim de actividade	----	----	----	----	
Inscrição de condutor	----	----	----	----	
Inscrição de condutor	----	----	----	----	

Figura 4 - Gestor de Notificações.

A Figura 4 apresenta o Gestor de Notificações, permitindo a adição de novos alarmes e a consulta do histórico de alarmes desencadeados, com o nome do alarme, do veículo, da baliza, do condutor, do grupo, como também a possibilidade de edição de cada alarme desencadeado para um eventual tratamento de forma manual.

2.1.2 Funcionamento

O funcionamento do iZiTraN Web começa pela instalação de um equipamento Tecmic, em cada veículo, composto por um recetor Global Positioning System (GPS) [8] e um módulo de comunicações General Packet Radio Service (GPRS) [9], que comunica minuto a minuto para um servidor central a localização e estado de cada veículo. Para consultar as informações recolhidas pelo equipamento Tecmic o operador de frota acede à página da plataforma iZiTraN Web. Cada operador de frota tem uma palavra-passe de acesso à plataforma, que lhe permite consultar unicamente os seus veículos. A Figura 5 apresenta a arquitetura iZiTraN Web.

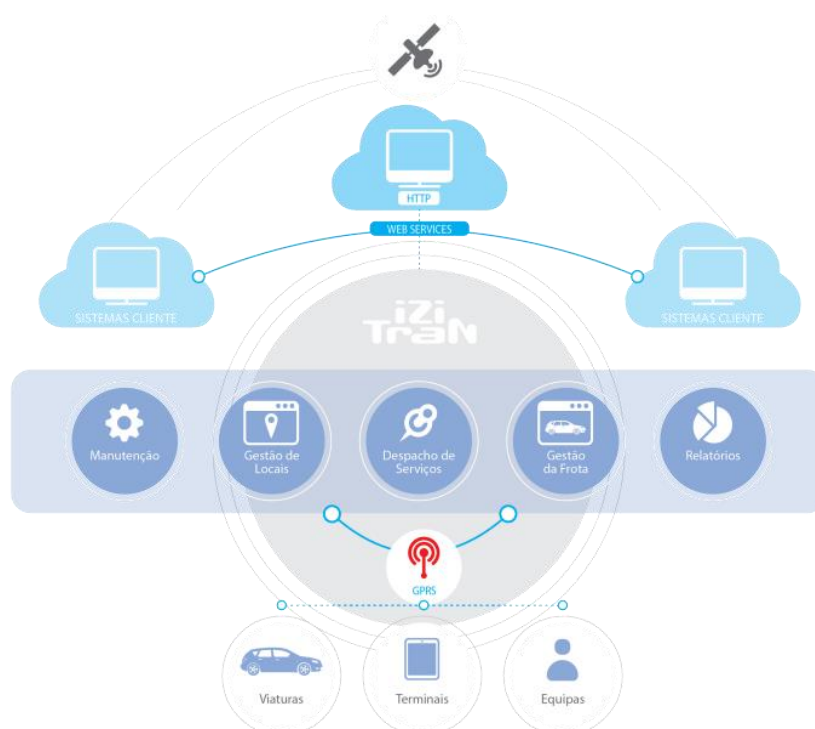


Figura 5 - Arquitetura iZiTraN Web [6].

A arquitetura do iZiTraN Web é bastante fácil de entender, uma vez que os equipamentos Tecmic presentes nos veículos comunicam via GPRS aos Web services de tempo real, alarmísticas, mensagens, entre outros, presentes nas instalações da Tecmic S.A., utilizando um modelo de comunicação cliente-servidor [10]. Depois deste procedimento, os operadores de frota acedem via *Web browser* ao iZiTraN Web que apresenta as várias informações recolhidas de forma perceptível e de fácil interpretação, podendo estes consultar e comunicar com o veículo.

2.2 Aplicações móveis concorrentes

Desde o princípio se soube que a aplicação móvel a desenvolver seria para monitorização de frotas e seria baseada no produto iZiTraN Web, no entanto foi também feito um estudo às aplicações móveis de empresas concorrentes de gestão de frotas, para que assim se analisasse outras funcionalidades e disposição de informação própria para dispositivos móveis. De seguida é feita uma breve apresentação de algumas das aplicações mais relevantes estudadas.

2.2.1 Telogis

A empresa Telogis [11] está sediada nos Estados Unidos da América e oferece produtos semelhantes aos da Tecmic S.A.. Tem soluções de localização para empresas em todo o

mundo, auxiliando as empresas na gestão efetiva de operações e de ativos móveis, através da integração de tecnologia de localização, informação e serviços. As soluções disponibilizadas permitem também a redução dos custos operacionais, como combustível e mão-de-obra, aumento da produtividade e da segurança das operações, além da disponibilização do apoio da tomada de decisões, fornecendo informações em tempo real, tudo isto integrado nos sistemas corporativos. De seguida, são apresentadas as aplicações móveis oferecidas pela Telogis.

2.2.1.1 Telogis Coach

A aplicação móvel Telogis Coach [12] está destinada a atribuir pontuações mediante a condução dos condutores de uma empresa, cujo objetivo é reduzir os maus hábitos de condução que até aqui podiam passar despercebidos, tentando assim diminuir o risco de acidentes de trabalho, ou situações piores. Esta aplicação funciona como um treinador, fornecendo *feedback* imediato aos condutores com base no seu desempenho real, oferecendo sugestões sobre formas de melhorar a sua pontuação de segurança. Uma característica da aplicação é conter um quadro de liderança, permitindo que os condutores possam competir com os colegas de trabalho para ser o condutor mais seguro na equipa e conseqüentemente a empresa possa atribuir prémios aos melhores condutores da mesma.

A aplicação tem implícito o uso de conceitos e metodologias dos jogos, processo chamado de Gamification [13]. Este processo usa a mecânica de jogos em que é preciso esforço para realização das tarefas para conquistar o objetivo, tornando-se numa estratégia de interação entre empresas e trabalhadores com base na oferta de incentivos que estimulem a melhoria no trabalho. A aplicação descrita é apresentada na Figura 6.

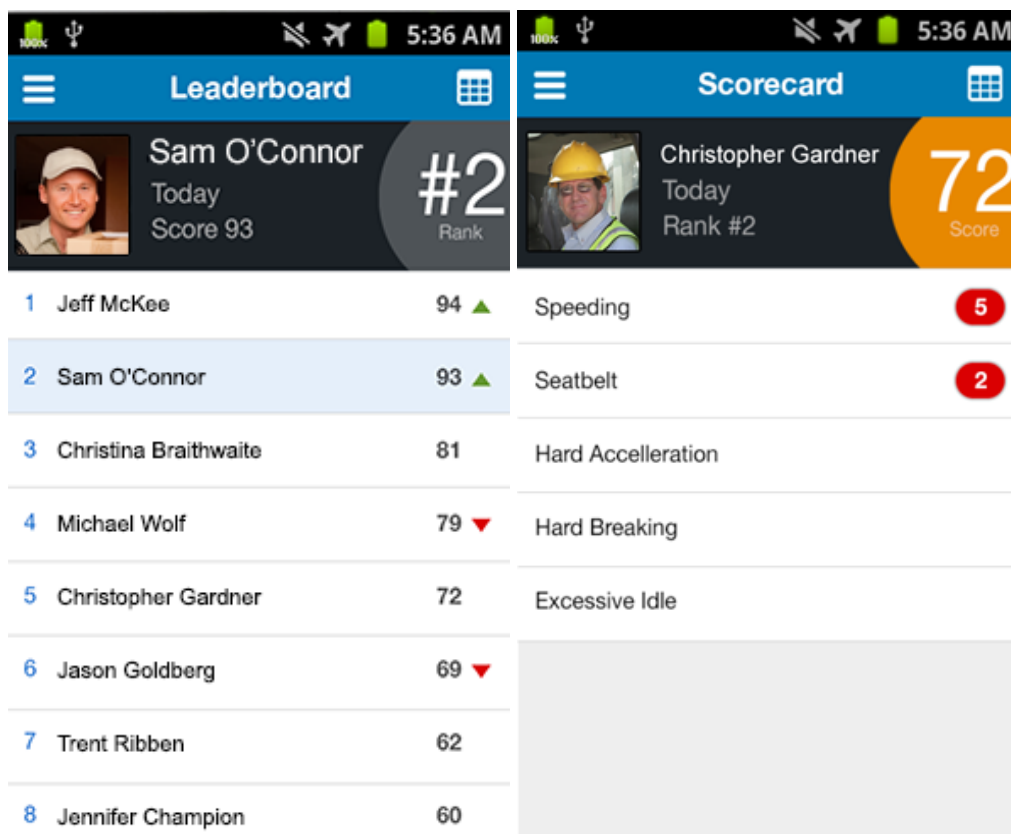


Figura 6 - Aplicação Telogis Coach apresentando o quadro de liderança [12] e o quadro individual de cada condutor [12].

A Figura 6 apresenta à esquerda o quadro de liderança da aplicação, onde estão todos os condutores com as respectivas pontuações e consequente posição, enquanto à direita apresenta a informação de cada condutor, com os respetivos maus hábitos de condução, em que cada mau hábito infligido pode ter um contador representativo do número de vezes que foi infligido.

2.2.1.2 Telogis Supervisor

Com a aplicação móvel Telogis Supervisor [14], os utilizadores podem visualizar a localização e estado atual da sua equipa de veículos. A aplicação ajuda os operadores de frota a alocar recursos de forma mais eficaz, fazendo agendamento de tarefas com base na localização do veículo. Uma vez conhecida a localização, a aplicação verifica o condutor que se encontra mais próximo desse local, e indica-o para lá. É ainda dada a possibilidade de o utilizador conhecer onde os veículos se encontram em relação aos locais de paragem para trabalho. A aplicação descrita é apresentada na Figura 7.

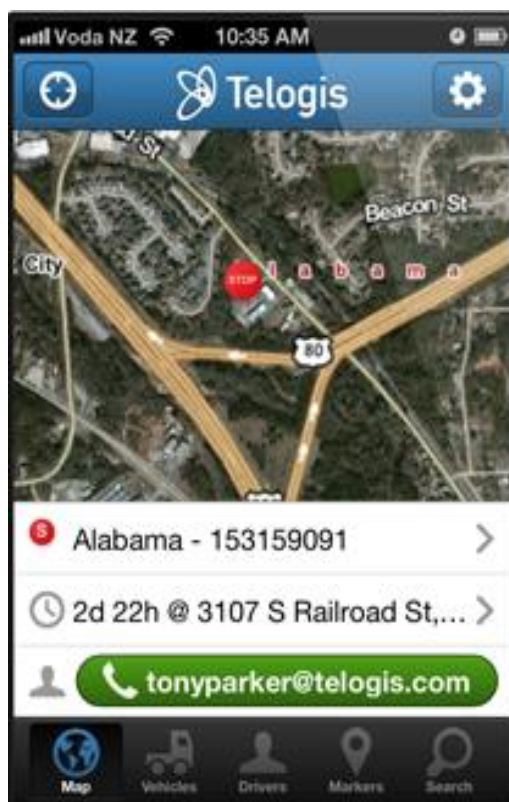


Figura 7 - Aplicação Telogis Supervisor [14].

A Figura 7 corresponde à visualização em formato mapa da localização dos veículos, podendo o operador de frota exercer ações sobre o veículo que seleciona, como fazer uma chamada para o veículo selecionado. De notar que a aplicação mostra uma barra em baixo, que permite, de forma simples, alternar entre os vários ecrãs da aplicação.

2.2.1.3 Telogis WorkPlan

A aplicação móvel Telogis WorkPlan [15] tem o intuito de gerir os planos de trabalho, trazendo ordens de serviço otimizadas para o campo, isto é, as tarefas agendadas são fornecidas diretamente para a aplicação do trabalhador no início do seu turno, para que este tenha uma lista simplificada das suas tarefas diárias. É por isso, uma aplicação forte na gestão de ordens de serviço, na priorização de tarefas, captura de dados de campo e inspeções a veículos, como também a atualização em tempo real do estado do trabalho de um trabalhador. Esta aplicação pode ajudar deste modo a resolver necessidades na gestão de ordens de serviço. A Figura 8 apresenta a aplicação descrita.



Figura 8 - Aplicação Telogis WorkPlan [15].

A Figura 8 condiz com a existência de um serviço por realizar. O trabalhador neste ecrã pode responder as questões daquele serviço e fazer a sua própria assinatura, para validar a sua identidade e anexar assim, a sua assinatura ao serviço.

2.2.2 Fleetmatics

A empresa Fleetmatics [16] também sediada nos Estados Unidos da América serve mais de 23.000 clientes, com mais de 500.000 veículos a circular em todo o mundo, cujo mercado de produtos disponíveis também é semelhante ao da Tecmic S.A. e da Telogis. Tem soluções de localização, eficiência energética, velocidade e quilometragem de veículos, dando às empresas uma visão mais completa do que acontece no campo, para depois poderem utilizar esse conhecimento para melhorar os seus negócios, reduzir os custos operacionais e de capital, bem como aumento de receita. De seguida, são apresentadas as aplicações móveis que são disponibilizadas pela empresa Fleetmatics.

2.2.2.1 Fleetmatics Reveal

A aplicação móvel Fleetmatics Reveal [17] tem o objetivo de monitorizar os veículos para que o utilizador da aplicação tenha a precisão, em tempo real, do local onde o condutor se

encontra. Assim o operador de frota pode estar informado do desempenho de cada veículo, compará-lo com os seus melhores indicadores, receber alertas de incidentes e se necessário, enviar o condutor mais próximo para um trabalho urgente. A aplicação descrita é apresentada com mais detalhe nas Figura 9 e Figura 10.

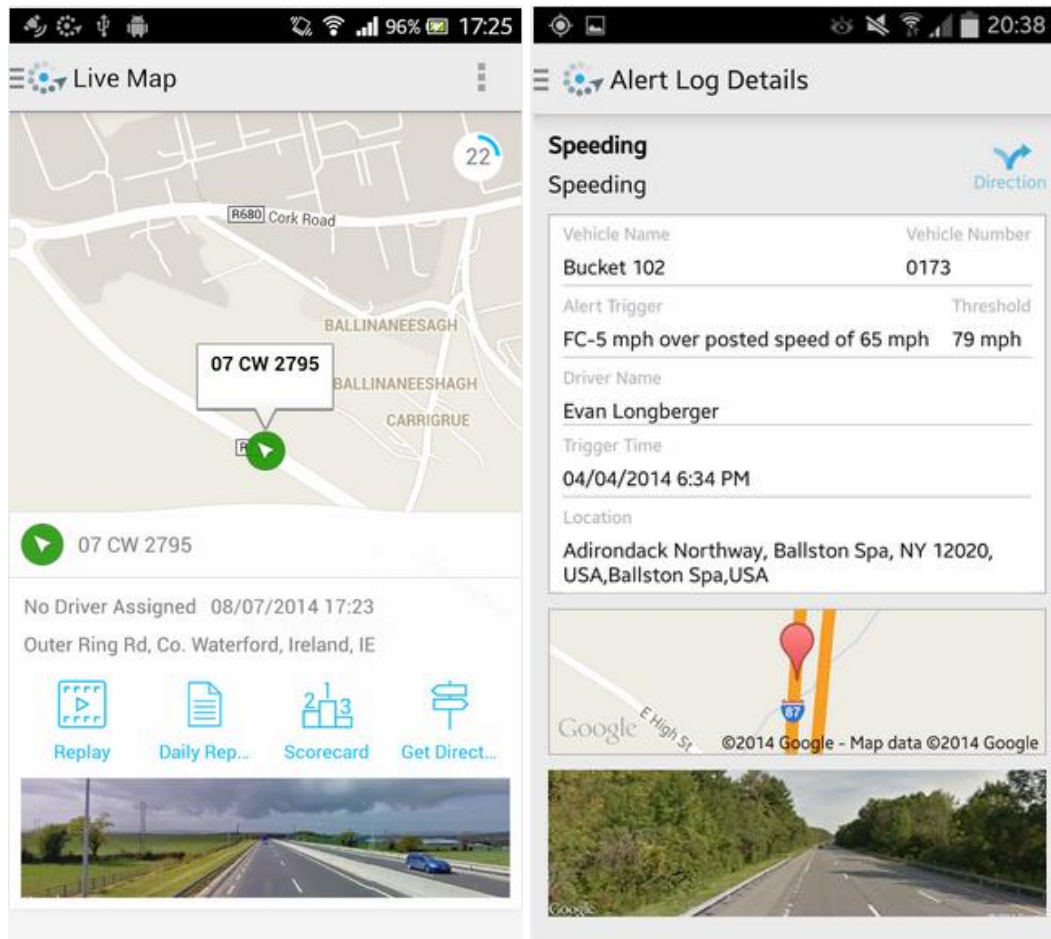


Figura 9 - Aplicação Fleetmatics Reveal apresentando mapa e alerta [18].

A Figura 9 apresenta à esquerda a visualização do veículo no mapa, mostrando o ponto onde este se encontra identificado com a sua matrícula. Em baixo é apresentada a localização real do veículo e algumas opções para visualizar outras informações com mais detalhe. À direita a Figura 9 apresenta um alerta que ocorreu por um condutor estar em excesso de velocidade, mostrando a informação do veículo, do condutor e da localização do mesmo, tanto escrita, como visualmente através do recurso ao mapa.

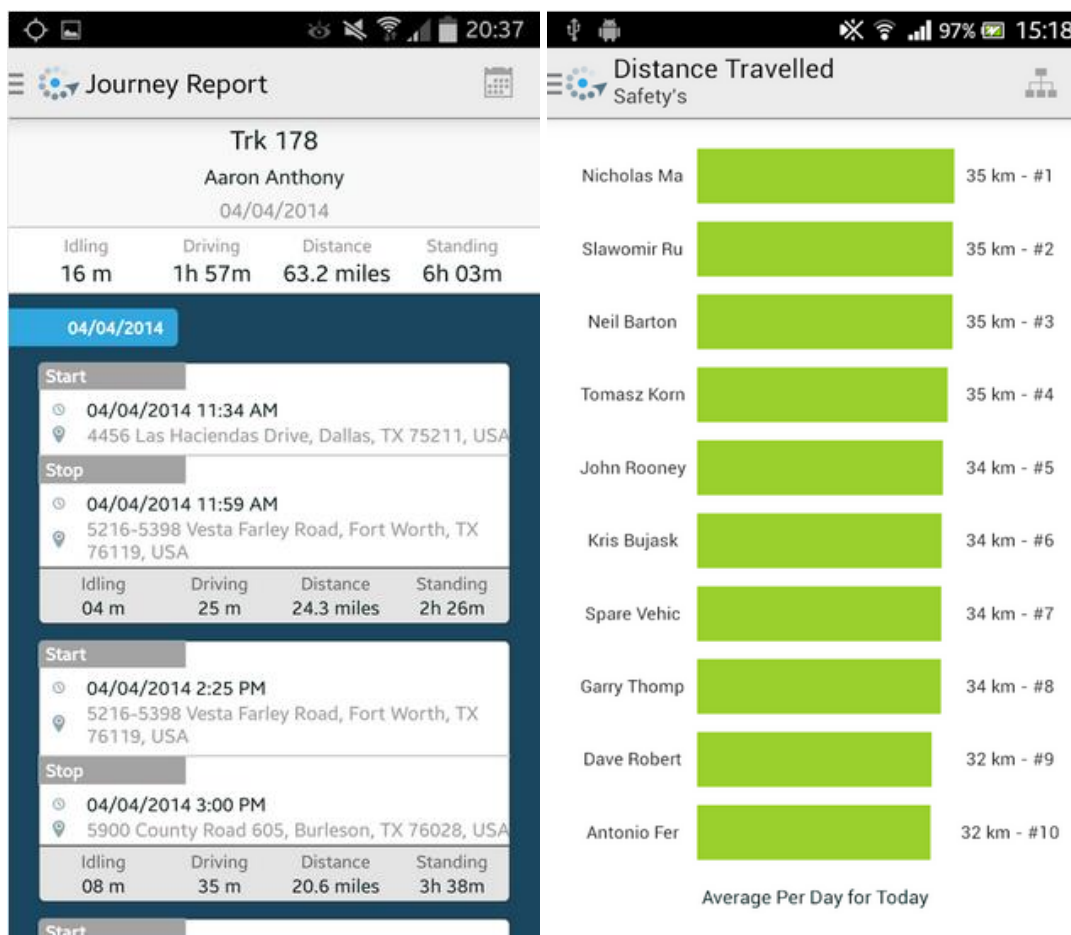


Figura 10 - A Aplicação Fleetmatics Reveal apresentando o histórico diário e a distância percorrida [18].

A Figura 10 destaca à esquerda o histórico diário do condutor selecionado, apresentando os indicadores totais de tempos inativo, condução e parado, bem como a distância percorrida naquele dia, mas também indicadores específicos em cada viagem, como os pontos de início e fim das mesmas. Enquanto à direita é apresentada a distância percorrida pelos condutores daquela frota, para que o operador de frota possa visualizar quais os condutores que fazem mais ou menos quilómetros diários.

2.2.2.2 Fleetmatics Reveal LogBook

A aplicação móvel Fleetmatics Reveal LogBook [19] obtém continuamente indicadores acerca do veículo, como indicadores do motor, combustível, velocidade, quilómetros percorridos, entre outros, diretamente para a aplicação. Esta aplicação é vocacionada para ser utilizada pelos condutores, tendo o intuito de permitir que os condutores possam controlar as suas horas de trabalho com simplicidade, para que deste modo saibam quanto tempo já conduziram ou descansaram, e assim possam saber se estão a cometer alguma infração à

legislação. De notar, que esta aplicação necessita que o utilizador tenha também a aplicação Fleetmatics Reveal instalada. Na Figura 11 é apresentada a aplicação.

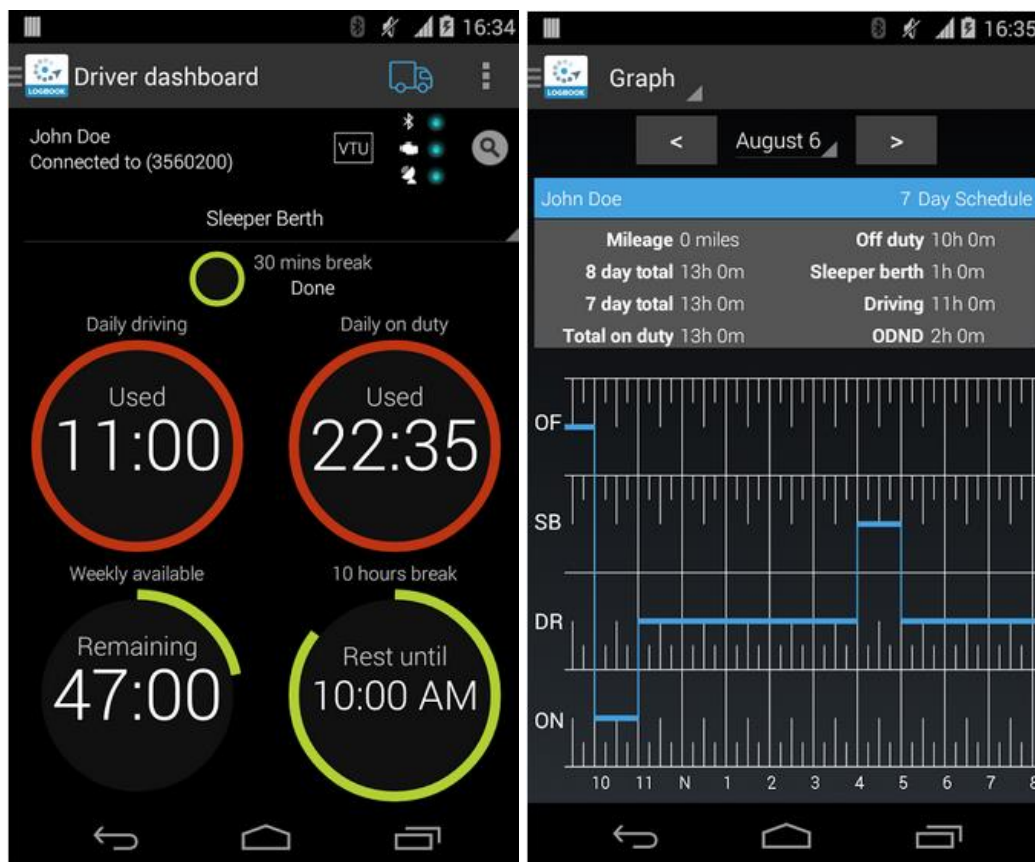


Figura 11 - Aplicação Fleetmatics Reveal LogBook apresentando os tempos de condução e descanso dos condutores [20].

A Figura 11 apresenta à esquerda os contadores de tempo de condução e descanso efetuados pelo condutor, bem como o tempo do próximo descanso e os tempos de descanso que ainda faltam cumprir durante a semana. Já à direita a figura apresenta tempos totalizadores, mas também um gráfico com o percurso de condução e descansos que o condutor realizou, bem como o tempo em que a ignição do veículo esteve ligada (ON) e desligada (OFF).

2.2.3 Scania

A Scania [21] é uma empresa que iniciou a sua atividade em 1891, com a produção de carruagens ferroviárias, e é hoje conhecida, mundialmente, pelos seus camiões de frotas. Esta empresa internacional está presente em mais de 100 países, sendo que a sua principal atividade se desenvolve ao nível da venda de veículos pesados, motores marítimos e industriais, acessórios e respetiva manutenção e reparação. A empresa tem uma reconhecida

experiência em veículos e também ela apostou numa aplicação móvel para gestão de frotas.

2.2.3.1 Scania Fleet Management

A aplicação móvel Scania Fleet Management [22] suporta a gestão de frotas e outros equipamentos onde quer que o utilizador esteja. A aplicação permite criar relatórios de avarias e tirar fotografias da situação de avaria, para posteriormente enviá-las junto com o relatório. Autenticado com o perfil de operador de frotas é possível também utilizar a aplicação para visualização da posição dos veículos num mapa, dados atuais de veículos e dados pormenorizados dos tempos de condução, atividades do condutor, bem como contactar os seus condutores diretamente por SMS ou telefone. Permite também que sejam enviadas mensagens entre utilizadores, onde também é possível utilizar o serviço de mensagens para enviar fotografias e indicações de posição. A aplicação descrita é apresentada com mais detalhe na Figura 12.

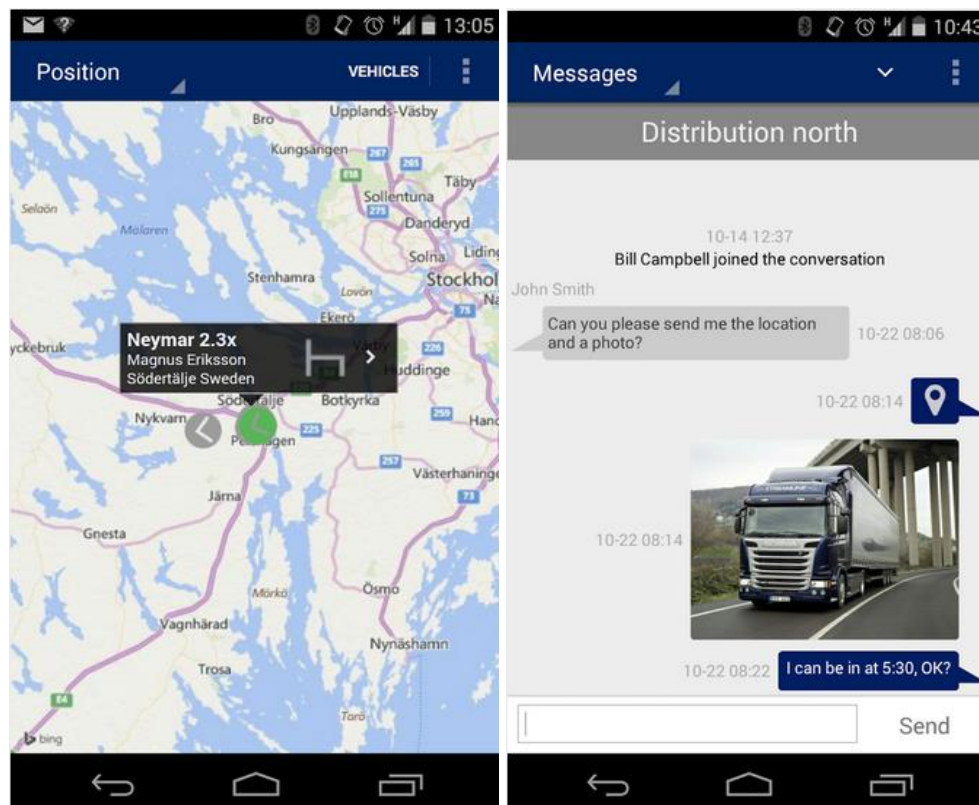


Figura 12 - Aplicação Scania Fleet Management apresentando o mapa e as mensagens [23].

A Figura 12 destaca à esquerda a visualização da localização de um veículo no mapa, enquanto à direita é possível visualizar uma troca de mensagens entre o utilizador da aplicação e outros utilizadores. Destaca-se também o facto de através das mensagens trocadas

ser possível ter vários utilizadores na mesma conversa, podendo assim enviar a mesma informação a vários utilizadores e existir a possibilidade de colocação da posição geográfica, bem como o envio de imagens.

2.2.4 Outras Aplicações

Além das aplicações apresentadas, foi também realizado um estudo mais alargado a outras, tanto para a plataforma Android [24], como para IOS [25] ou Windows Phone [26], sendo que muitas delas foram criadas para uso em mais que uma plataforma, para assim ampliarem o leque de oferta e abrangerem os utilizadores das várias plataformas.

Para a plataforma Android destacaram-se as aplicações: TSO Fleet Pro [27]; Fleet Manager [28]; GPS Insight Mobile [29]. Já para a plataforma IOS evidenciam-se as aplicações: TomTom WEBFLEET® Mobile [30]; MiX Fleet [31]. Enquanto para a plataforma Windows Phone destacou-se a aplicação Quartix [32].

Apesar destas aplicações não se terem destacado como as apresentadas anteriormente, também elas representaram um contributo (conhecimento de aplicações desenvolvidas em outras plataformas) para o estudo das aplicações concorrentes.

2.2.5 Comparação entre Aplicações

É possível notar que cada uma das aplicações móveis concorrentes tem as suas próprias funcionalidades e objetivos diferenciados. As empresas Telogis e Fleetmatics focaram-se na aposta em diversas aplicações de forma a responder a desafios distintos, contrastando com a empresa Scania que aposta numa aplicação com um completo conjunto de funcionalidades de forma a responder a todos os desafios.

A empresa Telogis tem uma forte aposta no desenvolvimento de diversas aplicações, como a aplicação Telogis Coach para ajudar a melhorar o desempenho dos condutores, a Telogis Supervisor para os operadores de frota monitorizarem o estado e localização dos veículos e a Telogis WorkPlan para os condutores gerirem os seus planos de trabalho, fazendo a gestão das suas ordens de serviço. A empresa Fleetmatics também aposta neste tipo de abordagem, com a sua aplicação Fleetmatics Reveal focando mais a parte de tempo real e da alarmística, e a aplicação Fleetmatics Reveal LogBook mais vocacionada para o condutor (indicadores do veículo, combustível, velocidade, quilómetros percorridos, tempos de condução e de descanso). Deve notar-se no entanto que a aplicação Fleetmatics Reveal LogBook necessita de ter a aplicação Fleetmatics Reveal instalada, provocando uma dependência obrigatória para

os utilizadores.

A empresa Scania tem na sua única aplicação Scania Fleet Management um completo conjunto de funcionalidades, como relatórios, visualização da localização de veículos em mapa, comunicação através de mensagens, entre outros, tornando esta aplicação a mais parecida com o produto iZiTraN Mobile.

A aposta em beneficiar de várias aplicações garante às empresas (Telogis e Fleetmatics) a existência de mais produtos para comercialização e uma melhor diversidade dos mesmos. No entanto poderão existir dependências entre aplicações, confundindo o utilizador relativamente a qual aplicação usar para aceder a uma determinada funcionalidade. Já a aposta numa única aplicação (Scania), permite que a empresa tenha uma aplicação completa, com um amplo conjunto de funcionalidades, garantindo ao utilizador acesso a todas as funcionalidades num único espaço. De referir que esta última aposta é a escolhida para a aplicação iZiTraN Mobile, devido às mais-valias explicadas anteriormente.

2.3 iZiTraN Mobile

O produto iZiTraN Mobile é a aplicação móvel desenvolvida durante o período de estágio que nasceu com o objetivo de estender o iZiTraN Web para as plataformas móveis de forma a incluir um conjunto de funcionalidades importantes, adaptadas para a realidade dos dispositivos móveis.

Foi através de várias reuniões de enquadramento no processo de negócio iZiTraN e da identificação do mercado-alvo deste novo produto, que se confirmou a necessidade real da utilização deste novo produto no dia-a-dia dos seus utilizadores. Como a aplicação iZiTraN Mobile é exclusivamente para ser usada em dispositivos móveis, foi essencial adquirir este conhecimento da plataforma iZiTraN Web para escolher as funcionalidades que melhor se aplicariam ao iZiTraN Mobile. Deste modo, foi decidido que devido às limitações de dimensões nestes dispositivos, toda a parte de gestão de relatórios não deveria ser incluída na aplicação móvel. Contudo, a parte de tempo real, mensagens e alarmes, seria o ponto forte desta aplicação.

O iZiTraN Web forneceu assim as funcionalidades base e o estudo das aplicações concorrentes serviu para analisar novas e futuras funcionalidades, bem como encontrar outras formas de navegação e interação com o utilizador. Com o iZiTraN Mobile surgem novos desafios que não existam no iZiTraN Web, ou pelo menos não de forma tão evidente e crucial. Alguns dos desafios que surgem estão relacionados com as diferentes características dos dispositivos móveis e dos poucos recursos que alguns deles possuem, como a baixa

qualidade de serviço ou a maior instabilidade associada às redes celulares, os ecrãs de pequenas dimensões, e a autonomia das baterias dos dispositivos móveis que suscitam vários desafios técnicos a serem resolvidos pela aplicação iZiTraN Mobile.

O próximo capítulo descreve a metodologia concebida para o desenvolvimento do projeto, bem como os aspetos genéricos de gestão do projeto.

Metodologia e Gestão de Projeto

Este capítulo descreve a metodologia utilizada durante o desenvolvimento do projeto Mobile Fleet Management, explicando quais são os pontos fortes da metodologia adotada e as adaptações que foram feitas ao longo do uso da mesma. No presente capítulo é também descrito como foi realizada a gestão do projeto descrevendo o âmbito e a gestão do tempo, onde é apresentado o planeamento realizado ao longo do período de estágio. É ainda apresentada a gestão de risco, mostrando as tarefas que se previu que poderiam trazer riscos, a gestão de recursos, onde são apresentados os recursos humanos e materiais utilizados no projeto e a gestão de comunicação, que explica a comunicação usada ao longo do mesmo.

3.1 Metodologia

Existem tipicamente duas etapas comuns a todos os projetos de desenvolvimento de *software*, uma primeira etapa de requisitos e uma segunda etapa de desenvolvimento que acontece sempre, independentemente do tamanho ou complexidade do projeto.

Devido à grande dimensão do projeto Mobile Fleet Management teve de se optar por uma metodologia dinâmica e flexível, visando um desenvolvimento rápido, atendendo às reais necessidades do cliente (peça chave no processo de desenvolvimento, tanto na construção e revisão dos requisitos, como na validação), e ainda que permitisse modificações, à medida que aparecessem novas necessidades [33]. Para tal, foi decidido seguir a metodologia eXtreme Programming (XP) [34] e adaptá-la às necessidades do projeto.

Além da comunicação, um dos principais motivos que levou a seguir esta metodologia foi o facto de o projeto conter um âmbito muito aberto, com vagos requisitos e em constante mudança, permitindo adotar uma estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de *software*. Outro dos motivos foi seguir iterações de trabalho em que se planeia as tarefas a realizar durante a iteração, para posteriormente serem implementadas durante a mesma.

É importante referir que existem valores [35], como a simplicidade, comunicação, *feedback*,

respeito e coragem, que alicerçam a metodologia pelos seguintes motivos:

- A simplicidade é necessária desde a forma como se recolhe requisitos até à codificação e aos testes do produto desenvolvido;
- A comunicação é obrigatória para que não haja deficiências em processos e problemas entre equipa e cliente;
- O *feedback* baseia-se em recolher informações entre os membros da equipa e também na relação com o cliente, desde responder *e-mails*, telefonemas, entre outros;
- O respeito deve permanecer sempre, respeitando a experiência dos clientes e vice-versa, aceitar responsabilidades e cumprir com o próprio trabalho;
- A coragem para saber dizer “NÃO” quando necessário, ou por exemplo, comunicar que o projeto vai demorar além do estimado, uma vez que novos requisitos necessitam de ser programados ou que o código já em funcionamento necessita de ser refeito.

Como se explicou, a metodologia adotada é dinâmica e flexível, pelo que permitiu corresponder da melhor forma às necessidades do projeto, porém é necessário muita disciplina e uma forte comunicação entre todas as pessoas envolvidas para aumentar o sucesso do projeto. Além dos valores que esta metodologia suporta, ainda traz consigo um fluxo de desenvolvimento que deve ser seguido [36], apresentado na Figura 13.

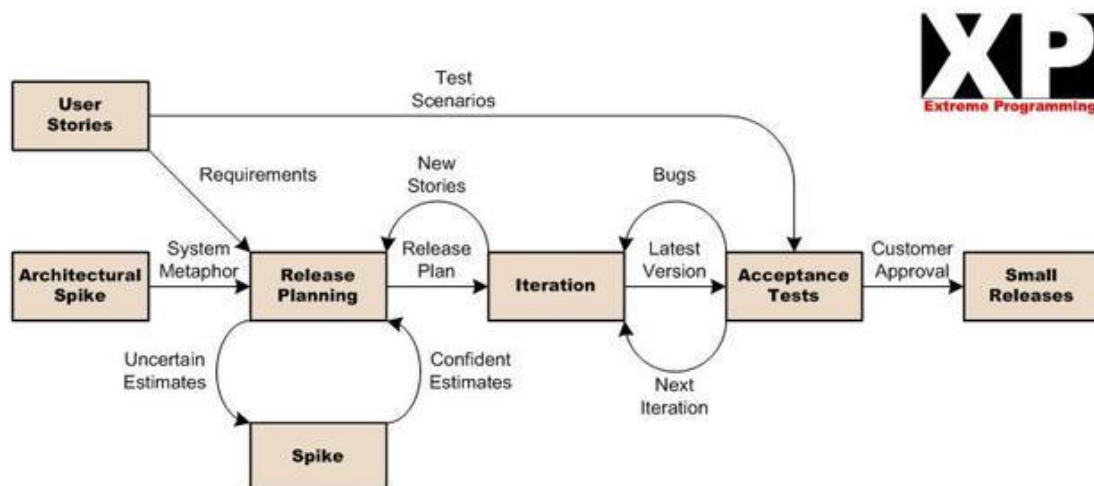


Figura 13 - Fluxograma com o fluxo de desenvolvimento [37].

O fluxo de desenvolvimento apresentado na Figura 13 é o fluxo geral da metodologia, no entanto a metodologia real aplicada teve algumas adaptações à medida que foram surgindo necessidades ao longo do projeto.

Na extração de requisitos, foi criado um documento simplificado com as *user stories*

(Apêndice I), mas nem todas seriam as mais indicadas para dispositivos móveis. Desta forma, houve uma fase de seleção de *user stories* que utilizou a matriz de Wieggers de Karl Wieggers [38] para auxiliar nesta seleção de importância de prioridades (Apêndice II). A matriz de Wieggers baseia-se na colocação em lista de todos os requisitos, para posteriormente os *stakeholders* estimarem, quatro variáveis: benefício, penalidade, custo e risco para que a matriz, através de fórmulas matemáticas, possa calcular a percentagem de prioridade atribuída a cada requisito. Primeiro estima-se os benefícios de cada requisito numa escala de 1 a 9. O valor 1 indica que ninguém considera o requisito útil e 9 significa que o mesmo é extramente importante. De seguida, estima-se a penalidade que existe no cliente e no processo de negócio caso o requisito não seja incluído. Novamente, é necessário utilizar a escala de 1 a 9, onde o valor 1 indica que nada será afetado se o requisito for excluído, e 9 indica um sério problema com o requisito. Requisitos que possuam baixo benefício e baixa penalidade adicionam custo ao projeto, uma vez que como têm pouco valor a sua execução em detrimento de outros traria elevado custo ao mesmo. Similarmente é atribuído um peso ao custo de implementação de cada requisito, novamente a escala de 1 (fácil e rápido) a 9 (difícil e demorado), a matriz calcula com base nos pesos atribuídos a percentagem do custo total que cada requisito tem. Geralmente a estimativa de custos é baseada na complexidade, desenho de *interfaces*, possibilidade de reutilização, quantidade de testes, documentação necessária, entre outros. Por fim, estima-se o risco relativo de cada requisito na escala de 1 a 9. De notar, que requisitos mal definidos podem requerer repetição de trabalho, uma vez que poderá haver trabalho feito desnecessariamente por não servir as necessidades do cliente final, acrescentando elevado risco ao projeto.

A utilização da matriz de Karl Wieggers trouxe vários benefícios, entre eles: um âmbito mais reduzido e bem definido, menor quantidade de defeitos nos requisitos, redução de repetição de trabalhos, desenvolvimento com menos requisitos desnecessários, diminuição de custos operacionais, desenvolvimento mais rápido, menos problemas de comunicação, estimativas mais confiáveis e maior satisfação dos clientes e membros da equipa.

As iterações seguiram um modelo incremental, aplicando o modelo espiral a cada iteração, repetindo as fases de análise, desenho, implementação e testes [39] [40]. Todas as semanas houve uma reunião entre a equipa de desenvolvimento e o gestor de projeto, onde foi discutido o trabalho já realizado e o trabalho a ser feito até à reunião seguinte. A validação de requisitos e os aspetos funcionais foram feitos tendo em consideração o conhecimento da empresa no setor de gestão de frotas, ao longo de mais de 25 anos.

Como é característica da metodologia XP, a validação dos requisitos desenvolvidos durante a

iteração ocorre através da execução dos testes de aceitação e dos testes unitários. De acordo com Beck [41], os programadores devem escrever os testes unitários e os clientes os testes de aceitação, sendo ambos escritos antes do processo de codificação e só posteriormente executados. No entanto, neste projeto específico, o programador fez a escrita de testes (secção 5.1), sendo que os testes de usabilidade (5.1.5) foram feitos às equipas de desenvolvimento da Tecmic S.A., tendo em conta o seu conhecimento no setor de gestão de frotas.

3.2 Gestão de Projeto

Esta secção descreve o projeto, começando por apresentar o seu âmbito e descrevendo os resultados pretendidos com o mesmo. Seguidamente é apresentada a gestão de tempo, onde são referidas as tarefas que foram ocorrendo durante o projeto, mostrando depois a gestão de risco que apresenta os riscos identificados e a importância da sua identificação para posteriormente os poder minimizar. É ainda descrita nesta secção a gestão de recursos, tanto humanos, como materiais e a gestão da comunicação, que apresenta os métodos usados na comunicação entre a equipa de desenvolvimento.

3.2.1 Descrição do Âmbito

Através da realização deste projeto, a Tecmic S.A. pode disponibilizar um novo produto vocacionado para dispositivos móveis que satisfaça as necessidades dos operadores de frota e cumpra determinados procedimentos exigidos por parte dos mesmos, que sentiam a necessidade de acompanhar em tempo real todos os veículos da sua frota.

O projeto consiste no desenvolvimento de uma aplicação móvel que permita visualizar, em tempo real, a posição precisa de veículos num mapa e informação dos mesmos, como dados pormenorizados dos tempos de condução, quilómetros efetuados, consumos de combustível, visualização de histórico de viagens do veículo, entre outros. Deste modo, tudo isto se torna possível devido à comunicação permanente entre a aplicação móvel e os seus Web services, que disponibilizam informações enviadas pelo equipamento Tecmic presente em cada veículo. Os operadores de frota podem contactar os seus condutores pela troca de mensagens, receber alertas dos mais variados incidentes que possam surgir e tratar dos mesmos. As funcionalidades referidas anteriormente são realizadas em tempo real e num dispositivo móvel, cujas características avançadas são variadas e a sua utilização tem a grande vantagem de poder ser acessível a partir de qualquer lugar.

3.2.2 Gestão de Tempo

A gestão de tempo foi planificada e estruturada de modo a que conjugasse o trabalho a realizar para a conclusão da unidade curricular de Estágio e o trabalho efetivamente realizado na empresa. Tendo o estágio decorrido num contexto empresarial, foi necessária a existência de um período de adaptação, com a finalidade de integração na equipa de desenvolvimento, nas principais instalações da empresa, no Taguspark (Oeiras).

O período de adaptação decorreu durante os dois primeiros meses de estágio, cujo objetivo principal foi a integração com a equipa de desenvolvimento. Neste período de tempo foram dadas várias ações de formação internas acerca dos vários produtos da empresa e procedeu-se à instalação de *software* de forma a criar uma base de trabalho sólida, com todas as ferramentas de trabalho necessárias. Desta forma, foi obtida uma melhor perceção acerca do modelo de negócio da empresa. Durante estes dois meses foi pedido para planificar as tarefas que viriam a ser realizadas ao longo do projeto, tendo sido a ferramenta Microsoft Project [42] a utilizada para criar o cronograma do mesmo (consultar Apêndice III para visualizar o Gráfico de Gantt [43] [44] correspondente), como é apresentado na Figura 14.

Task Name	Work	Duration	Start	Finish
iZiTraN Mobile	1.717,18 hrs	186 days	Mon 15-09-14	Mon 01-06-15
+ Gestão de Projeto	31,5 hrs	154,31 days?	Fri 24-10-14	Fri 29-05-15
+ Integração e Enquadramento	30 hrs	14,75 days?	Mon 15-09-14	Fri 03-10-14
+ Análise e Requisitos	236 hrs	24,38 days?	Mon 06-10-14	Fri 07-11-14
- Implementação e Desenvolvimento	1.236 hrs	156 days?	Sat 27-09-14	Mon 04-05-15
Elaboração de userstories	80 hrs	9,88 days?	Fri 07-11-14	Thu 20-11-14
+ Protótipo	168 hrs	32,63 days?	Wed 03-12-14	Fri 16-01-15
+ Pesquisa	72 hrs	21 days	Mon 01-12-14	Sat 27-12-14
Elaboração de contrato (dados e operações)	80 hrs	9,75 days?	Mon 19-01-15	Mon 02-02-15
+ Definição de Testes	172 hrs	12,96 days?	Thu 20-11-14	Tue 09-12-14
- Implementação	520 hrs	157 days	Sat 27-09-14	Mon 04-05-15
Criação de Webservice	74 hrs	9,25 days?	Thu 20-11-14	Wed 03-12-14
Integração entre Webservice com Aplicação	32 hrs	4 days?	Wed 03-12-14	Tue 09-12-14
Tempo Real	154 hrs	64,17 days?	Tue 09-12-14	Fri 06-03-15
Mensagens	112 hrs	46,67 days?	Tue 09-12-14	Wed 11-02-15
Alarmes	148 hrs	61,67 days?	Tue 09-12-14	Tue 03-03-15
+ Execução de Testes	88 hrs	111 days	Mon 01-12-14	Mon 04-05-15
Reuniões de validação	16 hrs	100 days?	Thu 20-11-14	Tue 07-04-15
Colocação em produção	40 hrs	5 days?	Tue 07-04-15	Tue 14-04-15
- Documentação	183,68 hrs	166,75 days?	Mon 06-10-14	Tue 26-05-15
Relatório	95,68 hrs	166,75 days	Mon 06-10-14	Tue 26-05-15
- Artefactos	88 hrs	97,25 days?	Fri 24-10-14	Tue 10-03-15
Especificação técnica	40 hrs	20 days?	Fri 24-10-14	Fri 21-11-14
Especificação do contrato de operações e dados	32 hrs	80 days?	Thu 20-11-14	Tue 10-03-15
Especificação de testes unitários	16 hrs	13,33 days?	Tue 09-12-14	Fri 26-12-14

Figura 14 - Cronograma do projeto Mobile Fleet Management.

A tarefa Gestão de Projeto engloba o planeamento do projeto, que representa o planeamento

de todas as tarefas enunciadas. Para que tudo ficasse organizado, antes e durante a elaboração do planeamento foi criado um documento auxiliar onde se descrevia todas as iterações e tarefas implementadas em cada iteração para que as mesmas não fossem esquecidas aquando do planeamento do projeto. A gestão de projeto engloba também a monitorização, durante todas as semanas, geralmente à sexta-feira, feita com a colocação do esforço despendido em cada tarefa e através de uma análise ao estado atual das tarefas planeadas.

A tarefa Integração e Enquadramento corresponde ao período de adaptação anteriormente referido, através das ações de formação internas acerca dos vários produtos e do conhecimento das ferramentas utilizadas pela empresa. Enquanto a tarefa de Análise e Requisitos teve como intuito a recolha e especificação de requisitos funcionais e não funcionais pela forma de *user stories*, cuja importância dada foi fulcral para se fazer um correto planeamento.

A tarefa Implementação e Desenvolvimento foi a maior e a que exigiu mais horas de dedicação, devido ao esforço requerido das suas subtarefas. Esta tarefa engloba a elaboração de *user stories* tão importantes para sintetizar os requisitos, como também do protótipo da aplicação móvel iZiTraN Mobile. O protótipo da aplicação móvel mostrou os requisitos esperados pelo cliente, a pesquisa feita às diretrizes (diretrizes para desenho de interfaces para a plataforma Android) e o conhecimento da plataforma a que o projeto se destina.

Um importante marco no projeto deu-se após a conclusão das tarefas enunciadas anteriormente, que ficou marcado pela mudança das principais instalações da empresa, no Taguspark (Oeiras), para continuar com o projeto nas instalações secundárias da empresa, na Incubadora D. Dinis (Leiria).

O trabalho realizado nas instalações secundárias da Tecmic S.A. iniciou-se pela tarefa da implementação do protótipo gráfico da aplicação móvel, com ecrãs que espelhavam o protótipo inicialmente desenhado. Seguiu-se a tarefa relativa à elaboração de contrato, que foi planeada para especificar os modelos de dados necessários e as operações a serem realizadas pela aplicação móvel e pelos seus Web services. A tarefa implementação tratou da parte de codificação da aplicação móvel iZiTraN Mobile, bem como da criação dos seus Web services e da correta integração entre ambos que é descrita no próximo capítulo. De referir que houve periodicamente reuniões de validação para acompanhar o trabalho realizado.

Na definição de testes foi esquematizado o conjunto de testes a serem realizados a cada módulo do projeto, para que desta forma e através desta documentação nada fosse esquecido durante a realização da tarefa de execução de testes, que como o nome indica trata-se da implementação e execução dos mesmos.

A tarefa Documentação engloba o presente documento, como também os apêndices e anexos deste, que foram documentados como suporte a tudo o que foi desenvolvido ao longo do período de estágio.

De notar que a gestão de todo o projeto teve de ser flexível mas rigorosa, uma vez que para além do Mobile Fleet Management, também houve o envolvimento na escrita de um artigo científico e ainda a participação em outros quatro projetos propostos pela empresa.

O artigo científico ficou nomeado de “iZiTraN Mobile Fleet Management” (Apêndice X), tendo sido submetido, avaliado e aprovado para apresentação na conferência internacional “CENTERIS - Conference on ENTERprise Information Systems” [45], entre os dias 7 e 9 de outubro de 2015.

Os quatro projetos propostos: Tempos de Condução (Anexo I), Avistamentos (Anexo II), Rent-a-Car (Anexo III) e Consola Gráfica para Transportes (Anexo IV) tiveram em conjunto com o artigo científico um esforço extra de trabalho, sendo que houve tarefas realizadas em simultâneo e tarefas que tiveram de ser paradas momentaneamente para se dar prioridade a outras. As alterações de prioridades das tarefas e o envolvimento de vários projetos em simultâneo não são resultado de um mau planeamento, mas sim do contexto empresarial que tem um dinamismo próprio para corresponder às necessidades do mercado.

3.2.3 Gestão de Risco

A gestão do risco de um projeto é bastante importante, sendo um processo sistemático de identificação, análise e resposta aos riscos do mesmo. Isto inclui maximizar a probabilidade e consequências de eventos positivos, que por sua vez minimiza a probabilidade e consequências que eventos adversos podem trazer aos objetivos do projeto.

Foram identificados vários riscos durante o planeamento, como o desconhecimento de produtos da empresa, ferramentas de trabalho utilizadas e o facto de o projeto ter sido desenvolvido internamente sem um cliente final, o que por si só já representou um importante risco assumido pela equipa de desenvolvimento. Outro risco identificado desde o início, foi o facto de o projeto estar inserido numa equipa de trabalho, implicando por isso fortes dependências entre os membros da equipa. Este risco só é minimizado se o plano de comunicação entre a equipa estiver bem articulado e se houver bom senso entre as pessoas que a compõem, preservando os valores da metodologia XP [35]. Um risco inicial que houve neste projeto foi o atraso no levantamento de todos os requisitos, bem como o não “congelamento” de requisitos. Este risco aparece facilmente em muitos projetos e pode ser desastroso, tendo de ser identificado atempadamente para que não se gastem recursos (tempo

e trabalhadores envolvidos) em tarefas que não fazem parte do âmbito do projeto. Este risco foi identificado e a maneira encontrada para o combater foi o uso da matriz de Wieggers (apresentada na secção 3.1). Foi assim possível priorizar requisitos e não perder tempo desnecessário com requisitos pouco importantes, que não constituíam valor para o cliente.

Atrasos no projeto são prováveis quando não se conhece bem as tecnologias envolvidas. Este tipo de risco foi identificado desde o início deste projeto nomeadamente, na utilização das *frameworks* de comunicação entre servidor e aplicação cliente e na tarefa de desenvolvimento de mapas. Para este último risco, foi desde logo identificado um conjunto de hipóteses que por ordem da melhor hipótese, para a pior, se decidiu ordenar: uso de APIs (Applications Programming Interfaces) dinâmicas de mapas; uso de APIs *web* de mapas; uso de imagens estáticas já tratadas pelo Web service, tornando o plano de gestão de risco com várias hipóteses alternativas, caso ocorressem falhas.

A existência do envolvimento de outros projetos a nível empresarial também foi desde logo considerada um risco, uma vez que existem momentos específicos que se tem de conceder prioridade a um projeto em detrimento de um outro. O plano para este risco foi de fazer os projetos em paralelo, reunindo esforços para o que fosse mais urgente, mas não esquecendo o segundo. Este risco não é fácil de gerir e os planos encontrados para minimizar este risco nem sempre são os melhores na altura que acontece, visto que por vezes a prioridade a ser dada a um projeto é numa altura delicada na vida de outro projeto, no entanto existem prioridades na própria empresa que têm de se respeitar.

Como se sabe os riscos são constantes e os imprevistos acontecem sempre em qualquer projeto, tornando assim a gestão de risco uma tarefa muito delicada. Sempre que era feita uma revisão do planeamento fazia-se também uma nova análise de risco, de modo a precaver novos riscos.

3.2.4 Gestão de Recursos

Os recursos utilizados para o projeto são essencialmente de dois tipos: humanos e materiais. Nos recursos humanos encontram-se os elementos constituintes da equipa de trabalho e os *stakeholders* diretamente envolvidos. A equipa de trabalho foi constituída por quatro elementos, sendo que os mesmos tiveram diferentes perfis. Na Tabela 2 são apresentados os recursos humanos e o seu respetivo perfil.

Recurso Humano	Perfis
João Domingos	Programador Estagiário
Pedro Marques	Gestor de Projeto
David Mendes	Programador Orientador de estágio na Tecmic S.A.
Vítor Basto	Orientador de estágio na ESTG-IPL

Tabela 2 - Recursos humanos e perfil de cada recurso.

Os recursos humanos apresentados foram os que constituíram a equipa de trabalho de todo este projeto de estágio, contribuindo cada um para o sucesso do mesmo através do perfil desempenhado por cada recurso.

No que diz respeito aos recursos materiais, estes foram essencialmente três, não esquecendo porém do recurso da energia elétrica, tipicamente tão útil e necessário, mas por vezes esquecido. Na Tabela 3 são apresentados os recursos materiais e a sua respetiva descrição.

Recurso Material	Descrição
Computador de Secretária	Computador utilizado para desenvolvimento
Telemóvel Inteligente	Telemóvel utilizado para desenvolvimento
Servidor	Computador utilizado para alojamento de Web services

Tabela 3 - Recursos materiais e sua descrição.

De notar, que a empresa já tinha o computador de secretária e o servidor, havendo os naturais custos de manutenção e licenciamento de *software*. No entanto adquiriu-se especificamente para este projeto um telemóvel inteligente, tendo este um custo extra para o projeto.

3.2.5 Gestão da Comunicação

A gestão da comunicação corresponde a um conjunto de processos de comunicação que são necessários para assegurar a criação, disseminação, armazenamento e disponibilização apropriada e atempada de informação acerca do projeto.

A comunicação utilizada pelos vários recursos humanos para enviar e receber informação pode ser muito variada. É através de uma boa identificação de recursos e de comunicações que se visa determinar quais as necessidades por parte dos recursos humanos do projeto e os processos de comunicação entre eles: quem precisa de informação, de que informação necessita, quando esta necessita de estar disponível, como será fornecida e por quem. Os métodos e tecnologias de comunicação utilizados neste projeto entre a equipa de trabalho são

descritos na Tabela 4.

Canal de Comunicação	Periodicidade	Tipo de Informação Transmitida	Recurso(s) Humano(s)
Reunião Estágio	Cada duas semanas	Ponto de situação do relatório; Definição de objetivos para a reunião seguinte.	João Domingos e Vítor Basto
Reunião Tecmic	Semanalmente	Ponto de situação do projeto; Definição de objetivos para a reunião seguinte; Atribuição de tarefas a cada elemento da equipa.	Equipa de desenvolvimento
Agenda	Obrigatório (anterior à reunião Tecmic)	Levantamento de tarefas atribuídas na reunião anterior; Registo de ações novas que serão analisadas na reunião seguinte.	David Mendes
Ata	Obrigatório (posterior à reunião Tecmic)	Levantamento de dados e factos referidos na reunião; Registo de todas as decisões tomadas na reunião.	David Mendes
<i>Online</i> (mensagens eletrónicas)	Diariamente	Mensagens curtas para esclarecimento de pequenas dúvidas existentes.	Equipa de desenvolvimento
Correio eletrónico	Semanalmente	Mensagens formais para registar momentos importantes do projeto.	Equipa de trabalho
Telefone	Semanalmente	Mensagens informais para dar o conhecimento de novos dados.	Equipa de desenvolvimento

Tabela 4 - Comunicação entre a equipa de trabalho.

É importante referir que não basta apenas elaborar um conjunto de métodos de comunicação, mas também cada elemento deve ir para a reunião previamente preparado para que não se percam tempos desnecessários. Essa preparação passa pela agenda disponibilizada com antecedência para ser devidamente analisada, para deste modo se ter um conhecimento de quais os tópicos que irão ser abordados na reunião. Durante a reunião deve-se falar apenas dos assuntos que originaram a mesma e em nada mais, tirando notas dos factos referidos para uma posterior análise.

O capítulo seguinte descreve em detalhe o desenvolvimento do projeto Mobile Fleet Management, justificando as opções tomadas durante o mesmo.

Desenvolvimento

Este capítulo apresenta o desenvolvimento do projeto, começando por mostrar de forma justificada as tecnologias utilizadas, de acordo com os seus benefícios para o projeto. De seguida é exibida a arquitetura do sistema informático iZiTraN em conjunto com a integração do projeto Mobile Fleet Management que foi desenhado de forma modular permitindo a sua escalabilidade e adaptação à evolução do processo de negócio no setor da gestão de frotas. O capítulo apresenta também o modelo de domínio onde foi analisado o processo de negócio deste setor, sendo ainda descritos com detalhe o desenvolvimento da aplicação servidora iZiTraN Mobile Services e da aplicação cliente iZiTraN Mobile.

4.1 Enquadramento Tecnológico

Entre os principais sistemas operativos para *smartphones*, a empresa optou que a aplicação móvel fosse desenvolvida nativamente para a plataforma Android [46] [47] com utilização em versões iguais ou superiores à versão KitKat 4.4 [48]. Esta plataforma é baseada no *kernel* Linux [49] para dispositivos móveis, sendo desenvolvida pela Open Handset Alliance [50], liderada pela Google [51]. Tem como objetivo ser uma plataforma flexível, aberta e de fácil migração para as diferentes marcas de dispositivos.

A principal razão desta escolha é o facto do sistema operativo da Google controlar atualmente a maior quota de mercado nesta área segundo um estudo realizado em 2014 pela International Data Corporation (IDC) [52]. A este estudo juntam-se vários outros, feitos por várias entidades, confirmando que desde 2011 a plataforma Android tem mantido a liderança e ao mesmo tempo a sua quota de mercado tem vindo a crescer. Numa análise mais focada no desenvolvimento para se conhecer as preferências dos programadores verificou-se que o Android também lidera nas preferências dos mesmos, como revela o estudo realizado em 2014 pela VisionMobile [53] a mais de 10.000 programadores de aplicações nas diversas plataformas. Pesou também o facto de a equipa de desenvolvimento já ter adquirido algumas bases de desenvolvimento nesta plataforma, assim como o suporte contínuo e evolutivo que a

Google fornece aos programadores desta. Outro motivo é a facilidade com que se programa para a plataforma, a forma como são construídos os *layouts* dos ecrãs, o modo de fazer a instalação para o *smartphone*, a experiência de utilização da aplicação e ainda a instrumentação para a realização de testes automáticos muito intuitiva.

Como a plataforma escolhida foi Android optou-se pelo uso da ferramenta Android Studio IDE [54] para proceder ao desenvolvimento da aplicação móvel iZiTraN Mobile, enquanto para os Web services desenvolvidos foi utilizada a ferramenta Microsoft Visual Studio [55] adotando a *framework* ASP.NET [56]. A equipa de desenvolvimento foi novamente um fator decisivo na escolha da *framework* ASP.NET devido à enorme experiência já adquirida na área, uma vez que muitos dos produtos existentes na empresa são desenvolvidos através desta.

4.1.1 Repositórios

Em relação ao código criado, este foi armazenado recorrendo ao repositório de controlo de versões Apache Subversion (SVN) [57] da própria Tecmic S.A.. O uso de controlo de versões é sempre útil em qualquer desenvolvimento de *software*, uma vez que permite a diversos programadores trabalharem sobre o mesmo conjunto de ficheiros ao mesmo tempo e minimiza o desgaste provocado por problemas com conflitos de edições. A maioria das implementações possibilita a divisão do projeto em várias linhas de desenvolvimento, que podem ser trabalhadas paralelamente, sem que uma interfira com a outra. Também existe um controlo de histórico que permite analisar o registo temporal dos desenvolvimentos, como também facilidade no resgate de versões mais antigas e estáveis. A maioria das implementações permitem analisar as alterações com detalhe, desde a primeira versão até a última. Tanto no sistema da Tecmic S.A., mas também na maioria dos sistemas SVN, estes permitem marcar a altura em que o ficheiro estava com uma versão estável, podendo ser facilmente recuperado no futuro.

No projeto foi também utilizado Apache Maven [58], uma ferramenta de automação de compilação. Através do repositório Maven da Tecmic, baseado no servidor Sonatype Nexus [59], é possível armazenar projetos Maven, bem como projetos NuGet [60]. As vantagens em utilizar Maven neste projeto, assim como para outros projetos, é que ele faz a gestão das dependências existentes, fazendo a gestão de todo o ciclo de vida do projeto e das etapas para construção do mesmo, que por padrão são limpar, compilar, testar, empacotar e distribuir. O Maven é útil porque faz a gestão de dependências e evita a existência da extensa e desorganizada pasta de bibliotecas, ou ainda a necessidade em passar bibliotecas para outros elementos da equipa, e a correspondente dificuldade no controlo de versões. O Android

Studio IDE utiliza Gradle [61], sistema próprio do IDE para automatização de *builds*, que com a simples adição do *plugin* Maven no ficheiro de configurações Gradle é possível obter todas as vantagens que o Maven disponibiliza. Com este *plugin* apenas é necessário identificar os repositórios que terão a biblioteca requerida e os identificadores da mesma (*group id*, *artefact id* e *version*) para que esta seja incluída na estrutura do projeto. Com a configuração realizada o Maven tenta encontrar primeiramente a biblioteca no repositório local, presente no computador, se não a encontrar acede ao repositório Maven da empresa e faz o *download* para o repositório local. Se a biblioteca também não estiver no repositório da empresa, o Maven tenta ainda procurar no seu repositório oficial, o Maven Central, e faz o *download* para o repositório local. Para fazer alterações de versão, basta modificar a mesma no ficheiro de configurações Gradle e submeter para o repositório. Deste modo, quem estiver a utilizar versões mais antigas não sofre quaisquer consequências, bastando apenas mudar para a versão mais recente quando assim for necessário.

4.1.2 Comunicação

No que diz respeito à comunicação, foi utilizada a arquitetura Representational State Transfer (REST) [62], sendo que para esta escolha pesou o facto de a equipa de desenvolvimento ter experiência adquirida. REST é simples de entender e pode ser adotado em praticamente qualquer cliente ou servidor com suporte a HTTP/HTTPS [63] [64]. Os programadores que o utilizam citam, como principais vantagens a facilidade no desenvolvimento, o aproveitamento da infraestrutura *web* existente e um esforço de aprendizagem pequeno. Sendo a maior vantagem da arquitetura REST a sua flexibilidade e o facto de dar a possibilidade ao programador de optar pelo formato mais adequado de acordo com as necessidades específicas. Os formatos mais comuns são JavaScript Object Notation (JSON), eXtensible Markup Language (XML) e texto puro, mas em teoria qualquer formato pode ser usado [65] [66]. Isso leva a outra vantagem, uma vez que quase sempre os Web services que usam o formato JSON (que é o utilizado) são mais “leves” isto é, necessitam de menor largura de banda, que é bastante útil nas comunicações em dispositivos móveis. No entanto, é de realçar o crescimento da utilização dos protocolos binários, assunto muito debatido na temática Internet of Things (IoT) [67] [68].

No projeto existem comunicações típicas unidirecionais (um sentido), mas também há a necessidade de comunicação em tempo real, requerendo-a de forma bidirecional (dois sentidos) entre a aplicação e cada Web service, como tal, de forma a agilizar o desenvolvimento para ambas as comunicações aplicaram-se as seguintes *frameworks*: Volley,

SignalR e ASP.NET Web API. A *framework* Volley [69] [70] é uma biblioteca HyperText Transfer Protocol (HTTP) que faz *networking* para aplicações Android, de forma a tornar a comunicação mais rápida e a facilitar a sua implementação. Esta biblioteca foi apresentada no evento Google I/O 2013 [71] e atualmente existe em muitas aplicações Android feitas pela própria Google. Com esta biblioteca é possível agilizar a implementação de pedidos de rede, utilizando várias ligações de rede simultâneas, através de pedidos que podem ser definidos em níveis de prioridade, bem como uma melhor gestão de memória cache. Por exemplo, para repetição e recuo na ligação, ordenando corretamente os pedidos, tornando-os mais fáceis no correto preenchimento das *interfaces* gráficas com os dados obtidos de forma assíncrona a partir da rede. O Volley destaca-se assim, em operações do tipo Remote Procedure Call (RPC) [72], com possibilidade de utilização para preenchimento de componentes interativos (listas, mapas, imagens, etc.), como por exemplo para obtenção de resultados de pesquisa com dados estruturados, integrando-se facilmente com qualquer protocolo [73]. Esta biblioteca tem suporte embutido para os recursos que são necessários para a comunicação, permitindo a redução de *boilerplate code* [74], dando a oportunidade ao programador de apenas se concentrar na lógica da aplicação.

Na aplicação móvel iZiTraN Mobile recorreu-se ao uso de Volley e no Web service iZiTraN Mobile Service utilizou-se a *framework* ASP.NET Web API [75] [76] para comunicar com a mesma. A Web API é uma das *frameworks* mais utilizadas na empresa, com ela torna-se mais fácil construir serviços HTTP que chegam a uma ampla gama de clientes, incluindo *Web browsers* e dispositivos móveis, permitindo que os pedidos sejam independentes da plataforma. Com a Web API pode negociar-se a representação de dados, ou seja, se o cliente está a solicitar dados a serem retornados como JSON ou XML, a Web API lida com o tipo de solicitação e retorna os dados de forma apropriada com base no tipo pretendido. O cliente pode assim, fazer um pedido GET, PUT, POST ou DELETE e obter a resposta ao pedido REST que será transmitida via HTTP/HTTPS.

Para corresponder às necessidades de todos os clientes consumidores da comunicação em tempo real, foi utilizada a *framework* SignalR [77], com uso tanto na aplicação, como no Web service iZiTraN Mobile RealTime. Esta escolha deveu-se ao facto de assim ser possível manter um certo volume de ligações no servidor, manipular a forma com que o servidor trata as requisições, garantir que o cliente esteja a receber dados no tempo correto e assim minimizar as dificuldades que a comunicação em tempo real pode provocar. Esta biblioteca tem código aberto, uma enorme facilidade na implementação de comunicação em tempo real e uma forma simples de tratar atualizações/notificações de forma assíncrona. Existem

algumas técnicas/tecnologias conhecidas de implementar uma comunicação em tempo real (WebSockets, Long Polling, entre outras), sendo que possuem uma complexidade ou limitação técnica. O SignalR propõe facilitar a sua implementação, uma vez que fica a cargo dele próprio encontrar o transporte mais eficaz entre um servidor e cliente. Sempre que a tecnologia WebSockets estiver disponível, é-lhe dada preferência, contudo se não existir esta disponibilidade o SignalR opta pelo Long Polling e as demais em sequência. O WebSocket [78] é uma tecnologia recente, estando a sua API já numa fase avançada para se tornar um padrão pelo World Wide Web Consortium (W3C) [79] e o seu protocolo está também a tornar-se padrão pelo Internet Engineering Task Force (IETF) [80], comunidade internacional preocupada com a evolução da arquitetura da Internet e do seu perfeito funcionamento. Esta tecnologia permite a comunicação bidirecional por canais *full-duplex* persistente e dedicado sobre um único *socket* Transmission Control Protocol (TCP) [81], cujo objetivo é obter um mecanismo para aplicações que necessitam de comunicação bidirecional sem terem que depender da abertura de múltiplas ligações HTTP. Isto significa que a comunicação entre cliente e servidor pode ocorrer nos dois sentidos, em simultâneo e assincronamente, e que a ligação é mantida aberta até que uma das partes envie, explicitamente, um pedido para o fecho da ligação. O servidor pode comunicar com o cliente assim que os dados forem alterados, sem ser necessário esperar por qualquer pedido do cliente. Existe também outra técnica que permite simular uma comunicação bidirecional, denominada Polling. Esta técnica consiste no envio periódico de requisições para o servidor para permitir que ele envie novos dados, caso existam. Desta maneira, pode-se simular uma ligação entre dois sentidos já que, do ponto de vista do utilizador, existem novos dados que lhe aparecem sem que ele tenha que explicitamente fazer uma nova requisição por eles. Esta técnica, entretanto, não é eficiente em termos de recursos computacionais, uma vez que serão realizadas inúmeras requisições inúteis quando o servidor não contém novos dados para responder. Também não representa uma via de comunicação de tempo real, já que quando houver um novo dado no servidor, ele só irá apresentar-se para o utilizador depois de decorrido o intervalo até à próxima requisição periódica da aplicação cliente. A técnica, que também existe no SignalR, denominada de Long Polling [82], consiste numa variação do conceito de Polling, porém cada requisição efetuada pelo cliente é mantida aberta até que o servidor tenha algum dado para enviar. Após isto, a ligação é fechada e o cliente deve estabelecer uma nova ligação. A vantagem neste caso é que não há requisições desperdiçadas enquanto o servidor não tem novos dados, uma vez que a ligação entre o cliente e o servidor se mantém aberta por um determinado período de tempo, até que seja enviada uma resposta através deste último. Como se disse anteriormente,

fica a cargo do SignalR encontrar o transporte mais eficaz entre um servidor e cliente, dando sempre preferência por WebSockets, seguindo-se pelo Long Polling e os demais em sequência, facilitando assim a implementação da comunicação em tempo real.

4.1.3 Mapas

Os mapas foram utilizados na aplicação móvel iZiTraN Mobile por forma a facilitar ao operador de frota a visualização da localização geográfica dos seus veículos. Como requisito, houve a procura da melhor solução para utilização de mapas *open source* que permitisse a utilização do serviço Web Map Service (WMS) [83] [84] que é o serviço utilizado pela cartografia Tecmic. Em conjunto com o requisito anterior foi pretendida a possibilidade de mudar a fonte de cartografia de uma maneira bastante simples, dando assim possibilidade da sua utilização noutras fontes, como Open Street Map [85], Google Maps [86], Bing Maps [87], entre outros.

Por forma a encontrar a melhor solução para o requisito descrito, foram estudadas várias *frameworks*: API Google Maps, gvSIG Mini Maps, Androzic, OpenLayers, OSMDroid, entre outras. Sendo o mapa para uso num contexto empresarial, a API Google Maps [88] não se apresentou como sendo uma solução viável, devido às restrições impostas pelo uso da mesma neste contexto [89]. A *framework* gvSIG Mini Maps [90] da empresa Prodevelop [91] foi vista durante muito tempo como a aposta ideal. Esta *framework* despertou interesse devido ao seu vasto conjunto de funcionalidades e pela sua disponibilização *open source* com uso de WMS, no entanto encontrou-se muitas dificuldades, uma vez que a mesma foi descontinuada, existindo pouca e fraca documentação, mas também pelos imensos erros de dependências para outras *frameworks*, originando assim um elevado custo na sua utilização. Foram feitas tentativas de comunicação com os seus programadores e com a sua comunidade, mas nunca houve uma resposta concreta, sendo possível verificar que existiam muitos utilizadores com os mesmos problemas. Já a *framework* Androzic [92] apresentou-se como sendo uma boa opção, tendo também um conjunto apreciável de boas funcionalidades (pontos de interesse com distância e direção, navegação, entre outras), no entanto não disponibiliza o serviço WMS, uma vez que utiliza o serviço Tile Map Service (TMS) [93]. Por este motivo tentou-se colocar a cartografia Tecmic a ser disponibilizada pelo serviço TMS, configurando o servidor GeoServer [94] da empresa para disponibilizar este serviço. Depois de configurado o GeoServer, o mapa foi corretamente apresentado na aplicação, no entanto este serviço apresentava alguma lentidão adicional, levando a que a *framework* fosse descartada. A *framework* OpenLayers [95] [96] também foi testada e funcionou bem, cumprindo os

requisitos pretendidos de *open source* e de serviço WMS. No entanto, apenas foi considerada uma hipótese válida se todas as restantes falhassem, uma vez que se trata de uma opção que não é nativa da plataforma Android e requer o uso adicional de uma *framework* do género de PhoneGap [97]. Também o facto de para ser corretamente implementada em dispositivos móveis tenha de se utilizar JavaScript com WebView, que tem alguns problemas de performance conhecidos o que tornaria a aplicação mais complexa e mais suscetível a falhas [98] [99].

Após muitas horas, dias, de estudo deu-se atenção à *framework* OSMDroid [100], no entanto já anteriormente tinha havido uma abordagem à mesma. Esta *framework* contrasta com a *framework* gvSIG Mini Maps, uma vez que tem uma comunidade ativa e com melhor documentação, contrastando também com a *framework* API Google Maps por ser inteiramente *open source*. Numa fase inicial foi colocada de parte por não disponibilizar o serviço WMS. Nesta nova abordagem, foi criado um mecanismo de WMS que consegue disponibilizar a cartografia da Tecmic e ao mesmo tempo usufruir das funcionalidades que a mesma oferece. Para tal, teve de se criar um adaptador personalizado para disponibilizar este serviço à *framework*, fazendo deste um adaptador universal que permite que a *framework* consuma vários serviços de mapas e assim possa ser utilizada, de forma simples, por várias fontes. Deste modo, encontrou-se na *framework* OSMDroid a melhor solução tanto para o requisito descrito, como também por se tratar de uma opção nativa para a plataforma Android.

4.2 Enquadramento Arquitetural do iZITraN

Como é natural a arquitetura que este projeto envolve teve que ser integrada na arquitetura iZiTraN existente, onde se encontram todos os sistemas da empresa. Esta integração envolveu um estudo sobre a arquitetura existente, sendo que deste modo se reconheceu que a mesma foi progredindo ao longo dos anos de existência da empresa.

A infraestrutura de base é constituída pelos vários dispositivos, terminais Personal Digital Assistant (PDA) e equipamentos Tecmic presentes nos veículos da frota, bem como os terminais PDA utilizados pelas equipas no terreno que comunicam através do serviço GPRS. É usado o serviço GPRS sobre as redes GSM existentes, cuja interligação à Internet possibilita o encaminhamento de dados ao servidor de comunicações (Communication Server), para que este trate das comunicações recebidas e as envie pela rede interna da Tecmic S.A. ao servidor aplicacional (Application Server). O Application Server assenta numa Service-Oriented Architecture (SOA) [101] que contém lógica de negócio distribuída, sendo responsável por centralizar toda a informação recebida, desde equipamentos Tecmic,

terminais PDA, como de outras integrações. Este utiliza a plataforma .NET, tendo a capacidade de alta disponibilidade, balanceamento de carga, redundância de informação, tolerância a falhas, entre outras.

A partir de uma certa altura a empresa sentiu a necessidade de desenvolver produtos *desktop*. Como tal a arquitetura progrediu para corresponder às necessidades que se foram criando, acrescentando assim, valências ao Application Server para que estes produtos pudessem interoperar e corresponder às necessidades da empresa. Seguidamente a empresa progrediu para soluções *web* (iZiTraN Web, entre outras), sendo os pedidos dos clientes enviados ao servidor *web* (Web Server) e deste comunicados ao Application Server. Na Figura 15 é apresentado o esquema da arquitetura do sistema informático iZiTraN antes do início do projeto Mobile Fleet Management.

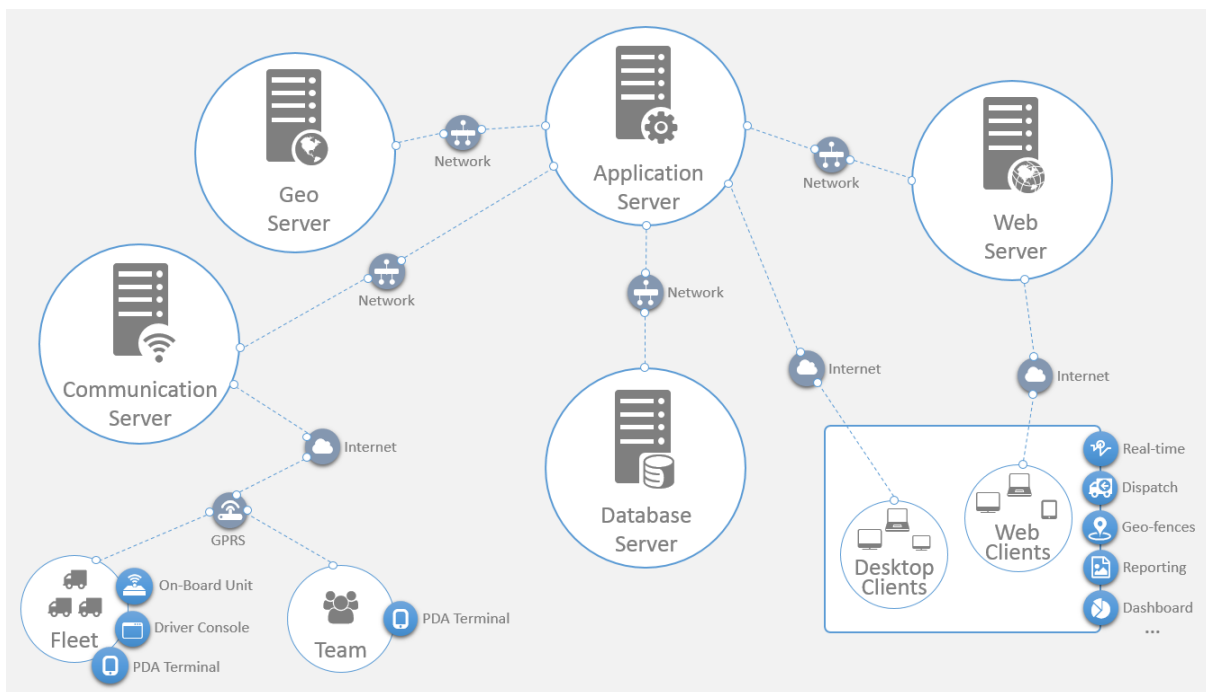


Figura 15 - Arquitetura do sistema informático iZiTraN antes do projeto.

O projeto Mobile Fleet Management focou-se no desenvolvimento de uma aplicação móvel, e consequente disponibilização para uso em dispositivos móveis (Mobile Clients), tornando-se estes em equipamentos adicionais consumidores da informação. Estes novos equipamentos comunicam com o servidor móvel (Mobile Server), criado especificamente para este projeto, que interage com o Application Server existente. De notar que não se utilizou o Web Server existente (utilizado pelo iZiTraN Web) por este não estar ajustado às necessidades da aplicação em dispositivos móveis, devido ao forte impacto de tempo real que esta requer.

O projeto teve a vantagem de ter as aplicações cliente-servidor desenvolvidas de raiz, no entanto houve necessidade de integrar o *software* desenvolvido na arquitetura já existente, em particular com as componentes associadas a equipamentos embarcados em veículos, que comunicam com o Application Server, e através destes disponibilizam informação aos dispositivos móveis. A Figura 16 apresenta o esquema da nova arquitetura descrita.

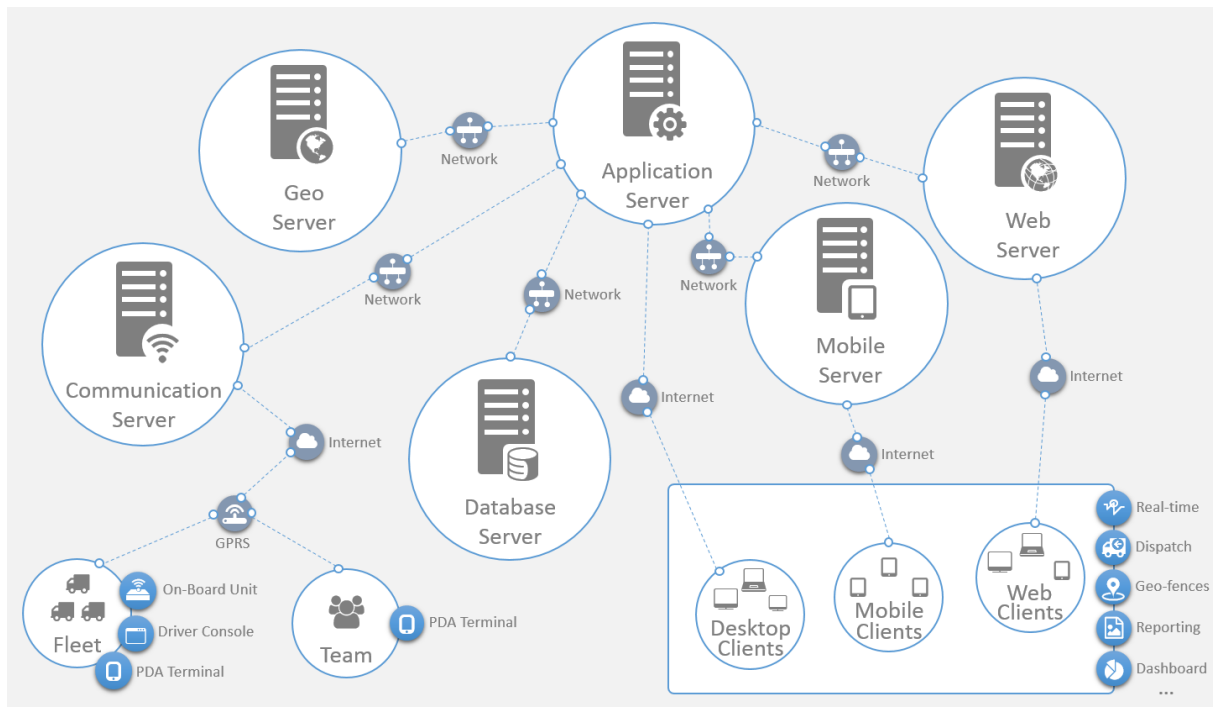


Figura 16 - Arquitetura do sistema informático iZiTraN com o Mobile Fleet Management integrado.

Na Figura 16 é possível observar que com a existência do projeto Mobile Fleet Management passaram a existir clientes móveis que contêm a aplicação iZiTraN Mobile a comunicar com o Mobile Server. Este servidor representa os Web services (iZiTraN Mobile Service e iZiTraN Mobile RealTime) que foram desenvolvidos para fazer a ponte entre os clientes móveis e o Application Server que disponibiliza os dados para serem consumidos.

4.3 Modelo de Domínio

O modelo de domínio foi desenhado para ser capaz de suportar a lógica do processo de negócio do projeto Mobile Fleet Management.

Numa fase inicial houve a ideia de adaptar o modelo existente do iZiTraN Web e adicionar apenas as novas entidades. No entanto, esta abordagem foi descartada após um estudo à mesma, uma vez que para além do tempo que seria necessário para compreender o modelo

existente e de quais as implicações da alteração de cada entidade, existia ainda toda a complexidade de dependências entre projetos (existentes no Visual Studio) que tornariam esta abordagem pouco viável. Um outro motivo que levou a descartar esta abordagem foi o facto de a lógica presente no iZiTraN Web enviar várias vezes informação redundante e desnecessária, tornando-o inviável para a aplicação móvel iZiTraN Mobile, uma vez que esta precisa de fazer comunicações que gastem poucos dados, tendo os mesmos de vir tratados para que não se tenha de fazer muito processamento no dispositivo móvel. Assim, foi desenhado um novo modelo de domínio capaz de suportar toda a lógica do Mobile Fleet Management, tendo sido estruturado de forma a ser o mais estável, flexível, extensível, manutenível e escalável possível, para que no futuro se possa adaptar a eventuais mudanças na lógica de negócio, sem prejuízo da lógica específica. O diagrama geral do modelo de domínio é apresentado na Figura 17.

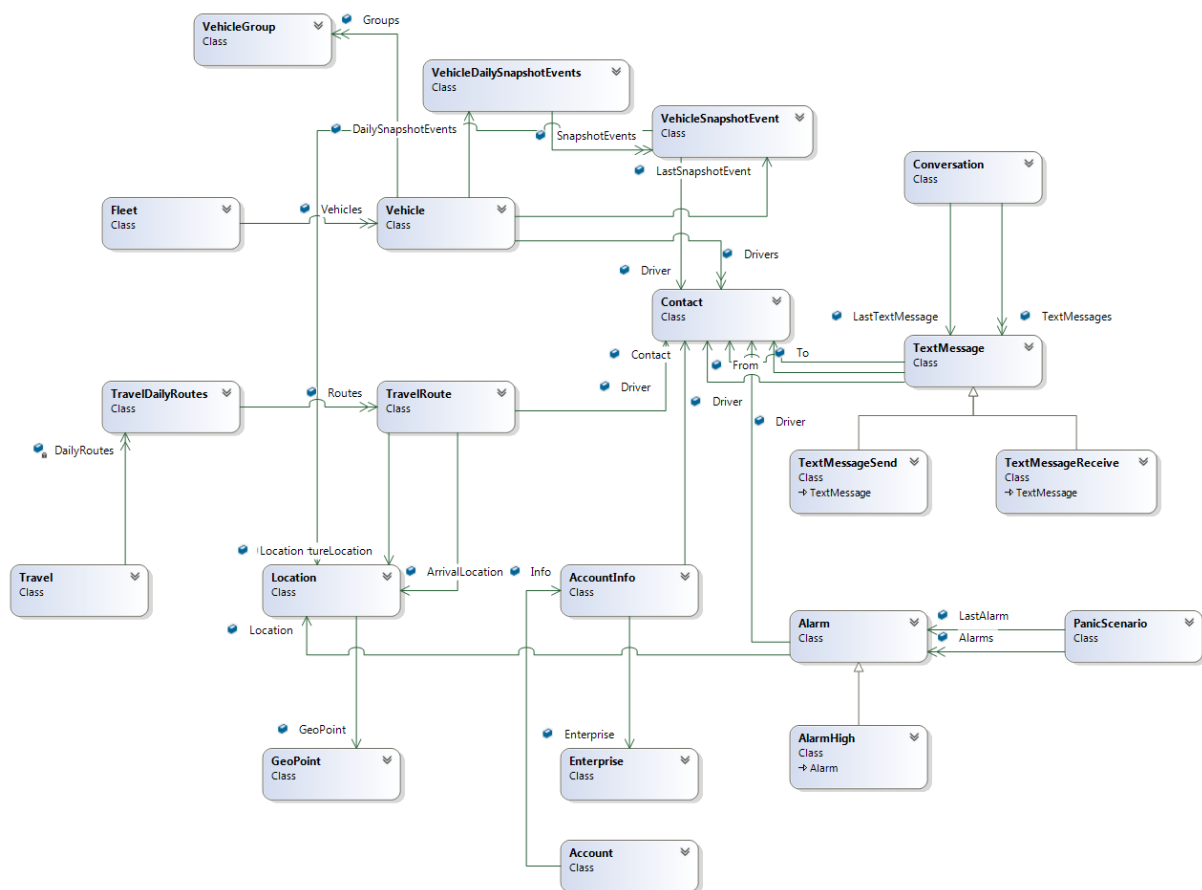


Figura 17 - Diagrama do modelo de domínio.

O elevado número de relações entre as classes do modelo de domínio revela a complexidade do processo de negócio apresentado na Figura 17. Por este motivo, esta secção fragmenta a

Figura 17, dando a conhecer o modelo e as decisões associadas. Esta secção não tem o intuito de dar a conhecer em detalhe todos os atributos das entidades mas, sim, descrever os atributos que compõem relações entre as diversas entidades do modelo de domínio:

- Account: esta entidade representa a conta do utilizador no sistema. As dependências desta entidade são apresentadas na Figura 18.



Figura 18 - Entidade Account.

A gestão de contas é importante para permitir a identificação do utilizador no sistema. A conta tem várias informações vinculadas a si, conforme representa o atributo Info na conta.

- AccountInfo: entidade que representa a informação abrangida pela conta, cujas dependências são apresentadas na Figura 19.

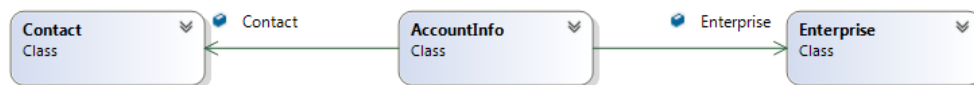


Figura 19 - Entidade AccountInfo.

A entidade AccountInfo não representa o conceito de conta, mas sim da informação associada à mesma. A informação da conta tem um contacto que identifica o utilizador, assim como a empresa a que esta pertence.

- Contact: esta entidade representa o contacto a partir do qual o tipo de contacto (Operador de Frota, Condutor, Veículo, Terminal) pode ser identificado e contactado. Esta entidade não depende de nenhuma outra do modelo de domínio.
- Enterprise: entidade que representa a empresa. Esta entidade não depende de nenhuma outra do modelo de domínio.

Como a lógica de negócio assenta sobre a gestão de frotas houve a necessidade da criação da entidade frota.

- Fleet: esta entidade representa a frota de veículos. As dependências desta entidade são apresentadas na Figura 20.



Figura 20 - Entidade Fleet.

A frota é um agrupamento de veículos, sendo estes um atributo fundamental em todo o modelo de domínio.

- Vehicle: entidade que representa o veículo, cujas dependências são apresentadas na Figura 21.

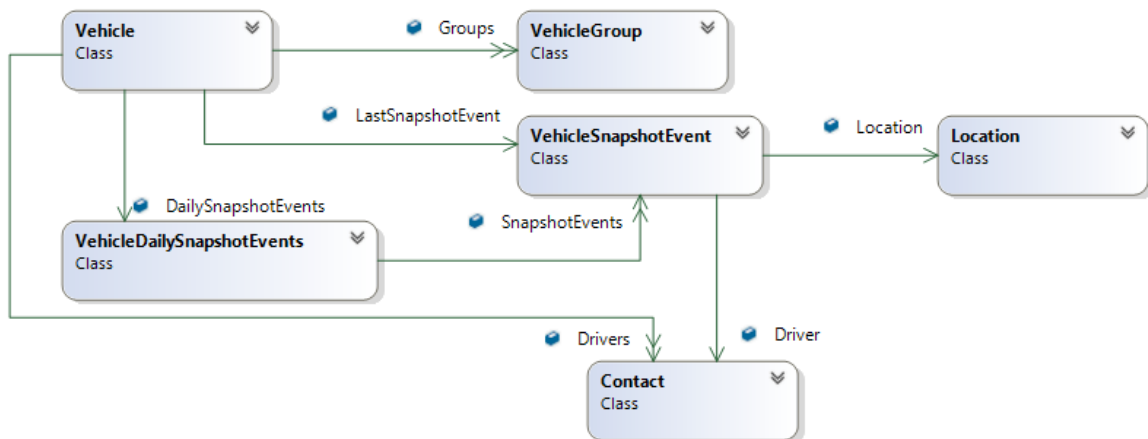


Figura 21 - Entidades Vehicle, VehicleGroup, VehicleDailySnapshotEvents e VehicleSnapshotEvent.

Um veículo pode pertencer a vários grupos que são criados para facilitar a sua correta organização por zonas. O veículo tem estatísticas diárias que são totalizadas através dos eventos ocorridos, pelo mesmo, ao longo do dia.

- VehicleGroup: esta entidade representa o grupo de veículos. Esta entidade não depende de nenhuma outra do modelo de domínio.
- VehicleDailySnapshotEvents: entidade que representa as estatísticas diárias do veículo. Tem o atributo SnapshotEvents, que cria uma dependência com a entidade VehicleSnapshotEvents, com o intuito de criar as estatísticas diárias.
- VehicleSnapshotEvent: esta entidade representa os indicadores instantâneos do veículo, como estado, velocidade instantânea, percentagem de combustível, condutor ativo, entre outros. Esta entidade tem dependência para Contact contendo o condutor e para Location.
- Location: entidade que representa a localização de um veículo, podendo ser de um dado instante, pontos de uma viagem ou de um alarme ocorrido. As dependências desta entidade geográfica são apresentadas na Figura 22.

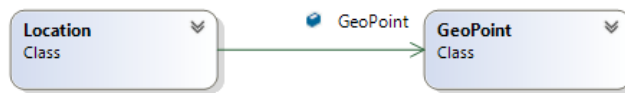


Figura 22 - Entidade Location.

Uma localização tem posição descritiva e coordenadas geográficas para representar a mesma.

- GeoPoint: esta entidade representa o ponto de localização na forma de coordenadas geográficas através da latitude e longitude. Esta entidade não depende de nenhuma outra do modelo de domínio.

Como os veículos efetuam ao longo do dia várias viagens houve a necessidade da criação da entidade viagem.

- Travel: entidade que representa a viagem efetuada por um veículo, cujas dependências são apresentadas na Figura 23.

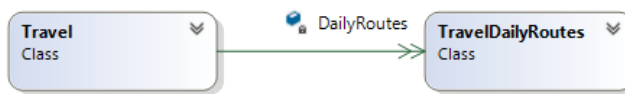


Figura 23 - Entidade Travel.

Os veículos efetuam viagens todos os dias, fazendo com que a entidade viagem seja um agrupamento de viagens diárias.

- TravelDailyRoutes: esta entidade representa a viagem diária que um veículo faz. As dependências desta entidade são apresentadas na Figura 24.

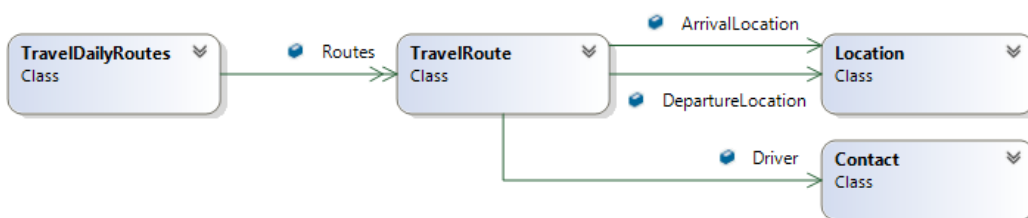


Figura 24 - Entidade TravelDailyRoutes e TravelRoute.

As viagens diárias são um agrupamento dos percursos (trajetos efetuados com a ignição ligada) efetuados pelos veículos naquele dia.

- TravelRoute: entidade que representa o percurso efetuado. Esta entidade depende da entidade Location para conhecer o local de chegada e destino, bem como da Contact para conhecer o condutor que efetuou aquele percurso.

Como os veículos podem ao longo do dia desencadear vários alarmes houve a necessidade da criação da entidade cenário de pânico.

- PanicScenario: esta entidade representa o cenário de pânico, sendo utilizada para agrupar alarmes, cujas dependências são apresentadas na Figura 25.



Figura 25 - Entidade PanicScenario.

O cenário de pânico é um agrupamento dos alarmes desencadeados pelos veículos. Esta entidade tem ainda o último alarme desencadeado.

- Alarm: entidade que representa o alarme desencadeado por um veículo. As dependências desta entidade são apresentadas na Figura 26.
- AlarmHigh: esta entidade representa o alarme com alta prioridade. Esta entidade herda de Alarm e tem como particularidade o facto de necessitar de ser tratado.

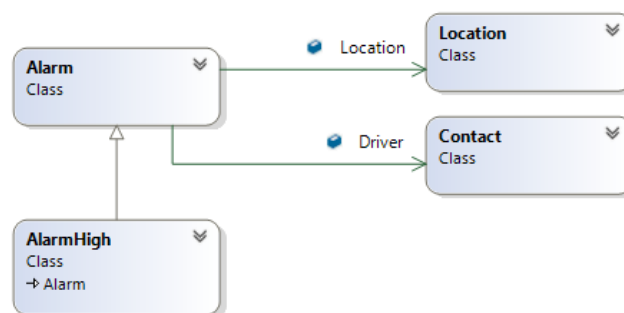


Figura 26 - Entidade Alarm e AlarmHigh.

Como os veículos e condutores são contactáveis, sendo que por este motivo podem enviar e receber mensagens de texto, houve a necessidade da criação da entidade conversa.

- Conversation: entidade que representa a conversa trocada através de mensagens, cujas dependências são apresentadas na Figura 27.

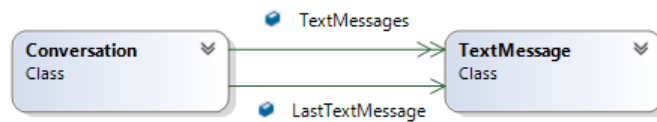


Figura 27 - Entidade Conversation.

A conversa é um agrupamento das mensagens enviadas e recebidas. Esta entidade tem ainda a última mensagem da conversa.

- TextMessage: esta entidade representa a mensagem de texto. As dependências desta entidade são apresentadas na Figura 28.
- TextMessageSend: entidade que representa a mensagem enviada. Esta entidade herda de TextMessage e tem como particularidade o facto ter os estados: criada, guardada, enviada, entregue, vista e de erro, bem como as datas de cada estado.
- TextMessageReceive: esta entidade representa a mensagem recebida. Esta entidade herda de TextMessage e tem como particularidade o facto ter os estados: recebida, não lida e lida, bem como as datas de cada estado.

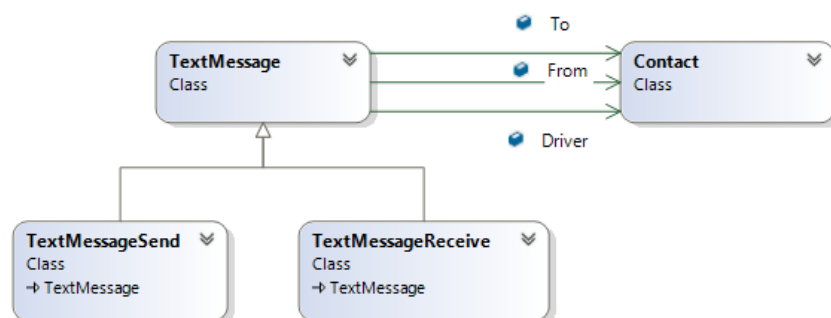


Figura 28 - Entidade TextMessage, TextMessageSend e TextMessageReceive.

Sendo o modelo estruturado de forma a ser o mais estável, flexível, extensível, manutenível e escalável possível, podem várias aplicações partilhar o mesmo modelo de domínio. A utilização do modelo de domínio originou um diagrama de classes, que depois de implementado deu origem à biblioteca iZiTraN Mobile Domain.

4.4 Servidor

As funcionalidades do projeto implicaram a existência de uma aplicação servidora, tendo esta sido construída na forma de um conjunto de Web services, apelidado por iZiTraN Mobile Services, como é apresentado nas seguintes subsecções.

4.4.1 Arquitetura

O iZiTraN Mobile Services tem incluído o iZiTraN Mobile Service para os serviços pedidos pela aplicação cliente e o iZiTraN Mobile RealTime para os serviços de tempo real, tendo ambos sido criados para serem utilizados pelos serviços consumidos pela aplicação cliente iZiTraN Mobile. O enquadramento arquitetural do iZiTraN Mobile Services é apresentado através da Figura 29.

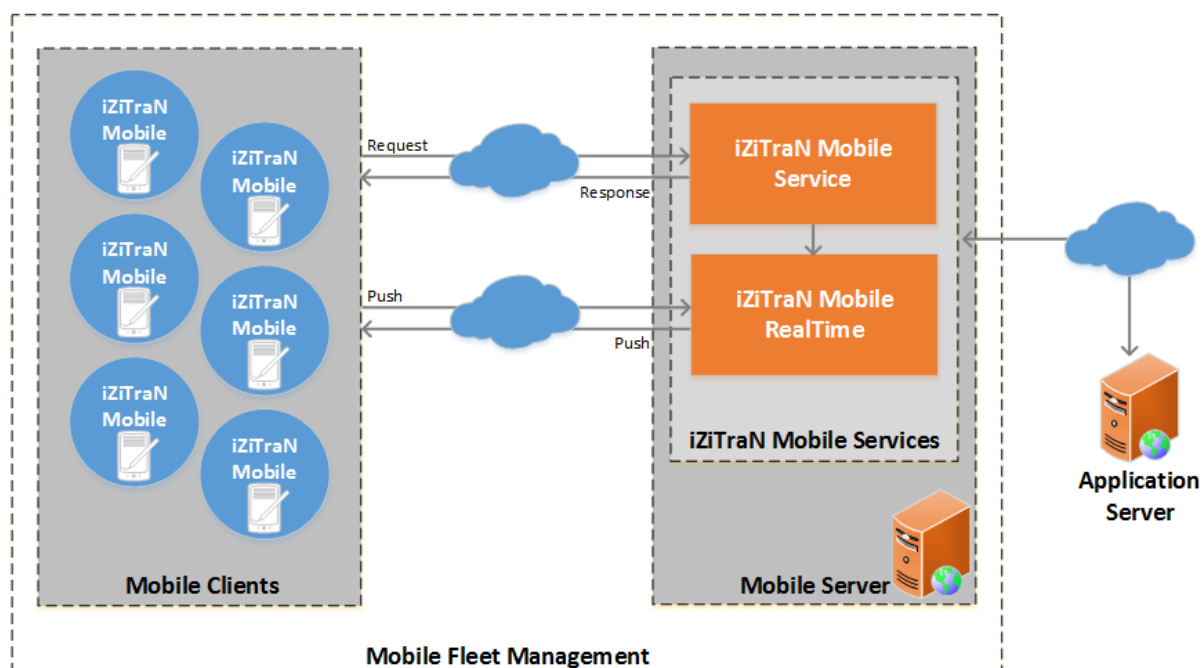


Figura 29 - Enquadramento arquitetural do iZiTraN Mobile Services.

A Figura 29 apresenta o enquadramento arquitetural, mostrando as aplicações cliente e os Web services que compõem o iZiTraN Mobile Services, cujo objetivo da sua criação foi a disponibilização de serviços utilizando o modelo de comunicação cliente-servidor [10].

A primeira grande decisão a tomar envolveu a escolha do tipo de Web service a implementar para o iZiTraN Mobile Service: REST [102] ou Simple Object Access Protocol (SOAP) [103]. Como explicado na secção 4.1.2, optou-se pela implementação da arquitetura REST devido à experiência adquirida da equipa de desenvolvimento, à sua maior escalabilidade, como pela possibilidade de seleção de formato de resposta e da conformidade de pedidos HTTP.

Depois da escolha da arquitetura REST para o desenvolvimento do iZiTraN Mobile Service, escolheu-se a tecnologia para utilização em tempo real no iZiTraN Mobile RealTime. Como

explicado na secção 4.1.2, optou-se pela utilização da *framework* SignalR por esta tratar de comunicações bidirecionais, encontrando o transporte mais eficaz entre servidor e cliente, dando preferência que seja por WebSockets.

Para a implementação destes Web services optou-se pela criação de projetos (no Visual Studio) utilizando o *template* ASP.NET Empty Web Application, com .NET Framework 4.5 e estrutura MVC 4 [76]. Partindo de um projeto baseado em Model-View-Controller (MVC) [104], e sendo MVC por si só uma arquitetura de desenho, é lógico que a arquitetura foi composta pelas três camadas do modelo MVC, que determinam uma separação lógica da responsabilidade de processamento e tratamento de informação, como é apresentado na Figura 30.

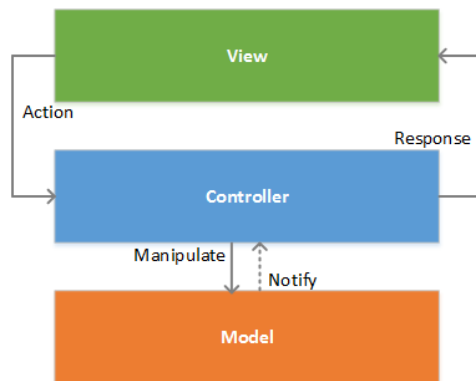


Figura 30 - Arquitectura da aplicação servidora.

Por ordem de apresentação na Figura 30, a arquitetura é composta pelas camadas de View, de Controller e finalmente pela camada Model. A camada View é composta pelos pedidos HTTP (GET, POST, PUT e DELETE) e pelos dados de retorno. Tratando-se de uma API, não são disponibilizadas páginas *web*, sendo devolvidos dados via protocolo HTTP. Por omissão, o sistema devolve dados no formato XML, contudo, o tipo de dados adotado foi JSON. A adoção de JSON foi baseada no facto de este ser um formato “leve” e mais rápido de processar do que o XML [65] [66], como descrito anteriormente na secção 4.1.2.

A camada Controller é composta por elementos denominados controladores, que respondem com base no *input* (HTTP) a ação a executar sobre o modelo de dados (Model) e determinam que informação deverá ser devolvida à camada View.

O Model é uma camada composta pelas classes que modelam os dados do sistema, ou modelo de domínio. Esta camada está implementada num projeto Visual Studio em separado (iZiTraN Mobile Domain) como especificado na secção 4.3.

De notar que não foi necessária uma camada de acesso a dados (Data Access Layer - DAL),

uma vez que o Mobile Server como representado na Figura 29, interliga-se com o Application Server que é responsável por fazer a gestão de acesso a dados.

4.4.2 iZiTraN Mobile Service

No Web service iZiTraN Mobile Service foi instalada e configurada a biblioteca Web API 2 [75]. Esta biblioteca permite estender os mecanismos internos da *framework* ASP.NET de forma a englobar o tipo de pedido HTTP (GET, POST, PUT ou DELETE), mapeando automaticamente o mesmo para a ação correspondente no controlador (Controller).

Nos controladores existe uma convenção de nomes, por exemplo, um pedido ao endereço “http://[ip]/api/Vehicles/GetVehicles/” realizado por GET é enviado automaticamente para o controlador correspondente, neste exemplo será para “VehiclesController”. Caso não exista uma classe que herde de ApiController com esse mesmo nome, o servidor rejeita o pedido com o código de resposta 404 (recurso não encontrado). Uma vez que o pedido é realizado por GET, o sistema procura no controlador um método cujo nome comece por “Get” como, por exemplo, “GetVehicles()” e executa o mesmo. Caso sejam especificados parâmetros no endereço, o sistema procura o método que receba o mesmo número de parâmetros. No entanto, neste tipo de rotas por omissão não são contempladas invocações para mais que um método com os mesmos parâmetros, uma vez que apenas parte do nome do método e do número de parâmetros são validados. O sistema fica sem saber qual dos métodos deve executar e lança uma exceção alertando para várias ações possíveis no mesmo pedido. Por outro lado, as rotas podem ser configuradas por ação, permitindo ter vários métodos com o mesmo número de parâmetros. Com esta abordagem, o endereço do pedido fica mais comprido, contendo mais informação visual, o que por si só pode ser indício de melhoria na usabilidade. As rotas utilizadas foram as rotas por ação e os métodos que foram criados para corresponder às necessidades deste Web service são apresentados na Tabela 5.

Rota	Descrição
api/Accounts/PostAuthCredentials/	Autentica as credenciais do operador de frota.
api/Vehicles/GetVehicles/	Obtém todos os veículos da frota.
api/Vehicles/GetVehicleDaily/	Obtém o diário do veículo.
api/Vehicles/PostVehicleUpdateFavorite/	Atualiza a opção de favorito do veículo.
api/Travels/GetTravels/	Obtém todas as viagens dos veículos.
api/Travels/GetTravelDaily/	Obtém a viagem diária do veículo.
api/Alarms/GetPanicScenario/	Obtém o cenário de pânico do veículo.
api/Alarms/GetAlarms/	Obtém os alarmes do veículo.
api/Messages/GetConversation/	Obtém a conversa do veículo.
api/Messages/GetMessages/	Obtém as mensagens do veículo.

Tabela 5 - Lista de serviços disponibilizados pelo iZiTraN Mobile Service.

4.4.3 iZiTraN Mobile RealTime

No Web service iZiTraN Mobile RealTime foi instalada e configurada a biblioteca SignalR [77]. Esta biblioteca simplifica o processo de adicionar funcionalidades de tempo real à *framework* ASP.NET, como descrito anteriormente na secção 4.1.2.

O SignalR contém dois modelos de comunicação entre clientes e servidores: Ligações persistentes e Hubs [105]. A ligação persistente (representado no código .NET pela classe `PersistentConnection`) dá ao programador acesso direto ao protocolo de comunicação de baixo nível que o SignalR expõe. Usando este modelo de comunicação torna-se mais familiar para os programadores que utilizam APIs baseadas em ligação, como o Windows Communication Foundation (WCF) [106]. Já com o modelo de comunicação por Hub, o programador tem acesso mais de alto nível sobre a API de ligação para que o cliente e servidor possam invocar os métodos diretamente. Utilizando o modelo por Hub torna-se mais natural para os programadores que utilizam APIs baseadas em invocação remota, como .NET Remoting [107] [108].

A opção do modelo de comunicação recaiu para a utilização por Hubs, devido à experiência adquirida da equipa de desenvolvimento e à sua utilização na maioria das aplicações, recomendada mesmo pela própria Microsoft [109].

Com o modelo Hub, que funciona como um controlador, o código do lado do servidor invoca um método no cliente, por exemplo, “`ReceiveTextMessage(TextMessage message)`” sendo um pacote enviado através do transporte ativo que contém o nome e os parâmetros do método a ser invocado (quando um objeto é enviado como um parâmetro do método, ele é serializado utilizando JSON). O cliente, em seguida, combina o nome do método enviado do lado do servidor com os métodos definidos no código do lado do cliente, utilizando `Dynamic Methods` [105]. Se houver uma correspondência, o método de cliente será executado usando os dados de parâmetros desserializados, o mesmo se aplica ao sentido contrário. Os métodos que foram criados para corresponder às necessidades deste Web service são apresentados na Tabela 6.

Rota	Descrição
Accounts/BroadcastUpdateFavorite/	Envio de atualização de favorito.
Vehicles/BroadcastVehicleSnapshotEvent/	Envio de evento de atualização do veículo.
Travels/BroadcastTravelRoute/	Envio de novo percurso efetuado.
Alarms/BroadcastAlarm/	Envio de alarme.
Messages/BroadcastMessageReceived/	Envio de mensagem enviada pelo veículo.
Messages/BroadcastMessageSend/	Envio de mensagem enviada por operador.
Messages/MessageSend/	Envio de mensagem escrita pelo operador.

Tabela 6 - Lista de serviços disponibilizados pelo iZiTraN Mobile RealTime.

4.4.4 Segurança

A segurança dos dados transferidos entre a aplicação servidora e cliente é feita através de uma ligação segura, que é assegurada pelo uso do protocolo HyperText Transfer Protocol Secure (HTTPS) [63] e da implementação de um mecanismo de autenticação baseado em *tokens*.

O protocolo HTTPS é uma implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL/TLS [64]. Ao utilizar HTTPS, é garantido que um pedido intercetado não possa ser lido de forma a obter informação útil, uma vez que os dados são cifrados durante a ligação. Para a utilização de HTTPS bastou ativar a opção “SSL Enabled” nos Web services desenvolvidos.

O mecanismo de autenticação utilizado foi baseado em *tokens*. Um *token* é uma sequência aleatória de dados, geralmente criada pelo servidor aquando a autenticação das credenciais de acesso (nome de utilizador e palavra-passe), com o intuito de identificar e autenticar os utilizadores. Com este mecanismo evita-se que sejam passadas as credenciais do utilizador durante as ligações entre aplicação cliente e servidor, tornando-se numa técnica de segurança que autentica os utilizadores, uma vez que os *tokens* são utilizados juntamente com a ligação segura descrita anteriormente. Os *tokens* utilizados envolveram algumas características como o facto de serem válidos unicamente para uma sessão, de terem uma expiração por inatividade, por terem uma expiração por tempo limite e por serem utilizados juntamente com técnicas de criptografia para se tornarem únicos e difíceis de reverter. Este mecanismo foi implementado segundo o fluxo definido pela RFC 2612 [110] e RFC 2617 [111]:

1. O cliente faz um pedido de autenticação com as credenciais do utilizador. O sistema valida as mesmas, criando depois um *token* recorrendo a sequências aleatórias e a técnicas de criptografia, devolvendo o mesmo durante a resposta (código 200).
2. O cliente faz um outro pedido com o *token* no campo Authorization do cabeçalho do pedido. O sistema obtém o *token*, descripta e valida o mesmo, identificando o utilizador, devolvendo depois a resposta ao pedido (código 200). Caso o cabeçalho do pedido não tenha o campo Authorization ou o *token* não seja valido, a resposta será de não autorizado (código 401).

Para proceder ao desenvolvimento do fluxo descrito foi implementado o AuthenticationFilter, que utiliza o atributo *Authorize* do ASP.NET [112]. Esta implementação resulta na execução do mecanismo de autenticação (validação do *token* presente no cabeçalho do pedido) antes do sistema chegar a executar as ações do controlador que é invocado, como é apresentado na

Figura 31.

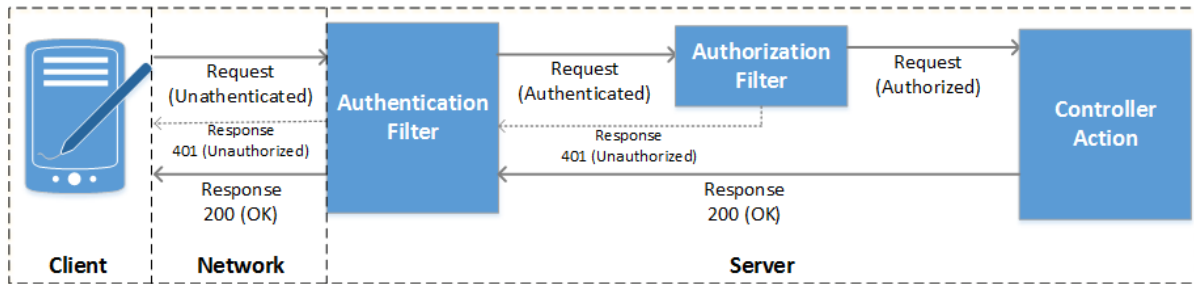


Figura 31 - Fluxo de segurança utilizando filtro de autenticação.

De notar que todos os mecanismos implementados, nomeadamente a utilização de HTTPS e a autenticação baseada em *tokens* têm de coexistir em simultâneo para que os dados sejam transmitidos de forma segura. Desta forma, há garantias de autenticidade, integridade, confidencialidade e disponibilidade, atributos da segurança da informação [113].

4.5 Aplicação Móvel

A aplicação cliente desenvolvida com o nome de iZiTraN Mobile é a aplicação móvel que consome os serviços disponibilizados pelos Web services apresentados anteriormente (secção 4.4) e que foi criada para que o operador de frota possa realizar as suas operações de forma rápida, simples e eficiente. A aplicação iZiTraN Mobile é apresentada nas seguintes subsecções.

4.5.1 iZiTraN Mobile

A aplicação iZiTraN Mobile permite ao operador de frota, com segurança, obter acesso ao estado dos veículos e realizar ações sobre os mesmos. Este pode visualizar em lista e em mapa o estado dos veículos (desligado, parado, em trânsito, em movimento), ter acesso a vários indicadores como localização, velocidade, quilómetros percorridos, consumos de combustível, entre outros. É também possível visualizar o histórico de viagens do veículo, o envio e receção de mensagens e os alarmes que podem ser desencadeados pelos veículos. Seguidamente é apresentada a arquitetura da aplicação iZiTraN Mobile.

4.5.1.1 Arquitetura

A aplicação móvel está organizada em camadas, de apresentação, negócio e comunicação que

se encontram dentro de *packages*, como é possível visualizar na Figura 32.

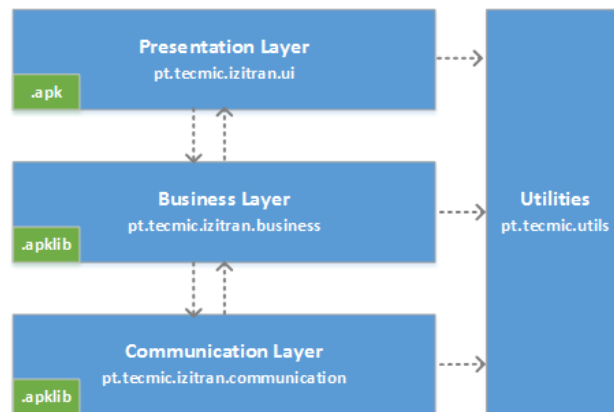


Figura 32 - Arquitetura organizada por *packages*.

Todas as camadas enunciadas têm a sua própria *package*, ou seja cada uma tem o seu próprio conjunto de ficheiros de código que compõem cada camada, servindo assim de contentor para uma melhor organização de código. As *packages* criadas começam, por convenção, pela identificação “pt.tecmic.izitrان” representando a estrutura de pastas. A *subpackage ui* inclui todas as classes necessárias para apresentar a *interface* gráfica e para tratar as interações do utilizador (View e Presenter), de acordo com o padrão Model-View-Presenter (MVP) [114]. Já a *subpackage business* contém a camada de negócio (Business) com as regras de negócio que determinam quais os dados criados, transformados ou persistidos que seguidamente poderão ir (dependendo da regra de negócio) para a camada de apresentação ou de comunicação. Desta forma, a camada de negócio funciona como um interpretador de dados. A *subpackage communication* inclui as classes que fazem parte da comunicação (Communication) entre a aplicação e os seus Web services, tanto pelo uso da *framework* Volley como da SignalR. De notar ainda a existência de uma outra *subpackage*, apelidada por *utils*, que contém os utilitários necessários e usados pelas camadas anteriormente descritas.

4.5.1.1.1 Separação da Lógica de Apresentação e de Negócio

Havendo a necessidade diminuir a complexidade e o esforço de manutenção da aplicação iZiTraN Mobile utilizou-se o padrão MVP [114] [115], devido à sua visão de separação entre a lógica de apresentação e a lógica de negócio. Desta forma, existe uma separação clara entre as implementações de *interface* e de negócio, além de permitir o desenvolvimento de testes unitários de forma simples.

O MVP é uma adaptação do conhecido padrão de *software* MVC, usado principalmente para

construir *interfaces* gráficas. No padrão MVP a camada Presenter assume a função de mediadora (desempenhada pelo Controller no MVC), atua sobre o Model e a View, tendo como função obter os dados dos modelos para depois formata-los e exibi-los para a View. A View é responsável por remeter os eventos de *interface* gráfica até à camada Presenter, enquanto o Model se torna estritamente num modelo de domínio. Uma característica muito importante é que a View deixa de ter referência ao Model, separando as regras de negócio da View, permitindo assim que qualquer alteração que lhe seja feita não influencie o modelo de negócio. Houve uma adaptação ao padrão genérico para que a View deixasse de ter uma referência ao Presenter e ao Model, permitindo assim que através da utilização de uma estrutura modular (apresentada na subsecção 4.5.1.1.2) qualquer alteração à mesma não influencie tanto a lógica de apresentação como a de negócio. Como neste padrão toda a lógica de apresentação é movida para o Presenter, os testes ao código são significativamente melhorados (como é possível observar através do teste da Figura 52), uma vez que não é necessário invocar a *framework* de *interface* gráfica, sendo este o grande problema que o padrão resolve.

4.5.1.1.2 Arquitetura Modular

A aplicação foi desenhada para ser modular, por forma a ser escalável com a adição e remoção de módulos, facilitando futuras manutenções, mudanças no código e agilizando a implementação do *software*, permitindo o desenvolvimento isolado nas diferentes partes da aplicação [116] [117].

A estrutura modular criada assenta numa Event-Driven Architecture (EDA) [117], satisfazendo a necessidade pretendida de adição e remoção de módulos de uma forma fácil e flexível [118]. Para fazer a gestão destes eventos recorreu-se à *framework* Guava EventBus [119] [120], cuja utilização permite o uso de eventos assíncronos. A biblioteca trabalha sobre o facto de existirem objetos que precisam partilhar dados. A forma utilizada para alcançar a partilha de dados é ter *listeners* de eventos, cujo único objetivo é exercer alguma ação quando ocorre um evento desejado, comunicando pela forma de publicação-subscrição. Esta biblioteca difere das restantes, pois permite que objetos utilizem a publicação ou subscrição de eventos, sem ter conhecimento explícito uns dos outros.

Num contexto relativo ao projeto, foram criados vários *broadcasts* (utilizando para a sua implementação o padrão de desenvolvimento Singleton [121]) que permitem a definição de um contrato de serviços e dados entre as camadas lógicas da aplicação. Com esta abordagem consegue-se menos dependências entre módulos, uma vez que o Business apenas tem a

responsabilidade de enviar dados para o *broadcast* de comunicações, sem se preocupar com o meio de comunicação, ou seja, apenas conhece o contrato de comunicação. Um outro grande benefício é a possibilidade de existirem vários módulos a poderem escutar o mesmo canal de *broadcast*. Desta forma, a aplicação quando necessita de enviar e receber dados faz uma publicação dos eventos para o *broadcast* desejado, sendo que os mesmos são recebidos pelos seus subscritores, que procedem ao tratamento do evento. A arquitetura do iZiTraN Mobile que faz uso desta estrutura modular é apresentada na Figura 33.

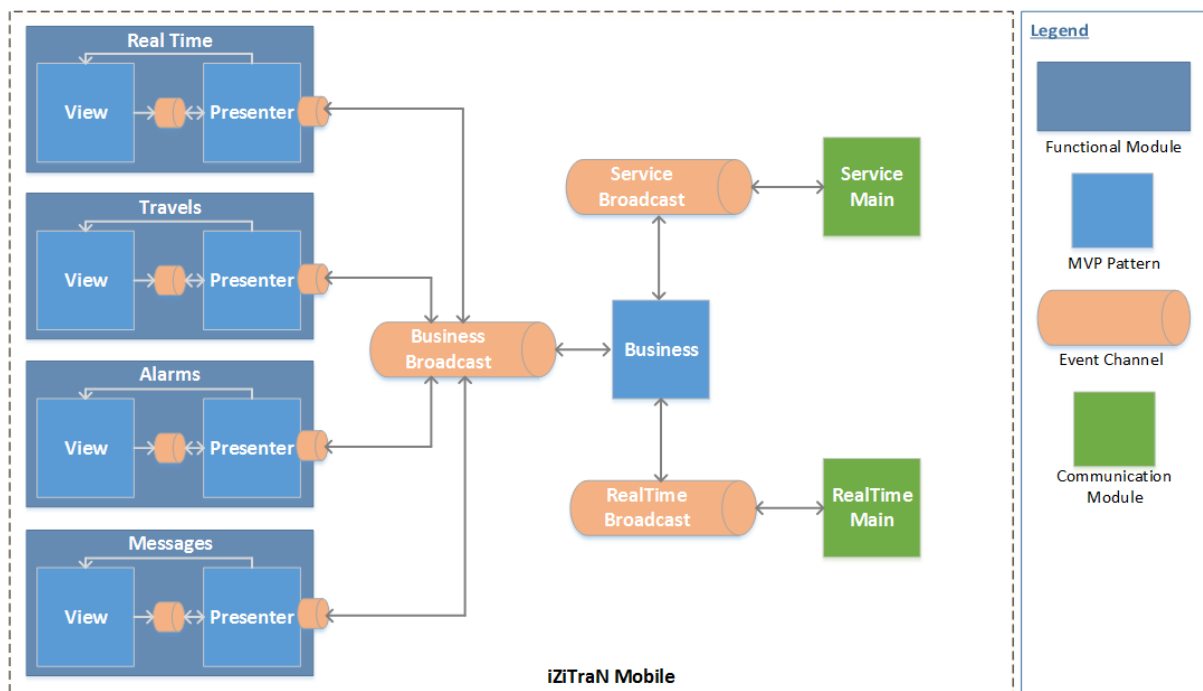


Figura 33 - Módulos da arquitetura iZiTraN Mobile.

A arquitetura apresentada na Figura 33 corresponde à estrutura modular criada para a aplicação cliente. Os módulos Real Time, Travels, Alarms e Messages contêm a camada de apresentação correspondente a cada uma, com a View a pedir ao Presenter que lhe apresente as ações pedidas. O canal *broadcast* entre ambos não é utilizado nos dois sentidos, uma vez que houve a intenção de tornar a aplicação ainda mais escalável com o Presenter a ter uma referência para a View, mas esta a não conter nenhuma referência para o mesmo. Com esta opção consegue-se adicionar ou remover Presenters tendo em conta as ações do utilizador, fazendo com que não exista a necessidade de mudanças em outros módulos. O Presenter de cada um está registado no canal Business Broadcast, podendo receber e enviar eventos para este. Quem estiver registado no canal está à escuta dos eventos que lhe são conhecidos e trata do evento recebido.

O Business é o módulo responsável por fazer a gestão de todos os dados, contendo toda a lógica de negócio da aplicação cliente. Pode persistir dados, validar a existência de ligação à Internet, processar informação que chegue do módulo de comunicações, entre outros. Este está registado nos canais Business Broadcast, Service Broadcast e RealTime Broadcast podendo receber e fazer *broadcast* dos eventos para os módulos registados nestes canais.

Os módulos Service Main e RealTime Main são responsáveis por fazer a gestão das comunicações com os Web services. O Service Main está registado no canal Service Broadcast, sendo por este canal que o mesmo recebe os pedidos a invocar ao Web service e envia as respetivas respostas, tudo sobre a forma de eventos. O processamento dos eventos recebidos consiste na criação de pedidos REST que são transmitidos para o iZiTraN Mobile Service via HTTPS, utilizando a *framework* Volley. Assim que o iZiTraN Mobile Service responde o Service Main cria um evento e publica-o no Service Broadcast. Por sua vez, o módulo RealTime Main está registado ao canal RealTime Broadcast, sendo por este canal que o mesmo publica a informação recebida do Web service iZiTraN Mobile RealTime e recebe os pedidos do Business. Os dados que são tratados por este módulo são transmitidos via WebSockets (utilizando SignalR). Desta forma a aplicação é alimentada com dados recolhidos em tempo real.

Com a arquitetura modular apresentada, a necessidade da adição e remoção de módulos de uma forma fácil e flexível é satisfeita. É ainda possível integrar um módulo implementado para plataformas distintas (aplicação Android, aplicação Java, entre outras), além de permitir substituir o consumo de um Web service específico por um repositório local de dados, agilizado assim o desenvolvimento. A adição de um módulo à arquitetura modular existente é apresentada na Figura 34.

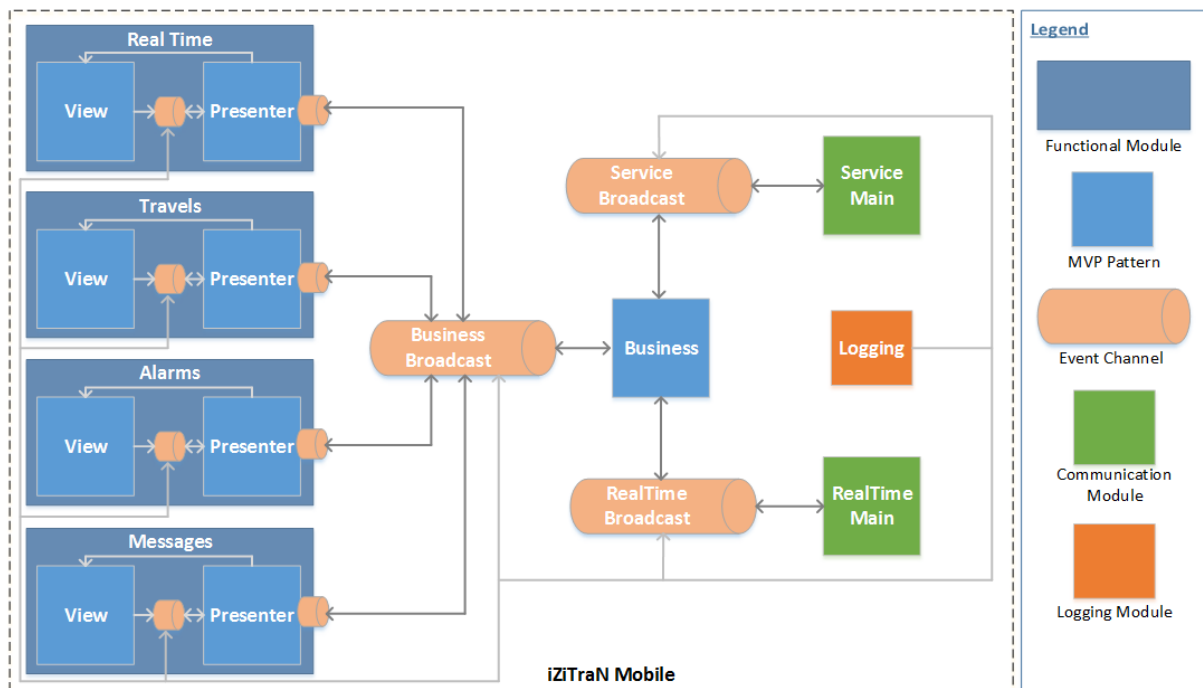


Figura 34 - Adição do módulo Logging à arquitetura iZiTraN Mobile.

A Figura 34 apresenta um exemplo da adição de um novo módulo, designado de Logging, sem influenciar os outros módulos pré-existent. Este módulo criado para este projeto, tem a função de recolher informações dos dados que passam pelos canais de eventos no qual esteja registado, podendo fazer recolha da aplicação toda ou de módulos específicos. Da mesma forma como foi simples a sua adição, também a sua remoção é simplificada, permitindo que possa ser feita em tempo de execução. Com esta estrutura modular, o módulo Logging consegue cumprir a sua função através do registo a todos os eventos que passam pelos canais *broadcast*, bastando para isso que a mesma se registre em todos os *broadcasts* criados. Se houver a necessidade de registar só os erros que ocorram basta fazer uma outra implementação, e se ainda houver a necessidade de enviar essa informação para um Web service, haverá outra implementação. No entanto, tudo é permitido sem influenciar módulos pré-existent.

A vantagem funcional do módulo Logging é de apresentar de forma muito simples as ações que o utilizador faz na aplicação, podendo a recolha das mesmas ser enviada para um Web service, de modo a se proceder a uma futura análise destas ações. A análise vai permitir a identificação de anomalias da aplicação e recolha das interações do utilizador com a mesma. Como resultado, sabe-se como priorizar quais as correções a fazer em primeiro lugar e quais as *interfaces* gráficas que podem ser melhoradas. Se a arquitetura apresentada não tivesse esta estrutura, haveria um maior esforço, pois tinha-se de ir a cada ação do utilizador (`onClick()`),

onLongClick(), onSelected(), entre outros) escrever uma linha para fazer esse mesmo registo e ainda invocar outros meios para enviar esse registo para um Web service, afetando módulos existentes e consequentemente estabilidade da aplicação.

4.5.1.2 Interface Gráfica

A *interface* gráfica da aplicação foi desenvolvida de modo a proporcionar ao utilizador a melhor experiência possível em termos de navegação, usabilidade e familiaridade com os componentes nativos da plataforma Android [24]. Foi pensada especificamente para dispositivos móveis, tendo o seu planeamento e implementação considerado as boas práticas sugeridas pelas diretrizes Android, o recente Material design [122]. Como explicado posteriormente na secção 4.5.1.3.

Antes da sua implementação foram desenhados *sketchings*, protótipos iniciais (Apêndice IV), de cada ecrã propostos para a aplicação, que tiveram em conta as diretrizes, como também todos os requisitos do cliente em relação a este projeto. Após análise dos *sketchings* com a equipa de desenvolvimento, estes foram aceites e passados para *mockups*, na forma de protótipos intermédios (Apêndice V). Após a sua validação procedeu-se à construção dos ecrãs gráficos da aplicação (Apêndice VI). De seguida são apresentadas algumas figuras (Figura 35, Figura 36, Figura 37, Figura 38, Figura 39 e Figura 40) dos protótipos, contrastando com os ecrãs finais da aplicação.

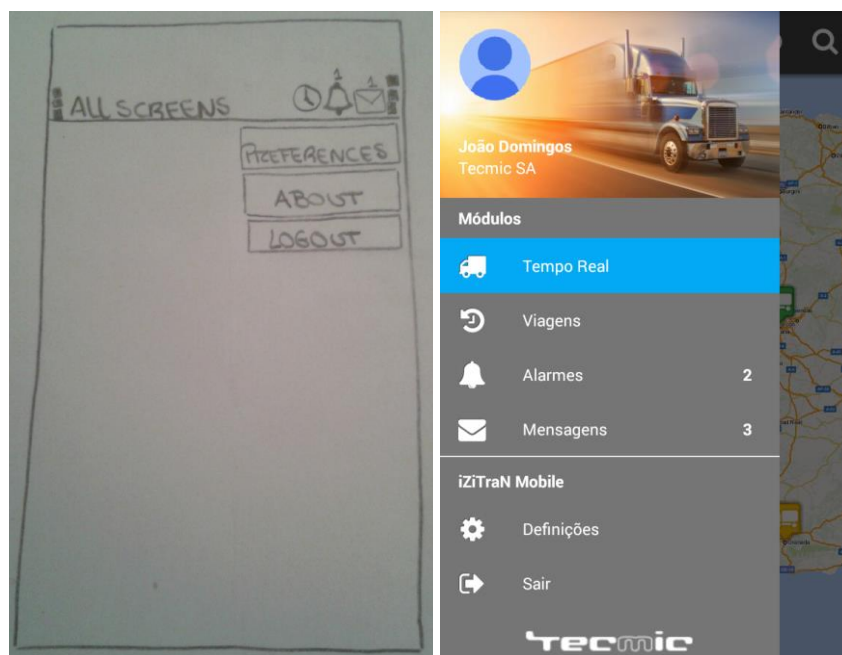


Figura 35 - Barra de Navegação com protótipo inicial (esquerda) e final (direita).

A Figura 35 apresenta a navegação, sendo que à esquerda é apresentado o protótipo inicial, enquanto à direita a Figura 35 apresenta a navegação final. Esta navegação inicialmente foi idealizada na forma de barra de navegação (protótipo inicial) para estar presente em todos os ecrãs da aplicação. No entanto, verificou-se através de testes de usabilidade (secção 5.1.5) que o utilizador facilmente se descontextualizava com o ecrã em que se encontrava, criando alguma confusão ao mesmo. Desta forma, seguindo o Material design [123] optou-se pela “navegação por painel” presente no ecrã final. Com esta navegação o utilizador pode navegar entre a lista de veículos, viagens, alarmes, mensagens e acesso aos ecrãs de preferências, versão de aplicação e terminar sessão, tendo a mesma os títulos e subtítulos contextualizados com o ecrã visualizado. Comparando o protótipo e o ecrã final, a barra do topo passou para navegação lateral, permitindo uma melhor organização dos módulos existentes e das predefinições da aplicação, tudo agrupado numa zona específica do ecrã e sem misturar a navegação às especificidades de cada ecrã, tornando o ecrã mais limpo e organizado.

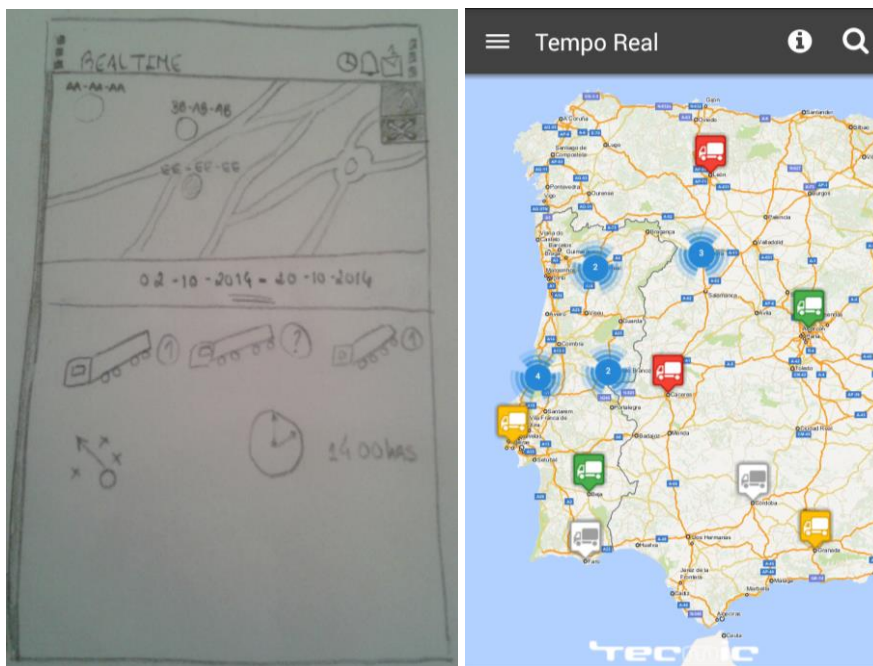


Figura 36 - Ecrã Principal dos Veículos com protótipo inicial (esquerda) e final (direita).

A Figura 36 apresenta à esquerda o protótipo inicial do ecrã veículos, enquanto que à direita apresenta o ecrã final. Este ecrã inicialmente foi idealizado para conter os veículos da frota no mapa, mostrando a sua localização e ao mesmo tempo o estado atual dos mesmos. Idealizado para também mostrar os indicadores atuais dos veículos, como o número de veículos em movimento, em trânsito e parados, os totalizadores de quilómetros percorridos e também as horas de condução efetuadas. No entanto, pela quantidade elevada de veículos presentes no

mapa e por forma a valorizar a simplicidade visual do ecrã optou-se por colocar os indicadores dos veículos acessíveis através da barra de topo deste ecrã. Comparando o protótipo e o ecrã final, estes diferenciam-se pelos indicadores dos veículos que passaram a ser acessíveis a partir da barra de topo, dando a possibilidade ao utilizador de conseguir utilizar a mesma forma de acesso para pesquisar por veículos.

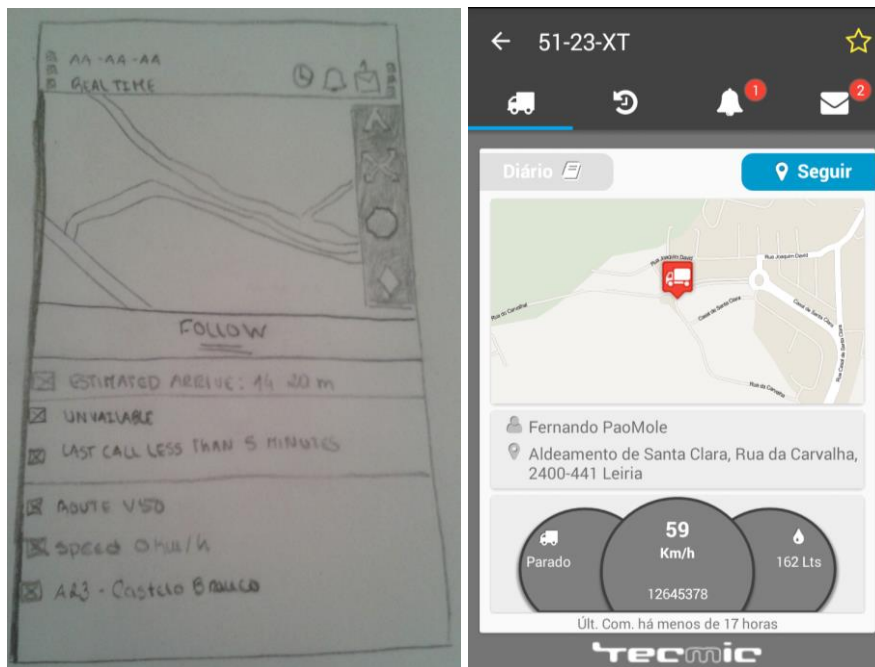


Figura 37 - Ecrã Tempo Real de um veículo com protótipo inicial (esquerda) e final (direita).

É apresentado na Figura 37 à esquerda o protótipo inicial do ecrã tempo real que foi desenhado inicialmente, enquanto à direita a Figura 37 apresenta o ecrã final. Este ecrã permite visualizar através de recurso a mapas a localização do veículo selecionado, permitindo também a visualização de vários indicadores através do menu de opções, que se encontra no lado direito no mapa. Este menu de opções permite que sejam atualizados os indicadores que se encontram abaixo do ecrã, tendo na sua essência três opções: Diário, Seguir e Ecrã Completo. A opção Diário corresponde aos indicadores diários do veículo, como quilómetros percorridos, velocidade média, consumo diário, entre outros. Já a opção Seguir (opção selecionada na Figura 37) corresponde a indicadores instantâneos (tempo real), como estado atual da ignição, atualização dos quilómetros percorridos, velocidade instantânea, consumo e localização atual. De notar que com esta opção o veículo vai atualizando a sua posição no mapa à medida que se desloca. Por último, a opção Ecrã Completo mostra o mapa em *fullscreen* para uma ampla e maximizada visão do mesmo. O protótipo e o ecrã final são similares, no entanto a disposição do menu passou do mapa para

a forma de dois botões no topo do mesmo, como também houve a alteração da disposição da informação dos indicadores dos veículos através da apresentação de um painel de indicadores.

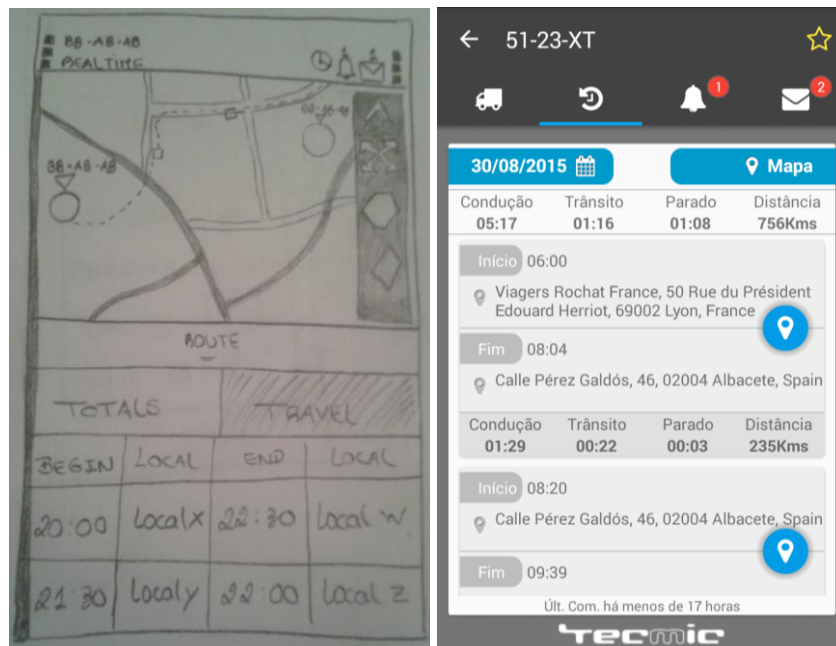


Figura 38 - Ecrã Viagens de um veículo com protótipo inicial (esquerda) e final (direira).

Na Figura 38 é apresentado o ecrã de viagens, à esquerda o protótipo inicial e à direita o ecrã final. Este ecrã permite visualizar as viagens que o veículo selecionado efetuou ao longo do dia atual, mas também a apresentação de viagens decorridas numa data específica, mostrando o seu trajeto em mapa e os respetivos indicadores. Estes indicadores correspondem a data, localização e tempos decorridos de início e fim de viagem que tanto podem ser de tempo em que o veículo esteve parado, em trânsito ou em movimento. Comparando o protótipo e o ecrã final, retirou-se a opção de alternar entre indicadores da viagem (totais e por percurso) para apresentar tudo no mesmo ecrã. Com esta alteração o mapa presente no protótipo, que não era a parte principal para este ecrã específico, passou a estar acessível através da opção “Mapa”, contendo ainda o ecrã final a opção para visualização em mapa dos diferentes percurso. Uma outra diferença foi a adição de escolha por dias anteriores, uma vez que os utilizadores têm a necessidade de consultar viagens de dias anteriores.



Figura 39 - Ecrã Alarmes de um veículo com protótipo inicial (esquerda) e final (direira).

É apresentado na Figura 39 à esquerda o protótipo inicial do ecrã alarmes que foi desenhado, enquanto à direita a Figura 39 apresenta o ecrã final. Este ecrã permite visualizar os alarmes que foram desencadeados pelo veículo selecionado, mostrando o nome do alarme localização e data de criação, podendo ainda visualizar a posição atual do veículo em mapa, como também a localização em mapa do alarme desencadeado. Comparando o protótipo e o ecrã final, estes ficaram bastante diferentes, uma vez que no ecrã final todos os alarmes ficam agrupados. Com esta alteração as mensagens trocadas aquando do alarme passaram a ser unicamente visualizadas e trocadas no módulo de mensagens, separando estes dois módulos, simplificando assim os ecrãs ao utilizador.

De notar que tanto o ecrã Alarmes, como o ecrã Mensagens têm um indicador a vermelho, no topo do ícone representativo de cada ecrã, para que este possa indicar ao utilizador o número de alarmes e de mensagens que o veículo selecionado contem que ainda estão por visualizar.

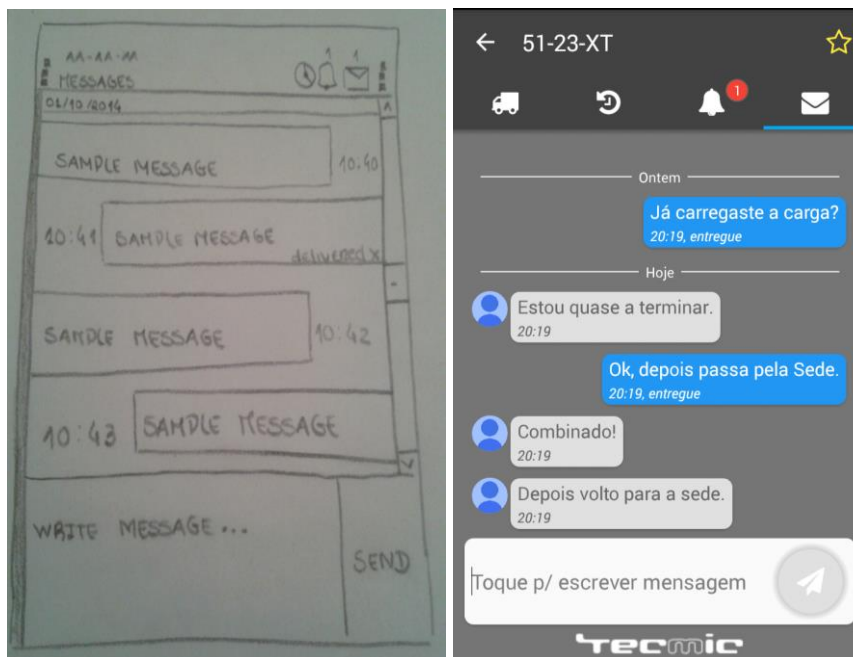


Figura 40 - Ecrã Mensagens de um veículo com protótipo inicial (esquerda) e final (direira).

A Figura 40 apresenta à esquerda o protótipo inicial do ecrã mensagens que foi desenhado, enquanto à direita a Figura 40 apresenta o ecrã final. Este ecrã permite receber e enviar mensagens entre o utilizador e o condutor do veículo, podendo ser visualizada a data e estado da mensagem (guardada, a enviar, enviada, entregue ou vista), sendo este ecrã bastante útil como meio de comunicação, entregando pequenas mensagens entre os intervenientes da conversa. Comparando o protótipo e o ecrã final, estes ficaram bastante similares, uma vez que apenas houve alterações ao nível de ícones, nomeadamente com a adição do ícone que mostra a imagem dos intervenientes da conversa e a data da mensagem que passou a ser apresentada por baixo da mesma.

Para a visualização de todos os protótipos iniciais, intermédios e ecrãs finais da aplicação, consultar os Apêndices IV, V e VI. De seguida são descritas as diretrizes para desenho de aplicações sobre a plataforma Android que foram adotadas neste projeto.

4.5.1.3 Diretrizes para desenho de interfaces para a plataforma Android

Ao longo do planeamento e do desenvolvimento do projeto foram estudadas e seguidas várias diretrizes utilizadas pela plataforma Android [124]. As diretrizes evidenciam claramente que há dispositivos de todos os tamanhos, modelos e orientações, apontando práticas para que

cada aplicação não se diferencie drasticamente das restantes (ícones, cores, tipografia, métricas, estilos de escrita, navegação, acessibilidade, entre outros), tornando assim as aplicações mais uniformes, consistentes e intuitivas para os utilizadores.

Como neste tipo de dispositivos os elementos visuais tocáveis são extremamente importantes, os ícones foram criados com pelo menos 48x48 density-independent pixel (dp) por forma a serem seleccionáveis e intuitivos para as funcionalidades que representam ao longo da aplicação. A Figura 41 apresenta os vários tamanhos de ícones seguidos pela aplicação móvel.

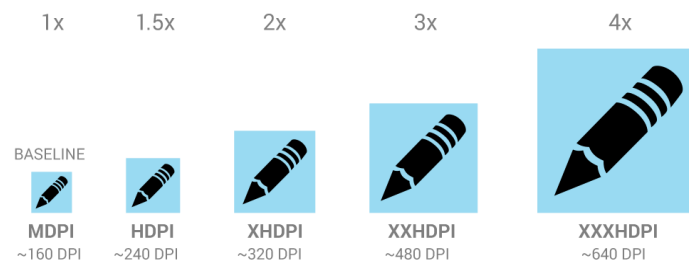


Figura 41 - Diretriz de ícones [125].

Os ícones da barra de menu são botões gráficos que representam as ações mais importantes que os utilizadores podem tomar dentro da aplicação. Cada um deve aplicar uma metáfora simples que representa um único conceito, para que os utilizadores possam intuitivamente conhecer o seu significado. Os ícones utilizados são do género Font Awesome [126], pelo facto de também ser utilizado nos produtos da empresa, tornando a sua utilização uniforme e constante a todas as plataformas. A fonte Font Awesome oferece uma grande coleção de ícones, com tamanhos infinitos, que podem ser utilizados em todas as cores, sem o menor esforço, apenas alterando a cor da fonte.

A cor predominante foi o cinzento-escuro, uma vez que é a cor dominante nos produtos da empresa e pelo facto de esta cor simbolizar estabilidade, sucesso e qualidade. Se bem utilizado, o cinzento oferece equilíbrio e flexibilidade por ser o equilíbrio entre o preto e o branco, expressando neutralidade na *interface* gráfica. O cinzento é também uma das cores em destaque na paleta de cores padrão do Android [127]. As cores usadas dão ênfase aos componentes e oferecem um bom contraste entre os componentes visuais.

Os elementos textuais (escrita) da aplicação seguiram também as diretrizes do Android, por forma a manter a *interface* concisa, simples e amigável. Na Figura 42 é apresentado um bom e um mau exemplo de uso das diretrizes Android para o estilo de escrita.

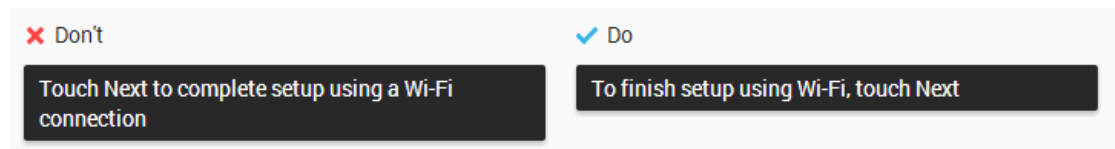


Figura 42 - Diretriz de estilo de escrita [128].

Deste modo, deve-se descrever apenas o que o utilizador precisa saber, eliminando a redundância, como títulos repetidos no conteúdo do corpo de uma mensagem numa caixa de informações. Tentou-se sempre manter o texto tão curto quanto possível, como ilustrado na Figura 43.

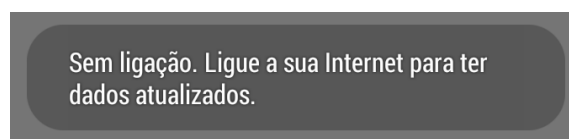


Figura 43 - Exemplo de mensagem com estilo de escrita.

Na Figura 43 pode-se observar uma mensagem utilizada na aplicação com o estilo de escrita adotado pelas diretrizes Android. Sabendo que a tipografia também é bastante importante numa aplicação, estudou-se a diretriz da mesma, uma vez que a implementação bem-sucedida é essencial para ajudar os utilizadores a entender rapidamente a informação visualizada. Para além de se seguir estas tipografias deu-se especial ênfase a tipos de letra sem serifas, o que permite ao utilizador ler a informação com plena facilidade e com um elevado conforto visual. Foram também seguidas as diretrizes para métricas e grelhas, que destacaram na aplicação o espaçamento entre componentes. Em média os componentes devem ter um espaçamento de 48x48 dp, que a traduzir para um tamanho físico é cerca de 9 milímetros, como é apresentado na Figura 44.

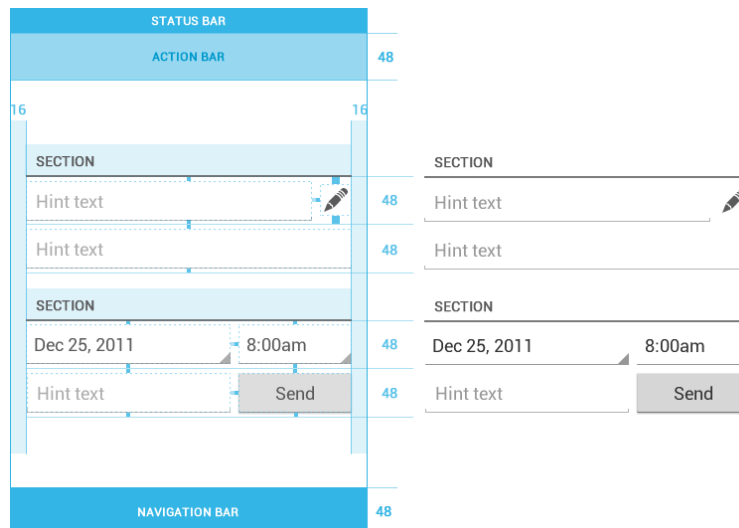


Figura 44 - Diretriz de espaçamento de componentes [129].

O tamanho de 48x48 dp é um tamanho confortável para os componentes *touchscreen* e deste modo, os utilizadores serão capazes de forma confiável e precisa aceder manualmente a todos os componentes tocáveis. Esta diretriz foi bastante utilizada na aplicação, como se pode verificar na Figura 45.

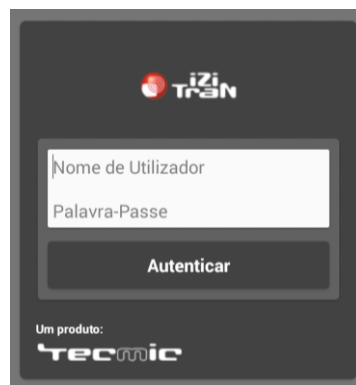


Figura 45 - Exemplo de espaçamento entre componentes.

Na Figura 45 está representado o espaçamento entre componentes usado na aplicação, mas esta figura consegue também demonstrar algumas “dicas”, como a descrição dos campos nome de utilizador e palavra-passe, para que assim o utilizador tenha uma maior e melhor perceção do que colocar no campo correspondente.

Na navegação entre os ecrãs da aplicação também se adotou uma diretriz Android, de forma que os utilizadores possam navegar intuitivamente através da barra de topo. Esta organização da *interface* ajuda os utilizadores a navegar diretamente pelos ecrãs hierárquicos da aplicação. A Figura 46 apresenta a diretriz de navegação padrão usada pela plataforma Android.



Figura 46 - Diretriz de navegação padrão [130].

A Figura 46 apresenta o estilo de navegação usado pelo Android, também usado na aplicação móvel implementada, como é apresentado pela Figura 47 que segue a mesma diretriz.

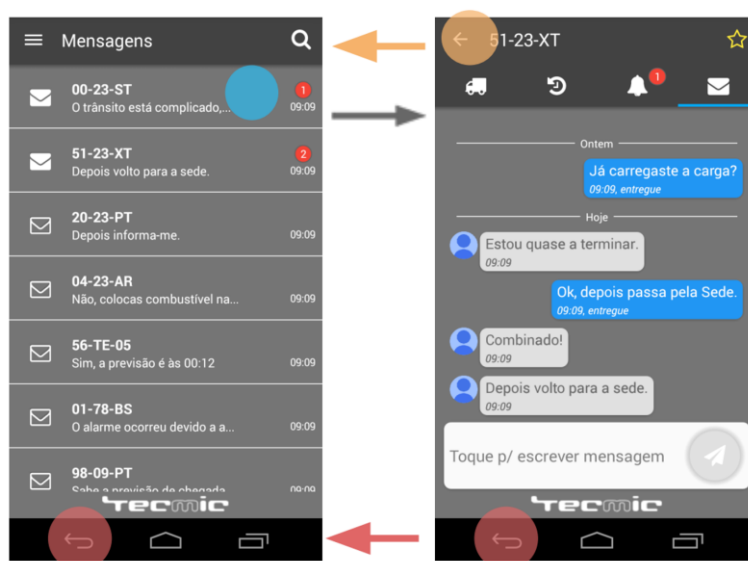


Figura 47 - Exemplo da diretriz de navegação padrão utilizada na aplicação.

Esta diretriz de navegação padrão foi utilizada de forma constante na aplicação, sendo ela o caminho principal para o utilizador visualizar os ecrãs pretendidos. Adicionalmente usou-se um painel para a transição a partir da lateral esquerda do ecrã que exhibe os principais módulos da aplicação, tornando-se também numa importante forma de navegação. Esta forma de navegação apelida-se de “navegação por painel”, sendo também uma diretriz de navegação

apresentada na Figura 48.



Figura 48 - Diretriz de navegação por painel [131].

A Figura 48 apresenta o estilo de navegação por painel utilizado pela plataforma Android, sendo esta uma forma de organizar o conteúdo que é frequentemente utilizado numa aplicação. Este estilo é também utilizado na aplicação móvel implementada, como é apresentado pela Figura 49.

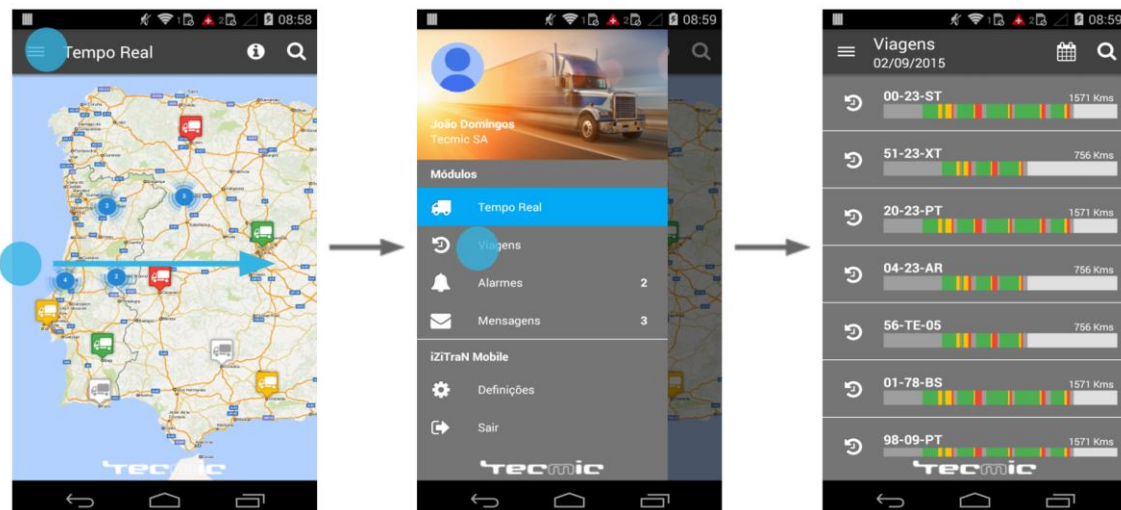


Figura 49 - Exemplo da diretriz de navegação por painel utilizada na aplicação.

Uma boa navegação é importante para qualquer aplicação, por isso, todo este estudo foi importante, por forma a garantir que as diretrizes do Android fossem aplicadas de forma correta.

A clareza e simplicidade das notificações e das ações que o utilizador poderá ter sobre as funcionalidades existentes (confirmadas ou reconhecidas por meio de texto) têm uma importância elevada na usabilidade da aplicação e estão também contempladas nas diretrizes do Android, tal como apresentado na Figura 50.

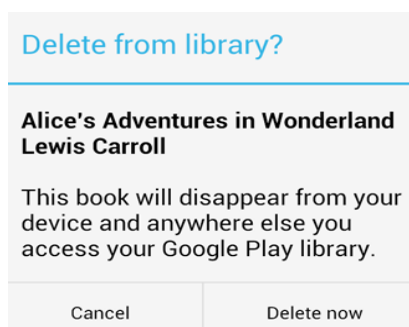


Figura 50 - Diretriz de notificação de confirmação [132].

Neste tipo de mensagens de notificação, o utilizador poderá visualizar a ação que irá ser confirmada, o nome do item que a ação recai, o texto descritivo de aviso das consequências da ação, bem como os botões de cancelar e o outro botão com o nome descritivo da ação que se realizará. Tentando seguir a mesma diretriz foi criada uma estrutura similar para a apresentação de notificações de confirmação, por forma a esclarecer e permitir ações do utilizador, tal como apresentado na Figura 51.

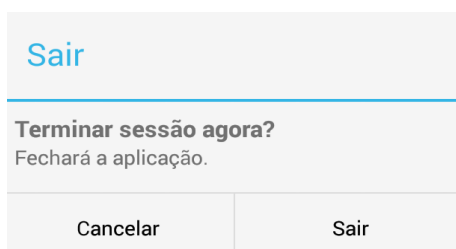


Figura 51 - Exemplo de notificação de confirmação utilizada na aplicação.

Deu-se também especial atenção à acessibilidade da aplicação. O utilizador poderá ativar o modo de acessibilidade presente no dispositivo móvel e ser guiado na aplicação através, das imagens, textos e itens presentes nas listas.

O próximo capítulo apresenta os testes realizados ao longo do projeto, nomeadamente os testes unitários, integração, carga, segurança, e os testes de usabilidade, tendo ainda na parte final uma discussão de resultados.

Testes

Este capítulo apresenta os testes realizados e os resultados obtidos, nomeadamente testes unitários, integração, carga, segurança, e os testes de usabilidade, finalizando o capítulo com a discussão de resultados.

5.1 Testes Realizados

Nas subsecções desta secção são apresentados os testes realizados [133] para provar conceitos adotados ao longo deste projeto, mas também para garantir que novos desenvolvimentos não danifiquem os existentes, diminuir custos associados à manutenção, aumentar a qualidade do produto desenvolvido, entre outros.

5.1.1 Testes Unitários

Os testes unitários foram desenvolvidos ao longo de cada iteração, tendo os mesmos sido bastante úteis para testar se as funcionalidades do *software* produzem o resultado esperado. O desenvolvimento destes testes consistiu em validar o código através da entrada e saída de dados num pequeno e isolado excerto de código. Devido à sua dimensão, são apresentados no Apêndice VII do presente documento.

5.1.2 Teste de Integração

O teste de integração é um teste de *software* usado com o intuito de testar a integração entre vários módulos [134]. Deste modo, efetuou-se o teste para envio de uma mensagem de texto e consequente resposta. Para além dessa funcionalidade foi testado simultaneamente o padrão MVP, a arquitetura modular por eventos e a *framework* SignalR que serviu para enviar e receber a resposta devolvida pelo servidor iZiTraN Mobile RealTime. A Figura 52 apresenta o resultado do teste descrito.

```

public class SendMessagesTest {
    @Test
    public final void testRequestSendMessage() {
        //set up params
        MessagesPresenter paramPresenter = new MessagesPresenter();
        TestMessagesView paramView = new TestMessagesView();
        MessagesModuleViewEvent paramRequestSendMessageEvent =
            new RequestSendMessageTestEvent().view(paramView)
                .identifier(1).descriptionMessage("Hello Driver");
        //expected result
        boolean expectedResult = true;
        //execute
        try {
            synchronized (paramView) {
                paramView.post(paramRequestSendMessageEvent);
                paramView.wait(5000);
            }
        } catch (Exception e) {
            fail(e.getMessage());
        }
        boolean result = paramView.result;
        //assert result
        assertEquals(expectedResult, result);
    }
}

private static class TestMessagesView implements SendMessagesTestAPI {
    protected boolean result;
    @Override
    public void responseMessage(boolean response) {
        result = response;
        synchronized (this) {
            notifyAll();
        }
    }
    public void post(MessagesModuleViewEvent messagesModuleEvent) {
        MessagesBroadcast.getInstance().post(messagesModuleEvent);
    }
}
}

```

ON SendMessagesTest (pt.tecmic.modules.messages)
ON All Tests Passed

Figura 52 - Teste de integração assíncrono para envio de mensagem e resultado do mesmo.

A Figura 52 apresenta o teste descrito e o resultado positivo que o mesmo teve. Com a arquitetura modular utilizada e com o módulo Logging a registrar-se em todos os canais de eventos existentes, obteve-se a recolha do fluxo dos eventos, que contem as informações recolhidas durante a execução do teste, como é apresentado na Tabela 7.

2015/08/22 13:46:46 INFO SendMessagePresenterEvent SendMessagePresenterEvent(identifier=1, message=Hello Driver)
2015/08/22 13:46:46 INFO SendMessageBusinessEvent SendMessageBusinessEvent(identifier=1, message=Hello Driver)
2015/08/22 13:46:46 INFO SendMessageCommunicationEvent SendMessageCommunicationEvent(identifier=1, message=Hello Driver)
2015/08/22 13:46:46 INFO ResponseMessageCommunicationEvent ResponseMessageCommunicationEvent(response=true)
2015/08/22 13:46:46 INFO ResponseMessageBusinessEvent ResponseMessageBusinessEvent(response=true)
2015/08/22 13:46:46 INFO ResponseMessagePresenterEvent ResponseMessagePresenterEvent(response=true)

Tabela 7 - Informação recolhida pelo registo de canais utilizando o módulo Logging.

5.1.3 Teste de Carga

O teste de carga apresentado corresponde ao pedido da lista de veículos da aplicação pela

framework Volley e a resposta devolvida pelo servidor iZiTraN Mobile Service. O teste que é apresentado na Figura 53 foi o escolhido por conter o maior volume de dados que são transferidos entre servidor e aplicação móvel.

```

public class VehiclesTest {
    @Test
    public final void testRequestVehicles() {
        //start
        Timer.start();
        //set up params
        VehiclesPresenter paramPresenter = new VehiclesPresenter(Mockito.mock(Context.class));
        VehiclesView paramView = new VehiclesView();
        VehiclesModuleViewEvent paramRequestVehiclesEvent =
            new RequestVehiclesTestEvent().view(paramView);
        //expected result
        int expectedResult = 1500;
        //execute
        try {
            synchronized (paramView) {
                paramView.post(paramRequestVehiclesEvent);
                paramView.wait(30000);
            }
        } catch (Exception e) {
            fail(e.getMessage());
        }
        int result = paramView.result;
        //assert result
        assertEquals(expectedResult, result);
        //end
        Timer.endAndPrintElapsedSeconds();
    }

    private static class VehiclesView implements VehiclesTestAPI {
        protected int result;
        @Override
        public void responseVehicles(List<Vehicle> response) {
            result = response.size();
            synchronized (this) {
                notifyAll();
            }
        }

        public void post(VehiclesModuleViewEvent vehiclesModuleEvent) {
            VehiclesBroadcast.getInstance().post(vehiclesModuleEvent);
        }
    }
}

```

OK VehiclesTest (pt.tecmic.modules.vehicles)
OK All Tests Passed

Figura 53 - Teste de carga assíncrono para pedido da lista de veículos e resultado do mesmo.

Com o teste apresentado na Figura 53 foi possível testar a recepção de vários veículos enviados pelo servidor que serviram para obter informações consideradas relevantes, nomeadamente o tamanho de dados recebidos e o tempo médio que demorou desde o pedido até à resposta. Os dados obtidos (Tabela 8) representam dados médios calculados através de dez repetições aos testes para maximizar a exatidão dos mesmos.

Número de Veículos enviados pelo Servidor	Tamanho Médio de Dados Recebidos (em caracteres)	Tempo Médio (em segundos)
25	52951	1.341

50	109801	1.975
75	170476	2.171
100	234601	2.641
250	699501	4.020
500	1524001	7.799
750	2661001	10.497
1000	3048001	13.149
1500	3072001	25.030

Tabela 8 - Dados médios obtidos do pedido e resposta para obtenção da lista de veículos na aplicação.

As condições que originaram os valores temporais apresentados na Tabela 8 correspondem ao dispositivo móvel Android Moto G (2ª Geração) com versão 4.4.2. Este dispositivo comunica por uma ligação sem fios IEEE 802.11g com um valor limite de largura de banda de 30Mbps [135], a um servidor com processador Intel(R) Core(TM) i7 de 3.40GHz, com 8GB de RAM. Após uma análise aos resultados da Tabela 8 verificou-se que os tempos médios obtidos poderiam ainda ser melhorados, face ao tamanho médio de dados recebidos. Deste modo procedeu-se à alteração no iZiTraN Mobile Service para que o mesmo envia-se apenas no objeto Vehicle o seu último VehicleSnapshotEvent do dia atual. Depois desta alteração foram repetidos os testes, verificando-se melhorias de 9.6%, como é apresentado na Figura 54.

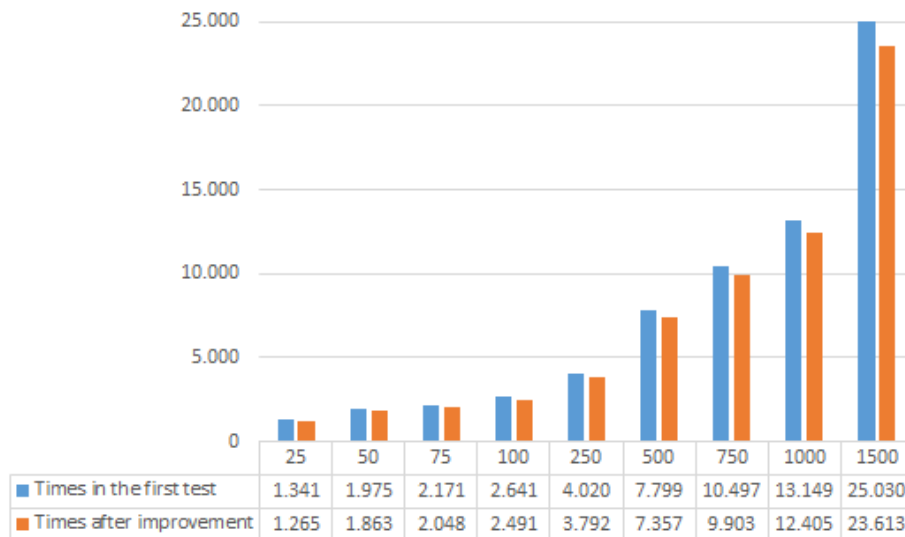


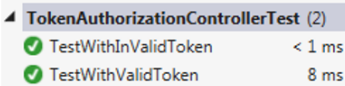
Figura 54 - Comparativo de melhoria de tempos médios obtidos pelos testes de carga.

5.1.4 Teste de Segurança

O teste de segurança apresentado corresponde à tentativa de acesso a um controlador do servidor iZiTraN Mobile Service que se encontra protegido pela autenticação por *token*. O teste que é apresentado na Figura 55 valida dois *tokens* que tentam aceder ao mesmo

controlador, sendo um válido e outro inválido.

```
[TestClass]
public class TokenAuthorizationControllerTest
{
    [TestMethod]
    public void TestWithInvalidToken()
    {
        // arrange
        var token = "fake";
        var expected = new HttpResponseMessage(HttpStatusCode.Unauthorized);
        var controller = new VehiclesController();
        controller.Request = new HttpRequestMessage();
        controller.Request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", token);
        // act
        var response = controller.GetVehicles();
        // assert
        Assert.AreEqual(expected.StatusCode, response.StatusCode);
    }
    [TestMethod]
    public void TestWithValidToken()
    {
        // arrange
        var token = "ES$48_2$;AG5&289~0ew@OG2$gV4f7";
        var expected = new HttpResponseMessage(HttpStatusCode.OK);
        var controller = new VehiclesController();
        controller.Request = new HttpRequestMessage();
        controller.Request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", token);
        // act
        var response = controller.GetVehicles();
        // assert
        Assert.AreEqual(expected.StatusCode, response.StatusCode);
    }
}
```



TokenAuthorizationControllerTest (2)	
✓ TestWithInvalidToken	< 1 ms
✓ TestWithValidToken	8 ms

Figura 55 - Teste de segurança para pedido da lista de veículos e resultado do mesmo.

5.1.5 Testes de Usabilidade

Foram realizados testes de usabilidade nas principais fases da aplicação móvel [136], tendo sido realizados pelas equipas de desenvolvimento da empresa devido ao seu forte conhecimento no setor de gestão de frotas. Primeiramente foi criado um guião de tarefas (Apêndice VIII) que serviu para indicar aos participantes os passos a serem seguidos ao longo dos testes. Após a elaboração do guião houve a realização de duas grandes fases de testes que serviram para analisar e registar os equívocos efetuados pelos participantes. As duas fases de testes são apresentadas nas seguintes subsecções.

5.1.5.1 Primeira fase

Foi realizada uma primeira fase de testes por forma a descobrir defeitos relativos à *interface* da aplicação. A equipa de desenvolvimento que realizou testes nesta fase foi a equipa de gestão de transportes pesados de mercadorias (6 pessoas), equipa a que este projeto se insere. Os dados retirados (Apêndice IX) pela análise aos testes revelaram resultados positivos, no entanto houve alguns equívocos que demonstraram alguma descontextualização em relação ao ecrã em que o participante se encontrava. Perante esta descontextualização foi refeita a parte

de navegação da aplicação móvel por forma a corrigir esta situação, como se pode comprovar na secção 4.5.1.2 que apresentou as melhorias dos ecrãs finais relativamente ao idealizado nos protótipos iniciais.

5.1.5.2 Segunda fase

Na segunda fase de testes foi a equipa de desenvolvimento de gestão de transportes de pesados de passageiros a realizar os mesmos (7 pessoas).

Os dados retirados (Apêndice IX) pela análise aos testes revelaram resultados muito positivos, uma vez que as tarefas do guião foram bem executadas, sem equívocos e com elevada percentagem de contextualização (95%), provando que a análise e consequentes alterações feitas após a primeira fase se revelaram acertadas.

5.2 Discussão de Resultados

Os testes efetuados apresentaram vários resultados animadores. Com os testes unitários apresentados conseguiu-se validar pequenos excertos de código com o objetivo de encontrar falhas dentro das várias funcionalidades da aplicação móvel, aumentando assim a qualidade e fiabilidade do código da mesma. O teste de integração apresentado conseguiu provar através do envio de uma mensagem que a integração entre as várias componentes (MVP, eventos e Web service) funciona corretamente. Já com o teste de carga foi mostrado que o limite de dados processados é o suficiente para a aplicação, uma vez que a métrica limite, estabelecida pela equipa de desenvolvimento, se encontrava na obtenção de 1500 veículos pelo tempo inferior a 30 segundos. Com o teste à segurança do servidor comprovou-se que o filtro de segurança por *tokens* está corretamente implementado.

Os testes de usabilidade efetuados ao longo das principais fases de desenvolvimento da aplicação conseguiram verificar que os protótipos não foram seguidos na sua totalidade, uma vez que devido à complexidade da aplicação houve a necessidade de melhorar alguns ecrãs. Para a concretização deste melhoramento houve um estudo adicional às diretrizes da plataforma Android (secção 4.5.1.3) para que os ecrãs finais satisfizessem todos os objetivos. De notar que ainda se terá de fazer os testes de aceitação ao cliente final, para depois se proceder à análise dos resultados dos mesmos.

No capítulo seguinte é feita a síntese e a análise global do trabalho. No mesmo capítulo também é apresentado o trabalho que se prevê desenvolver no futuro.

Conclusões e Trabalho Futuro

Para finalizar o presente documento são apresentadas neste capítulo as conclusões sobre o trabalho realizado no âmbito do projeto Mobile Fleet Management ao longo dos nove meses do mesmo. É também apresentado trabalho futuro que se prevê vir ainda a ser desenvolvido.

6.1 Conclusões

Com o projeto Mobile Fleet Management foi criada a aplicação móvel iZiTraN Mobile, para a plataforma Android, esta contém a apresentação de veículos em mapa, viagens, alarmes e mensagens com suporte a comunicações em tempo real. Este projeto representa uma aposta forte da Tecmic S.A. para plataformas móveis de forma a alargar e fazer evoluir as suas linhas de produtos (XTraN [137], 4Forces [138], entre outros).

Tal como apresentado ao longo deste documento, foi feita uma análise às aplicações móveis concorrentes e à plataforma iZiTraN Web. Esta análise foi essencial para escolher as funcionalidades que melhor se aplicariam ao iZiTraN Mobile, adaptando-as para a realidade dos dispositivos móveis. Foi também utilizada a metodologia XP que se tornou útil ao longo do projeto pela agilidade que a mesma proporcionou durante as fases de identificação, planeamento, execução e monitorização que são conhecidas na área da gestão de projetos.

Na fase de desenvolvimento de *software* foram tomadas inúmeras decisões, explicadas ao longo deste documento, e adquiridos conhecimentos úteis para a construção de um produto inovador com as qualidades necessárias e definidas pela Tecmic S.A. para todo o *software* que produz. Conseguiu-se uma arquitetura escalável que dá resposta às necessidades do cliente, como também possibilita futuras adições no processo de negócio do mesmo. Esta arquitetura utilizou no servidor tecnologias Microsoft, tendo em conta padrões de desenvolvimento e boas práticas de desenho arquitetural. A aplicação servidora, construída sobre a plataforma ASP .NET MVC disponibiliza um conjunto de Web services desenhados segundo os princípios arquiteturais SOA. A aplicação móvel foi criada para a plataforma Android e faz uso dos Web services criados. Com esta aplicação o operador pode ter um

contacto permanente com a sua frota, podendo assim usufruir de todos os benefícios que os dispositivos móveis oferecem. De destacar que a arquitetura apresentada representa uma importante inovação, uma vez que não é hábito o uso neste tipo de plataforma de uma estrutura modular baseada em eventos, mas foi mostrado ao longo do projeto os benefícios que este tipo de arquitetura EDA oferece, não tendo sido verificadas quaisquer perdas de performance com a arquitetura de eventos.

Deste projeto resultou ainda a produção de um artigo científico, aceite para publicação e apresentação na conferência internacional “CENTERIS - Conference on ENTERprise Information Systems” [45], entre os dias 7 e 9 de outubro de 2015.

6.2 Trabalho Futuro

Como trabalho futuro, existem vários pontos que se prevê virem a ser abordados num futuro próximo. De seguida são apresentados alguns deles:

- A atualização automática da aplicação móvel sem interação com o utilizador. Atualmente foi criada uma biblioteca que valida a versão atual e faz o *download* de uma nova versão (ficheiro .apk) procedendo depois à sua atualização no dispositivo móvel, ficando para trabalho futuro a sua integração com a aplicação móvel;
- A adição de mais módulos, como o de eficiência energética, de tempos de condução (Anexo I) e de pontuação de condutores, usando o conceito de Gamification [13];
- A adição de comunicações por voz para os veículos da frota através da API de voz do dispositivo móvel. Esta adição não foi considerada prioritária porque muitos dos veículos ainda não possuem comunicação por voz através da consola gráfica Tecmic;
- A identificação das ações que a aplicação pode fazer *offline*, para que se proceda à implementação de uma base de dados na aplicação móvel. Esta adição não foi considerada prioritária uma vez que o grande objetivo da aplicação é disponibilizar informação em tempo real, como localizações, alarmes, mensagens, entre outros;
- A execução dos testes de aceitação perante o cliente final e consequente análise dos resultados obtidos.

Por fim, nos próximos capítulos são listadas as referências consultadas ao longo do desenvolvimento do projeto e da redação do presente documento, os apêndices e ainda os anexos.

Referências

- [1] «TECMIC Inteligência e Inovação que otimizam a sua atividade». [Em linha]. Available: <http://www.tecmic.pt/>. [Acedido: 22-Set-2015].
- [2] «iZiTraN | Tecmic». [Em linha]. Available: <https://www.izitran.com/transportes-vidal-da-costa-figueiredo-utiliza-o-izitran/>. [Acedido: 22-Set-2015].
- [3] «Soluções - TECMIC». [Em linha]. Available: <http://www.tecmic.pt/solucoes/>. [Acedido: 22-Set-2015].
- [4] «Tecmic vence prémio de Melhor empresa Portuguesa na II Gala Anual do Empreendedorismo em Ciência e Tecnologia - TECMIC». [Em linha]. Available: <http://www.tecmic.pt/pt-br/tecmic-vence-premio-de-melhor-empresa-nacional-na-ii-gala-anual-do-empreendedorismo-em-ciencia-e-tecnologia/>. [Acedido: 22-Set-2015].
- [5] M. Wiberg, *Anytime, Anywhere Mobility*. 2009, pp. 164–168.
- [6] «Gestão de Frotas Online - iZiTraN - TECMIC». [Em linha]. Available: <http://www.tecmic.pt/portfolio/izitran/>. [Acedido: 22-Set-2015].
- [7] «Vodafone Portugal». [Em linha]. Available: <http://www.vodafone.pt/>. [Acedido: 22-Set-2015].
- [8] G. Taubes, «The Global Positioning System - The Role of Atomic Clocks», *Natl. Acad. Sci.*, pp. 2–8, 1997.
- [9] B. Walke, «The Roots of GPRS: The First System for Mobile Packet-Based Global Internet Access», *IEEE Wirel. Commun.*, pp. 15–17, 2013.
- [10] H. S. Oluwatosin, «Client-Server Model», *IOSR J. Comput. Eng.*, vol. 16, n. 1, pp. 67–71, 2014.
- [11] «One Platform for Connected Intelligence ~ Telogis». [Em linha]. Available: <http://www.telogis.com/>. [Acedido: 22-Set-2015].
- [12] «Telogis Coach». [Em linha]. Available: <http://www.telogis.com/solutions/mobile/coach>. [Acedido: 22-Set-2015].

- [13] G. Zichermann e C. Cunningham, *Gamification by Design - Implementing Game Mechanics in Web and Mobile Apps*. 2011, p. 208.
- [14] «Telogis Supervisor». [Em linha]. Available: <http://www.telogis.com/solutions/fleet/supervisor>. [Acedido: 22-Set-2015].
- [15] «Telogis WorkPlan». [Em linha]. Available: <http://www.telogis.com/solutions/mobile/work-order-management>. [Acedido: 22-Set-2015].
- [16] «GPS Fleet Tracking & Vehicle Tracking Solutions Fleetmatics». [Em linha]. Available: <http://www.fleetmatics.com/>. [Acedido: 22-Set-2015].
- [17] «Fleetmatics Reveal». [Em linha]. Available: <http://www.fleetmatics.com/reveal>. [Acedido: 22-Set-2015].
- [18] «Fleetmatics REVEAL Mobile». [Em linha]. Available: <https://play.google.com/store/apps/details?id=com.fleetmatics.presentation.mobile.android.sprite>. [Acedido: 22-Set-2015].
- [19] «Fleetmatics Reveal LogBook». [Em linha]. Available: <http://www.fleetmatics.com/reveal/logbook-compliance>. [Acedido: 22-Set-2015].
- [20] «Fleetmatics REVEAL LogBook Mobile». [Em linha]. Available: <https://play.google.com/store/apps/details?id=com.fleetmatics.redbull>. [Acedido: 22-Set-2015].
- [21] «Scania Group». [Em linha]. Available: <http://www.scania.com/>. [Acedido: 22-Set-2015].
- [22] «Scania Fleet Management». [Em linha]. Available: http://www.scania.com/products-services/services/fleet_management/. [Acedido: 22-Set-2015].
- [23] «Scania Fleet Management Mobile». [Em linha]. Available: <https://play.google.com/store/apps/details?id=se.scania.fms>. [Acedido: 22-Set-2015].
- [24] «Android». [Em linha]. Available: <https://www.android.com/>. [Acedido: 22-Set-2015].
- [25] «IOS - Apple». [Em linha]. Available: <http://www.apple.com/ios/>. [Acedido: 22-Set-2015].
- [26] «Windows Phone». [Em linha]. Available: <https://www.windowsphone.com/>. [Acedido: 22-Set-2015].
- [27] «GPS Fleet Tracking in your Mobile Device». [Em linha]. Available: <http://www.tsomobile.com/mobile-apps.html>. [Acedido: 22-Set-2015].
- [28] «COOLASIA - Fleet Management Solutions». [Em linha]. Available: <http://coolasia.net/>. [Acedido: 22-Set-2015].

- [29] «Manage your fleet from anywhere while on-the-go.» [Em linha]. Available: <http://www.gpsinsight.com/gps-fleet-tracking-functionality/mobile-access>. [Acedido: 22-Set-2015].
- [30] «WEBFLEET Mobile». [Em linha]. Available: http://business.tomtom.com/pt_pt/products/webfleet-mobile/highlights/. [Acedido: 22-Set-2015].
- [31] «MiX Fleet Manager». [Em linha]. Available: <http://www.mixtelematics.com/solutions/mix-fleet-manager>. [Acedido: 22-Set-2015].
- [32] «Dootrix delivers new vehicle tracking mobile app for Quartix». [Em linha]. Available: <http://dootrix.com/case-study-dootrix-delivers-new-vehicle-tracking-mobile-app-quartix/>. [Acedido: 22-Set-2015].
- [33] K. Beck e et al, «Manifesto for Agile Software Development», *Agile Alliance*, 2001.
- [34] WELLS, «D. Extreme Programming: A gentle introduction». [Em linha]. Available: <http://www.extremeprogramming.org/>. [Acedido: 22-Set-2015].
- [35] O. Hazzan e J. Tomayko, «Human Aspects of Software Engineering: The Case of Extreme Programming», 2004.
- [36] K. Beck e M. Fowler, *Planning extreme programming*. Addison-Wesley, 2004.
- [37] «EXTREME PROGRAMMING». [Em linha]. Available: <https://7bsp1018.wikispaces.com/eXtreme+Programming>. [Acedido: 22-Set-2015].
- [38] K. Wiegers, *More About Software Requirements: Thorny Issues and Practical Advice (Developer Best Practices)*. 2005.
- [39] N. Svensson e J. Gradén, «Software testing in extreme programming», pp. 1–16, 2009.
- [40] D. Talby, A. Keren, O. Hazzan, e Y. Dubinsky, «Agile Software Testing in a Large-Scale Project», *IEEE Computer Society*, pp. 30–37, 2006.
- [41] K. BECK, «Extreme Programming Explained», Massachusetts, 2000.
- [42] «Software de Gestão de Projetos | Microsoft Project». [Em linha]. Available: <https://products.office.com/project/>. [Acedido: 22-Set-2015].
- [43] D. Grover, «Graphical Project Planning Techniques: An Overview of Gantt, PERT, and CPM Charts», 2002.
- [44] *Developing a Gantt and PERT Chart*. Unitar, 2004.
- [45] «CENTERIS - Conference on ENTERprise Information Systems». [Em linha]. Available: <http://centeris.eiswatch.org/>. [Acedido: 22-Set-2015].
- [46] L. Ma, L. Gu, e J. Wang, «Research and Development of Mobile Application for Android Platform», *Int. J. Multimed. Ubiquitous Eng.*, vol. 9, n. 4, pp. 187–198, 2014.

- [47] «Welcome to the Android Open Source Project!». [Em linha]. Available: <http://source.android.com/>. [Acedido: 22-Set-2015].
- [48] «Android 4.4 Compatibility Definition», Google Inc, 2013.
- [49] C. L. G. Cabanban, «Development of Mobile Learning Using Android Platform», *Int. J. Inf. Technol. C omputer Sci.*, vol. 9, n. 1, pp. 98–106, 2013.
- [50] «Open Handset Alliance». [Em linha]. Available: <http://www.openhandsetalliance.com/index.html>. [Acedido: 22-Set-2015].
- [51] «Google - Company». [Em linha]. Available: <https://www.google.pt/intl/pt-PT/about/company/>. [Acedido: 22-Set-2015].
- [52] «Smartphone OS Market Share, Q3 2014». [Em linha]. Available: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. [Acedido: 22-Set-2015].
- [53] M. Wilcox e C. Voskoglou, «State of the Developer Nation Q3 2014», 2014.
- [54] «Android Studio Overview». [Em linha]. Available: <http://developer.android.com/tools/studio/index.html>. [Acedido: 22-Set-2015].
- [55] «Visual Studio - Microsoft Developer Tools». [Em linha]. Available: <https://www.visualstudio.com/>. [Acedido: 22-Set-2015].
- [56] «ASP.NET | The ASP.NET». [Em linha]. Available: <http://www.asp.net/>. [Acedido: 22-Set-2015].
- [57] B. Collins-Sussman, B. W. Fitzpatrick, e C. M. Pilato, *Version Control with Subversion*. 2011, pp. 1–14.
- [58] Srirangan, *Apache Maven 3 Cookbook*, PacktPub. 2011, p. Chapter No.2 «Software Engineering Techniques».
- [59] T. O’Brien, S. McCulloch, e B. Demers, *The Maven Cookbook*, A Sonatype. 2010, pp. 91–94.
- [60] «NuGet|Home». [Em linha]. Available: <https://www.nuget.org/>. [Acedido: 22-Set-2015].
- [61] T. Berglund e M. M. (O’Reilly), *Building and Testing with Gradle*, Gradle, In. 2011, pp. 47–68.
- [62] R. T. Fielding e R. N. Taylor, *Principled design of the modern Web architecture*. 2000, pp. 407–416.
- [63] E. Rescorla, «HTTP Over TLS», 2000.
- [64] E.Rescorla, *SSL and TLS - Designing and Building Secure Systems*. Addison-Wesley, 2001.

- [65] N. Nurseitov, M. Paulson, R. Reynolds, e C. Izurieta, *Comparison of JSON and XML Data Interchange Formats: A Case Study*. 2009.
- [66] D. Crockford, *JSON: The fat-free alternative to XML*. 2006.
- [67] D. Evans, «The Internet of Things How the Next Evolution of the Internet Is Changing Everything», *Cisco Internet Bus. Solut. Gr.*, 2011.
- [68] M. Butcher, «REST Without JSON: The Future of IoT Protocols», *IoT Zone*, 2015.
- [69] «Transmitting Network Data Using Volley». [Em linha]. Available: <http://developer.android.com/training/volley/index.html>. [Acedido: 22-Set-2015].
- [70] F. Kirkpatrick, *Volley - Easy, Fast Networking for Android*. Google, Inc., 2013.
- [71] «Google I/O 2013». [Em linha]. Available: <https://developers.google.com/events/io/>. [Acedido: 22-Set-2015].
- [72] A. L. Ananda, B. H. Tay, e E. K. Koh, «A Survey of Asynchronous Remote Procedure Calls», *Softw. Eng. J.*, pp. 92–109, 1987.
- [73] S. Bagchi, «The software architecture for designing interactive mobile computing applications», *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 10, n. 4, 2012.
- [74] S. E. Panitz, «The Scrap Your Boilerplate Pattern in Java», 2005.
- [75] G. Block, Pablo Cibraro, Pedro Felix, Howard Dierking, e Darrel Miller, *Designing Evolvable Web APIs with ASP.NET*. O'Reilly Media, 2014.
- [76] J. Kurtz, *ASP.NET MVC4 and the Web API - Building a REST Service from Start to Finish*. Apress, 2013.
- [77] J. M. Aguilar, *SignalR Programming in Microsoft ASP.NET*. Microsoft Press, 2014.
- [78] T. D. R. Varela e S. Loh, «Implementação e Análise da Utilização de WebSockets em Sistemas Computacionais», pp. 1–19.
- [79] «World Wide Web Consortium (W3C)». [Em linha]. Available: <http://www.w3.org/>. [Acedido: 22-Set-2015].
- [80] «About the IETF». [Em linha]. Available: <https://www.ietf.org/about/>. [Acedido: 22-Set-2015].
- [81] G. Thomas, «Introduction To The Transmission Control Protocol», *Introd. to Ind. Ethernet*, vol. 1, n. 5, pp. 1–4, 2000.
- [82] V. M. Themudo, «Implementação de um servidor de negociação em bolsa baseado em WebSocket», 2014.
- [83] «Web Map Service». [Em linha]. Available: <http://www.opengeospatial.org/standards/wms>. [Acedido: 22-Set-2015].

- [84] «WMS reference». [Em linha]. Available: <http://docs.geoserver.org/stable/en/user/services/wms/reference.html>. [Acedido: 22-Set-2015].
- [85] «OpenStreetMap». [Em linha]. Available: <https://www.openstreetmap.org/>. [Acedido: 22-Set-2015].
- [86] «Google Maps». [Em linha]. Available: <https://www.google.pt/maps>. [Acedido: 22-Set-2015].
- [87] «Bing Maps». [Em linha]. Available: <https://www.bing.com/maps/>. [Acedido: 22-Set-2015].
- [88] «API do Google Maps». [Em linha]. Available: <https://developers.google.com/maps/>. [Acedido: 22-Set-2015].
- [89] «API do Google Maps for Work». [Em linha]. Available: <https://www.google.com/work/mapsearch/products/mapsapi.html>. [Acedido: 22-Set-2015].
- [90] «gvSIG Mini». [Em linha]. Available: <https://confluence.prodevelop.es/display/GVMN/Home>. [Acedido: 22-Set-2015].
- [91] «Prodevelop - Integración de tecnologías». [Em linha]. Available: <http://www.prodevelop.es/>. [Acedido: 22-Set-2015].
- [92] «Androzic - Android navigation application». [Em linha]. Available: <http://androzic.com/>. [Acedido: 22-Set-2015].
- [93] «Tile Map Service Specification». [Em linha]. Available: http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification. [Acedido: 22-Set-2015].
- [94] «GeoServer is an open source server for sharing geospatial data». [Em linha]. Available: <http://geoserver.org/>. [Acedido: 22-Set-2015].
- [95] «OpenLayers». [Em linha]. Available: <http://openlayers.org/>. [Acedido: 22-Set-2015].
- [96] «Framework OL3». [Em linha]. Available: <https://github.com/netgis/ol3>. [Acedido: 22-Set-2015].
- [97] «PhoneGap». [Em linha]. Available: <http://phonegap.com/>. [Acedido: 22-Set-2015].
- [98] T. Luo, H. Hao, W. Du, Y. Wang, e H. Yin, «Attacks on WebView in the Android System», 2011.
- [99] T. Luo, X. Jin, A. Ananthanarayanan, e W. Du, «Touchjacking Attacks on Web in Android, iOS, and Windows Phone», 2011.
- [100] «OpenStreetMap - Tools for Android». [Em linha]. Available: <https://github.com/osmdroid/osmdroid>. [Acedido: 22-Set-2015].

- [101] D. Krafzig, K. Banke, e D. Slama, *Enterprise SOA: Service-Oriented Architecture Best Practices*. Prentice Hall, 2004.
- [102] «Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST)». [Em linha]. Available: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. [Acedido: 22-Set-2015].
- [103] «Simple Object Access Protocol (SOAP) 1.1». [Em linha]. Available: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>. [Acedido: 22-Set-2015].
- [104] J. Deacon, «Model-View-Controller (MVC) Architecture», 2013.
- [105] E. Ingebrigtsen, *SignalR: Real-time Application Development*. Packt Publishing Ltd., 2013.
- [106] «What Is Windows Communication Foundation». [Em linha]. Available: <https://msdn.microsoft.com/en-us/library/ms731082.aspx>. [Acedido: 22-Set-2015].
- [107] «.NET Remoting Overview». [Em linha]. Available: <https://msdn.microsoft.com/en-us/library/kwdt6w2k%28v=vs.71%29.aspx>. [Acedido: 22-Set-2015].
- [108] S. McLean, J. Naftel, e K. Williams, «Microsoft .NET Remoting», *Microsoft Press*, 2002.
- [109] «Microsoft». [Em linha]. Available: <https://www.microsoft.com/>. [Acedido: 22-Set-2015].
- [110] Adams e Gilchrist, «RFC 2612», 1999.
- [111] Franks e et al, «RFC 2617», 1999.
- [112] B. Lakshmiraghavan, *Pro ASP.NET Web API Security*. Apress, 2013.
- [113] G. Dhillon, *Principles of Information Systems Security: text and cases*. 2007.
- [114] F. S. Gharehchopogh e M. Abdollahias, «A New MVP Based Approach In Software Arquitecure For Time Reduction In Software Development», *Int. J. Found. Comput. Sci. Technol.*, vol. 3, 2013.
- [115] A. Syromiatnikov e D. Weyns, «A Journey Through the Land of Model-View-* Design Patterns», 2013.
- [116] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. 2006, p. 720.
- [117] B. M. Michelson, *Event-Driven Architecture Overview*. 2006.
- [118] K. Chandy e W. R. Schulte, *Event Processing: Designing IT Systems for Agile Companies*. 2009.

- [119] K. Bourrillion, «An Overview of Guava: Google Core Libraries for Java», *QCon*, 2012.
- [120] «Framework Guava». [Em linha]. Available: <https://github.com/google/guava>. [Acedido: 22-Set-2015].
- [121] E. Gamma, R. Helm, R. Johnson, e J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994, p. 416.
- [122] «Material design in Google design». [Em linha]. Available: <https://design.google.com/articles/>. [Acedido: 22-Set-2015].
- [123] «Material design guidelines». [Em linha]. Available: <https://www.google.com/design/spec/material-design/introduction.html>. [Acedido: 22-Set-2015].
- [124] «Android Design». [Em linha]. Available: <http://developer.android.com/design/index.html>. [Acedido: 22-Set-2015].
- [125] «Iconography». [Em linha]. Available: <https://developer.android.com/design/style/iconography.html>. [Acedido: 22-Set-2015].
- [126] «Font Awesome, the iconic font and CSS toolkit». [Em linha]. Available: <http://fontawesome.io/>. [Acedido: 22-Set-2015].
- [127] «Style Color». [Em linha]. Available: <https://www.google.com/design/spec/style/color.html>. [Acedido: 22-Set-2015].
- [128] «Writing Style». [Em linha]. Available: <https://developer.android.com/design/style/writing.html>. [Acedido: 22-Set-2015].
- [129] «Metrics and Grids». [Em linha]. Available: <https://developer.android.com/design/style/metrics-grids.html>. [Acedido: 22-Set-2015].
- [130] «Navigation with Back and Up». [Em linha]. Available: <https://developer.android.com/design/patterns/navigation.html>. [Acedido: 22-Set-2015].
- [131] «Navigation Drawer». [Em linha]. Available: <https://developer.android.com/design/patterns/navigation-drawer.html>. [Acedido: 22-Set-2015].
- [132] «Confirming & Acknowledging». [Em linha]. Available: <https://developer.android.com/design/patterns/confirming-acknowledging.html>. [Acedido: 22-Set-2015].
- [133] K. Beck, *Test-Driven Development By Example*. Addison-Wesley Professional, 2003, pp. 125–153.

- [134] H. Kim, B. Choi, e W. E. Wong, *Performance Testing of Mobile Applications at the Unit Test Level*, IEEE. Secure Software Integration and Reliability Improvement, 2009. SSIRI 2009. Third IEEE International Conference on, 2009, pp. 171–180.
- [135] «IEEE-SA -IEEE Get 802 Program - 802.11: Wireless LANs». [Em linha]. Available: <http://standards.ieee.org/about/get/802/802.11.html>. [Acedido: 22-Set-2015].
- [136] K. B. Lee e R. A. Grice, *Developing a new usability testing method for mobile devices*, IEEE. Professional Communication Conference, 2004. IPCC 2004. Proceedings. International, 2004, pp. 115–127.
- [137] «XTraN». [Em linha]. Available: <http://www.tecmic.pt/portfolio/xtran/>. [Acedido: 22-Set-2015].
- [138] «4Forces». [Em linha]. Available: <http://www.tecmic.pt/portfolio/4forces/>. [Acedido: 22-Set-2015].
- [139] T. Cata, P. S. Patel, e T. Sakaguchi, «QR Code: A New Opportunity for Effective Mobile Marketing», *J. Mob. Technol. Knowl. Soc.*, vol. 2013, p. 7, 2013.
- [140] M. Stigge, H. Plötz, W. Müller, e J.-P. Redlich, «Reversing CRC - Theory and Practice», *HU Berlin Public Rep.*, pp. 1–3, 2006.

Apêndices

Neste capítulo são apresentados os apêndices a este documento, que correspondem aos artefactos que foram criados ao longo da evolução do projeto Mobile Fleet Management.

Apêndice I - User Stories

O operador da frota é o utilizador da aplicação móvel, tendo como objetivo visualizar indicadores dos veículos da frota, agir em tempo real a situações de emergência e analisar o estado dos veículos. A extração de *user stories* foi feita por meio da identificação dos requisitos específicos do operador de frota, através do conhecimento da equipa de desenvolvimento no setor da gestão de frotas.

A Tabela 9 apresenta as *user stories* com as descrições das ações e funcionalidades que a aplicação móvel tentou corresponder às necessidades do operador de frota.

US#	User Story
US1	Como utilizador do sistema eu quero autenticar-me para que possa aceder em segurança à minha própria conta.
US2	Como utilizador do sistema eu quero terminar sessão para que possa sair em segurança da aplicação.
US3	Como utilizador do sistema eu quero alterar idioma para que possa utilizar o idioma mais compreensível para mim.
US4	Como utilizador do sistema eu quero alterar as coordenadas do mapa para que possa visualizar a zona que mais me importa.
<u>Veículos</u>	
US5	Como utilizador do sistema eu quero ter veículos favoritos para ter um grupo de veículos mais importantes.
US6	Como utilizador do sistema eu quero ver todos os veículos em mapa para que possa rapidamente ver a posição geográfica de todos os veículos.
US7	Como utilizador do sistema eu quero ver todos os veículos favoritos em mapa para que possa rapidamente ver a posição geográfica dos veículos favoritos.
US8	Como utilizador do sistema eu quero ver mapa em <i>fullscreen</i> para que possa ver o máximo número de veículos em todo o ecrã do meu dispositivo móvel.
US9	Como utilizador do sistema eu quero ver todos os veículos em lista para que possa seleccionar o veículo que desejar.
US10	Como utilizador do sistema eu quero ver todos os veículos favoritos em lista para que possa rapidamente seleccionar o veículo favorito que desejar.
US11	Como utilizador do sistema eu quero ver todos os veículos por grupo para que possa rapidamente e de forma organizada ver todos os veículos agrupados.
US12	Como utilizador do sistema eu quero expandir/colapsar grupos de veículos para que possa ver somente os veículos de um grupo específico.
US13	Como utilizador do sistema eu quero pesquisar veículo por nome.
US14	Como utilizador do sistema eu quero pesquisar veículo por matrícula.
US15	Como utilizador do sistema eu quero pesquisar veículo por localização.
US16	Como utilizador do sistema eu quero pesquisar veículo por condutor ativo.
US17	Como utilizador do sistema eu quero seleccionar veículo em mapa para que possa ver informação do veículo.
US18	Como utilizador do sistema eu quero seleccionar veículo em lista para que possa ver informação do veículo.
US19	Como utilizador do sistema eu quero seleccionar veículo em pesquisa para que

	possa ver informação do veículo.
US20	Como utilizador do sistema eu quero ver contador de número de veículos (disponíveis, em condução, parados, em estacionamento) para que possa rapidamente saber o estado atual da frota.
US21	Como utilizador do sistema eu quero ver estado de disponibilidade de veículo (disponível, em condução, parado, em estacionamento) para que possa rapidamente saber o estado atual do veículo.
US22	Como utilizador do sistema eu quero ver indicadores de veículo (velocidade média, distância percorrida, tempo de viagem) para que possa ver informações indicadoras em tempo real.
US23	Como utilizador do sistema eu quero ver localização descritiva de veículo para que possa facilmente saber onde este se encontra.
US24	Como utilizador do sistema eu quero marcar veículo como favorito para que possa ter um grupo simplificado com os veículos favoritos.
US25	Como utilizador do sistema eu quero desmarcar veículo como favorito para que possa remover veículos do grupo favoritos.
US26	Como utilizador do sistema eu quero seguir veículo para que possa ver informações do veículo em tempo real.
US27	Como utilizador do sistema eu quero ver tempo decorrido desde a última comunicação de veículo para que possa saber a margem de erro das informações.
<u>Viagens</u>	
US28	Como utilizador do sistema eu quero ver viagem de veículo para que possa ver informações atuais da viagem corrente do veículo.
US29	Como utilizador do sistema eu quero ver viagens de veículo por percurso para que possa ver informações de cada viagem já ocorrida com o veículo.
US30	Como utilizador do sistema eu quero ver previsão de chegada ao destino de veículo para que possa ter o conhecimento da hora prevista de chegada.
<u>Alarmes</u>	
US31	Como utilizador do sistema eu quero ver notificação de alarme recebido para que possa identificar a chegada de um alarme.
US32	Como utilizador do sistema eu quero receber alarme para que possa ter conhecimento das emergências alarmísticas do veículo.
US33	Como utilizador do sistema eu quero responder a alarme para que possa tratar das emergências alarmísticas do veículo.
US34	Como utilizador do sistema eu quero eliminar localmente alarme para que possa ver somente alarmes mais recentes e importantes.
US35	Como utilizador do sistema eu quero ver número de alarmes não visualizados para que possa identificar os alarmes que ainda estão por visualizar.
US36	Como utilizador do sistema eu quero ver listagem de todos os alarmes para que possa ver todos os alarmes e saber a sua ordem de ocorrência.
US37	Como utilizador do sistema eu quero ver remetente de alarme recebido para que possa facilmente identificar quem desencadeou o alarme.
<u>Mensagens</u>	
US38	Como utilizador do sistema eu quero ver notificação de mensagem recebida para que possa identificar a chegada de uma notificação de mensagem.
US39	Como utilizador do sistema eu quero receber mensagem para que possa ler o que o veículo me deseja comunicar.
US40	Como utilizador do sistema eu quero marcar uma mensagem como lida/não lida para me organizar melhor nas mensagens recebidas.
US41	Como utilizador do sistema eu quero escrever mensagem em rascunho para que a

	possa posteriormente enviar.
US42	Como utilizador do sistema eu quero enviar mensagem para que possa comunicar com o veículo.
US43	Como utilizador do sistema eu quero eliminar mensagem para que possa remover uma mensagem.
US44	Como utilizador do sistema eu quero ver número de mensagens não visualizadas para que possa identificar as mensagens que ainda estão por ler.
US45	Como utilizador do sistema eu quero ver listagem de todas as conversas para que possa ver todas as conversas e saber a sua ordem de ocorrência.
US46	Como utilizador do sistema eu quero ver listagem de todas as mensagens por conversa para que possa ver todas as mensagens e saber a sua ordem de ocorrência.
US47	Como utilizador do sistema eu quero ver remetente de mensagem recebida para que possa facilmente identificar quem enviou a mensagem.
US48	Como utilizador do sistema eu quero ver destinatário de mensagem enviada para que possa facilmente identificar a quem enviei a mensagem.
US49	Como utilizador do sistema eu quero ver confirmação de mensagem a enviar para que possa facilmente identificar o estado da mensagem.
US50	Como utilizador do sistema eu quero ver confirmação de mensagem enviada para que possa facilmente identificar o estado da mensagem.
US51	Como utilizador do sistema eu quero ver confirmação de mensagem entregue para que possa facilmente identificar o estado da mensagem.
US52	Como utilizador do sistema eu quero ligar para o veículo para comunicar rapidamente com o mesmo.

Tabela 9 - User Stories.

Apêndice II - Matriz de Wiegers

Por forma a apoiar a seleção de importância de prioridades nas *users stories* foi seguida a utilização da matriz de Wiegers, idealizada por Karl Wiegers. Esta é apresentada pela Tabela 10.

	Stakeholder 1 Pedro Marques		Stakeholder 2 David Mendes		Stakeholder 3 João Domingos					
Weights:	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		
US#	Relative Benefit	Relative Penalty	Relative Benefit	Relative Penalty	Relative Benefit	Relative Penalty	Total Benefit	Total Penalty	Total Value	% Value
US1	9	6	9	5	8	8	8,7	6,3	15,0	1,9
US2	9	7	9	5	8	9	8,7	7,0	15,7	2,0
US3	6	8	5	8	7	8	6,0	8,0	14,0	1,8
US4	5	7	5	8	6	6	5,3	7,0	12,3	1,6
<u>Veículos</u>										
US5	6	9	5	9	7	8	6,0	8,7	14,7	1,9
US6	4	6	3	6	5	7	4,0	6,3	10,3	1,3
US7	6	9	5	9	7	8	6,0	8,7	14,7	1,9
US8	8	8	8	7	8	9	8,0	8,0	16,0	2,0
US9	5	5	7	6	3	3	5,0	4,7	9,7	1,2
US10	7	6	7	6	6	5	6,7	5,7	12,3	1,6
US11	3	3	1	1	6	7	3,3	3,7	7,0	0,9
US12	8	8	9	8	7	7	8,0	7,7	15,7	2,0
US13	9	9	9	9	8	9	8,7	9,0	17,7	2,3
US14	9	9	9	9	8	9	8,7	9,0	17,7	2,3
US15	7	8	7	8	7	7	7,0	7,7	14,7	1,9
US16	5	7	5	8	5	6	5,0	7,0	12,0	1,5
US17	9	8	9	9	8	7	8,7	8,0	16,7	2,1
US18	9	9	9	9	8	8	8,7	8,7	17,3	2,2
US19	9	9	9	9	8	8	8,7	8,7	17,3	2,2
US20	7	7	7	8	7	7	7,0	7,3	14,3	1,8
US21	6	9	5	9	8	9	6,3	9,0	15,3	2,0
US22	8	8	9	9	7	6	8,0	7,7	15,7	2,0
US23	9	9	9	9	8	8	8,7	8,7	17,3	2,2
US24	7	9	7	9	8	9	7,3	9,0	16,3	2,1
US25	7	9	7	9	8	9	7,3	9,0	16,3	2,1
US26	8	8	9	9	6	6	7,7	7,7	15,3	2,0
US27	8	8	9	9	6	6	7,7	7,7	15,3	2,0
<u>Viagens</u>										
US28	8	8	9	9	7	7	8,0	8,0	16,0	2,0
US29	8	8	9	9	7	7	8,0	8,0	16,0	2,0
US30	7	8	7	8	7	7	7,0	7,7	14,7	1,9
<u>Alarmes</u>										
US31	7	9	9	9	4	8	6,7	8,7	15,3	2,0
US32	9	9	9	9	9	9	9,0	9,0	18,0	2,3

US33	7	9	7	9	8	8	7,3	8,7	16,0	2,0
US34	6	8	5	7	7	9	6,0	8,0	14,0	1,8
US35	8	9	9	9	7	8	8,0	8,7	16,7	2,1
US36	5	8	5	9	5	7	5,0	8,0	13,0	1,7
US37	8	9	9	9	6	8	7,7	8,7	16,3	2,1
<u>Mensagens</u>										
US38	9	9	9	9	8	9	8,7	9,0	17,7	2,3
US39	9	9	9	9	8	9	8,7	9,0	17,7	2,3
US40	1	1	1	1	1	1	1,0	1,0	2,0	0,3
US41	7	9	7	9	6	8	6,7	8,7	15,3	2,0
US42	9	9	9	9	6	9	8,7	9,0	17,7	2,3
US43	6	8	5	7	7	9	6,0	8,0	14,0	1,8
US44	8	9	9	9	6	7	7,7	8,3	16,0	2,0
US45	7	8	9	9	5	6	7,0	7,7	14,7	1,9
US46	7	8	9	9	5	6	7,0	7,7	14,7	1,9
US47	9	9	9	9	8	8	8,7	8,7	17,3	2,2
US48	9	9	9	9	8	8	8,7	8,7	17,3	2,2
US49	9	9	9	9	8	9	8,7	9,0	17,7	2,3
US50	9	9	9	9	8	9	8,7	9,0	17,7	2,3
US51	9	9	9	9	8	9	8,7	9,0	17,7	2,3
US52	6	6	5	5	7	7	6,0	6,0	12,0	1,5
Totals:	380	414	387	418	356	391	374,3	407,7	782,0	100,0

Tabela 10 - Matriz de Wieggers.

Apêndice III - Gráfico de Gantt

Por forma a apoiar a gestão do projeto Mobile Fleet Management foram planificadas as tarefas que viriam a ser realizadas ao longo do mesmo, tendo sido a ferramenta Microsoft Project a utilizada para criar o Gráfico de Gantt do mesmo, como apresentado na Figura 56.

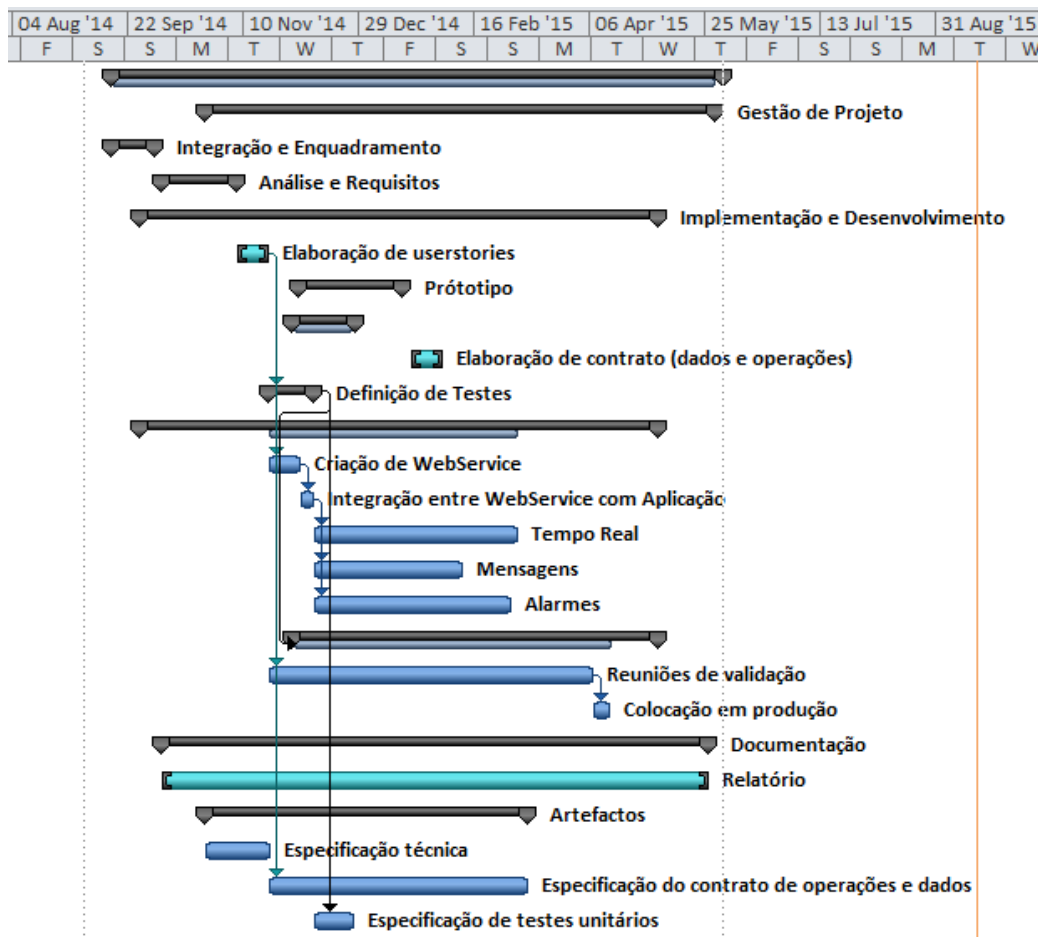


Figura 56 - Gráfico de Gantt com as principais tarefas.

Apêndice IV - Protótipos Iniciais

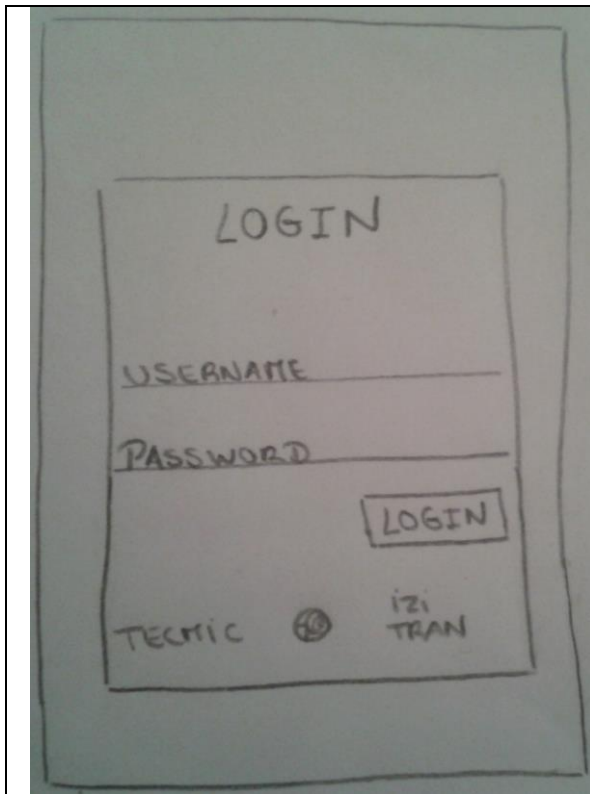


Figura 57 - Ecrã Login.

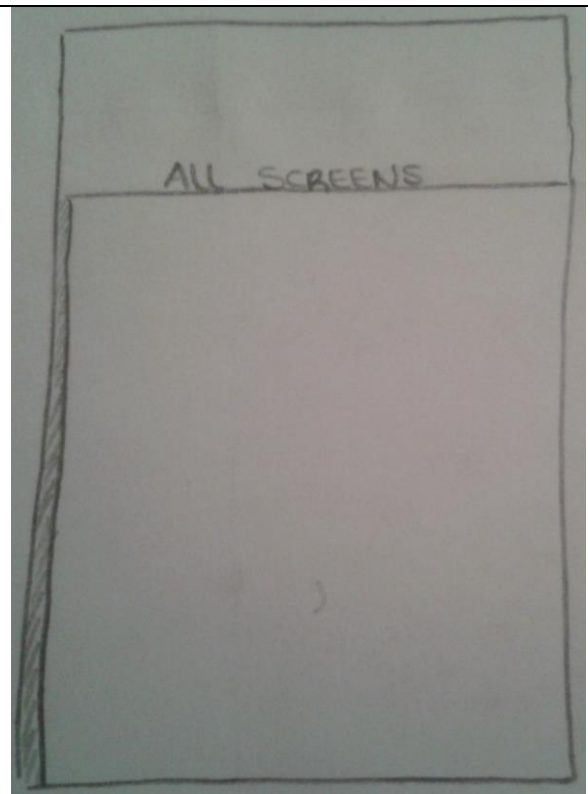


Figura 58 - Ecrã Painel Lateral.

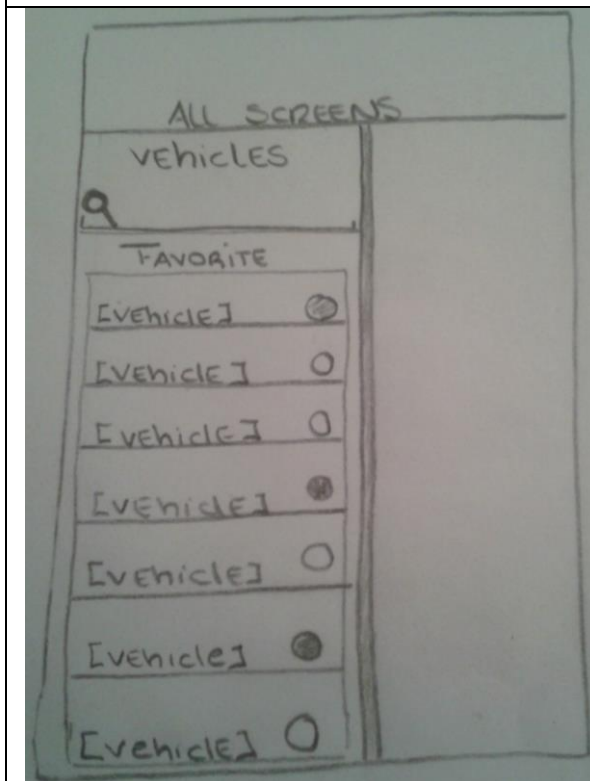


Figura 59 - Painel Lateral com dados.

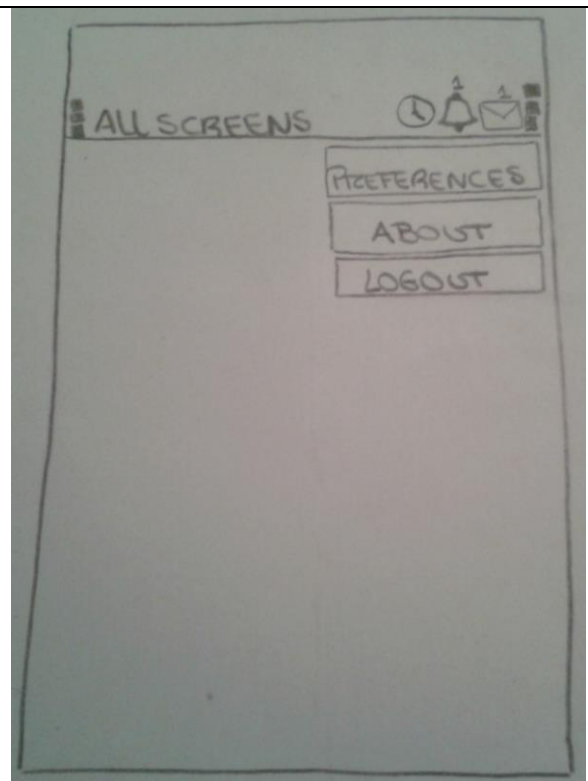


Figura 60 - Opções da Aplicação.

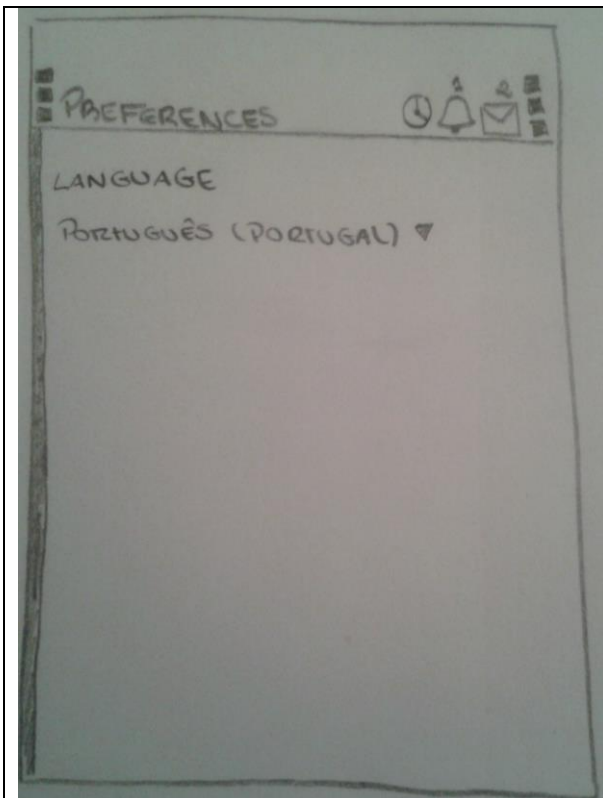


Figura 61 - Ecrã Preferências.

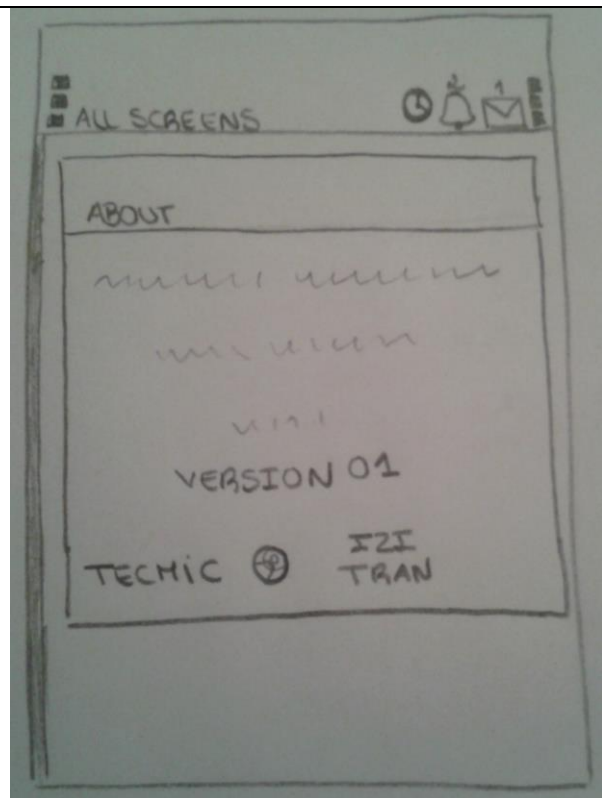


Figura 62 - Ecrã Sobre a Aplicação.

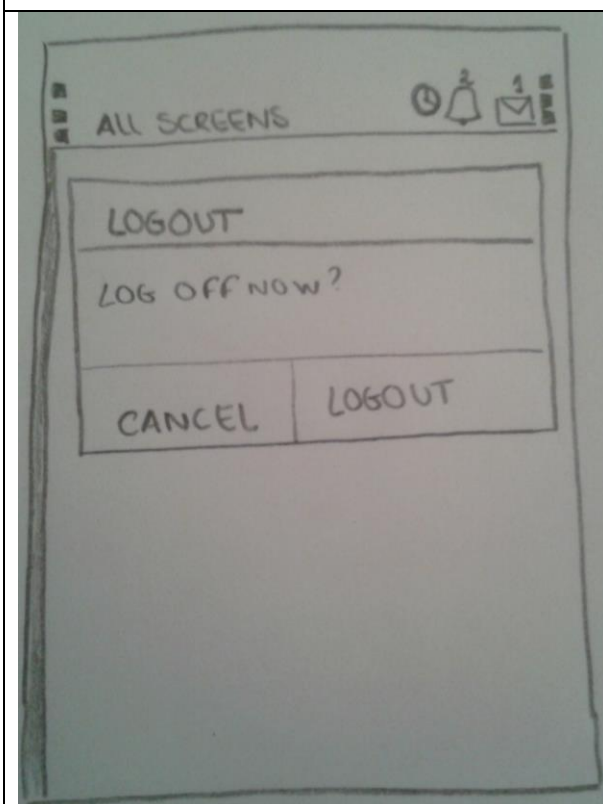


Figura 63 - Ecrã Terminar Sessão.

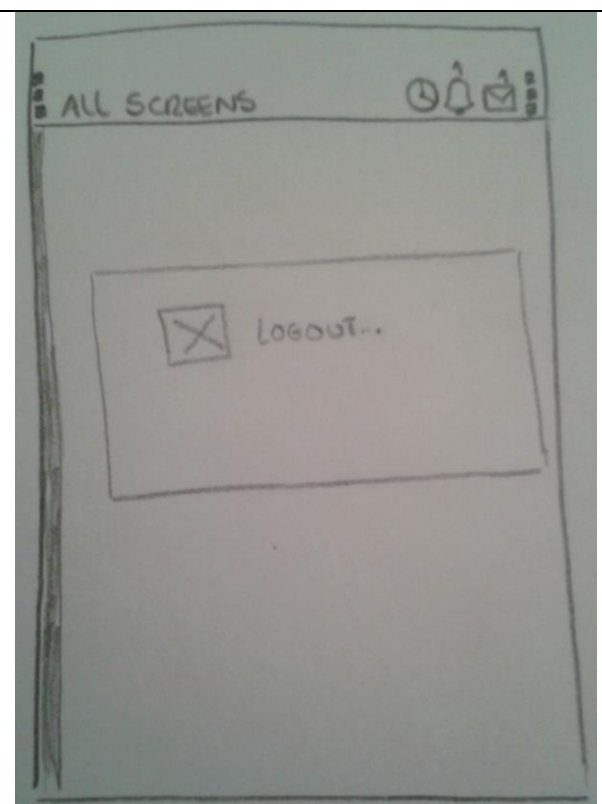


Figura 64 - Mensagem de Terminar Sessão.

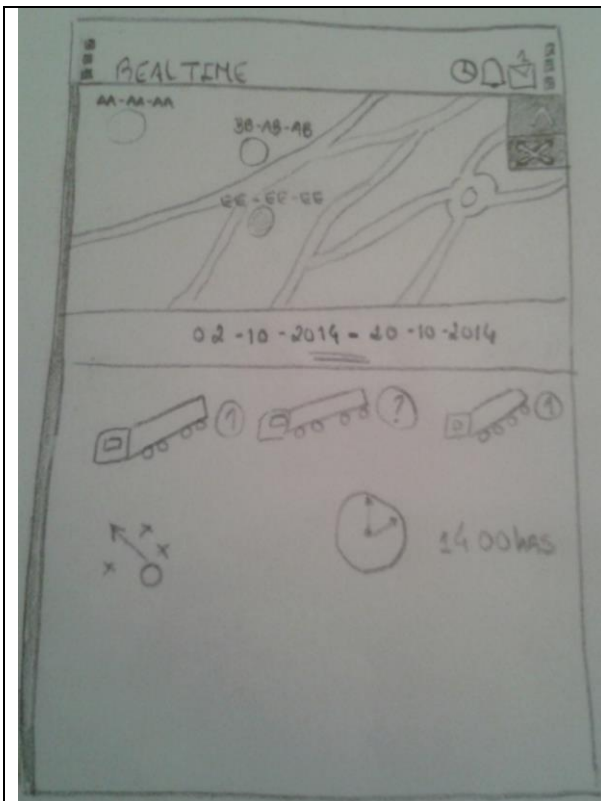


Figura 65 - Ecrã Real Time Geral.

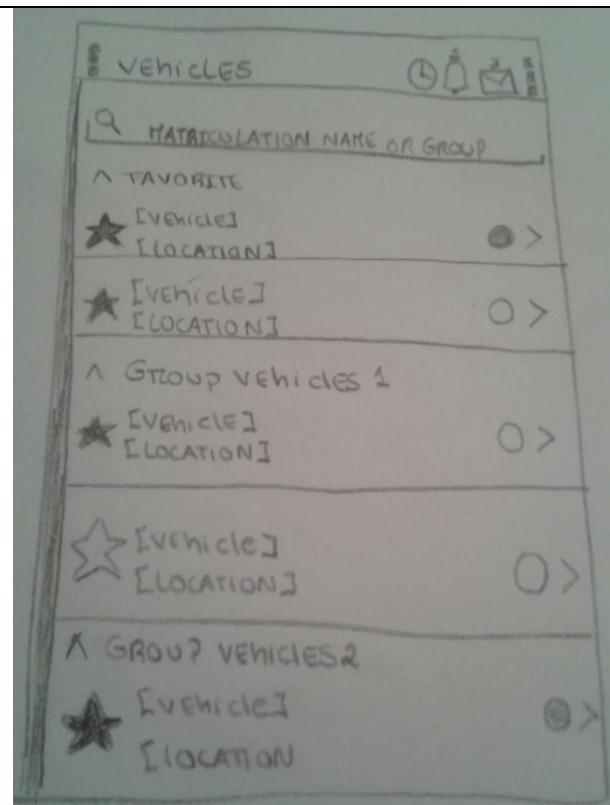


Figura 66 - Ecrã de Veículos.

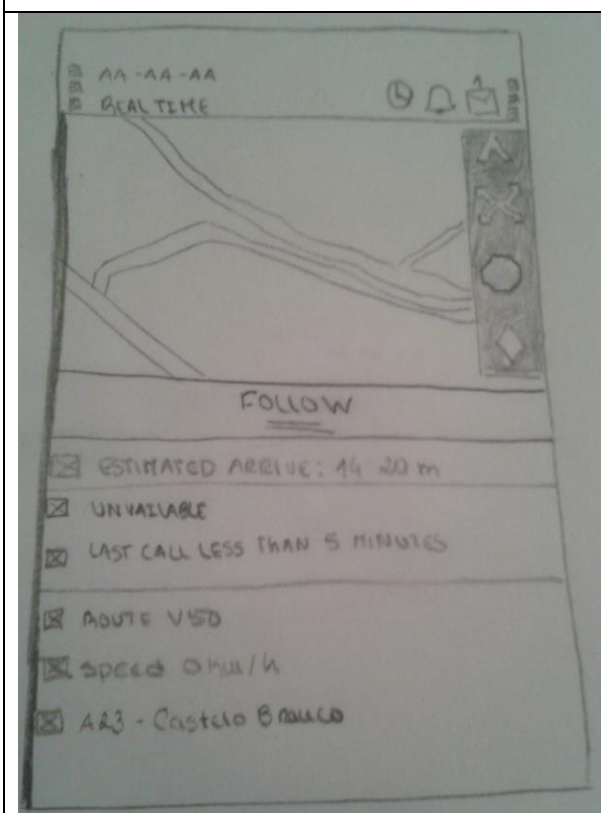


Figura 67 - Ecrã Seguir.

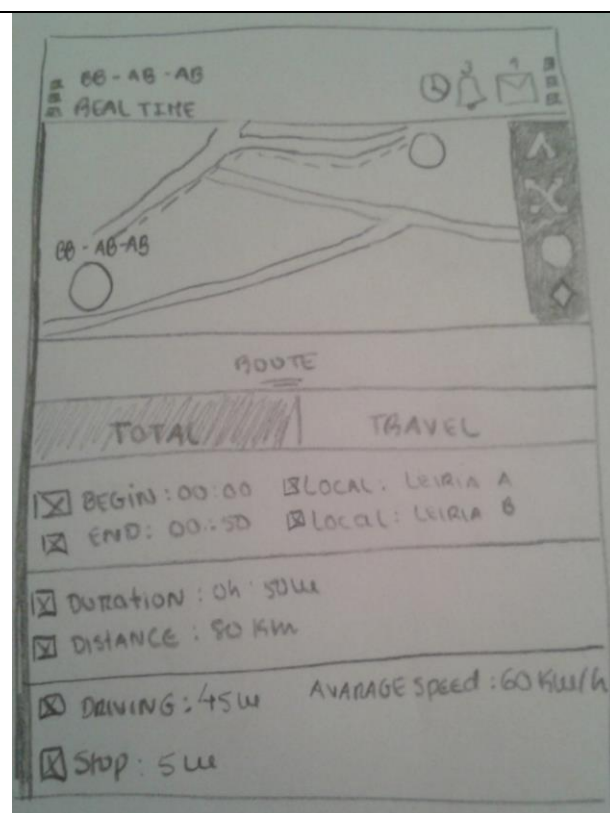


Figura 68 - Ecrã Diário.

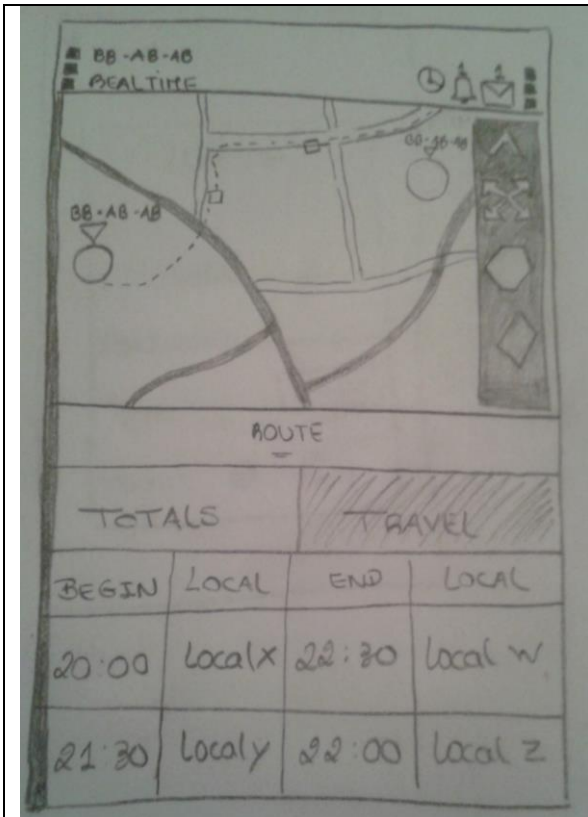


Figura 69 - Ecrã Viagens.

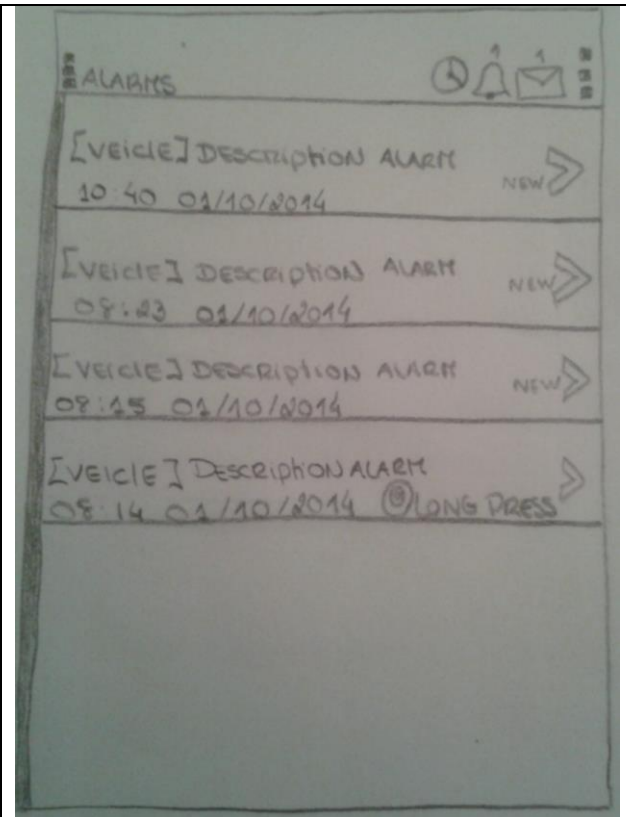


Figura 70 - Ecrã Alarmes.

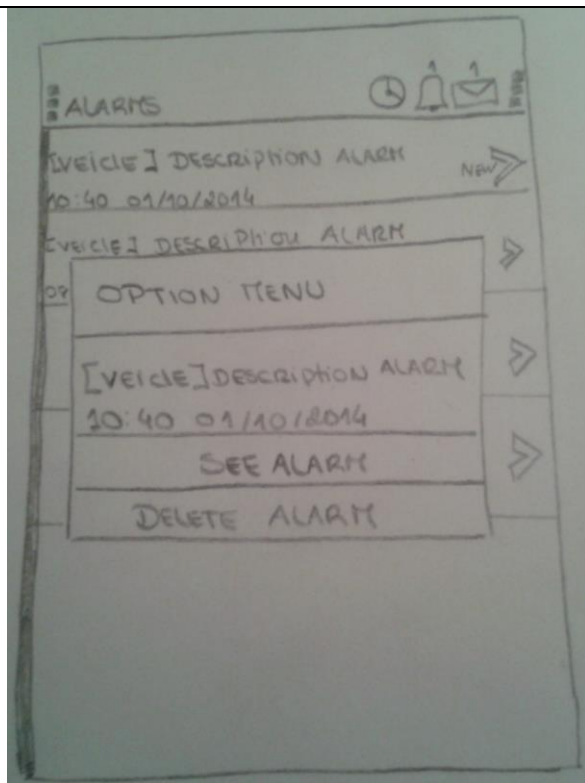


Figura 71 - Opções sobre Alarme.



Figura 72 - Ecrã Eliminar Alarme.

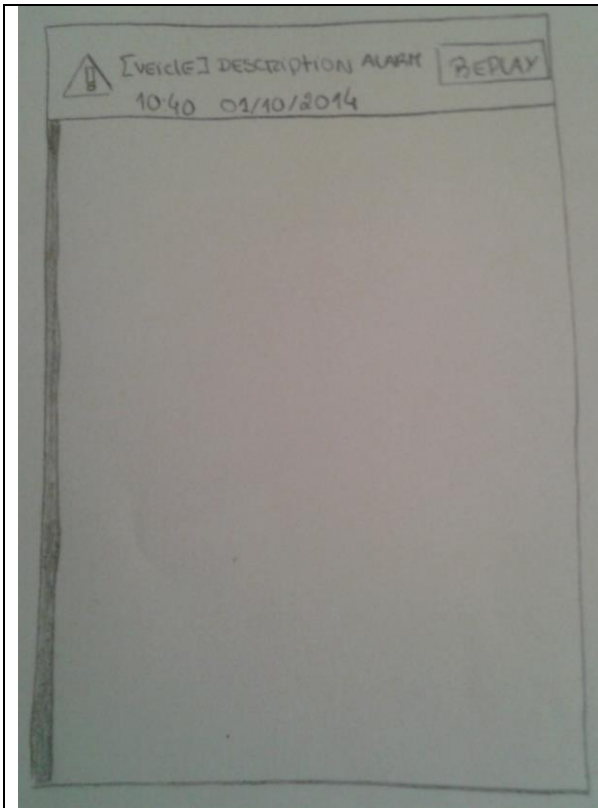


Figura 73 - Ecrã Notificação de Novo Alarme.

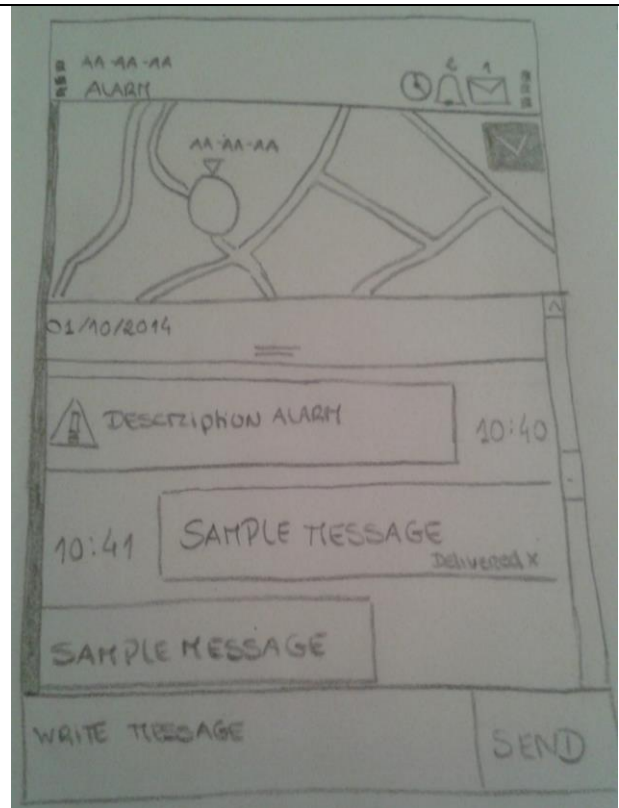


Figura 74 - Chat de Resposta a Alarme.



Figura 75 - Chat de Resposta a Alarme com teclado.

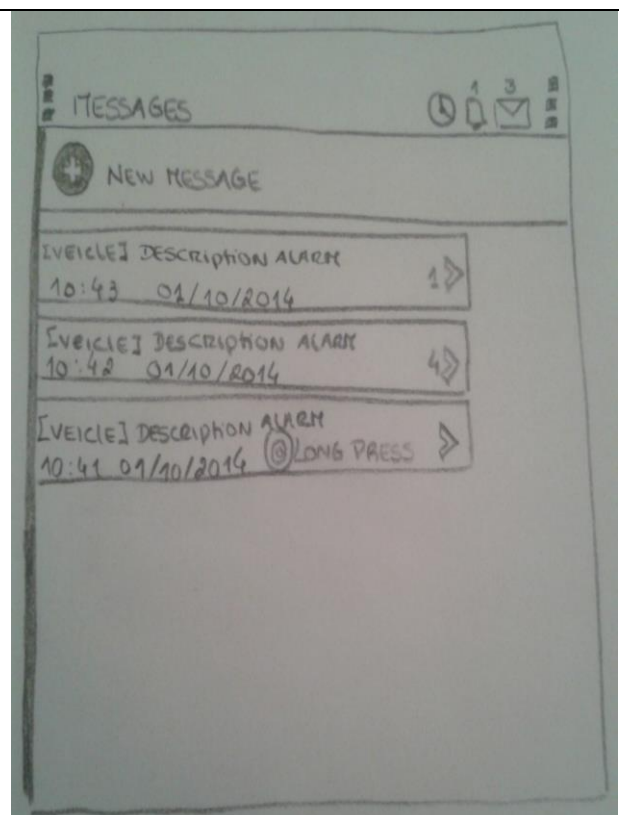


Figura 76 - Ecrã Mensagens.

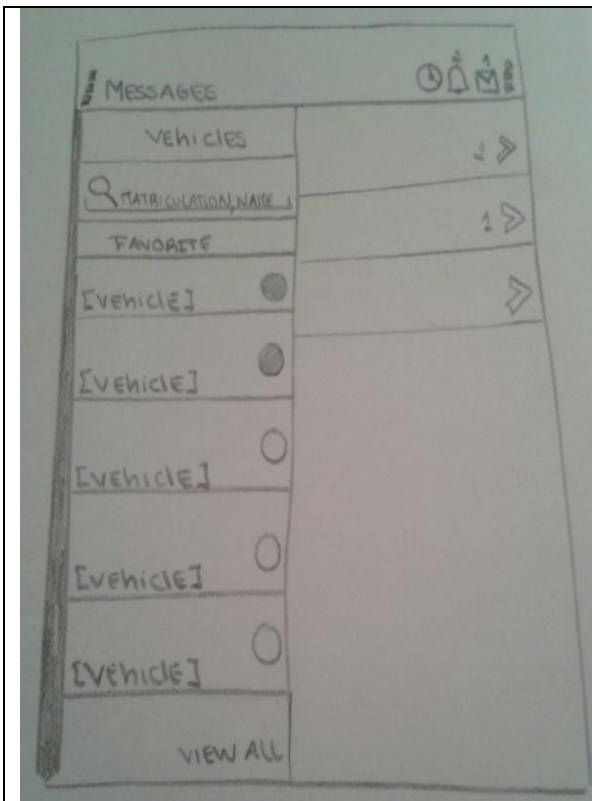


Figura 77 - Ecrã Escolha Veículo para Nova Mensagem.

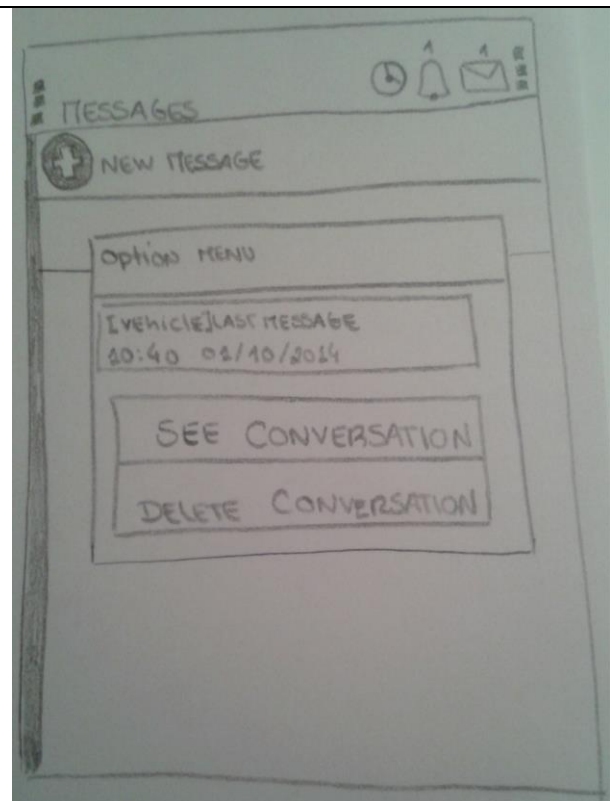


Figura 78 - Opções Sobre Mensagem.

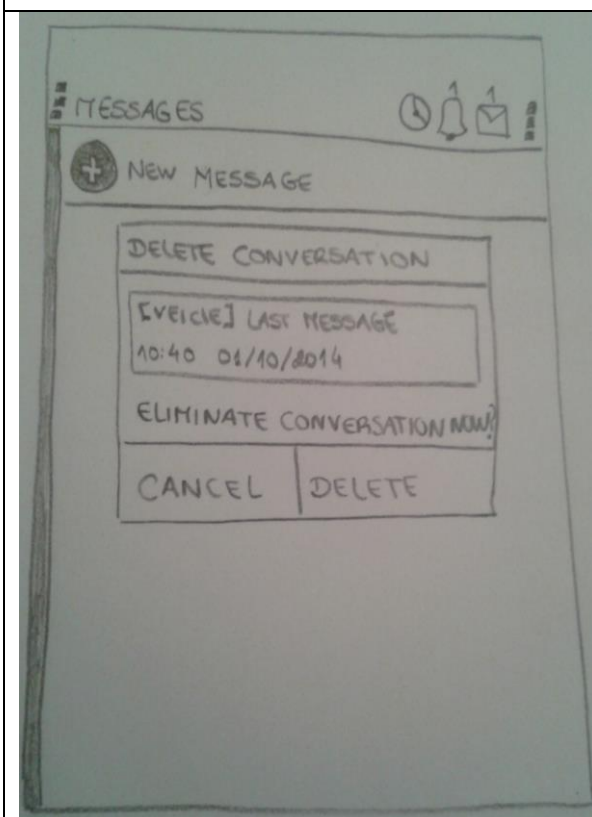


Figura 79 - Ecrã Eliminar Mensagem.

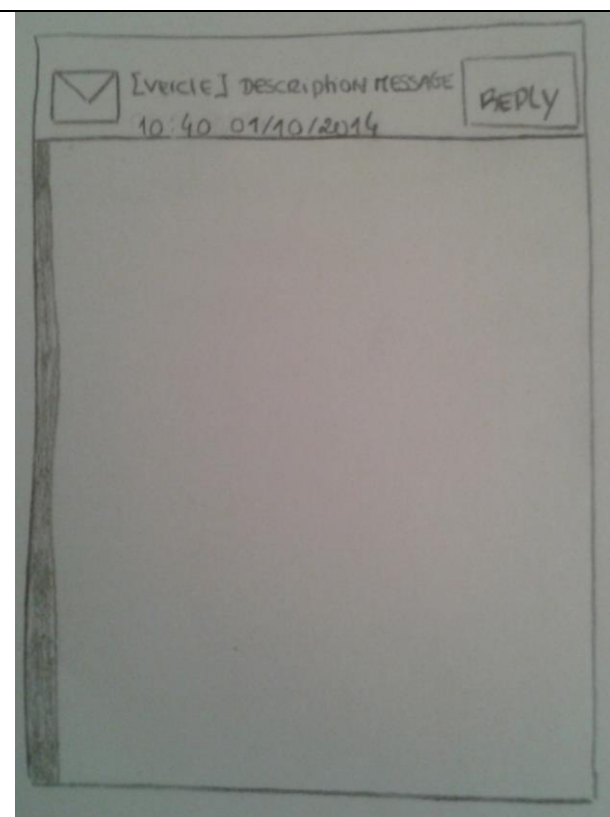


Figura 80 - Notificação de Nova Mensagem.

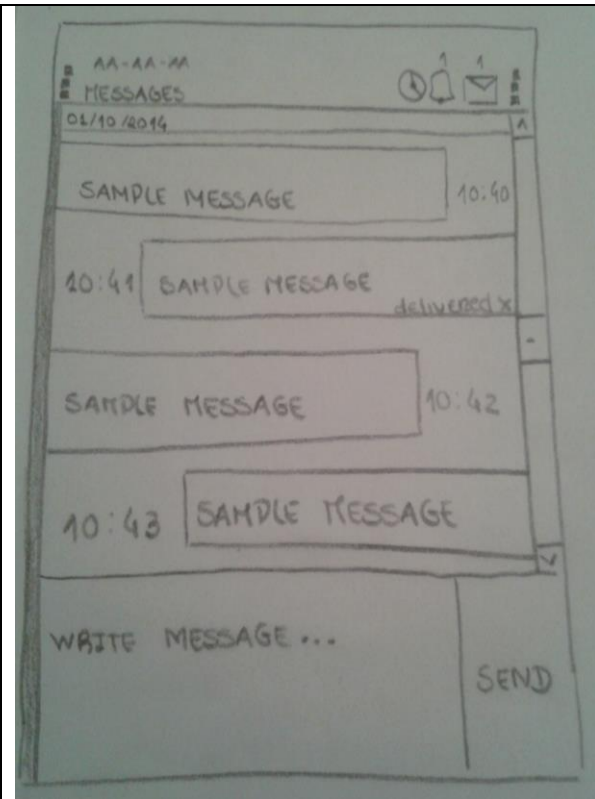


Figura 81 - Chat de Mensagens.

Apêndice V - Protótipos Intermédios

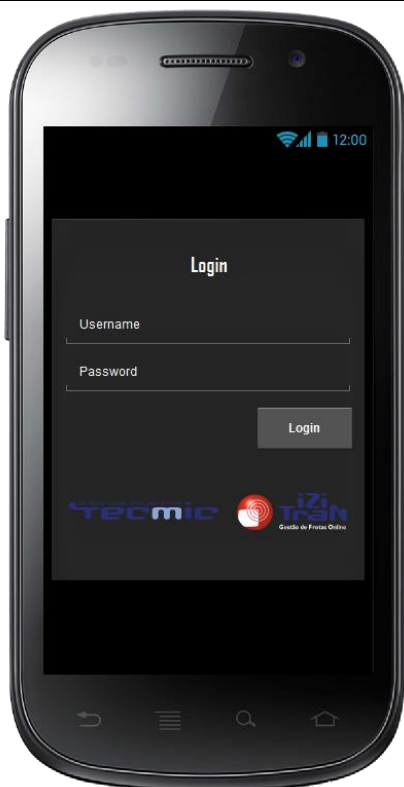


Figura 82 - Ecrã Login.



Figura 83 - Ecrã Painel Lateral.

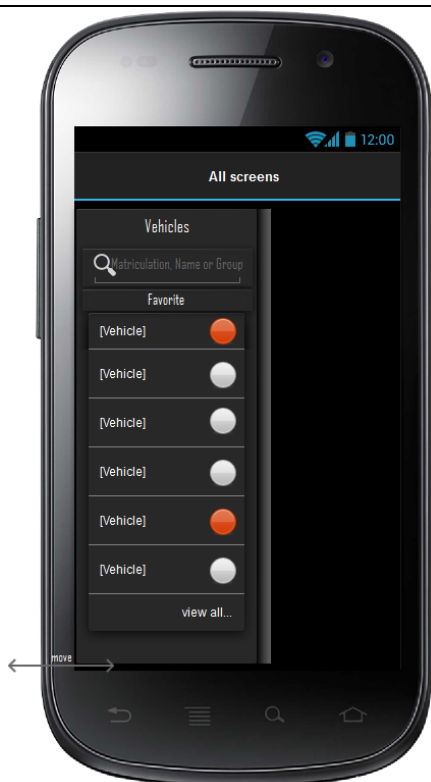


Figura 84 - Ecrã Painel Lateral com dados.

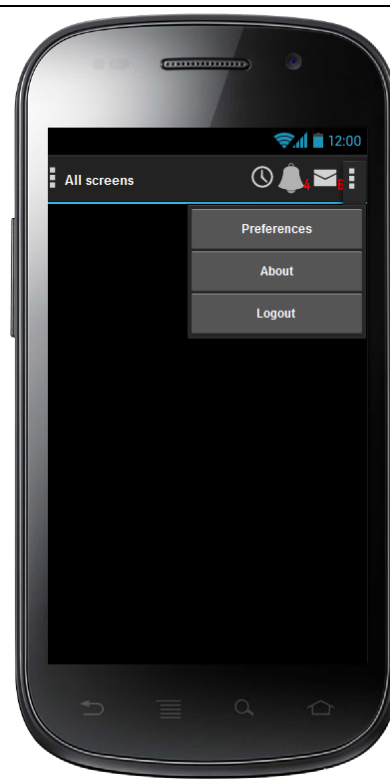


Figura 85 - Opções da Aplicação.

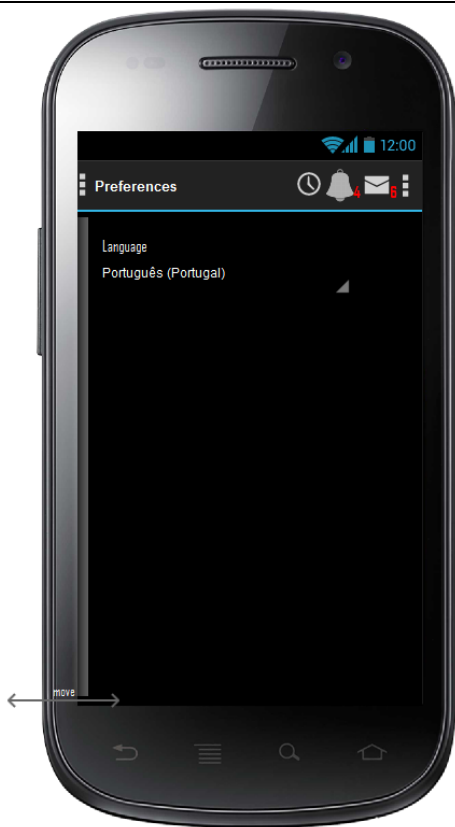


Figura 86 - Ecrã Preferências.

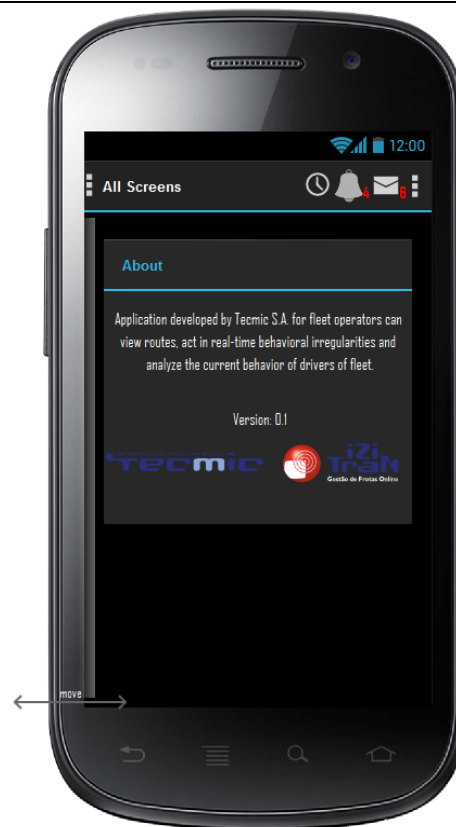


Figura 87 - Ecrã Sobre a Aplicação.

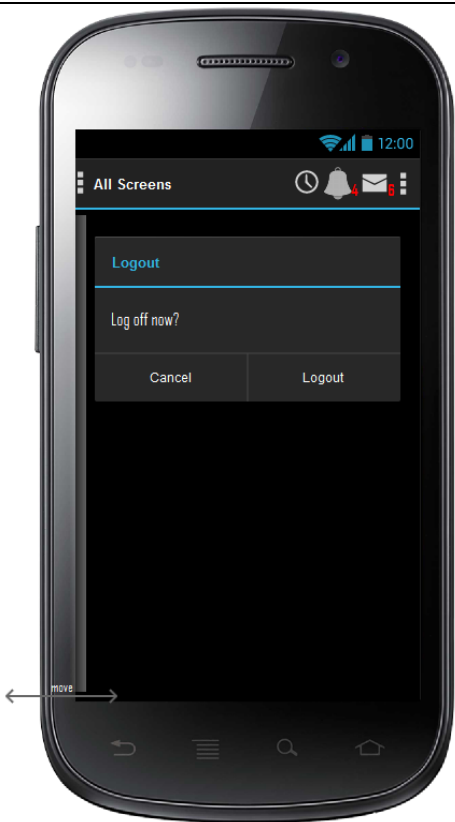


Figura 88 - Ecrã Terminar Sessão.

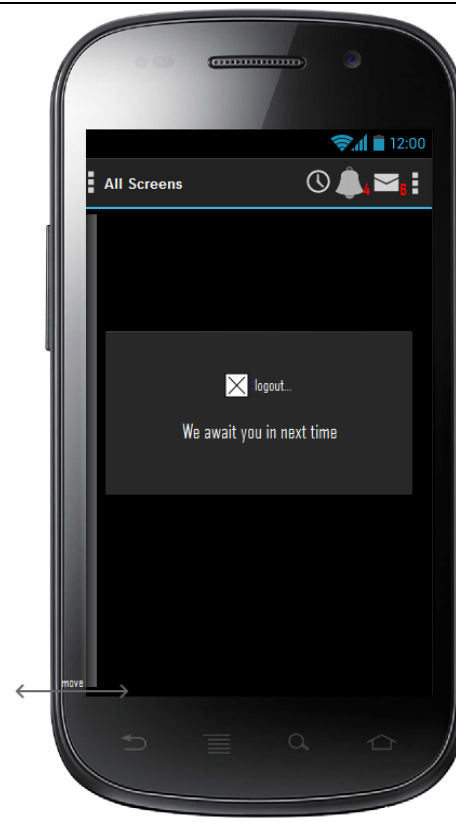


Figura 89 - Mensagem de Terminar Sessão.



Figura 90 - Ecrã Tempo Real Geral.



Figura 91 - Ecrã Veículos.

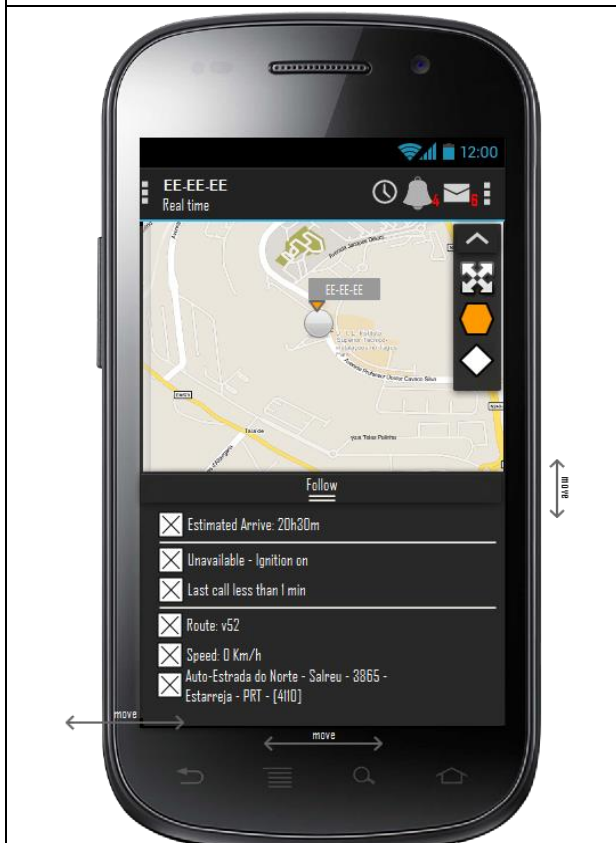


Figura 92 - Ecrã Seguir.



Figura 93 - Ecrã Diário.



Figura 94 - Ecrã Viagens.



Figura 95 - Ecrã Alarmes.

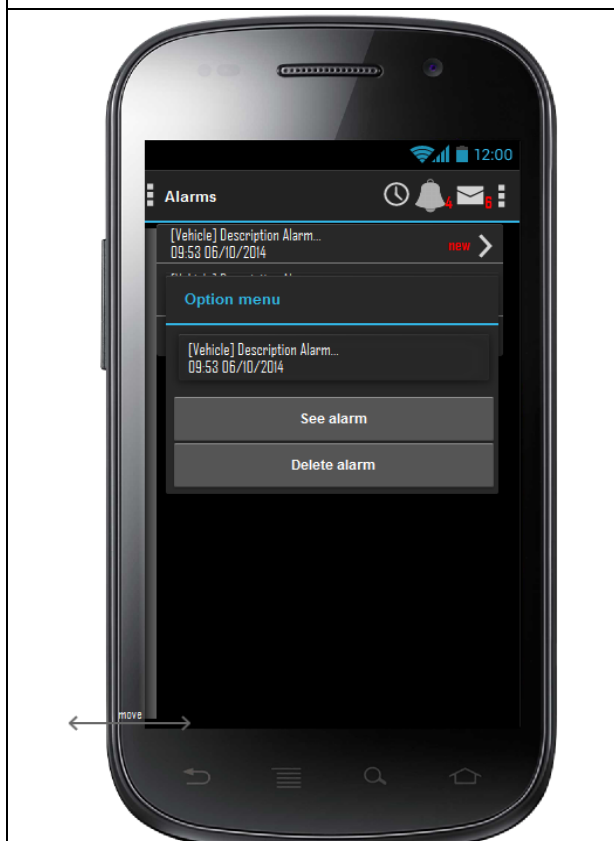


Figura 96 - Opções sobre Alarme.

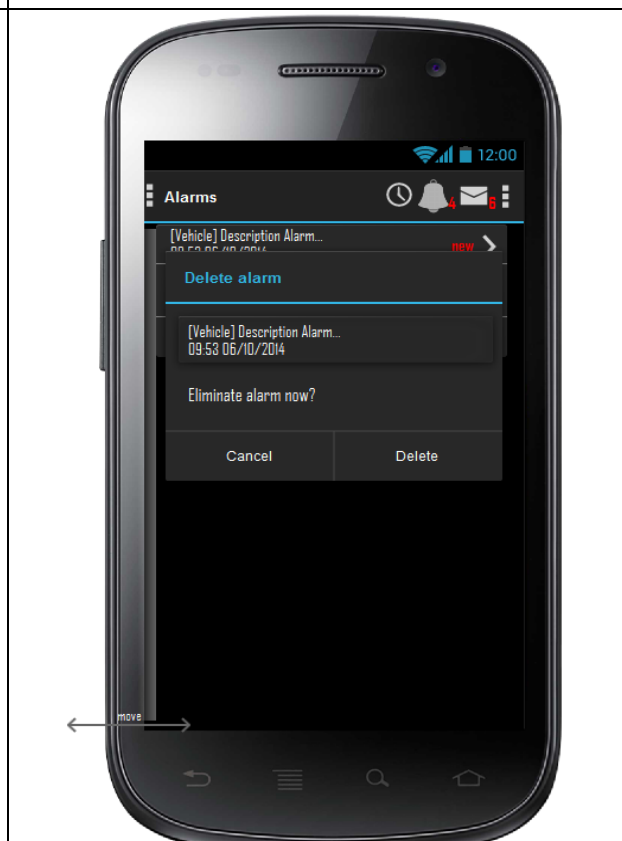


Figura 97 - Ecrã Eliminar Alarme.

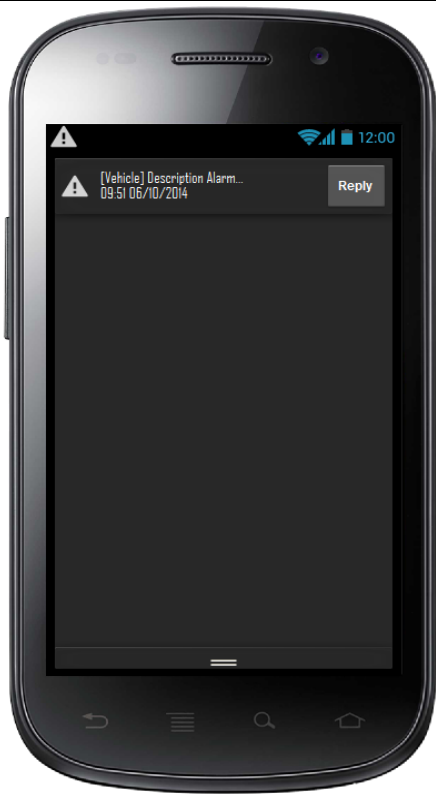


Figura 98 - Notificação de Novo Alarme.

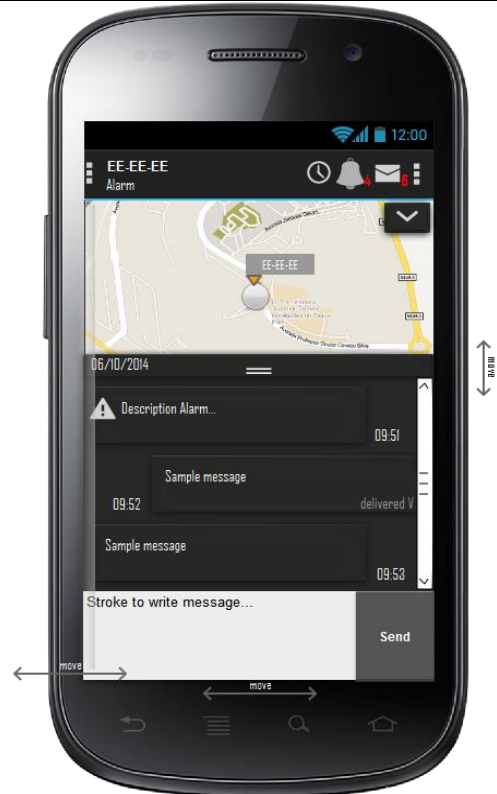


Figura 99 - Chat de Resposta a Alarme.

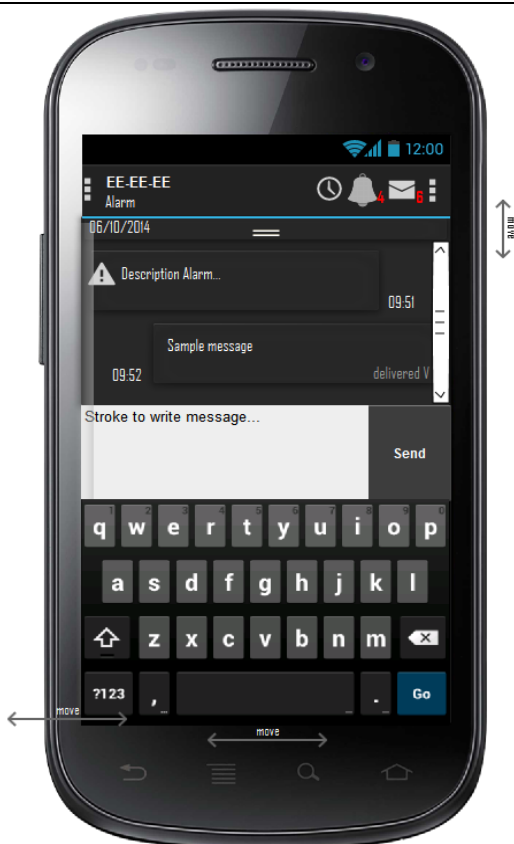


Figura 100 - Chat de Resposta a Alarme com teclado.

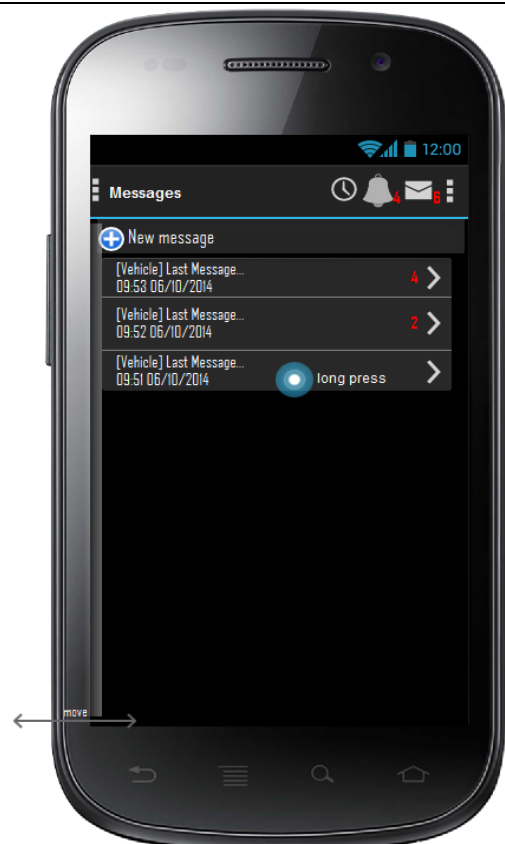


Figura 101 - Ecrã Mensagens.



Figura 102 - Ecrã Escolha Veículo para Nova Mensagem.

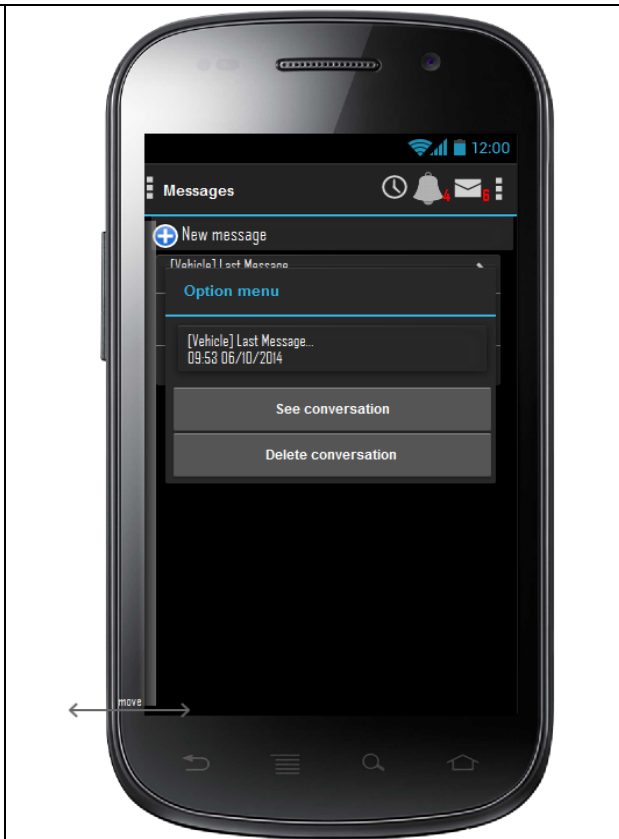


Figura 103 - Opções sobre Mensagem.

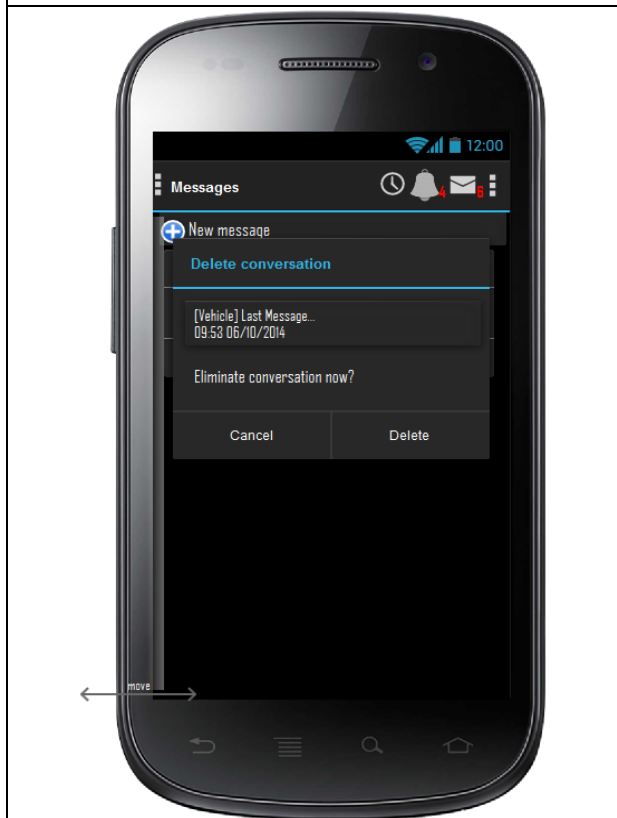


Figura 104 - Ecrã Eliminar Mensagem.



Figura 105 - Notificação de Nova Mensagem.



Figura 106 - Chat de Mensagens.

Apêndice VI - Ecrãs Finais



Figura 107 - Ecrã Login.

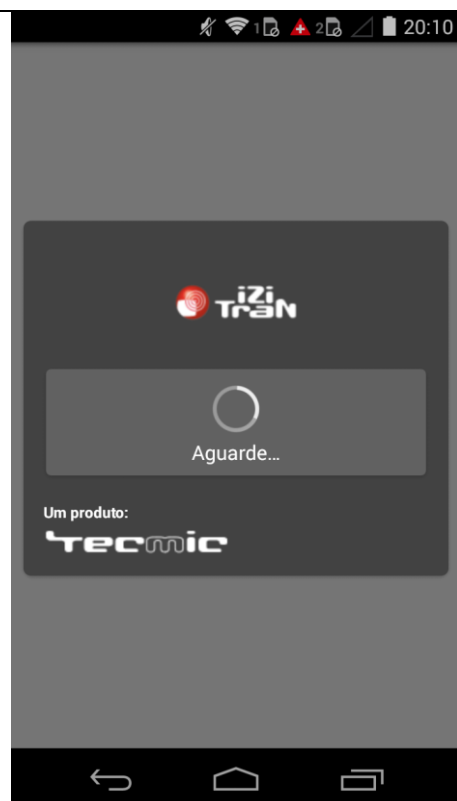


Figura 108 - Ecrã Login com Aguarde.

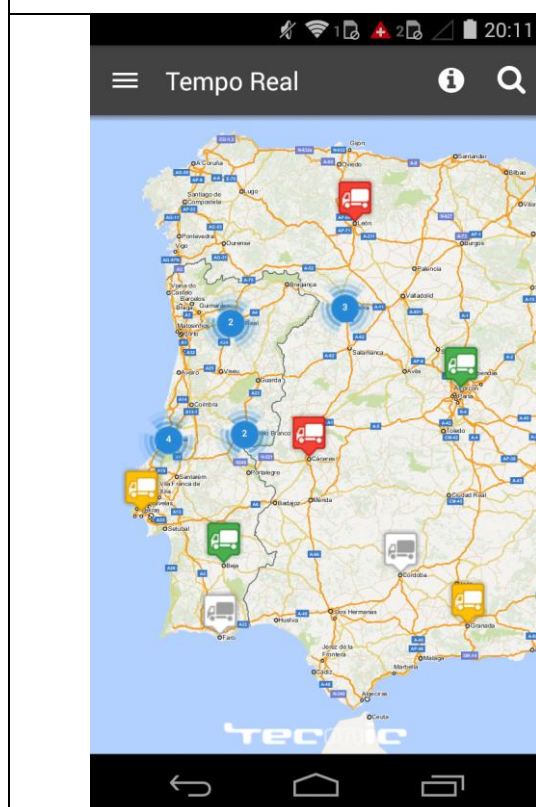


Figura 109 - Ecrã Principal.

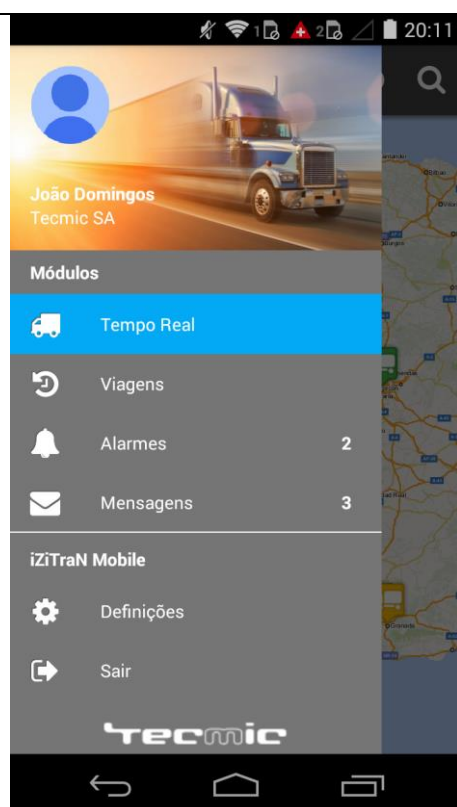


Figura 110 - Ecrã Painel Lateral com dados.



Figura 111 - Ecrã Definições.

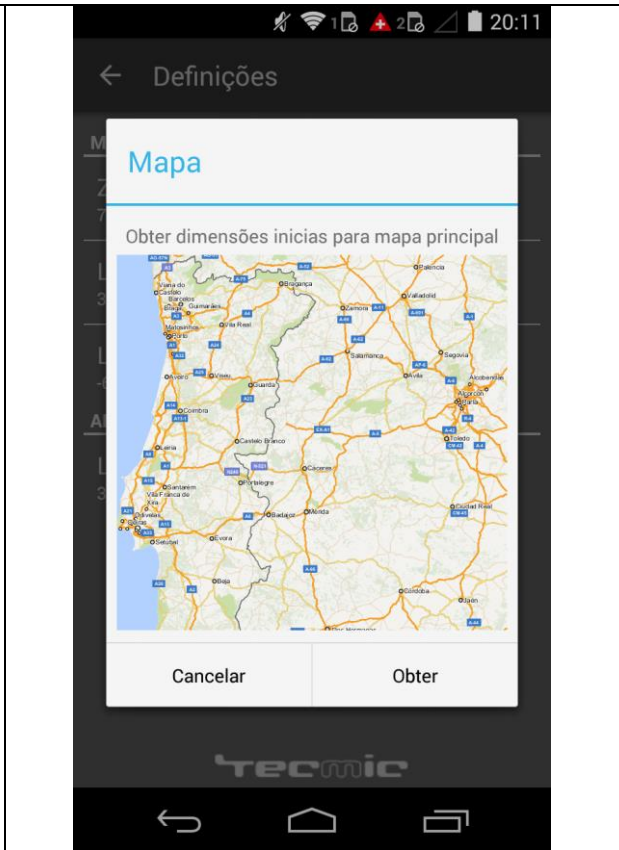


Figura 112 - Ecrã Definições de Mapa.

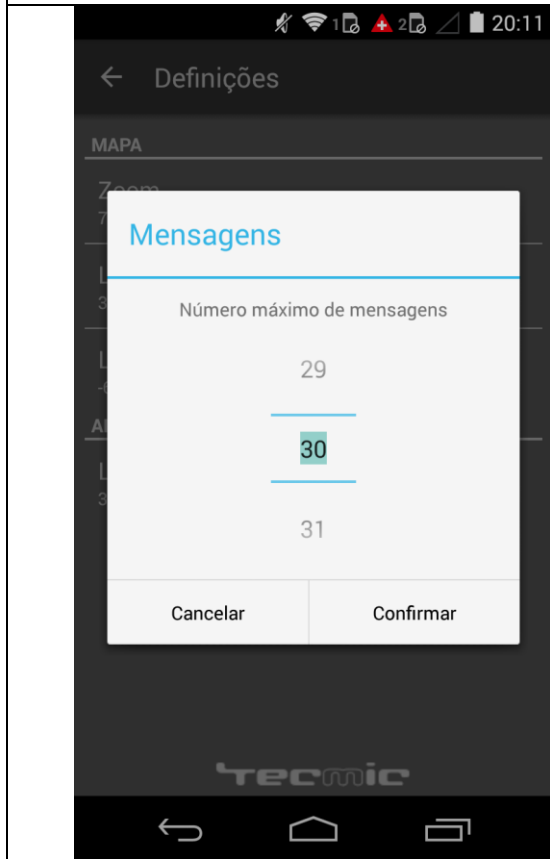


Figura 113 - Ecrã Definições de mensagens.

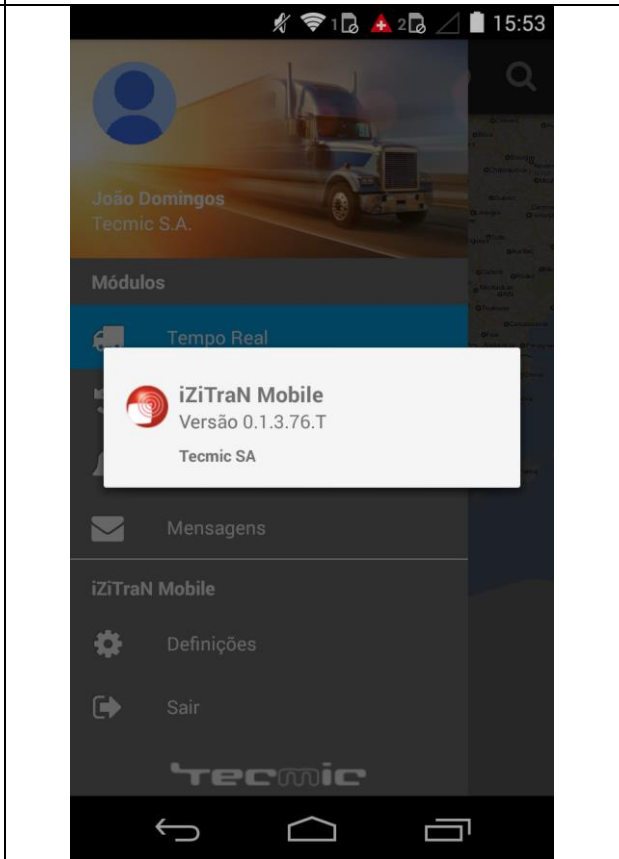


Figura 114 - Ecrã Sobre a Aplicação.

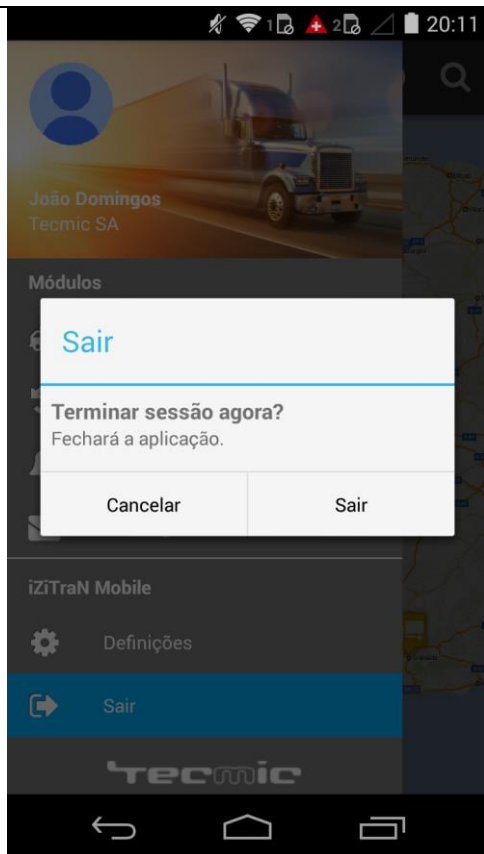


Figura 115 - Ecrã Terminar Sessão.

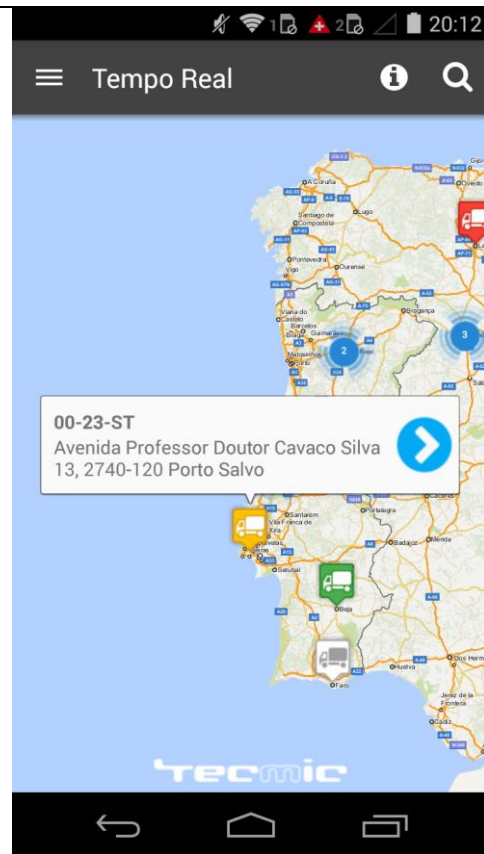


Figura 116 - Ecrã Tempo Real Geral.

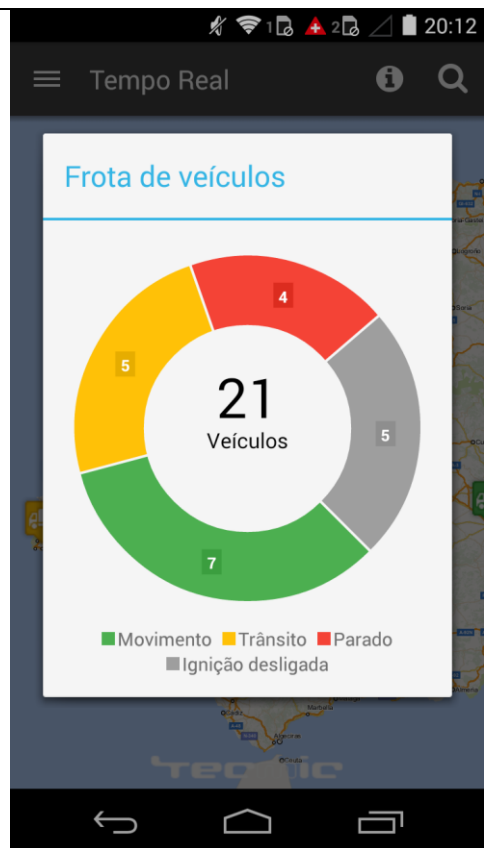


Figura 117 - Ecrã Frota de veículos.



Figura 118 - Ecrã Pesquisar.



Figura 119 - Ecrã Pesquisar com veículos.



Figura 120 - Ecrã Pesquisar por filtro.

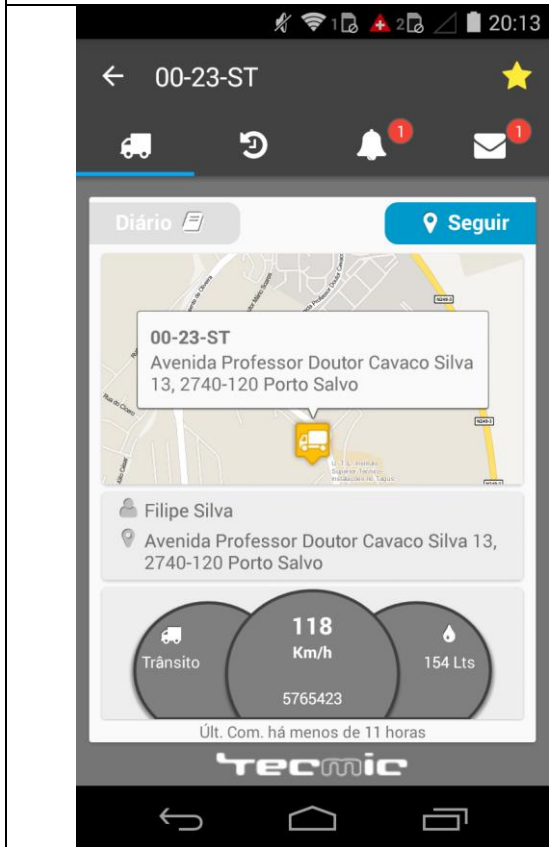


Figura 121 - Ecrã Seguir Veículo.

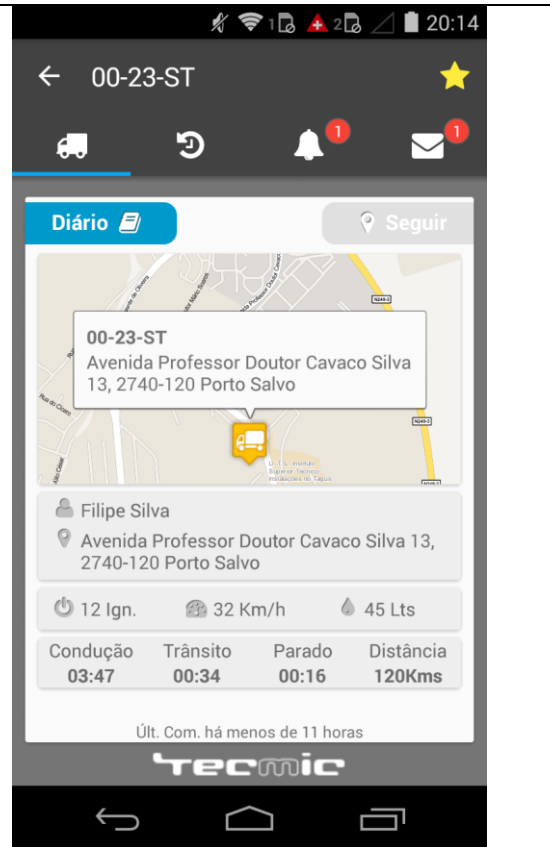


Figura 122 - Ecrã Diário de Veículo.

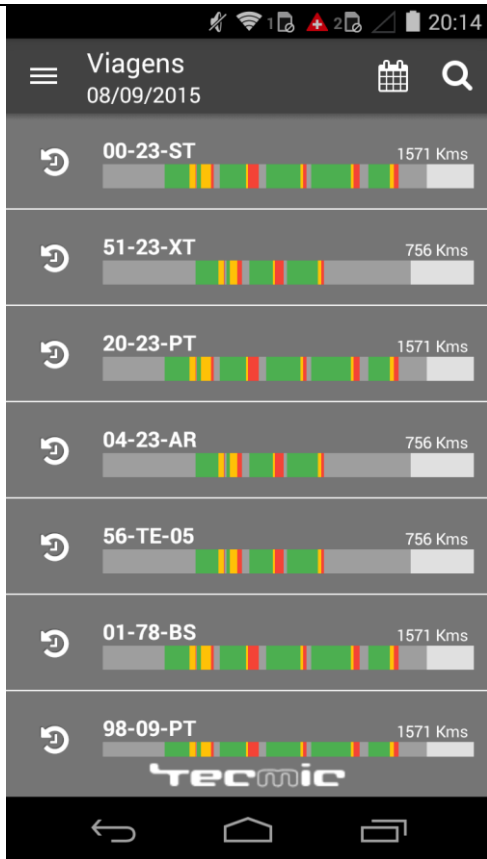


Figura 123 - Ecrã Viagens Geral.

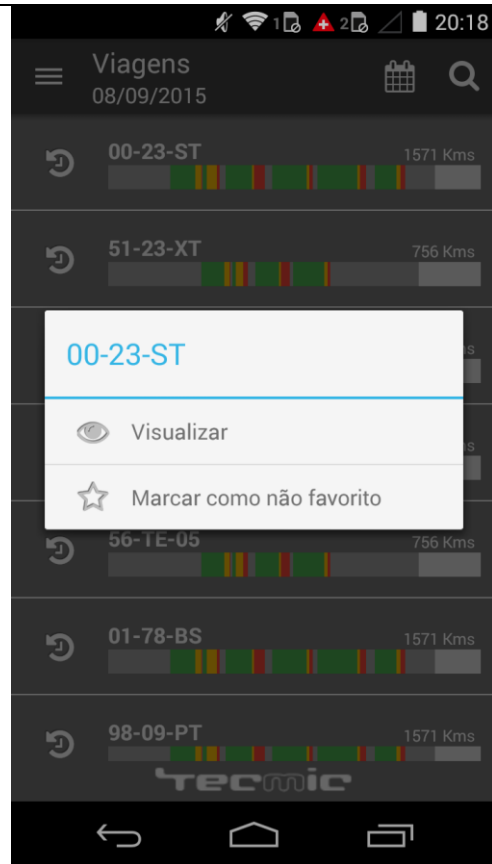


Figura 124 - Ecrã Viagens - Menu de Contexto.

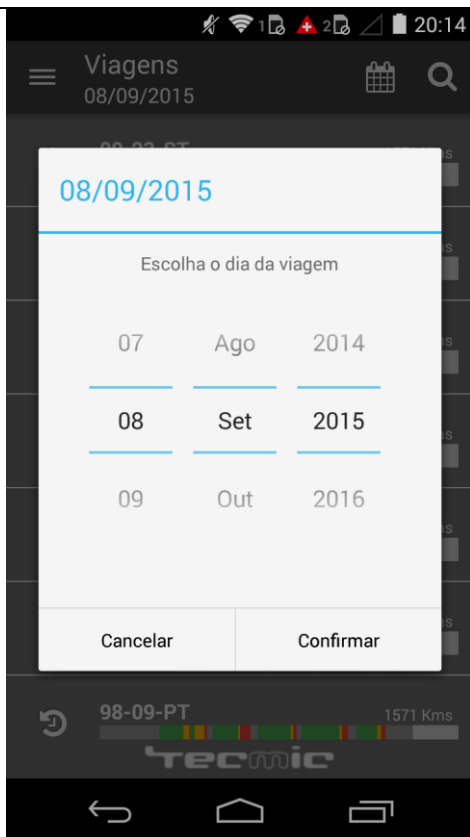


Figura 125 - Ecrã Viagens - Seleção de Dia.



Figura 126 - Ecrã Viagem.



Figura 127 - Ecrã Viagem - Seleção de Dia.

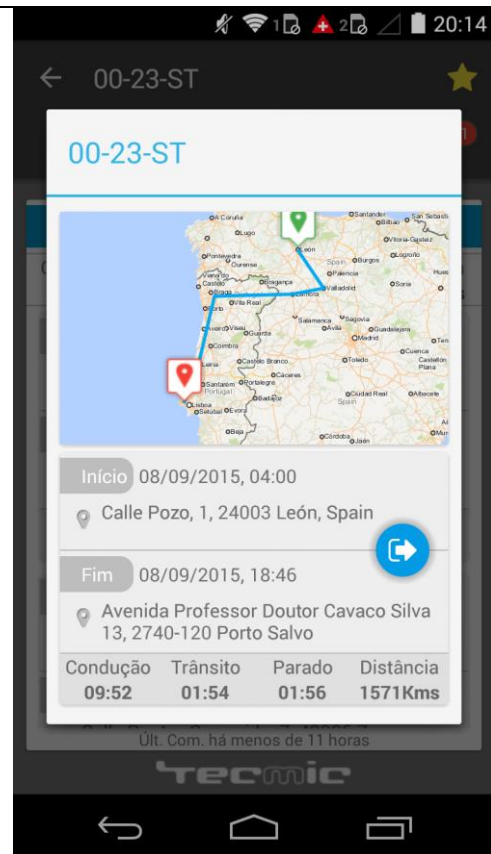


Figura 128 - Ecrã Viagem - Detalhe Global.

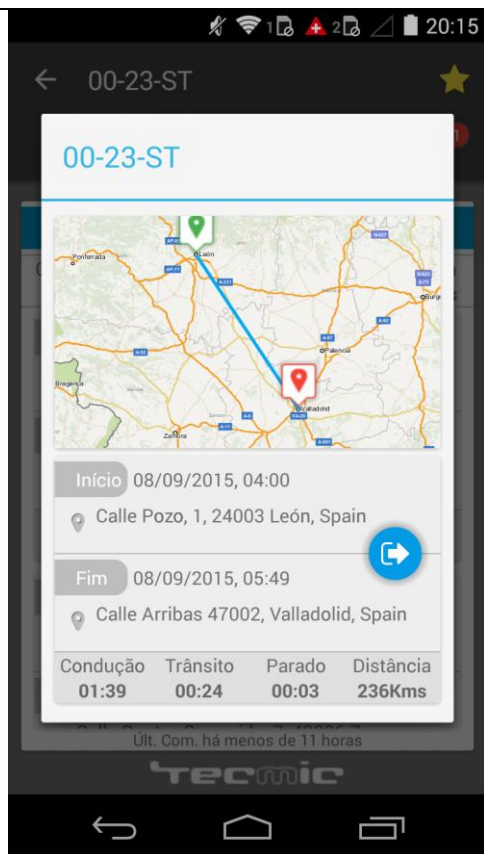


Figura 129 - Ecrã Viagem - Detalhe de Percurso.

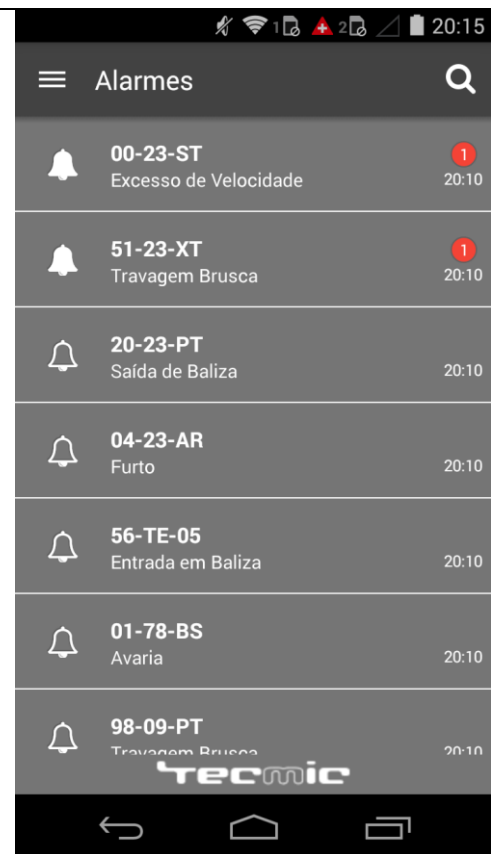


Figura 130 - Ecrã Alarmes.

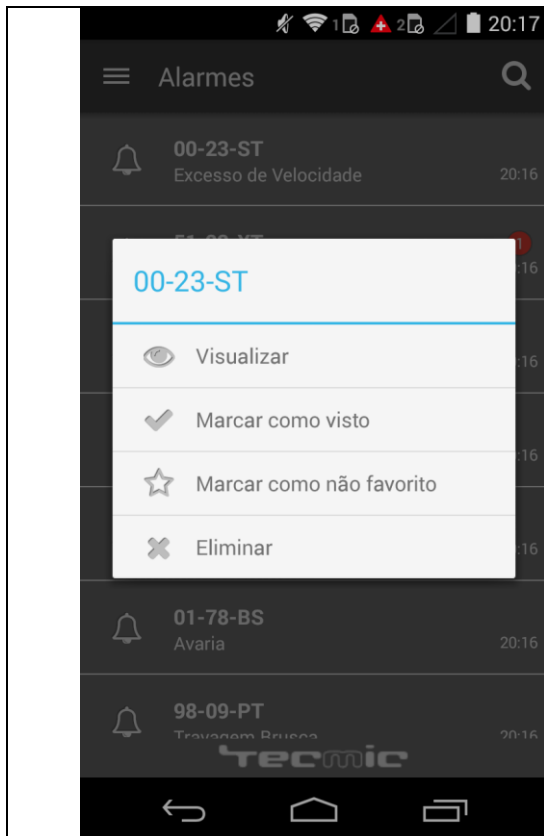


Figura 131 - Ecrã Alarmes - Menu de Contexto.

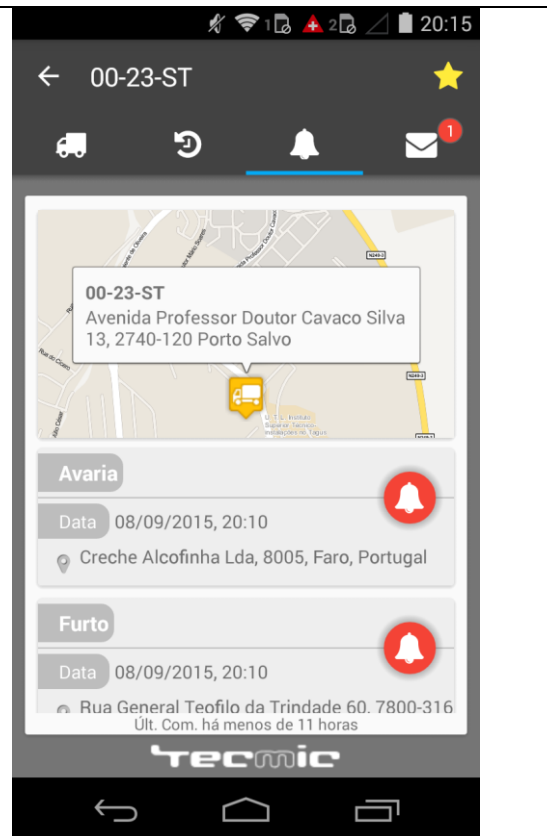


Figura 132 - Ecrã Alarmes do Veículo.

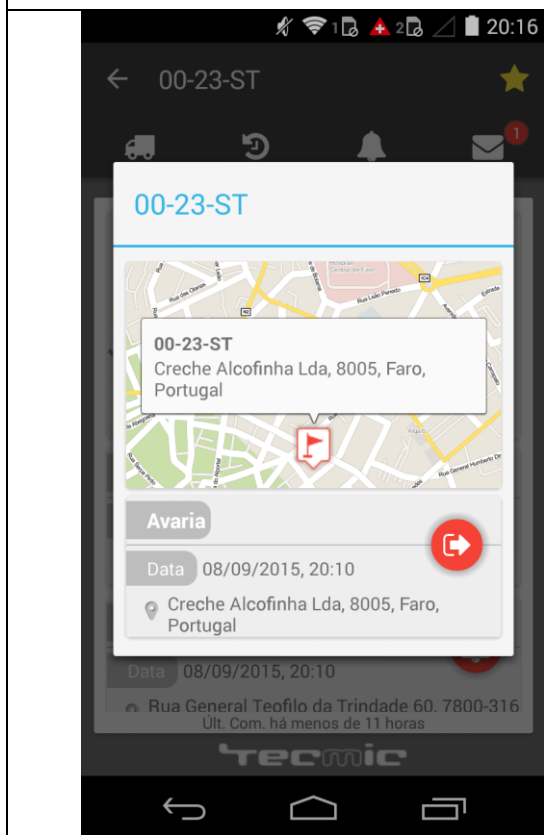


Figura 133 - Ecrã Alarme - Detalhe do Alarme.

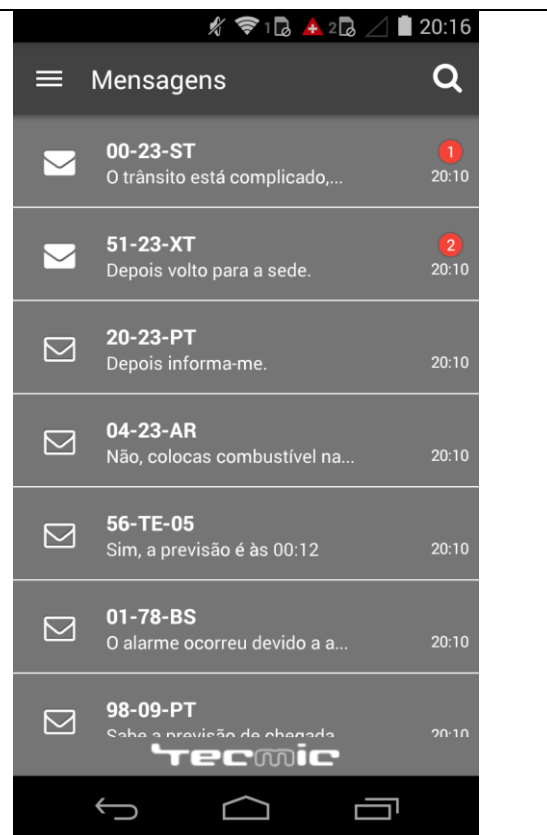


Figura 134 - Ecrã Mensagens.

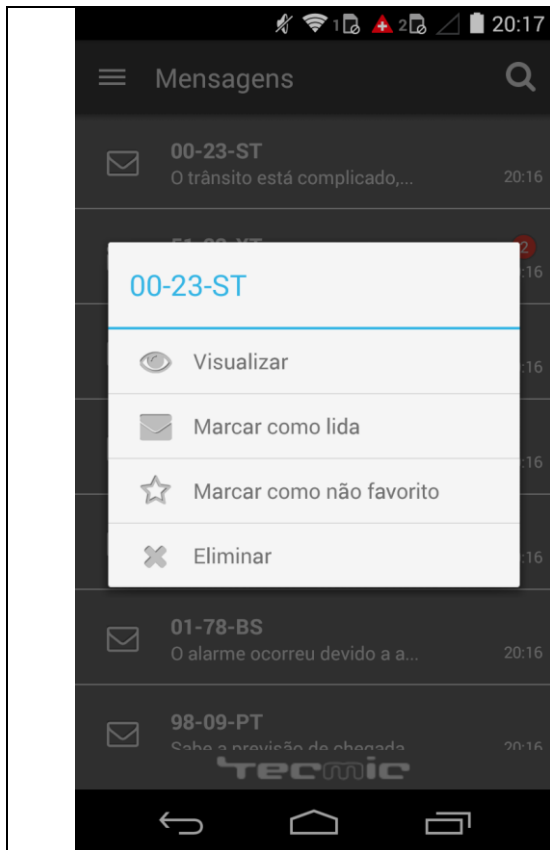


Figura 135 - Ecrã Mensagens - Menu de Contexto.

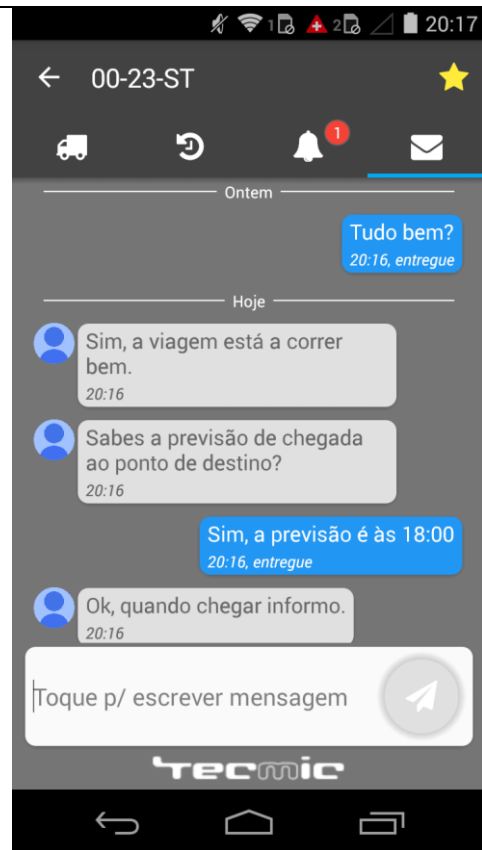


Figura 136 - Ecrã Chat de Mensagens.

Apêndice VII - Testes Unitários

Por forma a apoiar na realização dos testes unitários foram especificados um conjunto de testes contendo o seu identificador, descrição, parâmetros de entrada e resultado esperado. Estes testes e os resultados obtidos aos mesmos são apresentados na Tabela 11.

ID#	Descrição	Parâmetros	Resultado Esperado	Resultado
<u>Veículos</u>				
1	Total de veículos	Lista de Veículos	5	Passou
2	Total de veículos em movimento	Lista de Veículos	2	Passou
3	Total de veículos em trânsito	Lista de Veículos	1	Passou
4	Total de veículos parados	Lista de Veículos	2	Passou
5	Total de veículos desligados	Lista de Veículos	0	Passou
6	Condutor ativo	Lista de Condutores	“João Domingos”	Passou
7	Localização descritiva atual	Veículo	“Aldeamento Santa Clara, Rua da Carvalha, 570, 2400-441 Leiria, Portugal”	Passou
8	Estado atual	Veículo	“Movimento”	Passou
<u>Viagens</u>				
9	Data de partida	Lista de Viagens	“25-09-2015”	Passou
10	Localização descritiva de partida	Lista de Viagens	“Edifício Ciência II, Av. Prof. Dr. Cavaco Silva, nº13, 3A, 2740-120 Porto Salvo, Portugal”	Passou
11	Data de chegada	Lista de Viagens	“25-09-2015”	Passou
12	Localização descritiva de chegada	Lista de Viagens	“Aldeamento Santa Clara, Rua da Carvalha, 570, 2400-441 Leiria, Portugal”	Passou
13	Tempo total em movimento	Lista de Viagens	“01:46:56”	Passou
14	Tempo total em trânsito	Lista de Viagens	“00:16:32”	Passou
15	Tempo total parado	Lista de Viagens	“00:05:24”	Passou
16	Tempo total desligado	Lista de Viagens	“00:00:00”	Passou
17	Tempo total da viagem	Lista de Viagens	“02:08:52”	Passou
18	Total de ignições	Lista de Viagens	10	Passou
19	Total de quilómetros	Lista de Viagens	183	Passou
20	Média de velocidade	Lista de Viagens	87	Passou
21	Condutor com mais percursos	Lista de Viagens	“João Domingos”	Passou
<u>Alarmes</u>				
22	Último alarme	Lista de Alarmes	“Pânico”	Passou
23	Total de alarmes não vistos	Lista de Alarmes	0	Passou
24	Total de alarmes vistos	Lista de Alarmes	2	Passou
25	Total de alarmes com alta prioridade	Lista de Alarmes	1	Passou
<u>Mensagens</u>				
26	Última mensagem	Lista de Mensagens	“Olá”	Passou
27	Total de mensagens não lidas	Lista de Mensagens	1	Passou
28	Total de mensagens lidas	Lista de Mensagens	3	Passou

29	Total de mensagens enviadas	Lista de Mensagens	2	Passou
30	Primeiras mensagens de cada dia	Lista de Mensagens	1, 3	Passou
31	Mensagem recebida	Mensagem Recebida	Verdade	Passou
32	Mensagem enviada	Mensagem Enviada	Verdade	Passou

Tabela 11 - Testes Unitários apresentando o resultado obtido.

Apêndice VIII - Guião de Tarefas

Por forma a apoiar na realização dos testes de usabilidade foi elaborado um guião de tarefas para que os participantes dos mesmos o pudessem seguir. Este é apresentado pela Tabela 12.

T#	Tarefa
1	Inicie a sessão com nome de utilizador: “izitran” e palavra-passe: “mobile”.
2	Quantos veículos se encontram em Movimento?
3	Selecione o veículo com o seguinte nome: “AA-00-BB”.
4	Qual é o estado do veículo selecionado?
5	Quantas mensagens o veículo tem por ler?
6	Pesquise o veículo com o seguinte nome: “BB -11-CC”.
7	Coloque o veículo pesquisado como favorito.
8	Envie a mensagem “Hello World” para o veículo com o nome: “CC-22-DD”.
9	Qual foi o último alarme que o veículo selecionado desencadeou?
10	Navegue até ao ecrã Viagens.
11	Altere o número máximo de mensagens para “20”.
12	Termine a sessão na aplicação.

Tabela 12 - Guião de Tarefas para os Testes de Usabilidade.

Apêndice IX - Testes de Usabilidade

Primeira fase:

Os resultados do registo dos erros ocorridos durante a realização dos testes de usabilidade da primeira fase são apresentados na Figura 137.

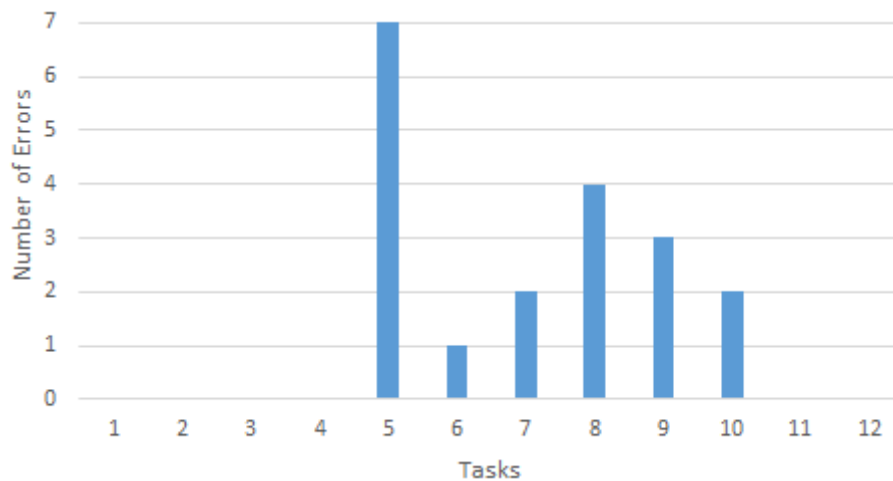


Figura 137 - Testes de Usabilidade da primeira fase apresentando o número de erros em cada tarefa.

Segunda fase:

Os resultados do registo dos erros ocorridos durante a realização dos testes de usabilidade da segunda fase podem ser visualizados através da Figura 138.

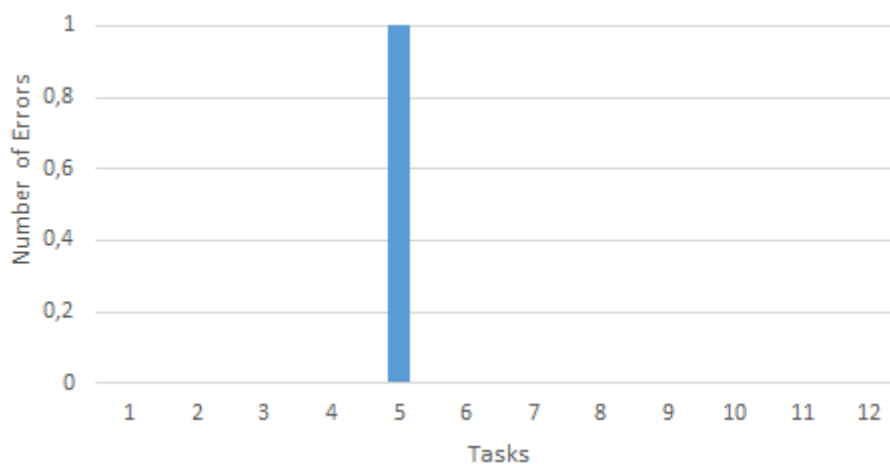


Figura 138 - Testes de Usabilidade da segunda fase apresentando o número de erros em cada tarefa.

Apêndice X - Artigo Científico

Conference on ENTERprise Information Systems / International Conference on Project
MANagement / Conference on Health and Social Care Information Systems and Technologies,
CENTERIS / ProjMAN / HCist 2015 October 7-9, 2015

iZiTraN Mobile Fleet Management

João Domingos^{a,b}, David Mendes^{a,b}, Pedro Marques^a, Vitor Basto-Fernandes^b

^a*Tecmic - Microelectronic Technologies, SA, 2740-120 Porto Salvo, Portugal*

^b*Computer Science and Communications Research Center, School of Technology and Management, Polytechnic Institute of Leiria, 2411-901 Leiria, Portugal*

Abstract

In this paper we present iZiTraN Mobile, a backend and mobile application allowing for precise location of vehicles in interactive maps, fleet operators communication with truck drivers by text messaging and voice calls, logistics, operational and emergency alert and reporting, following real time anytime/anywhere development principles.

The emerging technologies adopted to build iZiTraN Mobile, the software design assumptions together with agile software engineering development processes, ensured systems quality attributes enforcement such as modularity/layering, maintainability and upgrading, among other relevant system attributes¹, as well as project-based quality attributes such as ability for parallel development, cost effective development and customer satisfaction.

© 2015 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge.

Keywords: Mobile Application; Events Architecture; Fleet Management; Real Time; Messages; Alarms.

1. Introduction

iZiTraN Mobile application emerges as an answer of Tecmic - Microelectronic Technologies, SA to its customers' demands, following current economic, business, technology and social trends on anytime, anywhere any device information systems^{2,3}, for the specific domain of fleet management, which represents one of the Tecmic core businesses.

This application is the basis for a new Tecmic product, targeted for mobile devices users, allowing road delivery operators to monitor, in real time, all vehicles in the fleet. The application allows for precise location of vehicles on a map and also for the monitoring of several indicators associated with them, in a continuous real-time basis, supported by a communication channel between the application and a Web Service that retrieves data from Tecmic backend console, presented next.

iZiTraN Mobile enables fleet operators communication between operation officers and drivers, by exchanging text messages, by voice communication, or alerts triggering and processing on various fleet and vehicles incidents that may arise.

The paper is structured as follows. Section 2 presents methodology principles adopt for the project development, as well as project management decisions. Section 3 describes the software development process, technological options and architecture design. Section 4 introduces the application features and their business value. Section 5 presents the testing phase and Section 6 concludes the paper and describes the future work.

2. Methodology and project management

This section presents the software development methodology used in iZiTraN Mobile project, its strengths and methodology adaptations made, according to the project specific needs. Additionally, project management practices and options are also presented here.

2.1. Methodology

Due to the iZiTraN Mobile project size, development team size, software fast changing requirements, rapid and incremental development with strong client commitment, the chosen methodology should be able to support an agile construction and review of the software requirements and its validation. eXtreme Programming methodology (XP)⁴ was therefore adopted in order to allow for flexible modifications along the software development lifecycle. XP software development lifecycle is shown in Fig. 1.

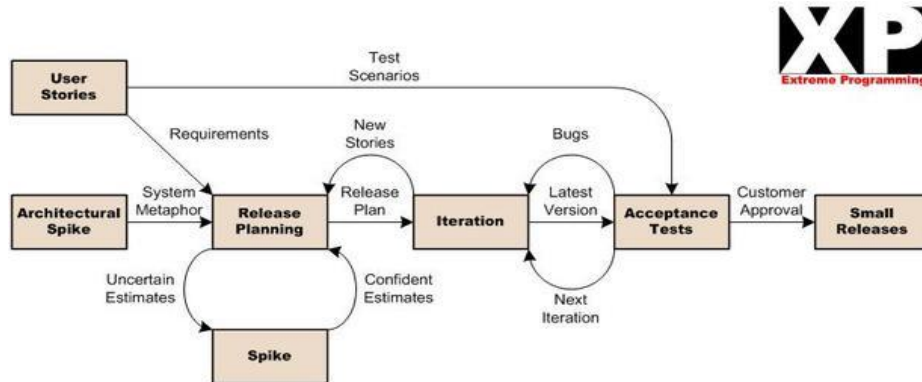


Fig. 1. XP Software development lifecycle

XP software development lifecycle presented in Fig. 1 is the general flow of the methodology. However, the applied methodology in iZiTraN Mobile had some changes according to the needs arose during the project. Requirements analysis and specification suffered considerable adaptations, since the Wiegiers matrix model, proposed by Karl Wiegiers⁵, was used to aid in priorities selection. This model revealed several benefits in other Tecmic projects, including low ratio of flaws with respect to initial project requirements, development and product requirement changes reduction, cost effectiveness, faster development, less communication problems, project management predictability, higher satisfaction for customers and development teams.

XP iterations followed an incremental spiral model at each iteration, by repeating the phases of analysis, design, implementation and testing. Weekly meetings addressing developed and planned features took place involving the development team and project manager. Requirements validation was performed against Tecmic internal knowledge and experience in the logistics and fleet management domain acquired over the last 25 years.

According to Beck, XP requirements and product validation must be done by the means of tests, written before the implementation process and only later executed. Also according to Beck⁶, developer's teams are responsible for writing unit tests and customers for writing acceptance tests.

In our project, the development team wrote both unit and acceptance tests, taking into account Tecmic expertise and experience on this specific problem domain. Acceptance tests were performed depending on the features and target technology.

2.2. Project management

iZiTraN Mobile project management addressed the following project dimensions, scope, time and resources management, risk management and communication management.

Risk management was dealt with special care, in order to maximize the success likelihood of the project, avoid harmful and unpredictable effects to the project objectives. Some scenarios and corresponding risk assessment was performed as well as preventive measures. The most relevant scenarios taken into account were requirements analysis and specification delays, and development team load distribution uncertainty among several projects. These risks were mitigated by an accurate planning and priority management with Wiegiers techniques⁵, with additional support from tools such as Microsoft Project® for project management specification, communication and follow up.

Special attention was put in the project communication dimension, ensuring proper identification of information needs, corresponding supporting communications channels development and communication tools deployment.

3. Software development

This section provides a detailed description of the technologies used, including operating systems, communication and distributed systems models and system architecture choices. Explanation on quality attributes that lead to these choices are also presented.

3.1. Technologies

Among the several choices of operating systems smartphones, we decided for Android native development. The main reason for this choice is that Google's operating system is currently the major player with the largest market share in this area, according to research conducted in 2014 by the International Data Corporation (IDC)⁷. In a more technology focused analysis, Android is also leading in the preferences of developers, according to the study conducted in 2014 by the VisionMobile⁸ over 10,000 developers who were familiar in developing for several existing platforms.

For software configurations and versions management Apache Maven was adopted. Apache Maven aims at "making the build process easy"⁹. The advantages of using Maven is related to its ability to manage project dependencies along the entire development life cycle and final product build. Tasks such clean, compile, test, packaging and distribution are well supported by Maven.

iZiTraN Mobile requires real-time one-way and two-ways (bidirectional) communication between the application and the Web Service. In order to speed up development for the project communications features, the following frameworks were used: Volley, SignalR e ASP.NET Web API.

Volley¹⁰ allows for automatic implementation of application network requests using multiple simultaneous network connections with possibility for priorities differentiation, and supports efficient caching features. Volley was used in the application side and for the Web Service ASP.NET Web API framework was chosen. This Web API makes easy the process of building HTTP services that reach a wide range of client applications, including browsers and mobile devices, enabling applications to be platform independent. The Web API also allows messages format negotiation to be adapted according to client application needs (e.g. JSON, XML).

To meet all real time communication needs, we use the SignalR¹¹ framework both in the application and Web Service sides. This choice was made due to the possibility of maintaining a high number of calls to the server, control the way server handles requests, ensure that the client is receiving data on time and minimize the effects of common weaknesses present in real-time communications. This is an open-source framework, with easy implementation of real-time communications, and a simple approach to handle asynchronous updates/notifications. SignalR also supports automatic choice of the most suitable and effective communication protocols to support the communication between the server and a client application (WebSockets^{12,13}, Long Polling¹³, etc.).

Guava¹⁴ framework consists in a set of Google libraries, including Guava EventBus library, which was used in iZiTraN for managing events (also asynchronous events). This library differs from others because it allows objects to use the publication or subscription events, without explicit knowledge of each other. iZiTraN broadcasts specific events every time it needs to send and receive data. The events reach the corresponding subscriber applications and are processed accordingly by them.

iZiTraN events management approach allows for easy addition and removal of application modules, achieving independent and dynamic modules composition in the application, with high levels of communication, technical and business agility^{15,16}.

For iZiTraN Mobile maps features, we searched for an open-source maps technology, compatible with the Web Map Service (WMS)¹⁷, which is currently used for Tecmic cartography, but also extensible for the eventual usage of other cartography sources. Several frameworks have been studied, such as Google Maps API, gvSIG Mini Maps, AndroziC, OpenLayers, among others.

Due to the business oriented purposes of iZiTraN, API Google Maps¹⁸ revealed no to be a feasible solution. Open-source gvSIG Mini Maps¹⁹ raised interest due to its wide range of functionalities and WMS usage, however this framework has poor documentation and is currently discontinued. AndroziC²⁰ provides a considerable number of important features but does not provide the WMS service, it uses the Tile Map Service (TMS)²¹ instead. The OpenLayers^{22,23} framework was also tested and worked well, fulfilling the desired requirements of open-source and WMS service. However, it is not natively supported by Android and requires an additional framework such as PhoneGap²⁴ to be implemented properly on mobile devices. It also uses JavaScript with WebView, which has known performance problems, it is a more complex solution and more likely prone to failures.

In contrast to gvSIG Mini Maps, OSMDroid²⁵ framework has an active supporting community and is well documented. Unlike Google Maps API it is open-source, but does not provide native support to the WMS

service. To overcome this drawback, we created a WMS mechanism able to use Tecmic maps, simultaneously able to explore all the features this framework offers. The built mechanism took the form of an adapter to provide this service to the framework, making it an universal adapter that enables the framework to consume various mapping services and thus can be used in a simple manner by various sources. OSMDroid was therefore selected as the best option, not only because of the above described requirements, but also because it's an Android native framework.

In order to separate presentation and business logic, but also due to easy units tests development, we used the Model-View-Presenter (MVP) pattern²⁶. MVP is an adaptation of the well-known Model-View-Controller (MVC)²⁶ pattern, mainly used for building graphical interfaces. The adaptation removed the View reference to the Presenter and Model, allowing for changes to the View not influencing the presentation and business logic. In this pattern the presentation logic is completely moved to the Presenter, easing considerably the testing phase, as calls to the GUI are not needed in the framework²⁷.

3.2. Architecture

The application was designed to be modular, following the layering approach of MVP pattern which separates and makes independent the view, presentation, business and communication layers. To ensure modularity a single main project option was made, containing the view and presentation layer, complemented with additional artifacts such as libraries, implementing the business and communication layers. View and presentation layers includes all classes needed for the Graphical User Interface (GUI) according to the MVP pattern, the business layer contains the business rules to interpret the data coming from the presentation towards the communication layer. The communication layer contains the classes taking part in the communication between the application and the Web Service, involving both the Volley and SignalR frameworks.

The layering architecture presented above provides support for an events based framework implemented to ease the integration process between architectural layers. Guava framework is used for events management. The overall architecture components and their interactions is shown in Fig. 2.

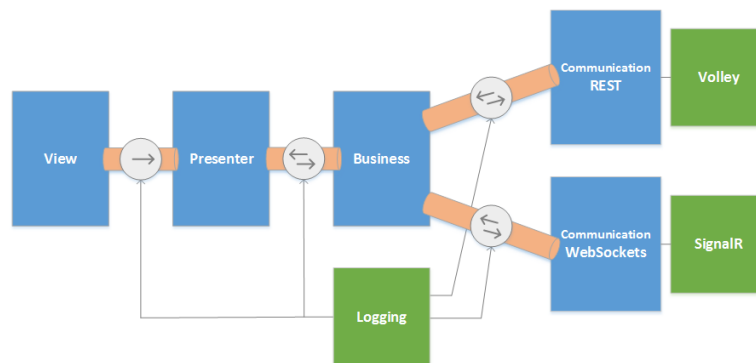


Fig. 2. Layering architecture model

Fig. 2 presents the event-based architecture that allows communications via REST (Volley) or WebSockets (SignalR) to feed the application. Logging artifact was created to support auditing, it records everything that goes through the events channels. The main advantage of this artifact is to present the user application iterations to the control center to be processed and analyzed. The analysis process allows for identification of abnormal user interactions or application behavior. As a result, we gain the knowledge of which corrections/changes should be applied and which interfaces should be improved. If the presented architecture didn't have this structure the effort needed to access the user's interactions would be greater. The `onClick()` and `onSelected()` events are examples of these interactions. If it weren't for the presented structure, we would need to write several lines to register each interaction, and then call the communication API to send it to the control center. Other benefits of the presented architecture are, the ability to add and remove modules without influencing or affecting what already exists (Fig. 3); the ability to easily integrate our modules in Android and Java applications.

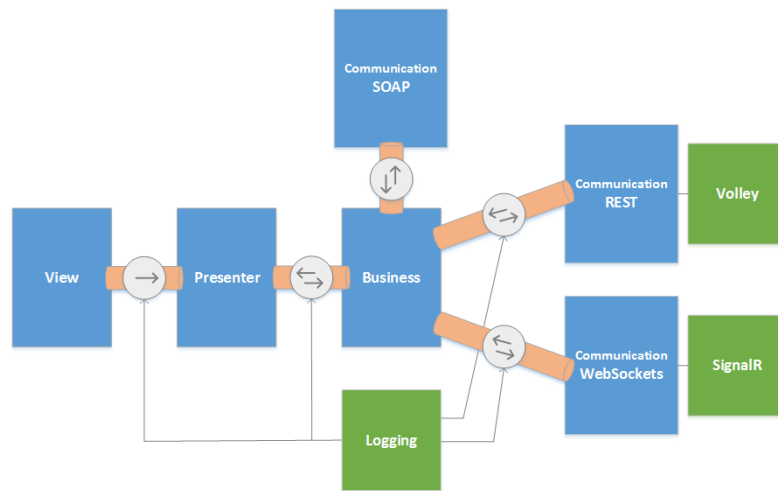


Fig. 3. Addition of a module Communication SOAP

Fig.3 presents an example of the addition of a module, named Communication SOAP, to the Business layer without influencing other pre-existent modules.

4. Application

This section presents the iZiTraN Mobile application and its role in the context of the overall iZiTraN project.

The GUI is designed to provide the best possible user experience for navigation, usability and integration with Android native GUI components. It was designed and implemented specifically for mobile devices following the best practices suggested by Android in the recent Material design²⁸. The GUI was developed in three stages, first sketchings of proposed initial screen were analyzed and selected by the development team, followed by mockups (intermediate prototypes) development and validation, and finally the GUI implementation of selected mockups. Fig. 4 and Fig. 5 present some prototype images and their final implementations.



Fig. 4. (a) Initial prototype navigation; (b) Final screen navigation; (c) Initial prototype real-time indicators; (d) Final screen real-time indicators.

Fig. 4 (a) and (b) shows the screen that allows navigation throughout the real-time alarms, messages, and access to screens preferences, application version and log off. Although navigation was initially proposed to the top of the screen, after GUI revision it was moved to the left side of the screen. The intention of the change was to optimize screen available space and introduce the possibility of adding more modules to the application.

Fig. 4 (c) and (d) shows the application main screen, displaying all fleet vehicles on the map. Although the initial sketches include vehicles control indicators to be shown also in the map, in the final version this information is accessed by a button present on the top bar of the screen. This option allows the map to fill the entire screen, easing the operator visualization of all fleet vehicles in the map and providing fast access to their real-time indicators, messages, alerts, etc.

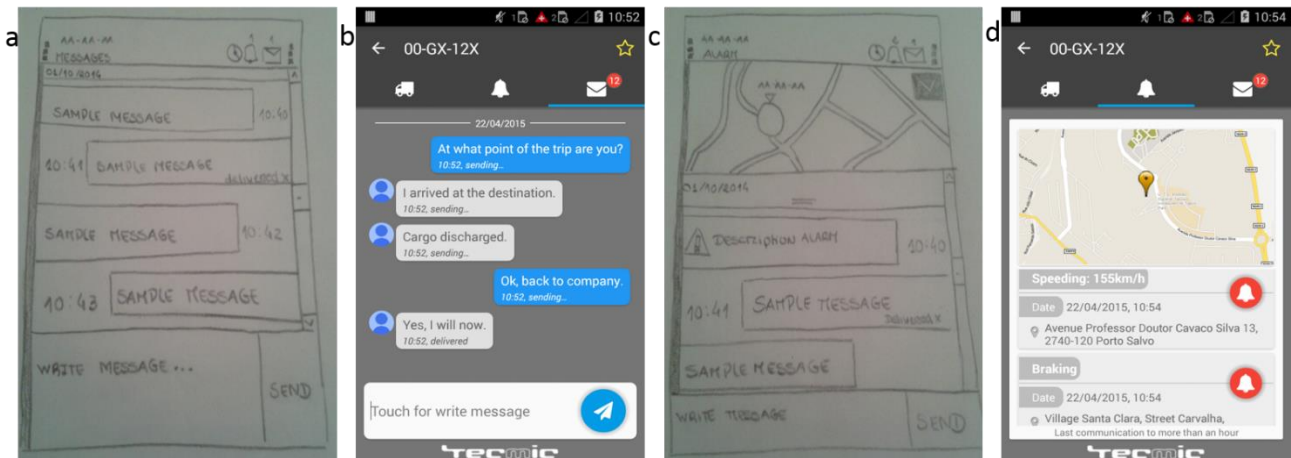


Fig. 5. (a) Initial prototype messages; (b) Final screen messages; (c) Initial prototype alarms; (d) Final screen alarms.

Fig. 5 (a) and (b) shows the screen that allows messages to be sent between fleet operators and vehicle drivers, including date and message status (outbox, sent, pending).

Fig. 5 (c) and (d) shows the screen intended for visualization of alarms originated by fleet vehicles. These alarms are generally related to travel plan faults such as excessive stopping time, geographically wrong locations or safety non-recommendable regions, but alarms related to travel plan compliance are also present, for example, planned arrival time confirmation. Alarms are prioritized in order to provide highlighted information to the fleet operator according to their relevance. Alarms information is made available to the fleet operator mobile device in real-time.

5. Tests

This section presents some of most relevant issues of iZiTraN Mobile project testing phase. Unit tests were developed along each iteration. Fig. 6 presents one unit test, showing the sending of a message and corresponding response, supported by iZiTraN events communication framework and MVP pattern.

```

public class MessagesFragmentTest {
    private static MessagesFragmentPresenter mPresenter;
    private static TestMessagesView mView;
    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
        // setup params
        mPresenter = new MessagesFragmentPresenter();
        mView = new TestMessagesView();
    }
    @Test
    public final void testSendMessage() {
        MessagesModuleEvent paramSendMessageEvent =
            new SendMessagePresenterEvent().view(mView).identifier(1).message("Hello Driver");
        try {
            //execute
            mView.post(paramSendMessageEvent);
        } catch (Exception e) {
            e.printStackTrace();
            fail(e.getMessage());
        }
    }
    private static class TestMessagesView implements MessagesFragmentAPI {
        @Override
        public void post(MessagesModuleEvent messagesModuleEvent) {
            MessagesViewBroadcast.getInstance().post(messagesModuleEvent);
        }
        @Override
        public void responseMessage(boolean response) {
            // compare result
            boolean expectedResult = true;
            assertEquals(expectedResult, response);
        }
    }
}

```

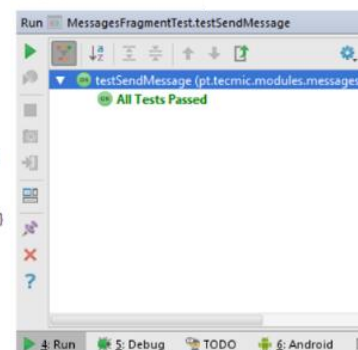


Fig. 6. iZiTraN messaging unit test.

Table 1 presents the information collected at various stages during the execution of the previous test, allowed by the Logging artifact, which registers events flowing in all existing channels.

Table 1. Messaging information collected by the Logging artifact

2015/04/22 13:46:46	INFO	SendMessagePresenterEvent	SendMessagePresenterEvent(identifier=1, message=Hello Driver)
2015/04/22 13:46:46	INFO	SendMessageBusinessEvent	SendMessageBusinessEvent(identifier=1, message=Hello Driver)
2015/04/22 13:46:46	INFO	SendMessageCommunicationEvent	SendMessageCommunicationEvent(identifier=1, message=Hello Driver)
2015/04/22 13:46:46	INFO	ResponseMessageCommunicationEvent	ResponseMessageCommunicationEvent(response=true)
2015/04/22 13:46:46	INFO	ResponseMessageBusinessEvent	ResponseMessageBusinessEvent(response=true)
2015/04/22 13:46:46	INFO	ResponseMessagePresenterEvent	ResponseMessagePresenterEvent(response=true)

Acceptance tests were performed by the development team in each iteration. Testing results were very positive, since the interfaces have been gradually improved with each iteration, as shown in Fig. 4 and Fig. 5.

6. Conclusions and future work

With iZiTraN Mobile project Tecmic SA created an Android application for the presentation of fleet vehicles on a map, with alarms and messages exchanges between fleet operators and drivers, supported by real-time communication. This project follows Tecmic SA technology choice of Android for mobile platforms and continues the evolution of its products lines XTraN²⁹, 4Forces³⁰, among others.

iZiTraN Mobile project brings all benefits of mobile devices features into this Tecmic SA business area. It also introduced technological innovation on Tecmic software development processes, since iZiTraN events based architecture revealed very positive system quality attributes such as maintainability and modularity, but also performance.

For future work we intend to add energy efficiency, driving time, driver's rankings and other application modules, as well as introduce Gamification³¹ principles for the GUI design, and increase testing efforts.

References

1. R. S. Pressman, *Software Engineering: A Practitioner's Approach*. 2006, p. 720.
2. M. Choi; E. Cho ; D. Park ; J. Bae ; C. Moon and D. Baik, "A Synchronization Algorithm of Mobile Database for Ubiquitous Computing," [Fifth International Joint Conference on INC, IMS and IDC, 2009], pp. 416-419, August 2009.
3. M. Fathy; M. Soryani; A. E. Zonouz; A. Asad and Seyrafi, M., "Some Enhanced Cache Replacement Policies for Reducing Power in Mobile Devices," [International Symposium on Telecommunications, 2008. IST 2008.], pp. 230-234, August 2008.
4. WELLS, «D. Extreme Programming: A gentle introduction». Available: <http://www.extremeprogramming.org/>>. [Access: 05-02-2015].
5. Karl Wiegers, *More About Software Requirements: Thorny Issues and Practical Advice (Developer Best Practices)*, December 2005.
6. K. BECK, «Extreme Programming Explained», Massachusetts, 2000.
7. «Smartphone OS Market Share, Q3 2014». Available: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. [Access: 05-02-2015].
8. M. Wilcox and C. Voskoglou, «State of the Developer Nation Q3 2014», 2014.
9. «What is Maven». Available: <https://maven.apache.org/what-is-maven.html>. [Access: 24-04-2015].
10. «Transmitting Network Data Using Volley». Available: <http://developer.android.com/training/volley/index.html>. [Access: 05-02-2015]
11. «What is ASP.NET SignalR». Available: <http://signalr.net/>. [Access: 24-04-2015].
12. T. D. R. Varela and S. Loh, «Implementação e Análise da Utilização de WebSockets em Sistemas Computacionais», pp. 1–19.
13. V. M. Themudo, «Implementação de um servidor de negociação em bolsa baseado em WebSocket», 2014.
14. «GuavaExplained». Available: <https://code.google.com/p/guava-libraries/wiki/GuavaExplained>. [Access: 24-04-2015].
15. K. Chandy and W. R. Schulte, *Event Processing: Designing IT Systems for Agile Companies*. 2009.
16. B. M. Michelson, *Event-Driven Architecture Overview*. 2006.

17. «WMS reference». Available: <http://docs.geoserver.org/stable/en/user/services/wms/reference.html>. [Access: 05-02-2015].
18. «API do Google Maps». Available: <https://developers.google.com/maps/?hl=pt-PT>. [Access: 05-02-2015].
19. «gvSIG Mini». Available: <https://confluence.prodevelop.es/display/GVMN/Home>. [Access: 05-02-2015].
20. «Androzic - Android navigation application». Available: <http://androzic.com/>. [Access: 05-02-2015].
21. «Tile Map Service Specification». Available: http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification. [Access: 05-02-2015].
22. «OpenLayers». Available: <http://openlayers.org/>. [Access: 05-02-2015].
23. «Framework OL3». Available: <https://github.com/netgis/ol3>. [Access: 05-02-2015].
24. «PhoneGap». Available: <http://phonegap.com/>. [Access: 05-02-2015].
25. «OpenStreetMap - Tools for Android». Available: <https://github.com/osmdroid/osmdroid>. [Access: 05-02-2015].
26. A. Syromiatnikov and D. Weyns, A Journey Through the Land of Model-View- Design Patterns.
27. M. Alles; B. Harleton and G. Pattison et al., Presenter First: Organizing Complex GUI Applications for Test-Driven Development.
28. «Material design». Available: <http://www.google.com/design/spec/material-design/introduction.html>. [Access: 24-04-2015].
29. «XTraN». Available: <http://www.tecmic.pt/portfolio/xtran/>. [Access: 24-04-2015].
30. «4Forces». Available: <http://www.tecmic.pt/portfolio/4forces/>. [Access: 24-04-2015].
31. G. Zichermann and C. Cunningham, Gamification by Design - Implementing Game Mechanics in Web and Mobile Apps. 2011, p. 208.

Anexos

Neste capítulo são apresentados os anexos deste documento, anexos esses que apresentam os projetos na área da gestão de frotas, que foram aparecendo ao longo do período de estágio.

Anexo I - Projeto Tempos de Condução

O projeto Tempos de Condução é apresentado como tendo aparecido naturalmente durante o período de estágio, sendo realizado para corresponder às necessidades de vários clientes, na área da gestão de frotas.

Este projeto baseou-se no desenvolvido de uma biblioteca, que através de dados recolhidos pelo tacógrafo do veículo (dispositivo utilizado em veículos, pesados de mercadorias, para monitorar o tempo de trabalho, condução e descanso) e da distância até ao destino, seja possível estimar o tempo que um condutor pode demorar até chegar ao destino. Numa fase inicial teve de se fazer um estudo exaustivo às regras que os condutores têm para os tempos de trabalho, condução e descanso, para que desta forma se cria-se um algoritmo para devolver o tempo estimado, tendo como parâmetros os tempos enviados pelo tacógrafo e a distância até ao ponto de destino.

De notar que este projeto ainda se encontra em fase de desenvolvimento e que se prevê que venha a ser integrado na aplicação móvel iZiTraN Mobile, uma vez que muitos utilizadores já pediram a sua inclusão. O protótipo criado com o intuito de integrar este projeto é apresentado na Figura 139.

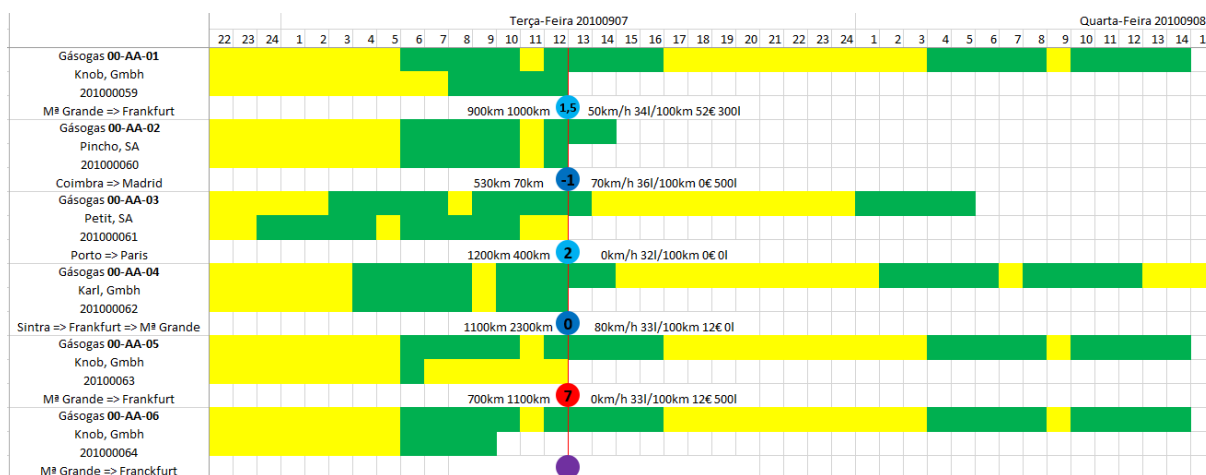


Figura 139 - Protótipo inicial para os Tempos de Condução.

Anexo II - Projeto Avistamentos

O projeto Avistamentos é apresentado como tendo aparecido naturalmente durante o período de estágio, sendo realizado para corresponder às necessidades reais de um cliente específico, na área da gestão de frotas.

Este projeto baseou-se no desenvolvimento de uma aplicação para a plataforma Android, cujo objetivo foi aceder aos avistamentos (termo usado no projeto para descrever ocorrências) apresentados na consola gráfica Tecmic presente no veículo (equipamento que disponibiliza *interface* gráfica para o condutor receber e realizar tarefas), para depois o utilizador os tratar. Os avistamentos são enviados pela consola em formato de Short Message Service (SMS), tendo a aplicação de ficar à escuta quando uma SMS, com o formato definido, tivesse sido enviada. Depois de detetada a nova SMS, a aplicação notifica o utilizador da chegada de um novo avistamento, indo depois para uma lista de avistamentos por tratar, onde por fim o utilizador precederá ao seu tratamento.

O tratamento do avistamento baseia-se em três pilares essenciais: registar avistamento através do formato de fotografia; recolha de coordenadas Global Positioning System (GPS) e ainda na existência de um campo observações. Para o registo de avistamento através do formato de fotografia, o utilizador poderá aceder à aplicação de fotografias, ou à galeria do dispositivo podendo assim tirar ou importar fotografias para a galeria da própria aplicação. Para a recolha de coordenadas GPS, a aplicação ao detetar que o GPS do dispositivo está desligado, pede que o utilizador o ligue, apresentando o ecrã com essa intensão. Na condição de o GPS estar ligado, a aplicação recolhe automaticamente as coordenadas onde o utilizador se encontra. O campo observações é um simples campo de texto para que o utilizador possa acrescentar mais alguma informação extra. Depois destes campos preenchidos, sendo os mesmos opcionais, o utilizador confirma o avistamento como tratado e este é guardado na base de dados da aplicação.

Após o tratamento feito pelo utilizador o avistamento é transferido para a lista de avistamentos tratados, tendo aqui a aplicação o papel fundamental de ser a responsável por enviar o tratamento para o seu Web service, uma vez que a aplicação fica à escuta que o dispositivo móvel tenha acesso à Internet para depois proceder ao envio de todos os avistamentos já tratados.

Anexo III - Projeto Rent-a-Car

O projeto Rent-a-Car é apresentado como tendo aparecido naturalmente durante o período de estágio, sendo realizado para corresponder às necessidades de vários clientes, na área da reserva de veículos.

Este projeto baseou-se no desenvolvimento de uma aplicação para a plataforma Android, cujo objetivo consistiu na reserva, *test drive*, empréstimo ou venda de veículos. O utilizador da aplicação (vendedor em rent-a-car) pode introduzir um veículo na aplicação através de um identificador QR Code [139] que identifica univocamente o veículo em que recairá a ação. A aplicação deteta o identificador QR Code através da manipulação de um *scanner* que faz a leitura do mesmo. Depois de introduzir o veículo o utilizador atualiza o estado atual do mesmo, através da contabilização do nível de enchimento do depósito, dos quilómetros, mas também através da possibilidade de identificar os riscos, mossas, algo partido, ou que esteja em falta. Esta possibilidade é feita através do recurso a um simples clique numa imagem representativa de um croqui, sendo que cada clique na zona acidentada do veículo permite ao utilizador poder ainda tirar uma fotografia, para que assim o ponto tenha uma correspondência à fotografia real do estado do veículo na zona selecionada.

Após o utilizador colocar todas as informações anteriores, coloca também alguns dados referentes ao cliente e escolhe a ação de reserva, *test drive*, empréstimo ou venda, indo depois estes dados para a base de dados e para o Web service do projeto, caso a Internet do dispositivo esteja ligada. O registo vai ainda para a lista de veículos da aplicação para que o utilizador possa ter o conhecimento do real estado dos veículos que se encontram à sua responsabilidade.

Anexo IV - Projeto Consola Gráfica para Transportes

O projeto Consola Gráfica para Transportes é apresentado como tendo aparecido naturalmente durante o período de estágio, sendo realizado para corresponder às necessidades de vários clientes, na área da gestão de frotas.

Este projeto baseou-se no desenvolvimento de uma aplicação Windows Mobile para consolas gráficas embarcadas (Windows CE) utilizadas pelos condutores de pesados de mercadorias. O utilizador da aplicação (conductor) pode identificar-se através do seu código de Cyclic Redundancy Check (CRC) [140] que é validado localmente através da execução de um algoritmo de *hash* que permite validar a integridade do código do condutor. Para além da identificação a aplicação permite receber e enviar mensagens para o centro de controlo e ainda visualizar uma listagem de todas as viagens associadas ao veículo. As viagens podem ser aceites, iniciadas, paradas ou recusadas, contendo a sua descrição, os serviços associados à mesma e uma listagem de paragens que o veículo pode seguir, tendo várias as operações associadas à paragem. As operações associadas às paragens podem ser chegada, iniciada, parada e partida, sendo que o condutor pode configurar os guias de remessa onde o condutor pode indicar o peso, volume, estrada, quantidade de carga e estado de execução da paragem.

A este projeto foi integrada a *framework* de navegação Aponia que permite navegação específica para veículos pesados de transportes. Com a integração desta *framework* o condutor pode visualizar todo o caminho que deve percorrer ao longo da viagem, sendo a própria aplicação desenvolvida a disponibilizar a esta *framework* as coordenadas da viagem seleccionada pelo condutor.