

XTraN Green Driving App

Módulo de Apoio à Condução Sustentável

João Eduardo Alves Nogueira

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Orientadores: Prof. Alberto Manuel Ramos da Cunha
Eng. João Miguel Henriques Domingos

Júri

Presidente: Prof. Paolo Romano
Orientador: Prof. Alberto Manuel Ramos da Cunha
Vogal: Prof. Miguel Filipe Leitão Pardal

Novembro 2022

Agradecimentos

No decorrer da unidade curricular de Dissertação com a qual concluo o Mestrado em Engenharia Informática e de Computadores no Instituto Superior Técnico, contei com o apoio de várias pessoas às quais gostaria de agradecer.

Primeiramente agradeço aos meus pais, não só pelas oportunidades que me disponibilizaram durante todo o meu percurso académico, mas também pelo constante apoio ao longo do mesmo. Queria também agradecer a um amigo que poderia ser meu irmão.

De seguida agradeço à Tecmic S.A. e aos seus colaboradores com os quais tive o privilégio de aprender ao longo do estágio nesta empresa que marca o meu começo no mundo profissional. Destaco evidentemente o coorientador deste projeto, o Engenheiro João Domingos, pela confiança que depositou em mim no início deste processo, bem como por toda a orientação, incentivo e encorajamento dado durante todo o mesmo.

Por fim agradeço também ao Professor Alberto Cunha pela ajuda disponibilizada relativa à parte académica deste projeto, pelas apreciações e comentários que elevam a qualidade deste mesmo documento.

Muito obrigado!

Abstract

In a world currently facing an energy crisis and still trying to fight off global warming, the levels of pollution caused by companies' vehicle fleets have a considerable negative impact on this problem. One way of mitigating this is for people to adopt environmentally friendly driving practices. This document describes the work developed as part of the *DriveGreen* project, proposed by the Portuguese company *Tecmic - Tecnologias de Microelectrónica, S.A.*.

The goal was the creation and integration of a new eco-driving support module in the company's *TeamMobile* mobile application. This application is used by company fleets' drivers belonging to Tecmic's clients, and provides features relating to GPS tracking and management of their overall daily work assignments. The new module receives the vehicle's sensor data from Tecmic's *XTraN* unit installed on it, and analyses and transforms that data into useful metrics related to ongoing and finished journeys: as a small set of charts in a bottom bar available during the trip, and through a detailed screen with suggestions on how to improve the user's driving style based on the finished journey data, respectively.

The module was developed and integrated in the *TeamMobile* application, but it is not yet finished, requiring additional tests related to its communication protocol with the *XTraN* unit and a new developed web service. After this, the new module will be subjected to internal usability tests, before being made available in a beta version to some of the company's clients. This will ultimately lead to the official release of this new version of the application.

Keywords

Mobile Application; Xamarin; Android; Eco-Driving.

Resumo

Num mundo atualmente marcado por uma crise energética e pelo aquecimento global, a poluição provocada por frotas de veículos terrestres tem um impacto negativo considerável. Uma das maneiras de o atenuar consiste na adoção de hábitos de condução mais sustentáveis. Desta maneira, o presente documento descreve o trabalho desenvolvido no âmbito do projeto *DriveGreen*, proposto pela empresa Tecmic - Tecnologias de Microelectrónica, S.A..

O objetivo foi a criação de um módulo de apoio à condução sustentável integrado na aplicação *TeamMobile* da Tecmic. Esta aplicação é utilizada por condutores de veículos de prestação de serviços pertencentes a clientes da empresa, e possui funcionalidades relacionadas com localização por GPS e gestão das viagens e do trabalho realizado por estes trabalhadores no dia a dia. Este módulo recebe os dados do veículo provenientes de uma caixa equipada *XTraN* da empresa, procedendo à sua análise e transformação em indicadores úteis a curto e longo prazo, respetivamente, indicadores gráficos de avaliação da viagem a decorrer, e resumos das viagens terminadas com sugestões de melhoria.

O módulo foi desenvolvido e integrado na aplicação *TeamMobile* encontrando-se neste momento numa última fase de desenvolvimento, relacionada com a testagem do protocolo de comunicação com a caixa e com o novo *WebService*. Concluído este processo, o módulo será alvo de testes de usabilidade internos, disponibilizado numa nova versão da *TeamMobile* a alguns dos clientes da empresa, e finalmente refinado antes do seu lançamento oficial.

Palavras Chave

Aplicação Móvel; Xamarin; Android; Condução Sustentável.

Conteúdo

1	Introdução	1
1.1	Enquadramento	3
1.2	Motivação e Objetivos	5
1.3	Estrutura do Documento	5
2	Estado da Arte	7
2.1	Sistemas Tecmic	9
2.1.1	Soluções XTraN	9
2.1.2	GreenDriving	9
2.1.3	XTP Ecodriver	11
2.1.4	TeamMobile	12
2.2	Aplicações Desenvolvidas por Terceiros	14
2.2.1	Car Scanner ELM OBD2	15
2.2.2	ECO-Driving Speedometer	16
2.2.3	Drive Eco	17
2.2.4	Ecologic	18
2.2.5	Torque Pro	19
2.2.6	DashCommand	20
2.2.7	Comparação	21
3	Proposta de Solução	23
3.1	Arquitetura e Design	25
3.1.1	Dados Recolhidos e ECSM	25
3.1.2	Apresentação da Informação	25
3.1.2.1	Histogramas	26
3.1.2.2	Indicadores em Segundo Plano	29
3.1.2.3	Ecrã de Resumo da Viagem	30
3.1.3	Protocolo de Comunicação	31
3.1.4	Base de Dados	32

3.2	Aspetos Sociais e Privacidade de Informação	33
3.3	Tecnologias e Ferramentas	34
3.3.1	Xamarin	35
3.3.1.1	Xamarin Nativo	35
3.3.1.2	Xamarin.Forms	35
3.3.1.3	MvvmCross	36
3.4	Planeamento	37
4	Desenvolvimento	39
4.1	Gestão e Metodologia	41
4.1.1	Gestão de Tarefas	41
4.1.2	Comunicação	41
4.1.3	Ambiente de Trabalho	42
4.1.4	Calendário	42
4.2	Etapas de Desenvolvimento	43
4.2.1	Desenho dos Ecrãs	43
4.2.1.1	Histogramas	44
4.2.2	Integração do Módulo	46
4.2.2.1	Ecrãs	46
4.2.2.2	Camada de Negócio	50
4.2.3	Protocolo de Comunicação	53
4.2.3.1	Via Tags	53
4.2.3.2	Via Webservice	56
4.2.4	Uniformização dos Ecrãs da Aplicação TeamMobile	59
5	Avaliação e Testes	61
5.1	Testes Funcionais e de Usabilidade	63
5.2	Aprovação e Lançamento	64
6	Conclusão	65
6.1	Conclusões	67
6.2	Trabalho Futuro	68
	Bibliografia	71
A	Anexo A	75
B	Anexo B	85
C	Anexo C	93

Lista de Figuras

2.1	Ecrã principal da aplicação <i>GreenDriving</i>	10
2.2	Ecrã de informação detalhada do módulo <i>XTP Ecodriver</i> , retirado do documento interno "Manual do XTraN Passenger Web".	11
2.3	Ecrã de listagem de viagens da aplicação <i>TeamMobile</i> , retirado do documento interno "Manual-Tablet Gestão de Equipas-IP".	13
2.4	Ecrãs da aplicação Car Scanner ELM OBD2.	15
2.5	Ecrã principal da aplicação ECO-Driving Speedometer.	16
2.6	Ecrãs da aplicação Drive Eco.	17
2.7	Ecrãs da aplicação Ecologic.	18
2.8	Ecrãs da aplicação Torque Pro.	19
2.9	Ecrãs da aplicação Dash Command.	20
3.1	Exemplo de um histograma de velocidade instantânea a ser apresentado no módulo <i>DriveGreen</i>	27
3.2	Exemplo de um histograma de velocidade constante a ser apresentado no módulo <i>DriveGreen</i>	28
3.3	Protótipo em papel da barra inferior (ecrã secundário) do DriveGreen.	29
3.4	Protótipo em papel do ecrã detalhado do <i>DriveGreen</i>	30
3.5	Arquitetura de comunicação entre caixa, aplicação, <i>WebService</i> e base de dados central. As ligações a verde correspondem à comunicação direta do módulo com a caixa (A), e as ligações violeta à comunicação através de um <i>WebService</i> à base de dados central (B).	31
3.6	Diagrama do modelo de dados na base de dados da aplicação <i>TeamMobile</i> , no contexto do novo módulo.	33
3.7	Comparação entre os paradigmas do <i>Xamarin Nativo</i> e <i>Xamarin.Forms</i>	36
4.1	Vista de tarefas semanais (<i>sprint</i>) na plataforma <i>Microsoft Azure DevOps</i> da Tecmic.	41
4.2	Gráfico de <i>Gantt</i> das etapas de desenvolvimento do projeto.	42

4.3	Protótipo em computador do ecrã principal do novo módulo.	43
4.4	Protótipo em computador do ecrã secundário (barra com gráficos em segundo plano) do novo módulo.	44
4.5	Gráfico desenhado com a biblioteca <i>MPAndroidChart</i>	45
4.6	Histograma resultante da aplicação do algoritmo de criação de gradiente.	45
4.7	Versão inicial do ecrã principal do novo módulo na <i>TeamMobile</i>	46
4.8	Versão inicial da barra inferior (ecrã secundário) do novo módulo na <i>TeamMobile</i>	47
4.9	Versão final da barra inferior do módulo <i>DriveGreen</i> no ecrã de atividade da <i>TeamMobile</i>	48
4.10	Versão final (legendada) do ecrã principal do módulo <i>DriveGreen</i>	49
4.11	Diagrama de sequência de receção de dados no módulo.	52
4.12	Diagrama de sequência de acesso do utilizador ao ecrã principal do módulo.	53
4.13	Diagrama de comunicação entre o módulo/aplicação e a caixa via <i>Tags</i>	55
4.14	Diagrama de classes da API de dados das viagens do <i>WebService</i>	56
4.15	Diagrama de comunicação do módulo via <i>WebService</i>	57
4.16	Diagrama de classes da API do catálogo de variáveis do <i>WebService</i>	58
4.17	Evolução do ecrã de atividade da aplicação <i>TeamMobile</i> visto no emulador de um telemóvel.	60

Lista de Tabelas

2.1	Comparação das características das aplicações analisadas.	21
-----	---	----

Listagens

4.1	Classe DriveGreenData.	51
4.2	Exemplo de variável do catálogo do <i>Webservice</i>	58
B.1	Código de teste da recepção de dados das viagens.	85

Acrónimos

API	Application Programming Interface
CEO	Chief Executive Officer
DAL	Data Access Layer
DI	Dependency Injection
ECSM	Energy Comfort and Security Module
GPS	Global Positioning System
IDE	Integrated Development Environment
IoC	Inversion of Control
JSON	JavaScript Object Notation
MVVM	Model-View-ViewModel
OBD	On-Board Diagnostics
REST	Representational State Transfer
TCP	Transmission Control Protocol
UI	User Interface
ULDA	Utilities, Logística, Distribuição e Ambiente
URL	Uniform Resource Locator
VPN	Virtual Private Network
XAML	Extensible Application Markup Language

1

Introdução

Conteúdo

1.1 Enquadramento	3
1.2 Motivação e Objetivos	5
1.3 Estrutura do Documento	5

O presente documento foi elaborado no contexto da unidade curricular de Dissertação de Mestrado em Engenharia Informática e de Computadores no Instituto Superior Técnico em Lisboa.

O projeto no qual se baseia foi proposto pela empresa Tecmic S.A., onde foi realizado um estágio académico com início em Outubro de 2021, altura do começo da unidade curricular de Projeto Integrador de 2º Ciclo em Engenharia Informática e de Computadores. O projeto inicialmente designado de *XTraN Green Driving App*, mais tarde nomeado apenas como *DriveGreen*, é um módulo aplicacional de apoio à condução sustentável integrado na aplicação *TeamMobile* da empresa. O módulo *DriveGreen* acrescenta à aplicação principal funcionalidades que permitem a recolha, análise e tratamento dos dados processados por uma caixa *XTraN* da empresa equipada no veículo, convertendo os mesmos em indicadores úteis a curto prazo, como pequenos gráficos que podem ser visualizados em segundo plano durante a viagem, e a longo prazo, através de um ecrã de resumo apresentado no fim das viagens, com diversos dados e sugestões.

1.1 Enquadramento

Um dos principais focos da Tecmic S.A. está na gestão e controlo de frotas de veículos terrestres. Através da aplicação *FleetMobile* [1] é possível ao operador de frota orientar e coordenar remotamente todos os veículos registados, obter a sua localização em tempo real por Global Positioning System (GPS), e receber informação sobre o percurso percorrido, dados diversos sobre a viatura, ou alertas para qualquer situação inesperada que tenha acontecido. Por sua vez, os condutores através da *TeamMobile* têm acesso a um sistema que lhes permite receber informação sobre as viagens e serviços a prestar, registar o trabalho efetuado, comunicar diretamente com o operador de frota através de mensagens de texto e enviar alertas, se necessário.

No entanto, a movimentação de frotas de veículos terrestres está associada a algumas das questões mais preocupantes atualmente à escala mundial, isto é a poluição, o aquecimento global e o consumo de energia. De forma a atenuar este problema, a prática de uma condução mais sustentável é apontada como uma solução eficaz, mas o que é ao certo condução sustentável ou *eco-driving*? Das várias definições possíveis, a mais abrangente classifica este termo como "a adoção de práticas e comportamentos que promovem uma condução mais eficiente (em termos energéticos), segura, e amiga do ambiente" [2].

Isto pode ser conseguido através de medidas em duas esferas distintas, a primeira relacionada com o veículo utilizado, e a segunda com os hábitos praticados durante a condução. No que diz respeito à primeira, o veículo em questão deve ser selecionado tendo em conta características como a sua eficiência de combustível e emissão de gases poluentes, além disso, deve ser feita a sua manutenção regular. Esta primeira esfera tem um maior impacto, devido à rápida evolução tecnológica no ramo,

contudo e como é evidente, não é útil no contexto deste projeto. Já a segunda esfera, de hábitos de condução, vai de encontro ao que o novo módulo pretende ensinar e corrigir, uma tarefa complexa, dado que os hábitos de condução das pessoas tendem a persistir, não evoluindo com a tecnologia dos seus veículos. [3].

Os fatores mais importantes na prática deste tipo de condução estão relacionados com a boa gestão da velocidade, aceleração, travagem e *ralenti*. A gestão da velocidade deve ser feita com o objetivo de manter uma velocidade constante, algo mais difícil de atingir em trajetos dentro de cidades ou com muito trânsito, mas mais fácil em troços como auto-estradas. De qualquer forma, esta é a principal meta na gestão da velocidade do veículo, sem esquecer naturalmente o cumprimento dos limites de velocidade. A gestão da aceleração e travagem baseia-se em evitar movimentos bruscos e súbitos durante a condução. Com este objetivo em mente o condutor deve praticar uma condução atenta e cautelosa, mantendo sempre uma distância de segurança dos outros veículos, de forma a antecipar possíveis problemas e adaptar a velocidade e mudanças ao trajeto que se avizinha. Finalmente, temos o *ralenti*, a imobilização do veículo com o motor a trabalhar. Este comportamento tem um impacto considerável no consumo de combustível do veículo e na emissão de gases poluentes, pelo que o condutor deve desligar o motor do veículo sempre que se encontre parado devido a situações como trânsito intenso ou longos tempos de espera em semáforos [4].

Relativamente à aplicação destas medidas e à análise dos seus resultados, podemos encontrar diversos estudos sobre o assunto. Por exemplo, na Finlândia [5], onde o apoio de uma aplicação de condução sustentável permitiu aos condutores de autocarros de Helsínquia uma redução de 8.9% no consumo de combustível, também como um impacto significativo na atenuação do excesso de velocidade e ainda no aumento do conforto reportado pelos passageiros. Também na Suíça [6], 50 condutores de uma empresa utilizaram uma aplicação móvel que transmitia *feedback* sobre aceleração, travagem, velocidade e mudanças adequadas, entre outros, resultando numa redução de combustível de 3.23%. Outros estudos referem as vantagens de pequenas formações sobre este tipo de condução: na Bélgica [7], uma formação de quatro horas revela efeitos positivos na condução diária dos voluntários que a realizaram, citando uma diminuição de 5.8% no consumo de combustível, uma maior eficiência na gestão das mudanças, e ainda uma redução nas acelerações e nas rotações do motor. Outro aspeto que todos estes estudos sugerem é que apesar da existência de um efeito positivo a longo prazo após os condutores deixarem de utilizar a aplicação ou acabarem a formação, esse efeito tende a esmorecer, sendo necessário incentivar periodicamente à prática desta forma de condução, de maneira a que os condutores não regressem à prática dos seus antigos hábitos.

1.2 Motivação e Objetivos

A Tecmic tem estado atenta a esta problemática, e possui já soluções por exemplo no contexto de uma aplicação de gestão de transporte de passageiros (secção 2.1.3), e deste modo, com o intuito de ajudar o planeta e ao mesmo tempo reduzir os custos de combustível das frotas dos seus clientes, aposta no desenvolvimento de um módulo de condução sustentável, *DriveGreen*, para a sua aplicação *TeamMobile*. Este projeto tem como principais características:

- Transmissão de dados entre o veículo e aplicação de duas maneiras distintas: (a) Comunicação direta entre a caixa *XTraN* equipada no veículo e a consola de bordo onde está a aplicação; (b) Comunicação dos dados do centro de controlo da Tecmic (enviados pela caixa) para a aplicação através de um novo *WebService*. Isto acontece quando a aplicação não pode comunicar diretamente com a caixa;
- Tratamento e apresentação dos dados recebidos de duas formas: (a) Por meio de um ecrã principal a que o condutor possa aceder quando pretender, contendo a informação completa sobre a sua condução desde o início da viagem, e um histórico dos dados de viagens passadas. Neste ecrã será também disponibilizado *feedback* que indique ao condutor formas de melhorar a sua condução no futuro; (b) Através de pequenos indicadores (gráficos) mantidos em segundo plano nos vários ecrãs da aplicação principal, de forma discreta, durante a viagem a decorrer;
- Integração e testagem do referido módulo na aplicação da Tecmic, *TeamMobile*.

1.3 Estrutura do Documento

Este documento está organizado da seguinte maneira: o presente Capítulo 1 dá a conhecer o contexto do projeto e as suas principais características e objetivos. No Capítulo 2 é apresentado o Estado da Arte, onde foram estudadas aplicações relevantes para a problemática em questão, e feitas comparações com o projeto corrente. No Capítulo 3 estão presentes os detalhes da proposta de implementação e planeamento do projeto. O Capítulo 4 contém os detalhes do desenvolvimento prático do módulo. O Capítulo 5 apresenta a metodologia de testes e avaliação de usabilidade do módulo. E por fim o Capítulo 6, onde são apresentadas as conclusões e indicado o trabalho futuro.

2

Estado da Arte

Conteúdo

2.1	Sistemas Tecmic	9
2.2	Aplicações Desenvolvidas por Terceiros	14

Neste capítulo é analisado o Estado da Arte relativamente a projetos e aplicações no domínio do módulo *DriveGreen*. São primeiro dadas a conhecer algumas soluções da Tecmic neste contexto, bem como a aplicação alvo de integração do módulo, a *TeamMobile*. Seguidamente são avaliadas aplicações móveis desenvolvidas por terceiros disponíveis para *Android* ou *iOS* e feitas comparações de acordo com os critérios estabelecidos para este projeto.

2.1 Sistemas Tecmic

Dentro da esfera de soluções de transporte *XTraN* da Tecmic, existem já algumas com funcionalidades dedicadas ao apoio à condução sustentável, como a *XTraN Passenger* (secção 2.1.3), e outras ainda sem essas funções, como a *TeamMobile* (secção 2.1.4). Nesta secção 2.1 é primeiro explicado o conceito *XTraN*, seguido da apresentação de duas soluções previamente desenvolvidas no âmbito da condução sustentável: a aplicação móvel *GreenDriving* e o módulo *XTP Ecodriver*. É ainda apresentada a aplicação *TeamMobile* onde se pretende integrar o novo módulo *DriveGreen*.

2.1.1 Soluções XTraN

O termo *XTraN* é mencionado repetidamente ao longo deste projeto, pelo que é útil ser desde já apresentado. Este termo está relacionado com soluções da empresa ligadas ao registo de informação dos sensores do veículo e à sua localização por *GPS*, estando presente em duas sub-áreas da Tecmic, a gestão profissional de frotas [8] e a gestão de transporte de passageiros [9]. Da primeira área fazem parte aplicações como a *TeamMobile* (secção 2.1.4) e a *FleetMobile* [1], a caixa *XTraN*, presente em qualquer veículo com acesso a estes sistemas, e ainda a consola de bordo *XTraN* presente em alguns deles. Da segunda área faz parte a solução *XTraN Passenger* e o respetivo módulo *Ecodriver*, alvo de análise mais à frente neste capítulo (secção 2.1.3). Como tal, o desenvolvimento deste novo módulo *DriveGreen* está aliado a todas estas soluções de *hardware* e *software* pertencentes à empresa.

2.1.2 GreenDriving

Entre 2017 e 2018, sob a alçada da Tecmic, foi desenvolvida uma nova aplicação, *GreenDriving* [10], pelo Engenheiro Luís Henrique Santos Torres. O objetivo era a criação de uma aplicação móvel de apoio à eficiência energética que recolhesse e transformasse os dados do veículo em tempo real e os mostrasse ao condutor para que este pudesse adaptar a sua condução. A transmissão dos dados foi feita através da comunicação da aplicação móvel com uma caixa *MK3* da Tecmic. Existiam três principais tipos de informação disponibilizada: (a) Dados em tempo real como as rotações do motor, a velocidade instantânea e o consumo de combustível; (b) Contadores como o número de acelerações

bruscas e excessos de velocidade; (c) É uma secção de alarmes que reportava anomalias no veículo. Na figura 2.1 é possível ver o *layout* final da aplicação com as várias categorias de informação descritas.



Figura 2.1: Ecrã principal da aplicação *GreenDriving* [10].

A solução foi desenvolvida em *Xamarin* com uma arquitetura Model-View-ViewModel (MVVM), testada e lançada como uma aplicação independente. Esta última característica define desde logo uma das principais diferenças entre o projeto corrente e esta aplicação, já que o novo módulo *DriveGreen* tem o objetivo de ser integrado na aplicação já existente da Tecmic, a *TeamMobile*. No que toca à transmissão de dados entre o veículo e a aplicação, neste anterior projeto a comunicação era efetuada através de uma caixa *MK3*, ao contrário do pretendido neste módulo, com uma caixa *XTraN*. Serão assim novamente estudados os protocolos envolvidos nesta troca de informação entre a caixa *XTraN* do veículo e a consola de bordo. Outro foco do presente módulo é a discrição nos ecrãs e dados apresentados, dado que o objetivo não é que o condutor esteja constantemente focado na informação apresentada, mas sim que possa obter indicadores úteis enquanto observa os outros ecrãs da aplicação principal durante a viagem, ou mais tarde, no fim da viagem, através de um ecrã mais completo, de resumo. Também o tipo de informação disponibilizada irá ser diferente, desviando o foco de indicadores como a velocidade e as rotações do motor que já se encontram presentes no painel de instrumentos do veículo, para *feedback* simples que o condutor possa utilizar para melhorar a sua condução.

2.1.3 XTP Ecodriver

No contexto da plataforma Tecmic *XTraN Passenger* [9] de gestão profissional de transporte de passageiros, foi desenvolvido um módulo de eficiência energética, o *XTraN Passenger Ecodriver*. A solução *XTraN Passenger*, é um sistema *Web* que traz funcionalidades tais como o controlo e gestão das viagens, localização geográfica dos autocarros, mecanismos de videovigilância e alarmes, contagem de passageiros, entre outros. O módulo *Ecodriver* tem o objetivo de fornecer informação sobre a condução efetuada durante as viagens. A recolha de dados é feita através do sistema *CAN-Bus* do autocarro, e após o seu tratamento, são disponibilizados tanto na consola de bordo do condutor, como na aplicação central *Web* onde podem ser acedidos por gestores e supervisores. Toda a informação fica guardada persistentemente na plataforma através do armazenamento em base de dados, o que posteriormente permite a criação de relatórios sobre a eficiência energética da condução dos veículos.

O módulo disponibiliza na consola de bordo diversos indicadores: (a) Gerais como o consumo de combustível, distância percorrida e tempos de viagem; (b) Contadores de excessos cometidos, como travagens, acelerações e velocidade; (c) E histogramas, gráficos que através de um sistema simples de cores representam o desempenho do condutor ao longo do tempo, mais uma vez relacionado com métricas como a velocidade, travagem e aceleração. O condutor tem acesso na consola a estes dados de duas formas: os dados acumulados mais recentes, disponíveis permanentemente em segundo plano nos outros ecrãs da aplicação, e através de um painel detalhado de consulta que pode ser acedido para visualizar os dados recolhidos desde o início da viagem. Este ecrã de consulta pode ser visto na figura 2.2.

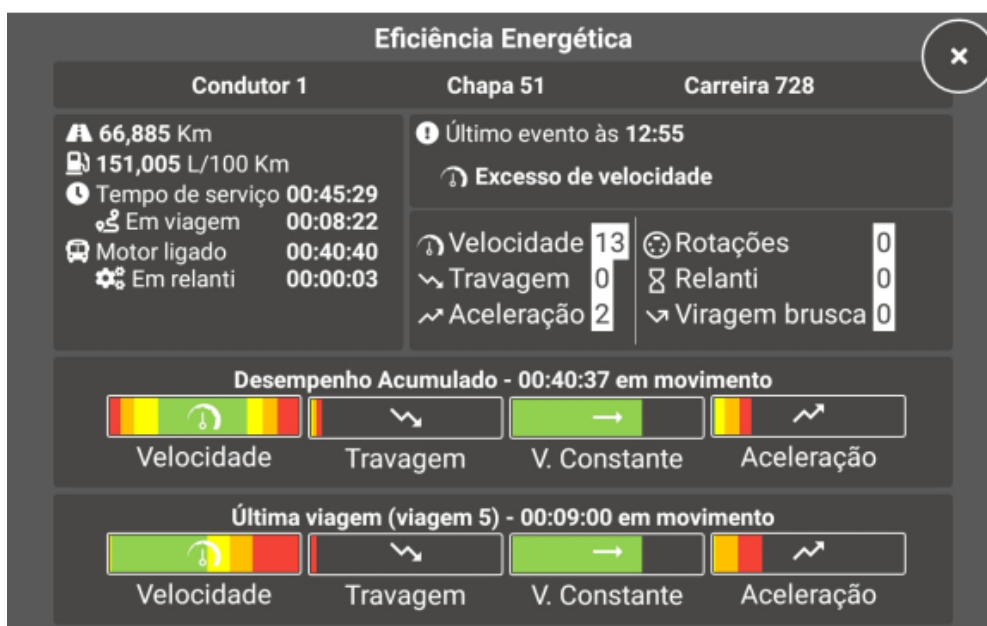


Figura 2.2: Ecrã de informação detalhada do módulo *XTP Ecodriver*, retirado do documento interno "Manual do *XTraN Passenger Web*".

Algumas das diferenças do módulo *Ecodriver* acima descrito em relação ao módulo *DriveGreen* deparam-se no protocolo de comunicação, já que no módulo *DriveGreen* a transmissão de informação é feita através da caixa equipada *XTraN*, que já realiza a recolha dos dados do veículo, não sendo necessário comunicar diretamente com o controlador do mesmo. Relativamente à apresentação de informação, o *Ecodriver* aproxima-se de vários dos requisitos pretendidos neste projeto, como a existência de dois tipos de ecrãs (uma vista detalhada num ecrã principal, e a apresentação de pequenos indicadores em *background* da aplicação) e da relevância do tipo de dados fornecidos. Uma das funcionalidades que no entanto não se encontra presente no *Ecodriver* é a disponibilização de *feedback* para que o condutor possa melhorar a sua forma de conduzir. Um dos objetivos do módulo *DriveGreen* é que através da análise dos dados recolhidos durante a viagem, como contadores de excesso e evolução temporal das métricas, possam ser automaticamente extraídas indicações úteis apresentadas no ecrã de resumo no final da viagem, que auxiliem o condutor a praticar uma condução mais sustentável.

2.1.4 TeamMobile

A *TeamMobile* é aplicação utilizada pelos condutores de veículos de prestação de serviços, servindo de contraparte à aplicação *FleetMobile* [1] usada pelos operadores de frota. A aplicação tem três ecrãs principais: o ecrã de *Atividade* onde o condutor e equipa do veículo se registam para começar o trabalho; o ecrã de *Mensagens* onde podem ser trocadas informações de forma manual ou automática com o operador de frota; e o ecrã de *Viagens* onde estão presentes todas as viagens, paragens e operações a realizar e onde são registados todos os detalhes sobre as mesmas. Na figura 2.3 pode ser vista a secção inicial do ecrã de *Viagens* da aplicação.

Desta maneira, o fluxo normal de uso desta aplicação é o seguinte:

1. O condutor efetua o registo da equipa no ecrã de *Atividade*, onde são também configurados outros parâmetros específicos como listas de equipamentos. Este processo é similar ao de um *login*;
2. É utilizado o ecrã de listagem de viagens onde é selecionada e iniciada a viagem a executar. Cada viagem pode ter várias paragens associadas, entenda-se, múltiplos locais de operação. Existe também disponível um módulo de navegação que serve de auxílio ao condutor para se deslocar aos vários destinos destas viagens, embora o trajeto a percorrer não seja restrito nem registado pela aplicação. As paragens da viagem estão ordenadas de modo a otimizar o tempo e combustível gasto, contudo esta ordem não é obrigatória, podendo o condutor efetuar as paragens por outra ordem caso seja necessário;
3. Em cada paragem é efetuado o serviço definido e registado em formulários o trabalho relacionado com o mesmo;
4. Ao fim do dia é terminada a atividade no primeiro ecrã.

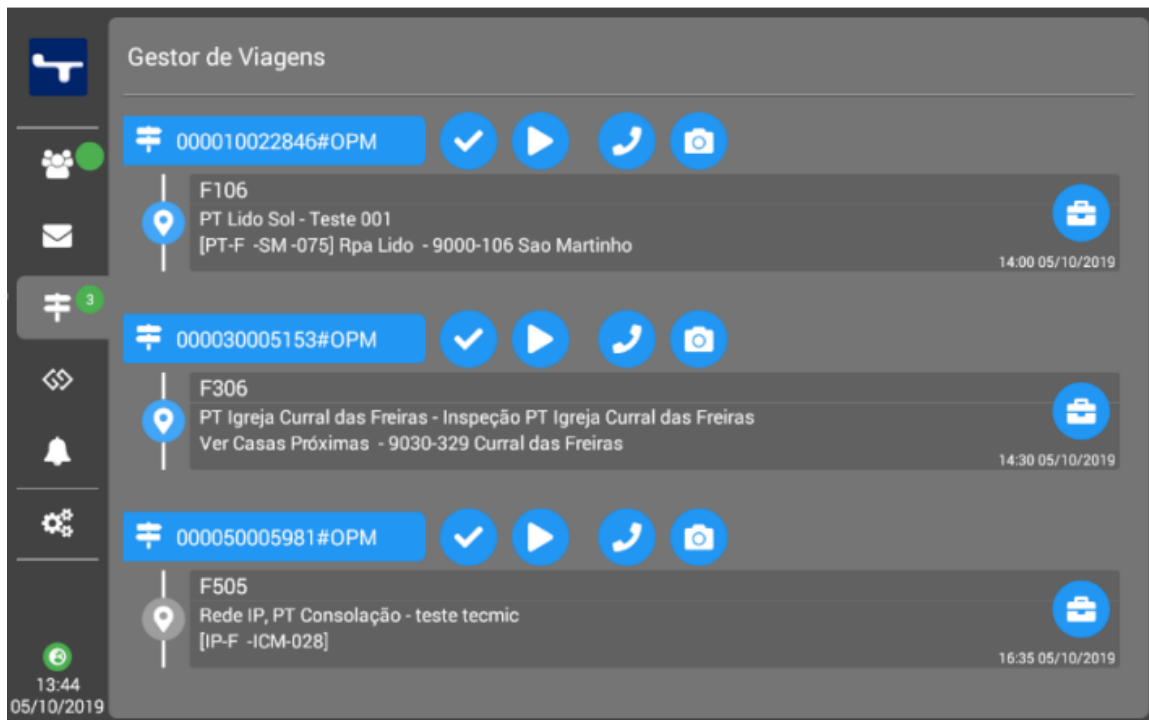


Figura 2.3: Ecrã de listagem de viagens da aplicação *TeamMobile*, retirado do documento interno "Manual-Tablet Gestão de Equipas-IP".

Dada a importância que esta aplicação tem na gestão de todos os aspetos do trabalho e viagens realizadas pelos trabalhadores, um dos grandes desafios deste projeto é a realização de uma integração simples e eficiente de um novo módulo que consiga expandir as funcionalidades da aplicação sem adulterar o seu atual funcionamento. Não esquecer que, para além da criação de novos ecrãs, como o de resumo do final da viagem, será também necessário adicionar elementos na *layout* de várias secções já existentes, servindo como indicadores em segundo plano que avaliam a condução praticada durante o período de tempo mais recente. Estes aspetos são abordados em detalhe no capítulo 3 do presente documento.

2.2 Aplicações Desenvolvidas por Terceiros

Fora do âmbito da Tecmic, foi também feita uma tentativa de investigação de outras soluções empresariais criadas no domínio do problema, como a da *Samsara* [11] ou da *INGTECH* [12], mas como seria de esperar a maioria dos detalhes não é disponibilizada ao público, pelo que este estudo não revelou resultados úteis. Optou-se então pela análise de aplicações móveis relevantes no contexto do módulo *DriveGreen*. Isto é, aplicações disponíveis para *Android* ou *iOS* cuja principal função seja a aquisição, transformação e apresentação de dados do veículo e que auxiliem no âmbito da condução sustentável. Esta análise e posterior comparação com o projeto corrente foi feita com base numa visão geral das características funcionais e de *design* dos ecrãs das aplicações, e não de acordo com as suas arquiteturas, ou protocolos que não foram objeto de estudo detalhado.

2.2.1 Car Scanner ELM OBD2

Disponível gratuitamente para *Android* e *iOS*, *Car Scanner* [13] é uma aplicação baseada no conceito On-Board Diagnostics (OBD) tradicional, utilizando todos os sensores disponíveis no veículo para facultar dados estatísticos, códigos de falha, desenhar gráficos e monitorizar o seu estado. A aplicação apresenta uma grande quantidade de informação que pode ser personalizada de acordo com as necessidades do utilizador, criando um *dashboard* com os parâmetros desejados. Apesar de gratuita a aplicação contém compras *in-app* que desbloqueiam funcionalidades adicionais. No entanto, a aplicação não é dirigida a condutores de veículos empresariais nem indicada para auxílio à condução sustentável.

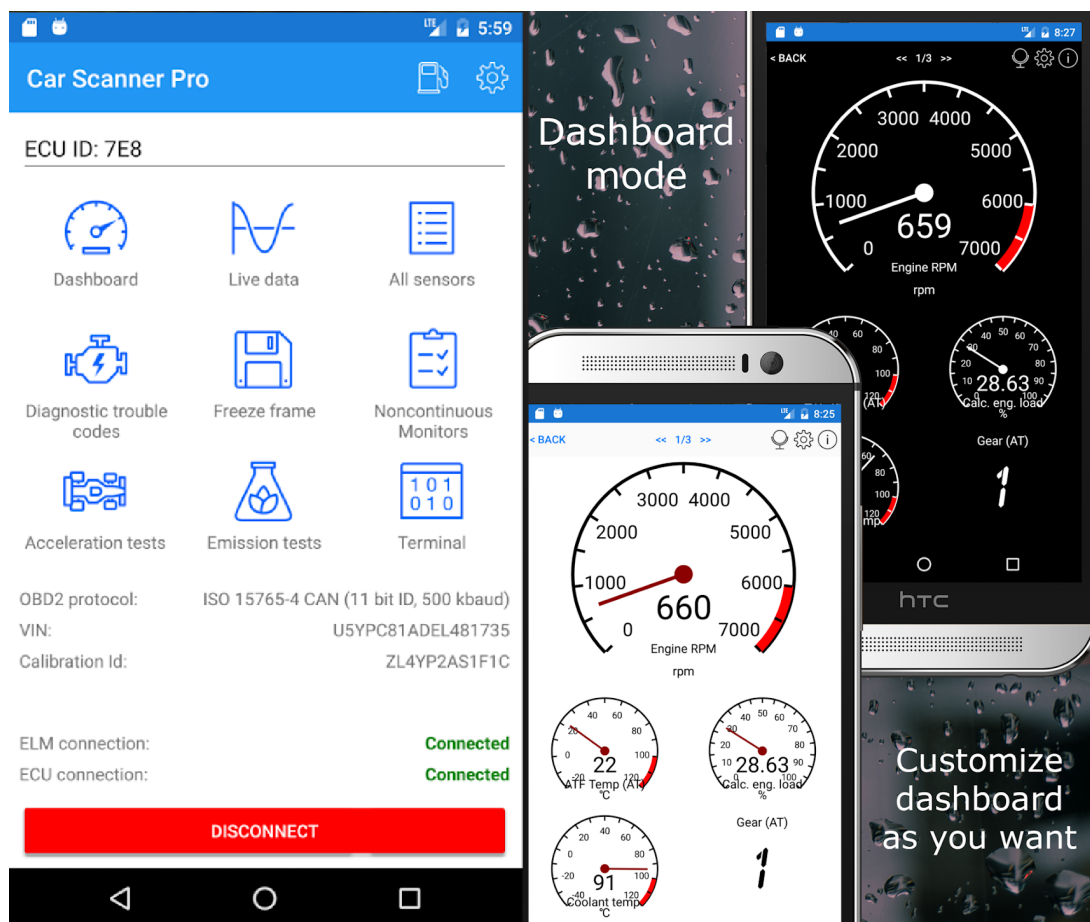


Figura 2.4: Ecrãs da aplicação Car Scanner ELM OBD2 [13].

2.2.2 ECO-Driving Speedometer

Esta aplicação [14] presente no *Google Play* de forma gratuita, parece ser algo simples, apresentando apenas um ecrã principal e algumas janelas de configurações. É referida com uma ferramenta acessível de condução económica, podendo ser facilmente instalada e imediatamente utilizada, oferecendo indicadores sobre velocidade, travagem, aceleração, distância percorrida, e dando a oportunidade de personalizar alertas sonoros. No entanto, os dados recolhidos vêm diretamente do dispositivo utilizado (telemóvel ou tablet) e não do próprio veículo.

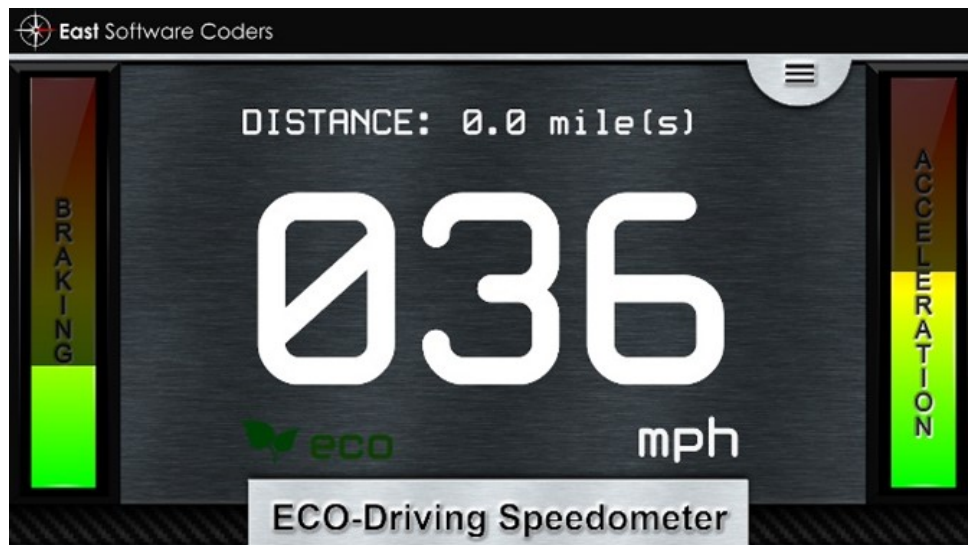


Figura 2.5: Ecrã principal da aplicação ECO-Driving Speedometer [14].

2.2.3 Drive Eco

Drive Eco [15], disponível gratuitamente para *Android*, é uma aplicação que pretende ajudar na prática de condução sustentável. Recolhe dados sobre inclinações, altitude, consumo e custo de combustível, velocidade e distância percorrida, entre outros, gerando gráficos e tabelas com base nos mesmos. Contudo, toda esta informação é um pouco falível, já que, mais uma vez, os dados são recolhidos através dos sensores do telemóvel e não pelo próprio veículo (OBD). A principal característica relacionada com condução sustentável parece ser a criação de gráficos e estatísticas sobre eficiência de combustível.



Figura 2.6: Ecrãs da aplicação Drive Eco [15].

2.2.4 Ecologic

Ecologic [16], disponível gratuitamente para *iOS* ou *Android*, é uma aplicação dirigida a empresas e gestão de frotas, disponibilizando maneiras de supervisionar grupos de trabalhadores e analisar os seus hábitos de condução. Através da recolha de dados por OBD apresenta os habituais indicadores de condução, alarmes, e avalia condução sustentável. A aplicação mantém dados persistentes sobre as viagens passadas, que podem ser acedidos através de um portal *Web*.

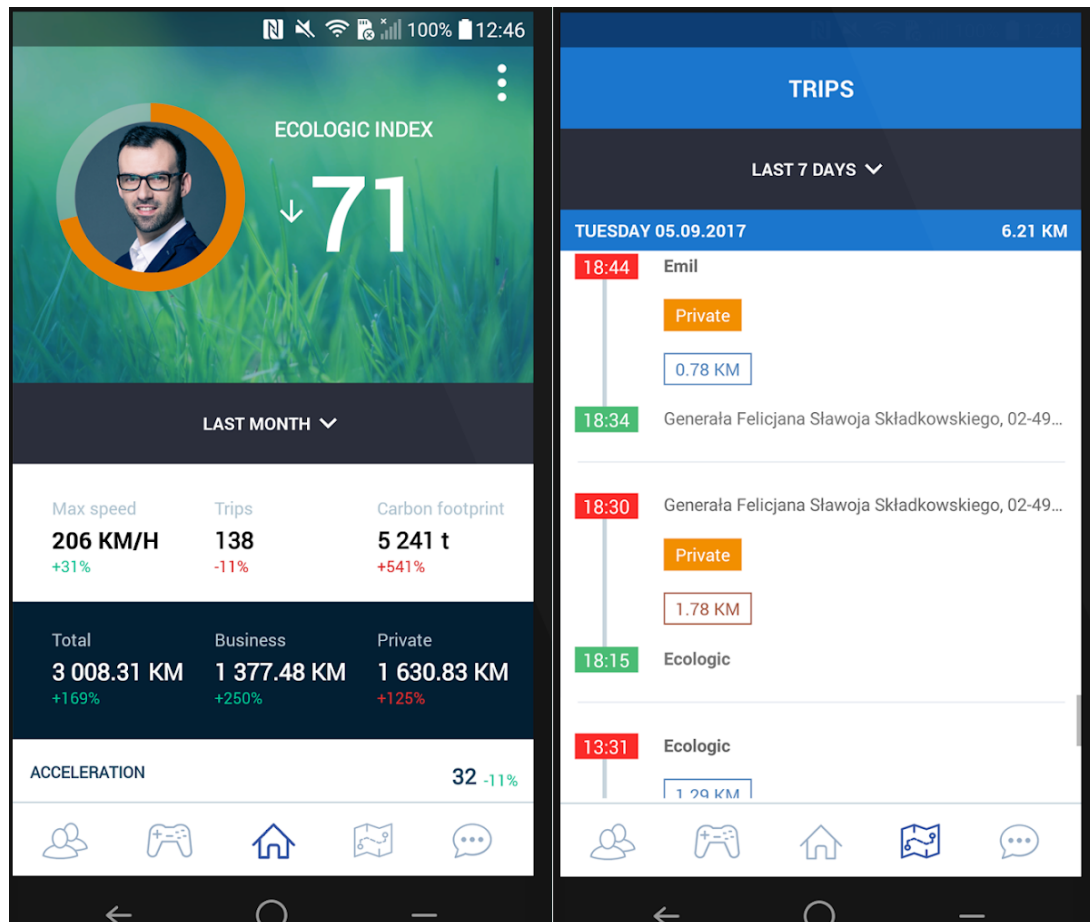


Figura 2.7: Ecrãs da aplicação Ecologic [16].

2.2.5 Torque Pro

Torque Pro [17] é uma aplicação paga disponível no *Google Play* e na *Apple Store*, disponibilizando uma grande quantidade de informação recolhida através de OBD. Inclui também alertas, gráficos e *dashboards* personalizáveis. Apesar de não ser diretamente indicada para ajudar na condução sustentável, apresenta alguns dados relevantes como a emissão de gases CO₂ e a eficiência de consumo de combustível. Existe também uma versão gratuita da aplicação, *Torque Lite* [18], possuindo inferior compatibilidade com diferentes modelos de veículos e uma menor gama de funcionalidades disponíveis [19].



Figura 2.8: Ecrãs da aplicação Torque Pro [17].

2.2.6 DashCommand

DashCommand [20], disponível para *Android* e *iOS*, é outra aplicação relacionada com OBD que recolhe uma grande quantidade de dados e os transforma em tabelas e gráficos. Em relação à condução sustentável, está presente informação sobre acelerações, travagens, economia de combustível e, possivelmente uma das funções mais distintas, um mapa que traça a *performance* de condução ao longo do percurso do veículo de acordo com um sistema simples de cores, que pode ser visualizado na última imagem da figura 2.9. Apesar de estar disponível para transferência imediata a aplicação corre apenas em modo de avaliação e para a desbloquear é necessário fazer uma compra *in-app*.

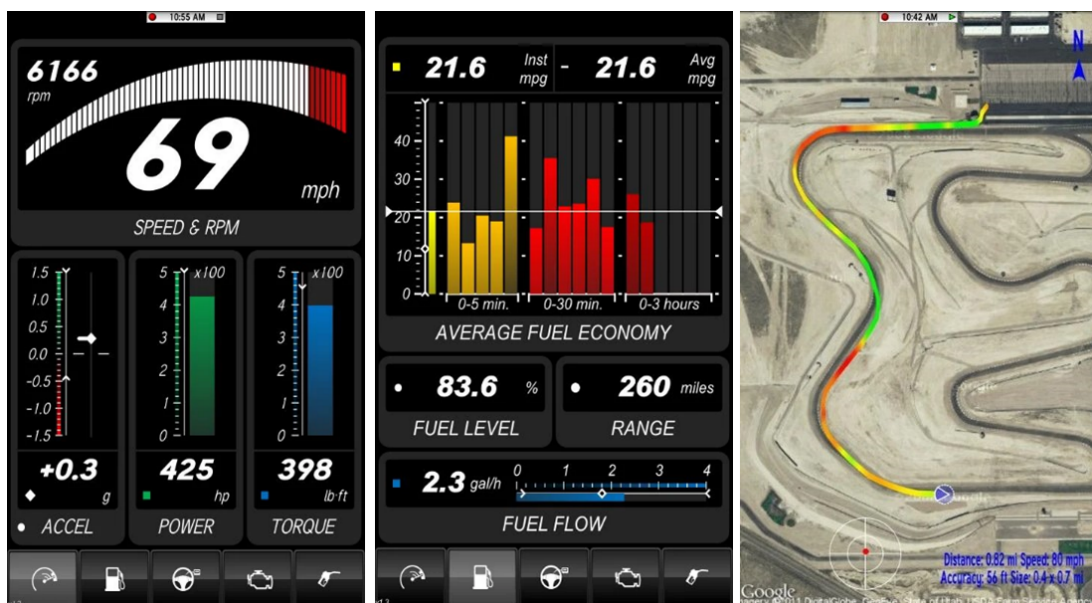


Figura 2.9: Ecrãs da aplicação Dash Command [20].

2.2.7 Comparação

Na tabela 2.1 é possível ver algumas das características chave necessárias neste projeto (colunas), e a sua presença nas aplicações analisadas (linhas).

	Dados em Tempo Real	Dados Persistentes	Aquisição de Dados do Veículo	Interface Simples/Discreta	Condução Sustentável
Car Scanner	x		x		
Speedometer	x			x	x
Drive Eco	x	x			x
Ecologic	x	x	x		x
Torque Pro	x		x		
Dash Command	x		x		x
DriveGreen	x	x	x	x	x

Tabela 2.1: Comparação das características das aplicações analisadas.

Podemos concluir que nenhuma destas aplicações cumpre todos os requisitos desejados para o módulo *DriveGreen*. A maior parte delas foca-se na recolha de dados através de OBD, mostrando diversos números, gráficos e texto a que o utilizador não irá aceder durante a viagem, podendo até vir a ser motivo de distração. Nem todas recolhem dados que possam ser analisados em futuras viagens ou no fim da mesma, e embora muitas tenham funcionalidades dedicadas à condução sustentável, tendem a ficar aquém do desejado.

Uma aplicação como a *ECO-Driving Speedometer* [14], embora bastante trivial, vai de encontro à ideia de uma aplicação simples e discreta que mostra dados sobre condução sustentável, apesar de não utilizar dados provenientes do próprio veículo mas sim do dispositivo móvel. Já uma aplicação mais complexa como a *Ecologic* [16], embora apresente dados relevantes sobre condução sustentável, provenientes dos sensores do veículo, e até histórico de informação passada, falha na perspetiva do contexto do módulo pretendido e sua complexidade, já que contém bastante informação e dados que nunca serão vistos pelo condutor durante, ou até no fim da sua viagem.

Pretende-se assim a criação de um módulo *DriveGreen* que através da recolha em tempo real de dados do veículo, os transforme e apresente ao condutor de uma forma simples, acessível e auto-explicativa, que permita não só a análise da sua eficácia de condução sustentável, mas também que através de sugestões baseadas nessa análise contribua para uma melhoria contínua dessa mesma condução.

3

Proposta de Solução

Conteúdo

3.1	Arquitetura e Design	25
3.2	Aspetos Sociais e Privacidade de Informação	33
3.3	Tecnologias e Ferramentas	34
3.4	Planeamento	37

De acordo com o objetivo de criação de um módulo de apoio à condução sustentável, *DriveGreen*, integrado na aplicação *TeamMobile* da Tecmic, são apresentados neste capítulo tópicos relacionados com a especificação, arquitetura, *design*, tecnologias e planeamento do projeto.

3.1 Arquitetura e Design

O objetivo é a criação de um módulo que utiliza os dados do veículo obtidos através da caixa *XTraN*, para os disponibilizar ao condutor em forma de contadores, gráficos e sugestões, guardando ainda esta informação na base de dados da aplicação. Ao longo desta secção são apresentados alguns detalhes relativamente à transmissão de dados, indicadores produzidos e elaboração dos ecrãs do módulo.

3.1.1 Dados Recolhidos e ECSM

Os dados do veículo obtidos através da caixa *XTraN* são primeiro analisados pelo módulo Energy Comfort and Security Module (ECSM) presente na mesma, que os transforma em indicadores úteis, alguns relacionados com o tema da condução sustentável. Da lista facultada com mais de 100 indicadores, são mencionados alguns daqueles que serão úteis para este projeto:

- Gerais: Consumo de combustível, distância total percorrida e tempo total em movimento;
- Contadores: Tempos e distâncias totais em movimento, em marcha de inércia, em *ralenti*, em excesso de rotações, em excesso de velocidade e com velocidade constante;
- Excessos: Número total de excessos de velocidade, aceleração, travagem, rotações do motor, rotações em *ralenti*;
- Categorias: Tempos em aceleração, travagem e velocidade contidos em escalões predefinidos (utilizados para o desenho dos gráficos).

Estes indicadores serão usados: (a) de forma direta para apresentação no ecrã; (b) de forma indireta, usando os seus valores por exemplo para representação dos gráficos; (c) e como base para gerar novos indicadores, como a pontuação da viagem e as sugestões.

3.1.2 Apresentação da Informação

Um dos principais fatores a ter em conta no desenho dos ecrãs deste módulo tem que ver com o contexto da aplicação *TeamMobile* e com o seu tipo de utilizador. Dado que este se encontrará grande parte do tempo a conduzir é essencial que este módulo seja discreto e não intrusivo, para que o utilizador possa concentrar-se na sua condução e nas tarefas que têm a realizar. Por outro lado, o tipo

de utilizadores que usa esta aplicação pode não possuir um vasto conhecimento tecnológico, nem ter tempo para investigar a fundo ecrãs muito complexos. Para tal é preciso que os ecrãs sejam simples, com pouca interação, e fáceis de perceber, disponibilizando ajuda se necessário.

A informação será então apresentada de duas maneiras: presente em segundo plano nos vários ecrãs da aplicação, de forma mais discreta para não retirar ao condutor o foco do seu trabalho diário, e de forma detalhada num ecrã à parte, contendo uma maior quantidade de dados, mais complexos, relativos à viagem terminada. Antes de passarmos à descrição destes dois tipos de ecrãs, seria útil apresentar em maior detalhe um dos elementos mais úteis e possivelmente mais complexos que estarão presentes na aplicação, os histogramas.

3.1.2.1 Histogramas

De entre todos os indicadores a apresentar, os histogramas serão os mais complexos, mas serão também certamente os mais úteis e visualmente fáceis de perceber. Estes histogramas são gráficos coloridos que apresentam informação referente aos valores de um atributo específico como a velocidade, travagem, aceleração ou velocidade constante, numa determinada janela temporal.

O tipo de gráfico aqui considerado tem semelhanças com outras três categorias de gráficos existentes: (a) Gráficos de barras; (b) Histogramas; (c) e *Gauges* ou Medidores Radiais. O gráfico pretendido é essencialmente um gráfico de barras, disposto na horizontal, com uma só barra, e com várias categorias. Cada barra (gráfico) corresponde a uma só propriedade como velocidade, aceleração e travagem, e cada categoria corresponde a um intervalo de valores dentro dessa propriedade, representada por uma cor que transmite a avaliação do desempenho do condutor (fraco, médio, bom).

Naturalmente, apresenta também semelhanças com um histograma, gráfico que representa a distribuição de uma variável em múltiplas categorias divididas em intervalos. O gráfico que estamos à procura junta as várias barras de um histograma numa única barra, atribuindo uma cor diferente a cada categoria, e desenhando o comprimento das várias secções em proporção ao total.

Por outro lado, apresenta também algumas semelhanças com *gauges* ou medidores radiais, um gráfico que engloba valores numa dada escala, dividida em várias categorias e que vai aumentando ou diminuindo de intensidade consoante pretendido, de maneira similar por exemplo aos velocímetros presentes num automóvel.

Contudo este último tipo de gráficos, embora visualmente semelhante, não se adequa ao propósito funcional desejado, de maneira que tendo em vista a sua implementação técnica na secção 4.2.1.1 este gráfico é equiparado a um gráfico de barras com múltiplas categorias, ou *Stacked Bar Chart*, sendo no entanto designado por *Histograma* ao longo de todo este projeto, por uma questão de simplicidade.

O módulo irá apresentar quatro gráficos, de forma similar ao encontrado no módulo *Ecodriver* (secção 2.1.3), relacionados com os seguintes atributos: (a) Velocidade instantânea; (b) Aceleração;

(c) Travagem; (d) Velocidade constante.

O gráfico de velocidade instantânea será provavelmente o mais fácil de perceber: para o construir dividimos os diferentes valores possíveis de velocidade do veículo em intervalos correspondentes a velocidades inadequadas (cor vermelha), pouco adequadas (cor amarela), e adequadas (cor verde). Imaginemos por exemplo um veículo ligeiro numa auto-estrada, com limite de velocidade de 120 Km/h. Se circular a 90 Km/h ou 110 Km/h a velocidade seria adequada. No entanto, se circular a 130 Km/h ou 50 Km/h já seria pouco adequada, e no caso de valores como 150 Km/h ou 30 Km/h, seria inadequada. Estes valores são apenas a título de exemplo, até porque a responsabilidade de definição destas categorias está a cargo do já referido módulo *ECSM*, presente na caixa *XTraN*, e não do módulo *DriveGreen*.

Consideremos agora o gráfico gerado a partir dos dados de condução dos últimos cinco minutos de viagem, presente na figura 3.1. Ao analisar este gráfico, o condutor pode concluir que passou aproximadamente metade do tempo analisado (dois minutos e meio) a praticar valores de velocidade inadequados (cor vermelha) e pouco adequados (cor amarela), e a outra metade já com valores de velocidade considerados adequados (cor verde).

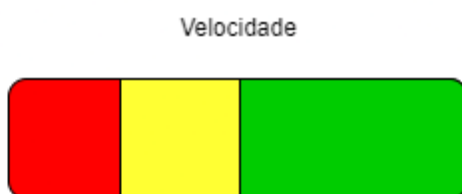


Figura 3.1: Exemplo de um histograma de velocidade instantânea a ser apresentado no módulo *DriveGreen*.

Os gráficos de aceleração e travagem são em tudo similares ao de velocidade instantânea, apresentando várias categorias associadas ao mesmo esquema de cores, consoante o comportamento praticado pelo condutor no período analisado. Já o gráfico de velocidade constante apresenta um aspeto ligeiramente diferente, isto porque os valores correspondentes a este comportamento pertencem a apenas duas categorias, a presença ou ausência de velocidade constante. Isto significa que com base na avaliação do módulo *ECSM* para cumprimento deste critério são gerados apenas dois valores, traduzidos no gráfico com uma cor verde e uma cor transparente ou cinzenta, como é possível ver na figura 3.2.

Velocidade Constante



Figura 3.2: Exemplo de um histograma de velocidade constante a ser apresentado no módulo *DriveGreen*.

Através deste modelo simples e intuitivo, será fácil para o condutor olhar de relance para estes gráficos presentes em segundo plano e assim ajustar a sua condução em relação ao atributo medido.

Segue-se agora a descrição das duas maneiras de apresentação da informação já mencionadas.

3.1.2.2 Indicadores em Segundo Plano

Nos ecrãs principais das várias secções da aplicação serão mantidos pequenos indicadores que avaliam o desempenho nos últimos minutos de condução, de maneira bastante similar à encontrada no módulo *Ecodriver*. Deste modo, estarão presentes ao longo de uma barra na parte inferior do ecrã pequenos histogramas representativos da velocidade instantânea, velocidade constante, aceleração e travagem. Considera-se que estes são os indicadores mais úteis que através do uso de um sistema de cores simples e intuitivo ajudarão o condutor a corrigir o seu comportamento e verificar o seu impacto em tempo real. Nos protótipos em papel o sistema de cores está substituído por diferentes padrões. Estes indicadores são puramente informativos e não é possível interagir com os mesmos. Na figura 3.3 é possível ver um protótipo destes indicadores num dos ecrãs da aplicação. A parte central do ecrã encontra-se omitida já que a única secção relacionada com o módulo a desenvolver é a barra inferior.

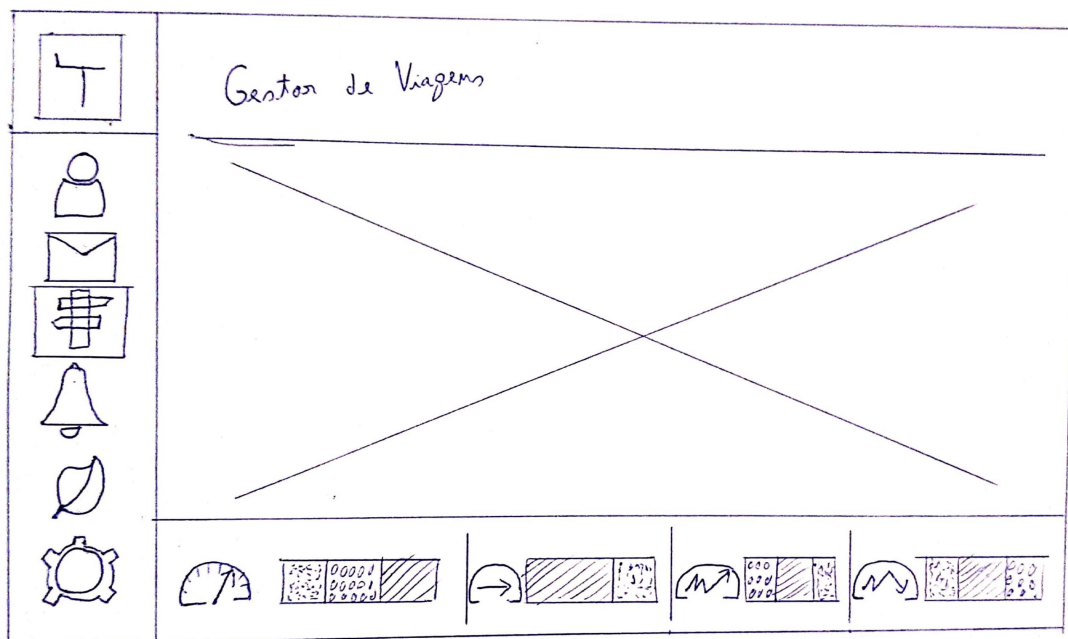


Figura 3.3: Protótipo em papel da barra inferior (ecrã secundário) do DriveGreen.

3.1.2.3 Ecrã de Resumo da Viagem

Este ecrã estará disponível no fim de cada viagem mas também em qualquer outro momento caso seja pretendido pelo condutor. Apresenta dados básicos da viagem como a duração, distância e consumo de combustível, contadores de excessos cometidos, histogramas semelhantes aos encontrados anteriormente em segundo plano, e sugestões de como melhorar o desempenho do condutor com base nos dados dessa viagem. O ecrã faz uso de vários ícones cujo significado pode ser explicado carregando no botão de ajuda no canto superior direito. Na figura 3.4 é possível ver um protótipo do ecrã descrito.

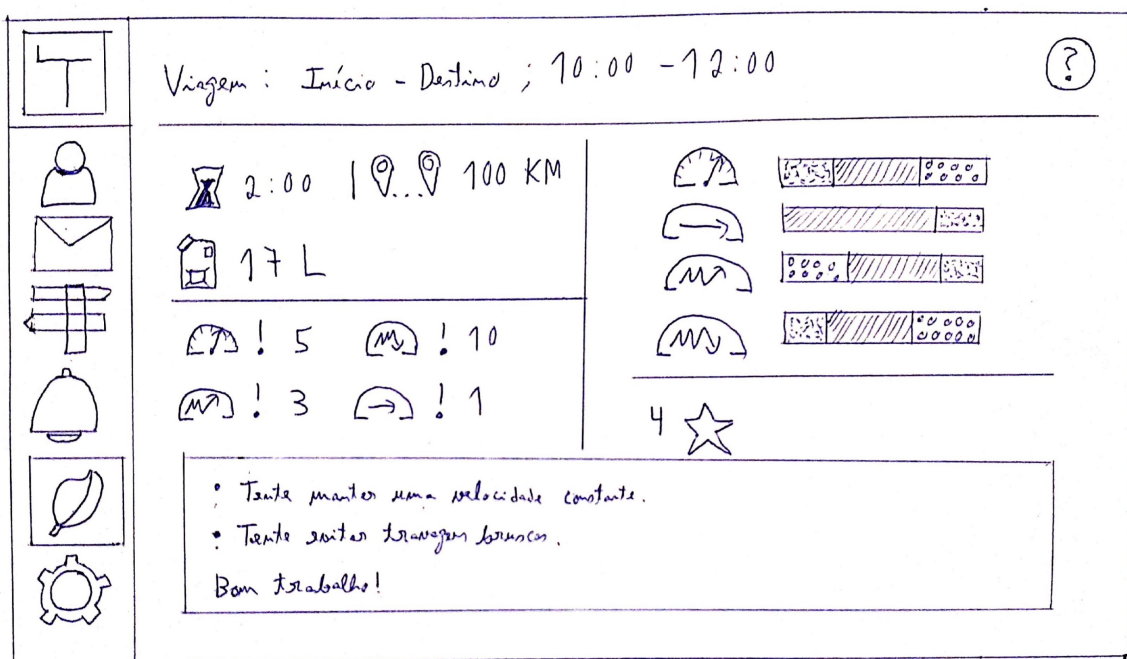


Figura 3.4: Protótipo em papel do ecrã detalhado do *DriveGreen*.

Através da ligação à base de dados da aplicação será possível ver neste ecrã não apenas as estatísticas da última viagem, mas também de uma qualquer viagem dos últimos sete dias ou até uma combinação de dados de várias viagens.

Uma outra funcionalidade para este ecrã que poderá ser implementada no futuro, é que o condutor possa aceder a um mapa para visualizar o seu desempenho ao longo das várias secções do trajeto percorrido, de maneira similar à encontrada na última imagem da figura 2.9 da aplicação *DashCommand*.

3.1.3 Protocolo de Comunicação

No contexto deste projeto, a transmissão de dados entre a caixa equipada *XTraN*, o módulo *DriveGreen*, e a base de dados central da empresa será efetuada de duas maneiras distintas:

- (A) Diretamente entre a caixa *XTraN* e o módulo *DriveGreen* através de um protocolo definido entre a caixa equipada do veículo e a consola de bordo do condutor. Esta ligação é feita através de um canal em série que transmite a informação necessária em *bytes* entre as duas partes;
- (B) Da caixa *XTraN* para a base de dados central da Tecmic, e daí para o módulo *DriveGreen* através de um novo *WebService*.

A figura 3.5 apresenta o diagrama da arquitetura destas duas abordagens.

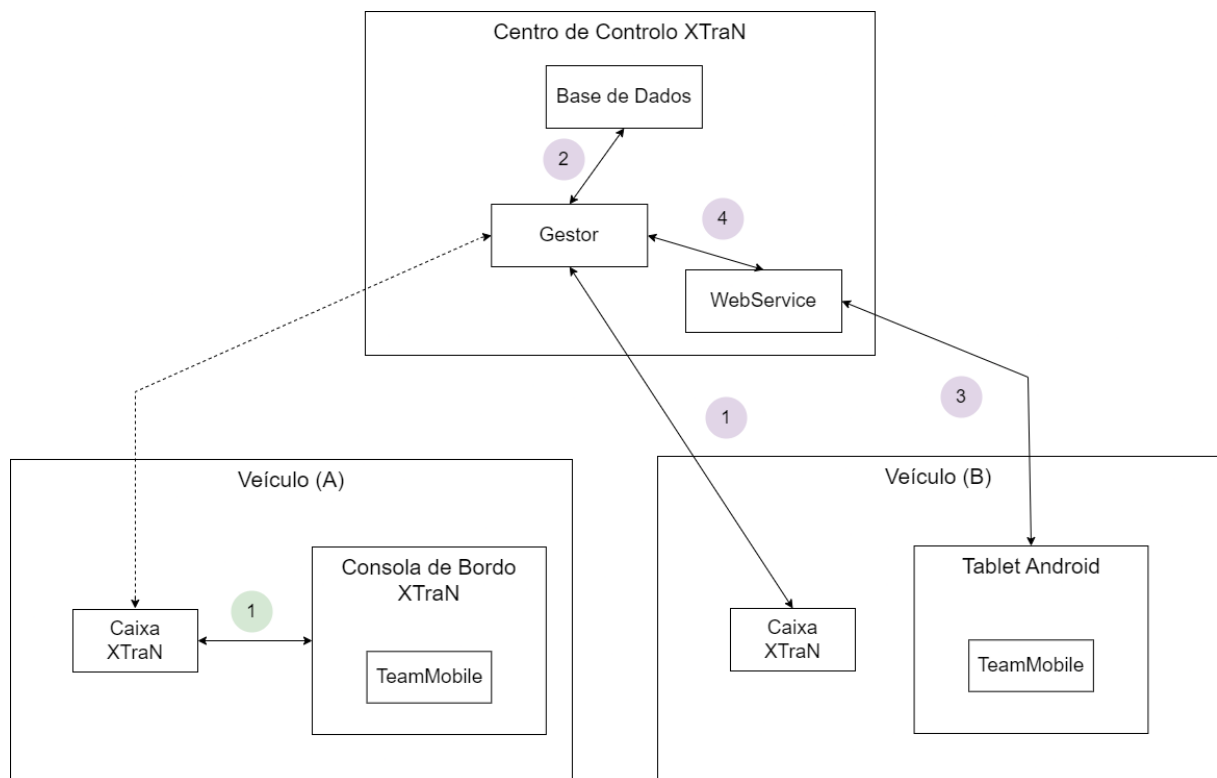


Figura 3.5: Arquitetura de comunicação entre caixa, aplicação, *WebService* e base de dados central. As ligações a verde correspondem à comunicação direta do módulo com a caixa (A), e as ligações violeta à comunicação através de um *WebService* à base de dados central (B).

As duas abordagens são mutuamente exclusivas, sendo utilizadas em diferentes contextos. A primeira (A) requer a existência de uma consola de bordo *XTraN* presente no veículo, onde está por sua vez hospedada num sistema operativo *Android* a aplicação *TeamMobile*. É assim possível estabelecer um protocolo de comunicação que permite a transmissão direta de dados entre a caixa *XTraN* e a aplicação. Isto faz com que todos os indicadores necessários para o módulo *DriveGreen* estejam

acessíveis de forma fácil e rápida para serem atualizados, por exemplo de minuto a minuto, no caso dos histogramas em segundo plano, ou no fim da viagem no caso do ecrã de resumo. Esta abordagem utiliza apenas uma ligação (1, verde) em ambos os sentidos, que será estudada mais em detalhe no capítulo 4. Existe também uma ligação da caixa ao centro de controlo, presente em ambas as abordagens, mas só relevante para a segunda.

Já na segunda abordagem (B) a aplicação *TeamMobile* presente num comum tablet *Android* não tem acesso direto aos dados obtidos pela caixa equipada do veículo. A caixa utiliza assim a ligação ao centro de controlo (1, violeta) para enviar para a base de dados central todos os dados necessários no contexto da aplicação, incluindo os indicadores que o módulo *DriveGreen* precisa. Esta ligação é feita através de um canal Transmission Control Protocol (TCP) para um componente designado de *Gestor* presente no servidor, que coloca a informação em base de dados (ligação 2, violeta). Para lhes poder aceder, é necessário a criação de um novo *WebService* que sirva de intermediário entre a aplicação e a base de dados central, mais particularmente, entre a aplicação e o *Gestor*, visto que este é o mediador do acesso à base de dados. Deste modo, para receber tanto os dados em segundo plano, como os de resumo das viagens, são enviados pedidos à Application Programming Interface (API) do novo *WebService* (ligação 3, violeta) para que este, auxiliado pelo *Gestor*, obtenha e responda com os dados necessários à medida que ficam disponíveis na base de dados central (ligações 4 e 2, violeta).

3.1.4 Base de Dados

Como já foi referido, o módulo *DriveGreen* irá guardar os seus dados numa base de dados *SQLite* da aplicação, de modo a que possa aceder aos mesmos a qualquer momento, para visualizar os relatórios de condução sustentável mais recentes no ecrã de informação detalhada.

Na figura 3.6 é possível ver o diagrama da estrutura da base de dados.

À base de dados existente na aplicação será assim adicionada uma tabela principal de relatórios (*ReportData*) onde se encontrará toda a informação dos mesmos. Essa informação inclui um ID próprio auto-gerado, o ID do condutor do veículo, os códigos (ID's) da viagem e ainda a data de início da viagem, sendo estes três últimos atributos chaves estrangeiras, associados a valores já existentes nas outras tabelas da aplicação (*Person* e *Journey*). Por último, cada item da tabela tem também associado todos os valores necessários para a elaboração destes relatórios (as variáveis obtidas durante a viagem). Em termos do modelo de dados da aplicação, estes valores estarão guardados numa lista de objetos *ECSMVariableValue*, uma classe simples com um ID único associado à variável medida e o respetivo valor registado. Contudo na base de dados, devido às limitações do *SQLite*, este campo terá que ser armazenado através de um *string* serializado em *JSON*.

Desta forma a apresentação dos relatórios de um condutor fica extremamente simplificada, e é também muito fácil gerar relatórios conjuntos. Se pedirmos por exemplo todas as viagens efetuadas

por um condutor num certo dia, filtrando pelos atributos de *DriverId* e *StartDate*, basta agora somar os valores das diferentes variáveis registadas nessas viagens para consolidar toda a informação num só relatório apresentado no ecrã detalhado.

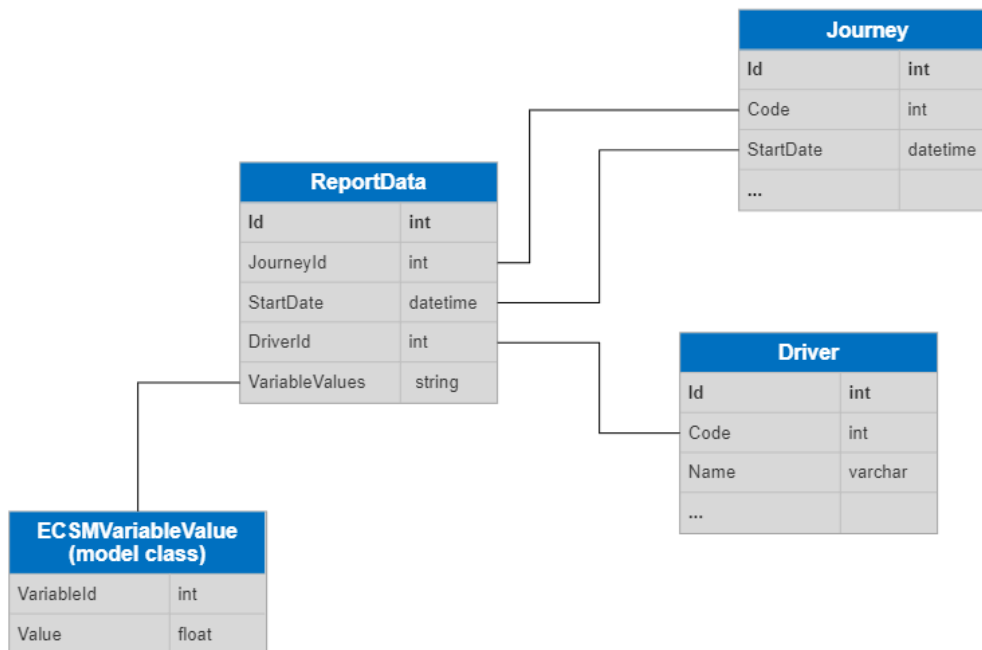


Figura 3.6: Diagrama do modelo de dados na base de dados da aplicação *TeamMobile*, no contexto do novo módulo.

3.2 Aspetos Sociais e Privacidade de Informação

Nesta curta secção são apresentadas algumas considerações sobre a esfera social e de privacidade de dados que envolve este projeto. O módulo desenvolvido utiliza, armazena e transmite os dados referentes à condução do utilizador, pelo que é preciso ter em conta que:

1. O utilizador principal da aplicação (o condutor) tem que ser informado que os dados relativos à sua condução estão a ser registados e guardados, com o seu consentimento. É de notar que nesta primeira versão da implementação do módulo, em que o seu objetivo principal passa por criar uma ferramenta de auto-ajuda para o condutor, os seus dados não estarão acessíveis de qualquer outra maneira para além de na própria aplicação. Contudo, é provável que no futuro sejam desenvolvidas funcionalidades que permitam a criação de relatórios de condução disponibilizados na plataforma *Web* da Tecmic, de forma similar ao que já acontece no *Ecodriver* (secção

- 2.1.3). Estes dados poderão ser acedidos pelos supervisores dos condutores, se requisitado, caso em que cada empresa visada, cliente da Tecmic, tem a responsabilidade de informar aos seus trabalhadores sobre a forma como os seus dados serão utilizados;
2. Os dados recolhidos no contexto do módulo são enviados pela caixa *XTraN* periodicamente para a base de dados central da Tecmic. No caso da arquitetura de transmissão por *WebService* serão ainda comunicados através desse canal para a aplicação *TeamMobile*. É assim necessário garantir que os dados estão protegidos e não são acedidos de forma maliciosa, tanto no servidor da empresa como nas comunicações intermédias;
 3. Os dados recolhidos serão apresentados no ecrã de resumo do módulo para analisar as várias viagens de acordo com critérios ligados à condução sustentável. Isto pode fazer surgir diversos sentimentos no utilizador, pode sentir-se vigiado e avaliado, e ainda elogiado ou criticado consoante os resultados visíveis. É assim importante ser cuidadoso na forma como este módulo é apresentado, desenhado e utilizado pelos condutores, de forma a garantir que não é motivo de frustração ou *stress*.

3.3 Tecnologias e Ferramentas

De forma a cumprir com a arquitetura descrita, durante o desenvolvimento deste projeto serão utilizadas as seguintes tecnologias e ferramentas:

- *Microsoft Visual Studio*, como ambiente de desenvolvimento, onde será utilizado *Xamarin* para efetuar a criação de um módulo *Android* integrado na aplicação *TeamMobile*, com linguagem C#. Com a mesma linguagem e Integrated Development Environment (IDE) será também desenvolvido o *WebService*, em *ASP.NET CORE*;
- *Protocol Buffers* da *Google* para serializar os dados enviados na troca de mensagens entre a aplicação e a caixa *XTraN* do veículo;
- Biblioteca *MvvmCross* para facilitar a implementação de uma arquitetura *MVVM* na conceção do módulo;
- *SQLite* para guardar e gerir os dados criados pelo módulo na base de dados da aplicação;
- *Microsoft Azure DevOps* para organizar todas as tarefas e *sprints*, gerir e integrar as várias versões do código e automatizar a testagem e *deployment* do projeto, e *Microsoft Teams* para efeitos de comunicação e reuniões.

Seguidamente é apresentada a plataforma *Xamarin*, base do desenvolvimento deste novo módulo.

3.3.1 Xamarin

A plataforma *Xamarin* é de entre as tecnologias referidas a mais importante no processo de desenvolvimento do projeto, e como tal, é nesta secção feita uma pequena análise sobre a mesma no contexto da aplicação *TeamMobile*.

Antes de apresentar qualquer definição é importante referir que existe uma diferença substancial entre falar de *Xamarin Nativo* e *Xamarin.Forms*. Nesta primeira parte é apresentado o *Xamarin Nativo*, designado apenas por *Xamarin*, plataforma que serve como base deste projeto e da aplicação *TeamMobile* hoje em dia. Numa segunda parte é também brevemente apresentado o *Xamarin.Forms*, através das suas principais diferenças com o primeiro, devido à sua relevância no contexto da plataforma *Xamarin* como um todo, e pela sua presença noutras aplicações móveis da Tecmic.

3.3.1.1 Xamarin Nativo

Xamarin é uma plataforma *open-source* pertencente à *Microsoft* usada para desenvolver aplicações móveis multi-plataforma. Utiliza a linguagem *C#*, orientada a objetos, e o ambiente de desenvolvimento *Visual Studio* (quando desenvolvido em *Windows*) pertencente à mesma empresa. A sua principal característica, a criação de aplicações móveis multi-plataforma, assenta na vantagem de todo o código da aplicação correspondente ao *backend* poder ser escrito numa única versão, sendo apenas necessário separar o código relativo à interface do utilizador, por cada plataforma desejada [21, 22].

Presentemente, a aplicação *TeamMobile* utiliza *Xamarin Nativo*, e neste caso concreto apenas *Xamarin.Android*, dado que a aplicação só está disponível para este sistema operativo. Desta forma, a *TeamMobile* não faz proveito deste principal benefício da plataforma *Xamarin*, mas no caso de uma eventual disponibilização para *iOS* terá o seu desenvolvimento facilitado, tanto através da incorporação de um projeto *Xamarin.iOS*, ou possivelmente através da sua migração para *Xamarin.Forms*.

3.3.1.2 Xamarin.Forms

Xamarin.Forms é outra plataforma *Xamarin* com ainda mais ênfase no paradigma de partilha de código entre as plataformas destinadas à aplicação. Isto significa que em vez de partilhar apenas o código de *backend*, com o *Xamarin.Forms* é também possível partilhar o código relativo à interface da aplicação, o que torna o processo de desenvolvimento muito mais célere, sendo possível a partilha de até 90% a 95% de todo o código da aplicação [23]. *Xamarin.Forms* utiliza assim uma linguagem única para definição da interface, Extensible Application Markup Language (*XAML*), que embora torne todo o processo mais rápido, é relatado apresentar menor *performance* em ecrãs mais complexos. Esta metodologia bem como a diferença em comparação com *Xamarin Nativo* pode ser vista na figura 3.7.

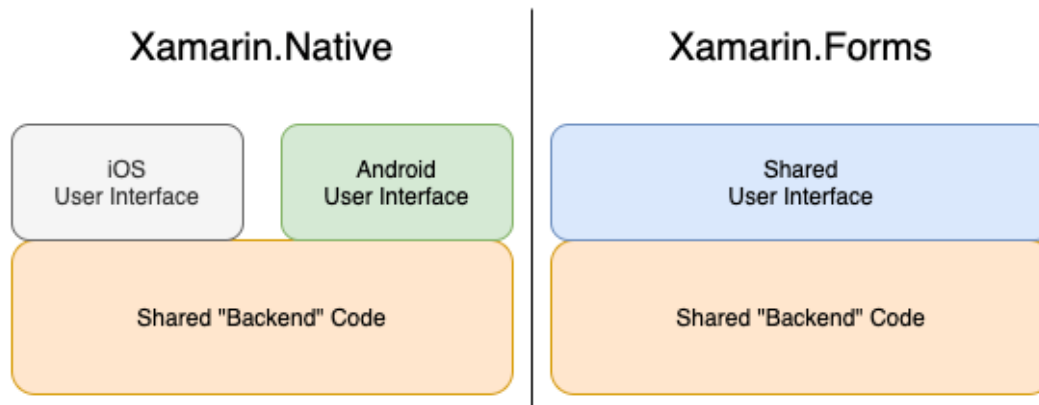


Figura 3.7: Comparação entre os paradigmas do *Xamarin Native* e *Xamarin.Forms* [24].

Xamarin.Forms não é desta maneira um *upgrade* ao *Xamarin Native*, são utilizados por razões diferentes, sendo que *Xamarin Native* é preferível para aplicações mais complexas a nível de *backend* com funcionalidades mais avançadas, enquanto que *Xamarin.Forms* é superior para o desenvolvimento rápido de aplicações mais simples sem muito trabalho em segundo plano.

3.3.1.3 MvvmCross

MvvmCross é uma biblioteca que permite a integração de uma arquitetura MVVM no desenvolvimento de aplicações multi-plataforma *Xamarin*. MVVM é uma arquitetura que separa as responsabilidades da aplicação por vários componentes, o *Model* que contém a lógica da camada de negócio da aplicação (*business layer*), a *View* que é a interface onde são apresentados os dados, e o *View-Model* que liga estas duas camadas, definindo a lógica de operação e transmissão dos dados [25].

Esta biblioteca, presente na aplicação *TeamMobile*, tem como principais características [26]:

1. Desenvolvimento de uma arquitetura MVVM, como já referido;
2. *Binding* de dados, que permite ligar valores em constante atualização no *View-Model* a propriedades da User Interface (UI) como por exemplo texto, cor ou visibilidade. Pode ainda ser feito uso de conversores e/ou combinadores de valores que adicionam uma maior flexibilidade ao tipo de dados utilizados e à maneira como são vistos pelo utilizador;
3. Inversion of Control (IoC) e Dependency Injection (DI), termos relacionados com a capacidade de injetar objetos dependentes da classe na mesma, normalmente através de uma *framework*, em vez de serem criados pela própria;
4. Ferramentas de implementação de testes unitários, entre outras.

A presença desta biblioteca na aplicação muda substancialmente a maneira como o código é desenvolvido, apresentando diferenças nas três camadas da arquitetura, desde versões específicas da biblioteca em elementos da *View*, como por exemplo *RecyclerViews* próprias, ao *binding* de dados definidos no *View-Model*, à injeção de dependências em classes do *Model*.

3.4 Planeamento

Tendo como base a prévia análise e especificação teórica, a fase de desenvolvimento prático apresentada no seguinte capítulo 4 foi planeada com as seguintes etapas:

- Prototipagem em computador e desenho em *Android* dos ecrãs do módulo;
- Estudo do funcionamento técnico da aplicação *TeamMobile* e do *Xamarin*;
- Estudo dos protocolos envolvidos na comunicação entre a aplicação, caixa equipada e centro de controlo, incluindo o desenvolvimento do novo *WebService*;
- Desenvolvimento e testagem do módulo *DriveGreen*;
- Obtenção e avaliação de *feedback* do módulo por parte dos utilizadores, numa versão *Beta* da aplicação;
- *Deployment* da primeira versão oficial do módulo integrado na aplicação *TeamMobile*.

4

Desenvolvimento

Conteúdo

4.1	Gestão e Metodologia	41
4.2	Etapas de Desenvolvimento	43

Neste capítulo são apresentadas em detalhe as diferentes fases de desenvolvimento do projeto bem como a gestão aplicada ao longo do processo.

4.1 Gestão e Metodologia

Esta secção detalha brevemente a gestão do trabalho realizado ao longo do desenvolvimento deste projeto, como o planeamento e registo de tarefas e comunicação.

4.1.1 Gestão de Tarefas

Todo o desenvolvimento foi registado na plataforma *Microsoft Azure DevOps* da empresa. Nela são criadas semanalmente tarefas correspondentes a cada etapa, às quais são atribuídas as respetivas horas gastas diariamente. Na figura 4.1 é possível ver as tarefas desenvolvidas durante um dos *sprints* semanais.

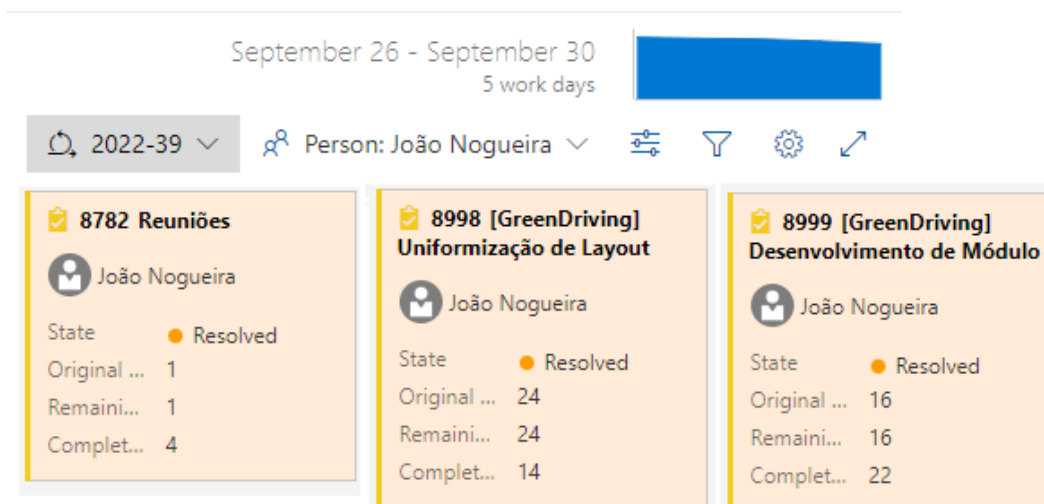


Figura 4.1: Vista de tarefas semanais (*sprint*) na plataforma *Microsoft Azure DevOps* da Tecmic.

4.1.2 Comunicação

Diariamente são realizadas reuniões com a equipa da unidade de desenvolvimento Utilities, Logística, Distribuição e Ambiente (ULDA) da Tecmic através do *Microsoft Teams*. Nestas reuniões cada elemento descreve brevemente o dia de trabalho anterior e o plano para esse dia, mencionando também problemas existentes que o possam impedir de avançar. Para além destas foram também realizadas múltiplas outras reuniões de modo a definir as diferentes fases do projeto bem como esclarecer dúvidas quanto às mesmas.

4.1.3 Ambiente de Trabalho

Durante a maior parte do processo de desenvolvimento, o projeto foi realizado de forma remota, através da ligação por Virtual Private Network (VPN) ao computador presente nas instalações da empresa em Lisboa. Contudo, já em Junho foi desencadeado um processo de forma a dar início a um estágio profissional nesta empresa, que acabou por começar apenas em Setembro de 2022, a ser praticado presencialmente a partir das instalações de Leiria.

4.1.4 Calendário

Foi criado um gráfico de *Gantt* com a representação do tempo gasto nas várias etapas de desenvolvimento deste projeto. A primeira etapa consistiu num estudo inicial do *Xamarin*, através do desenvolvimento de uma *Spike Solution* para implementação dos histogramas, e também no desenho em computador dos ecrãs da aplicação (secção 4.2.1). A segunda etapa incidiu no estudo da aplicação *TeamMobile* e integração e desenvolvimento do módulo (secção 4.2.2). As duas restantes etapas, efetuadas em várias partes e intercaladas entre si, foram a tarefa de uniformização dos ecrãs da aplicação *TeamMobile* (secção 4.2.4), e o desenvolvimento do protocolo de comunicação entre o módulo, caixa *XTraN*, *WebService* e base de dados central (secção 4.2.3). Este gráfico pode ser visto na figura 4.2.

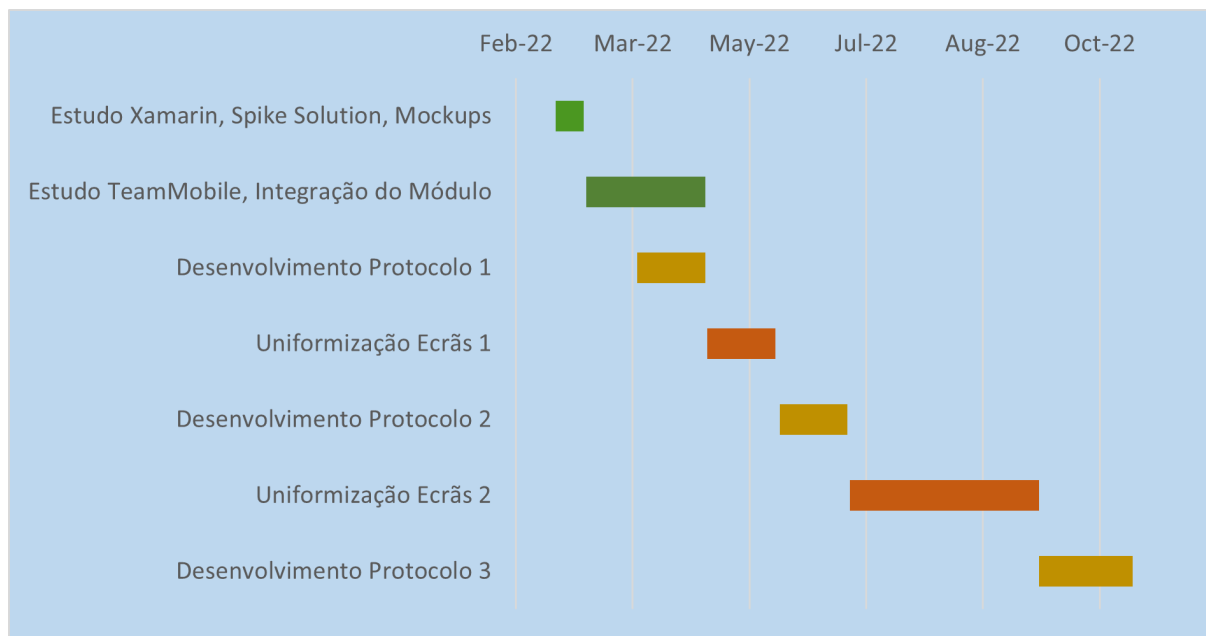


Figura 4.2: Gráfico de *Gantt* das etapas de desenvolvimento do projeto.

É de notar neste gráfico que a etapa de uniformização dos ecrãs da *TeamMobile* foi a que demorou mais tempo em todo o processo. Este desenvolvimento não estava planeado inicialmente no âmbito

do projeto (secção 3.4), mas é extremamente relevante no contexto da aplicação, e naturalmente, do trabalho desenvolvido durante este percurso.

4.2 Etapas de Desenvolvimento

Nesta secção são descritas as várias etapas de desenvolvimento apresentadas anteriormente.

4.2.1 Desenho dos Ecrãs

Seguindo os protótipos desenhados à mão apresentados na secção 3.1.2, e tendo em conta o *layout* da aplicação *TeamMobile*, foram criadas versões em computador destes ecrãs através da plataforma *Moqups* [27] que podem ser vistos nas figuras 4.3 e 4.4.

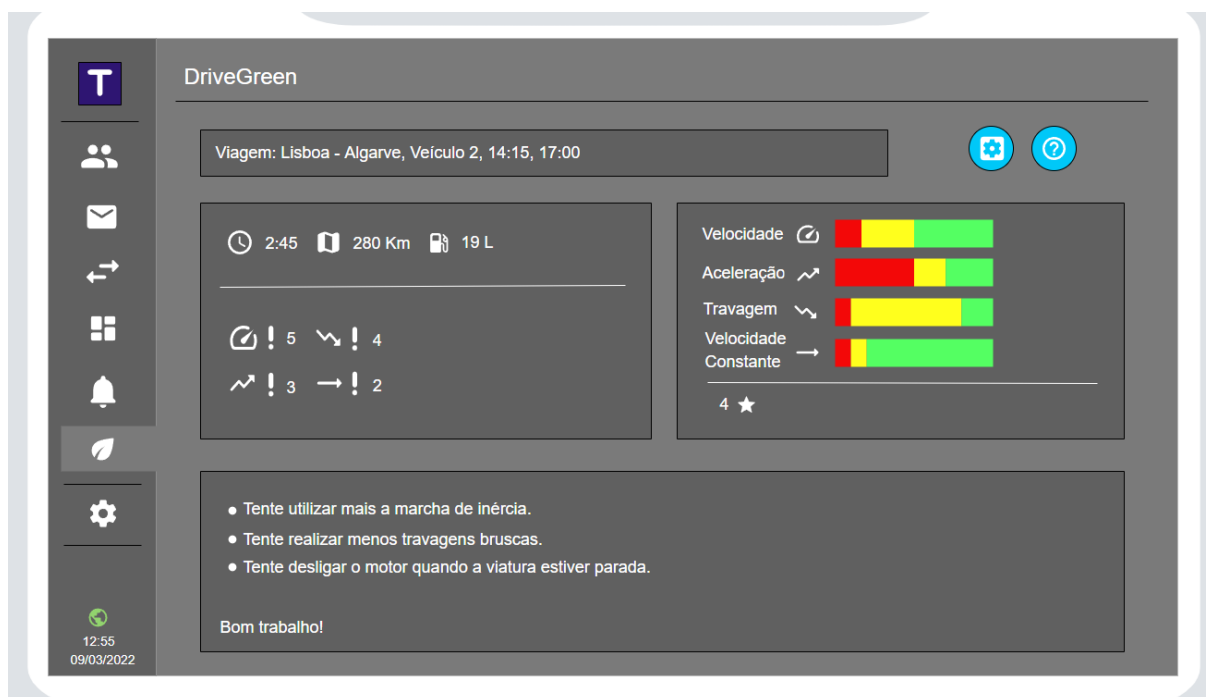


Figura 4.3: Protótipo em computador do ecrã principal do novo módulo.



Figura 4.4: Protótipo em computador do ecrã secundário (barra com gráficos em segundo plano) do novo módulo.

De entre os vários elementos apresentados o único que poderia trazer desafios para a sua representação era o gráfico colorido nomeado de histograma. Nesse sentido o primeiro desenvolvimento prático em *Xamarin* foi a criação de uma *Spike Solution*, isto é um projeto simples, à parte da aplicação principal, onde foi estudada a melhor maneira de realizar esta implementação.

4.2.1.1 Histogramas

De acordo com o explicado na secção 3.1.2.1 a principal *keyword* de pesquisa utilizada para este estudo foi a de *Stacked Bar Chart*. No entanto, chegar a uma solução final não foi tarefa simples, por várias razões. Logo à partida, por ser um tipo de gráfico de barras um pouco fora do normal, um gráfico com múltiplas categorias na mesma barra. Para além disso existiam várias restrições, estar disponível para *Xamarin Native*, e ser possível desenhá-lo com uma só barra, horizontal, com várias cores correspondente a cada categoria, e sem quaisquer eixos, valores, títulos ou legendas, algo que não costuma ser o objetivo das bibliotecas disponibilizadas para desenho de gráficos.

Foram assim testadas diversas *libraries*, algumas pagas como a *SyncFusion* [28] e a *SciChart* [29], que apresentam vários tipos de gráficos e maneiras de os customizar ao pormenor. Foram também analisadas *libraries* gratuitas como a *OxyPlot* [30], a *Microcharts* [31] e a *MPAndroidChart* [32], sendo que apenas esta última, ao fim de muitos ajustes, apresentou o resultado desejado e acabou por ser a alternativa escolhida. O gráfico, isolado e com as mínimas margens possíveis (visíveis através da cor

de *background*), pode ser visto na figura 4.5.

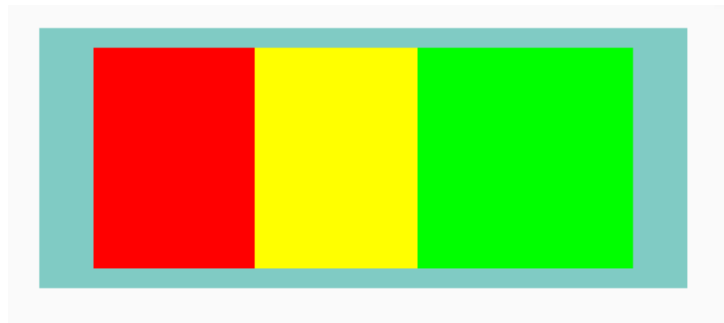


Figura 4.5: Gráfico desenhado com a biblioteca *MPAndroidChart*.

Mais tarde foi detetado um outro problema neste tipo de gráfico: ao utilizar apenas três cores sólidas para as suas categorias, poderia levar o utilizador a crer que representam intervalos bruscos dos valores medidos, quando de facto a gama de valores de cada uma pode ser considerável. Para resolver este problema era necessário aplicar gradientes às cores utilizadas, de forma a representar uma mudança contínua ao longo das mesmas, mas tal era impossível de implementar com as funcionalidades da biblioteca escolhida.

Como solução foi desenhado um pequeno algoritmo que divide cada categoria principal (cada cor) em múltiplas pequenas categorias, cada uma com uma parte do valor total, atribuindo cores ligeiramente diferentes a cada uma delas, criando o aspeto de um gradiente contínuo. O algoritmo utiliza os parâmetros *steps* para definir o número de cores diferentes em cada gradiente/categoria, e os valores iniciais e finais das cores dos gradientes a serem gerados. Por exemplo, em vez de representar a cor vermelha uma vez, associada ao valor *value*, são criados $x=steps$ tons de vermelho, cada um associado ao um valor de $y=value/steps$ partes iguais. Ao definir as cores iniciais e finais do gradiente pretendido, é utilizada uma função *GetGradients()* para gerar as várias gamas de cores, código adaptado desta publicação [33].

O resultado está presente na figura 4.6. O aspeto é facilmente ajustável alterando os parâmetros referidos: o número de *steps* e as cores iniciais e finais de cada categoria.

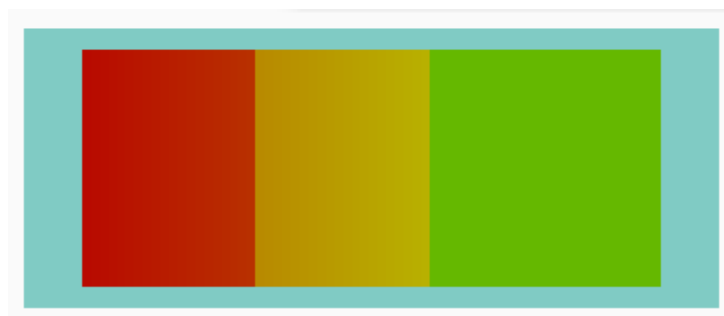


Figura 4.6: Histograma resultante da aplicação do algoritmo de criação de gradiente.

4.2.2 Integração do Módulo

Foi então começada a integração e desenvolvimento do módulo *DriveGreen* na aplicação *TeamMobile*, acompanhado de um estudo contínuo e progressivo da mesma. Nesta secção é apresentado o desenvolvimento e funcionalidades dos ecrãs, e a camada de negócio do módulo incluindo aspetos como o modelo de dados e comunicação interna dos mesmos para a base de dados da aplicação e para a interface. O protocolo de comunicação com o exterior, entenda-se, com a caixa *XTraN* e a base de dados central e *WebService* é especificado na secção 4.2.3.

4.2.2.1 Ecrãs

Após os ecrãs estarem desenhados e a biblioteca para implementação dos histogramas definida, seguiu-se o primeiro contacto com a aplicação *TeamMobile*, através da integração dos novos ecrãs. Nas figuras 4.7 e 4.8 podem ser vistas as versões iniciais destes ecrãs (barra inferior e ecrã principal) na aplicação.

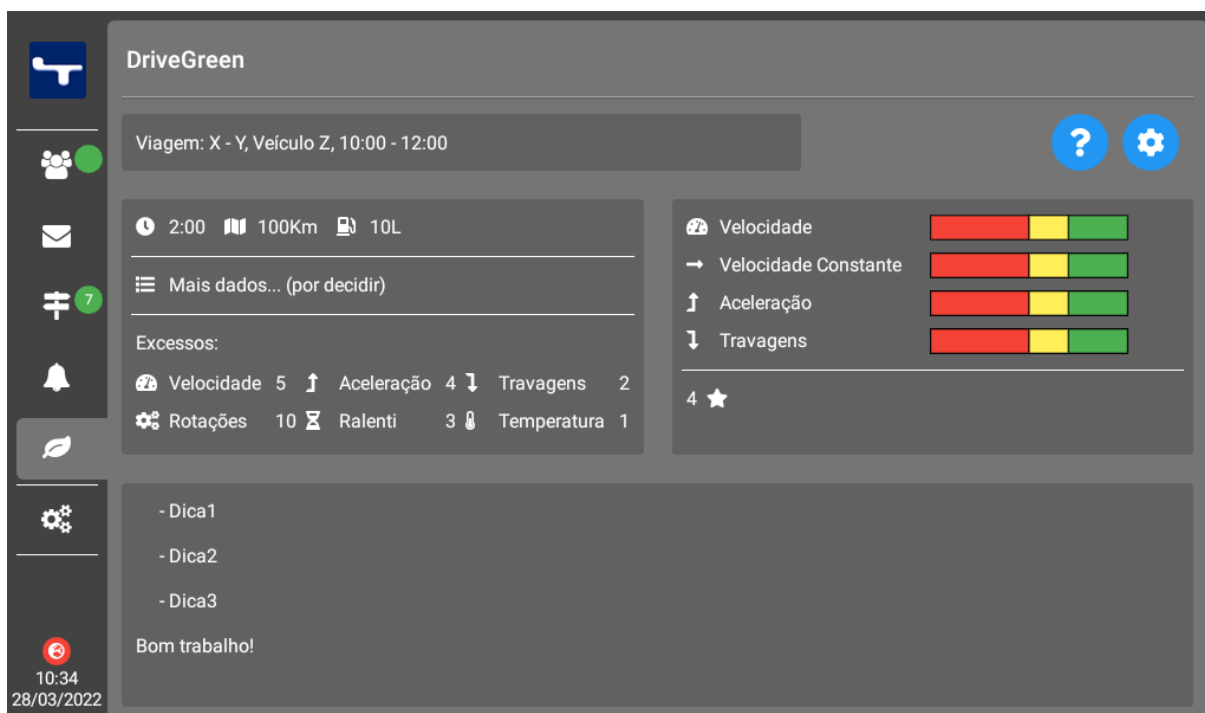


Figura 4.7: Versão inicial do ecrã principal do novo módulo na *TeamMobile*.



Figura 4.8: Versão inicial da barra inferior (ecrã secundário) do novo módulo na *TeamMobile*.

O ecrã secundário, correspondente à barra inferior presente nas várias secções da aplicação, é onde os quatro gráficos referentes à velocidade instantânea, aceleração, travagem e velocidade constante são atualizados sempre que recebidos novos dados do veículo sobre a viagem a decorrer. Em termos de implementação, este ecrã é relativamente simples, por um lado, porque não é alvo de qualquer interação por parte do utilizador, servindo apenas para consulta, e por outro pela maneira como os ecrãs da aplicação se encontravam já estruturados. Isto quer dizer que a aplicação *TeamMobile* utilizava já uma arquitetura com um par *View-Model* e *View* principais, as classes *MainViewModel* e *MainActivity*. Estas classes, em termos de *layout*, contêm os elementos base presentes em qualquer ecrã da aplicação, como por exemplo a barra lateral esquerda de navegação entre módulos. Desta maneira foi simples adaptar estas classes para incluírem a nova barra inferior do módulo, que completa este espécie de moldura onde é mostrado o conteúdo dos ecrãs. Um aspeto a ter em atenção é a necessidade de esconder esta barra em determinados contextos: quando o condutor não está em viagem, quando o ecrã apresenta formulários relacionados com os serviços prestados, ou até durante a visualização do ecrã principal do módulo.

Este ecrã não sofreu alterações significativas até ao fim do projeto, e pode ser visto na figura 4.9, no ecrã de atividade da aplicação.

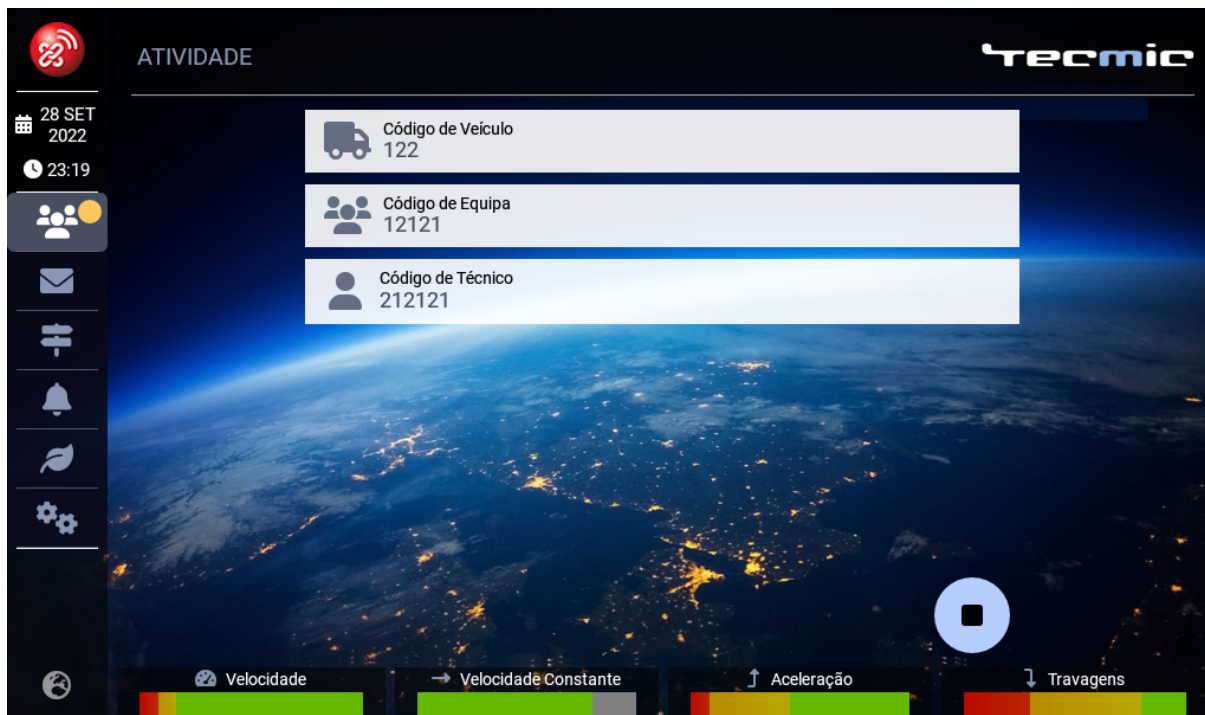


Figura 4.9: Versão final da barra inferior do módulo *DriveGreen* no ecrã de atividade da *TeamMobile*.

Já o ecrã principal foi através de um novo *View-Model* e *View*. Este ecrã, mais complexo, foi alvo de várias modificações e a última versão pode ser vista na figura 4.10, legendada para facilitar a descrição dos seus elementos. Os dados apresentados na figura são apenas a título de exemplo, não correspondendo a um caso real. Este ecrã, disponível através da barra lateral da aplicação, mostra ao utilizador os dados recolhidos e transformados durante as viagens efetuadas, através dos seguintes elementos:

1. Identificadores do condutor e da viagem;
2. Seletor que permite escolher a viagem ou conjunto de viagens apresentadas neste ecrã. Quando o utilizador acede ao ecrã é sempre apresentada a viagem mais recente, mas é possível mudar entre qualquer outra viagem do condutor registada nos últimos sete dias, bem como aceder a um resumo de todas as viagens efetuadas num determinado dia;
3. Identificador com o *timestamp* da última atualização dos dados do ecrã. Sempre que são recebidos novos dados de uma viagem, caso o utilizador esteja nesse momento neste ecrã, o ecrã é automaticamente atualizado, bem como este *timestamp*;
4. Botão de ajuda que mostra ao utilizador um pequeno guia sobre o funcionamento e objetivo do módulo *DriveGreen*, bem como o significado associado a alguns ícones;
5. Informação geral da viagem, tal como tempo e distância totais, tempo com o motor ligado durante

a condução e tempo com o motor ligado com o veículo parado (*ralenti*), e consumo e eficiência de combustível;

6. Contadores de vários tipos de excessos cometidos: limites de velocidade, limites de acelerações e travagens bruscas, limite de rotações do motor, limite de tempo em *ralenti*, e limite de temperatura do motor;
7. Histogramas de velocidade instantânea, velocidade constante, aceleração e travagem, tal como vistos na barra inferior do ecrã secundário, desta vez relativos ao total da viagem;
8. Pontuação da viagem. Este valor é calculado de forma similar à utilizada no módulo *Ecodriver* da *XTraN Passenger*. Para tal é analisado o total de excessos e com base num limiar definido de por exemplo, vinte excessos por cada cem quilómetros, é calculada a pontuação representada pelo número de estrelas amarelas no ecrã;
9. Sugestões de melhoria da condução. Sendo uma nova funcionalidade neste tipo de módulo as sugestões calculadas inteiramente pelo *DriveGreen* (para além da pontuação, que embora também calculada pelo módulo já tinha uma fórmula pré-definida), com base nos restantes dados recebidos da caixa. O algoritmo, detalhado abaixo, indica um máximo de três sugestões, geradas com base no estudo apresentado sobre os fatores com mais impacto para uma condução sustentável analisados na secção 1.1.



Figura 4.10: Versão final (legendada) do ecrã principal do módulo *DriveGreen*.

Como foi referido o algoritmo de sugestões indica um máximo de três sugestões geradas em três etapas, dado que em cada etapa pode ou não ser gerada uma sugestão. Na primeira etapa são analisados os valores de tempo total da viagem e tempo total em *ralenti*. Se este último valor for superior a um certo limite, por exemplo, ultrapassar cinco por cento do primeiro valor, será mostrada a sugestão relativa a este comportamento, como se pode ver na figura (primeira sugestão). Na segunda etapa são analisados os valores dos vários excessos cometidos, e utilizado o maior valor de entre os seis, que ultrapasse um limite pré-definido baseado na fórmula de cálculo da pontuação. No caso da figura, podemos deduzir que o limite excessivo de acelerações foi ultrapassado, e foi gerada a segunda sugestão da lista. Finalmente, de forma similar à segunda etapa, são analisados os quatro gráficos presentes no relatório, e se em alguns deles o valor correspondente a um comportamento incorreto (valor das barras vermelhas no caso da velocidade, aceleração e travagem) ultrapassar o limite definido, é gerada outra sugestão. No caso da figura, foi gerada uma terceira sugestão relativa à *performance* de travagem.

Este algoritmo, foi desenhado tendo em conta vários critérios. O limite de três sugestões assegura que o utilizador não se sente excessivamente criticado pelo módulo, e a linguagem utilizada (uso do verbo "tentar" e saudação final) visa a ideia do módulo como um instrumento de *self-improvement* do condutor. As sugestões geradas estão diretamente relacionadas com os dados visíveis no ecrã, e focam na primeira etapa um comportamento muito prejudicial mas fácil de remediar, o *ralenti*, na segunda etapa a análise do excessos cometidos, indicador relevante usado também no cálculo da pontuação, e na terceira etapa a *performance* representada nos gráficos, que inclui também um comportamento não presente nas etapas anteriores, a prática de uma velocidade constante.

4.2.2.2 Camada de Negócio

A *business layer* engloba as classes e interfaces necessárias para o funcionamento do módulo, tal como objetos do modelo de dados, eventos, serviços e controladores do fluxo de receção, interpretação, armazenamento e envio de dados.

Primeiro temos a classe *DriveGreenData*. Esta classe, cuja definição pode ser vista na listagem 4.1, é a base do modelo de dados do módulo. Dos seus atributos fazem parte um ID, os dados identificadores de uma viagem como os seus códigos e a data de início, o condutor associado, e ainda uma lista com todas as variáveis medidas que são usadas nos relatórios de resumo. Cada variável da lista é uma classe simples (também visível na listagem) com dois atributos, um ID único associado à variável em concreto, e o valor registado. Ainda relacionado com o modelo de dados existe uma classe *DriveGreenDB*, equivalente à anterior, com a exceção da sua lista de variáveis estar serializada para um *string*, dado que esta classe é armazenada na base de dados *SQLite*, que não suporta listas de objetos.

```

1 public class DriveGreenData : IDriveGreenData
2 {
3     public int Id { get; set; }
4     public int DriverId { get; set; }
5     public string JourneyId { get; set; }
6     public string SecondaryJourneyId { get; set; }
7     public DateTime JourneyStart { get; set; }
8     public List<EcsmVariableValue> VariableValues { get; set; }
9
10    //...
11 }
12
13 public class EcsmVariableValue
14 {
15     public int VariableId { get; set; }
16     public float Value { get; set; }
17
18    //...
19 }

```

Listagem 4.1: Classe DriveGreenData.

Segue-se a classe de acesso à base de dados e à nova tabela do módulo, *DriveGreenDAL*. Nas Data Access Layer (DAL) são definidos os métodos/*queries* para acesso às tabelas com base nos atributos fornecidos. No caso deste módulo são úteis métodos que acedam aos relatórios de todas as viagens de um condutor, ou a uma viagem em particular.

Existem também classes de *eventos*, um objeto de C# utilizado para enviar notificações e/ou dados para outras classes [34]. No contexto do módulo um evento é disparado quando a aplicação recebe novos dados de uma viagem e precisa de atualizar elementos no ecrã. Por exemplo, ao receber informação periódica de uma viagem em curso irá disparar um evento para atualizar os gráficos em segundo plano, e se receber novos dados enquanto o utilizador está no ecrã principal do módulo, é disparado um evento para enviar esses dados para o ecrã e o atualizar automaticamente.

Existe ainda um serviço, *DriveGreenApiService*, alvo da secção 4.2.3.2, mas dentro de todo este conjunto, o controlador é sem dúvida o elemento principal e mais complexo, responsável por coordenar a interação entre todas estas classes.

Podemos assim considerar dois principais fluxos de comunicação no que diz respeito ao funcionamento interno do módulo, o primeiro, relacionado com a receção de novos dados de viagens, que pode ser visto no diagrama da figura 4.11.

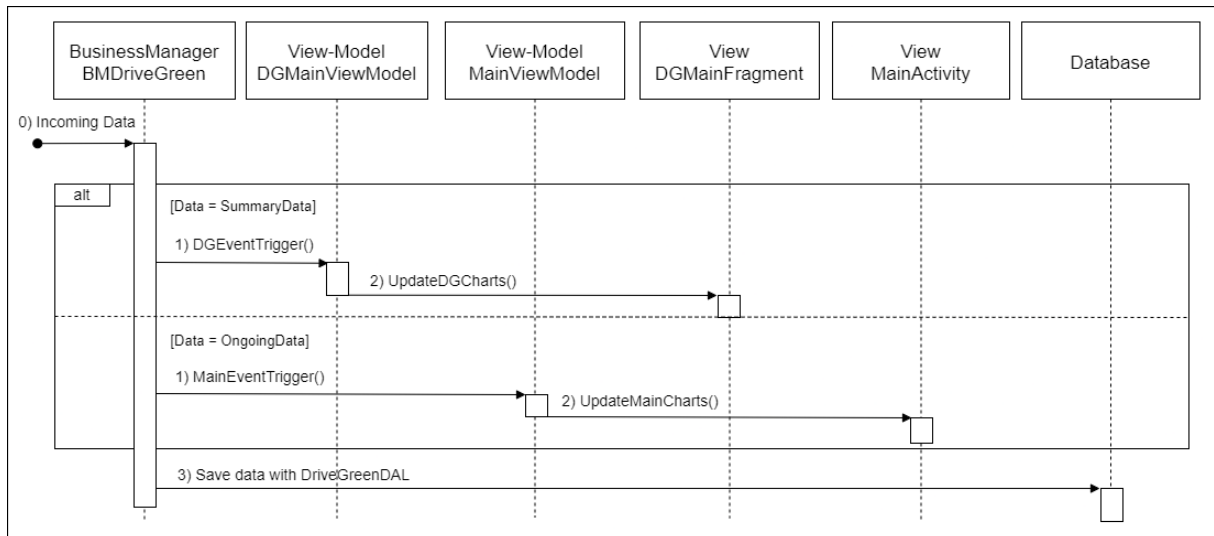


Figura 4.11: Diagrama de sequência de receção de dados no módulo.

Após o módulo receber novos dados de uma viagem (0), tanto através da caixa como do *WebService*, os mesmos são interpretados e transformados. Recordo que a maneira de receção e interpretação dos dados é estudada na secção à frente (4.2.3). De seguida, o controlador, *DriveGreenBusinessManager*, com os dados já convertidos para a classe *DriveGreenData*, envia esta informação para o *View-Model* correspondente (1). Isto é, se forem dados periódicos, com a informação dos últimos minutos de viagem, dispara o evento necessário para notificar a classe ligada ao ecrã principal da aplicação, *MainViewModel*. Por outro lado, se os dados recebidos forem de resumo da viagem e o utilizador esteja nesse momento no ecrã principal do módulo, é disparado um outro evento para a classe *DriveGreenMainViewModel*. Cada um destes *View-Models* é responsável por atualizar os ecrãs da sua *View* correspondente (2), os gráficos da barra inferior na *MainActivity*, ou os dados do ecrã principal do *DriveGreenMainFragment*.

Para além disso, o controlador guarda também os dados recebidos (periódicos ou de resumo) na base de dados, através da DAL já mencionada (3).

O outro fluxo de informação, visível na figura 4.12, acontece quando o utilizador acede ao ecrã principal do módulo (1). Neste caso o *View-Model* faz uma chamada ao método *GetDriverData()* do controlador (2), que através da DAL obtém os dados de viagens realizadas pelo utilizador ativo (3,4) e devolve essa informação ao *View-Model* (5) que atualiza a respetiva *View* (6), forçando a atualização dos dados do ecrã.

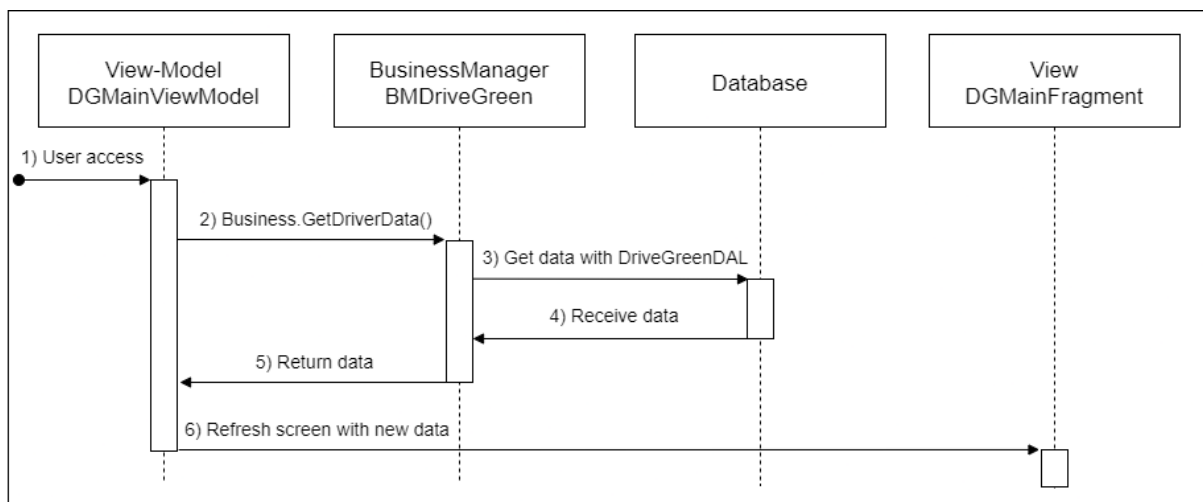


Figura 4.12: Diagrama de sequência de acesso do utilizador ao ecrã principal do módulo.

4.2.3 Protocolo de Comunicação

O próximo passo foi o estudo da transmissão de dados do veículo para o módulo *DriveGreen*. Como explicado na secção 3.1.3, este protocolo está dividido em duas partes distintas, a comunicação direta com a caixa *XTraN* através de um sistema de *Tags* quando a aplicação está presente numa consola de bordo da empresa, ou através de um *WebService* ligado à base de dados central da Tecmic, quando usada num comum tablet *Android*.

4.2.3.1 Via Tags

O sistema de comunicação através de *Tags* é usado no envio de mensagens de dados entre a caixa equipada do veículo e a aplicação instalada na consola de bordo. Este é um mecanismo simples e assíncrono, através de um canal de comunicação em série. As *Tags* são classes correspondentes a objetos e atributos normais, interpretadas de forma especial pelo serializador e formatador *XTraN*. Essencialmente, quando a aplicação é iniciada são registadas as *Tags* que pretende receber da caixa, enviando essa informação para a mesma. Após isso, e até ser desligada ou receber novas instruções, a caixa manda periodicamente informação das *Tags* escolhidas para a consola/aplicação. Estes dados são transformados em *bytes* aquando do seu envio e decodificados no recetor.

Contudo, este sistema de comunicação não estava presente hoje em dia na aplicação *TeamMobile* para *Android*, mas sim numa versão antiga da mesma que corre em *Windows CE*. Foi desta maneira necessário migrar o código antigo para esta versão da aplicação, e fazer uso de cinco *Tags*, quatro delas novas, para a comunicação do módulo *DriveGreen*. As cinco *Tags* utilizadas são as seguintes:

1. *TagEcoDynamic*, onde é enviada informação sobre o estado dinâmico, em tempo real, do veículo, neste caso os dados que permitem atualizar os histogramas presentes em segundo plano;

2. *TagEcoSummary*, com a informação de resumo da viagem, enviada no fim da mesma ou quando pedida através da *TagEcoRequest*. Esta informação é usada para preencher o ecrã principal do módulo;

3. *TagEcoEvent*, que envia para a aplicação informação sobre eventos anómalos ocorridos em tempo real, tais como excessos de velocidade, aceleração, travagens, entre outros;

4. *TagEcoRequest*, usada para pedir à caixa informação sobre a última viagem efetuada, no caso de por exemplo falhas na entrega da última *TagEcoSummary* ou de os dados disponíveis na aplicação estarem por alguma razão desatualizados;

5. *TagService*, uma *Tag* antiga recuperada para este desenvolvimento que troca com a caixa informação sobre o estado atual da viagem. Deste modo a caixa pode saber a janela de tempo de cada viagem efetuada de modo a enviar os dados relativos a cada uma.

O novo módulo está assim preparado para enviar à caixa as atualizações do estado das viagens através da *TagService*, para pedir à caixa informação completa da última viagem com a *TagEcoRequest*, ou para receber informação da caixa através das restantes três *Tags*. Esta comunicação está representada na figura 4.13.

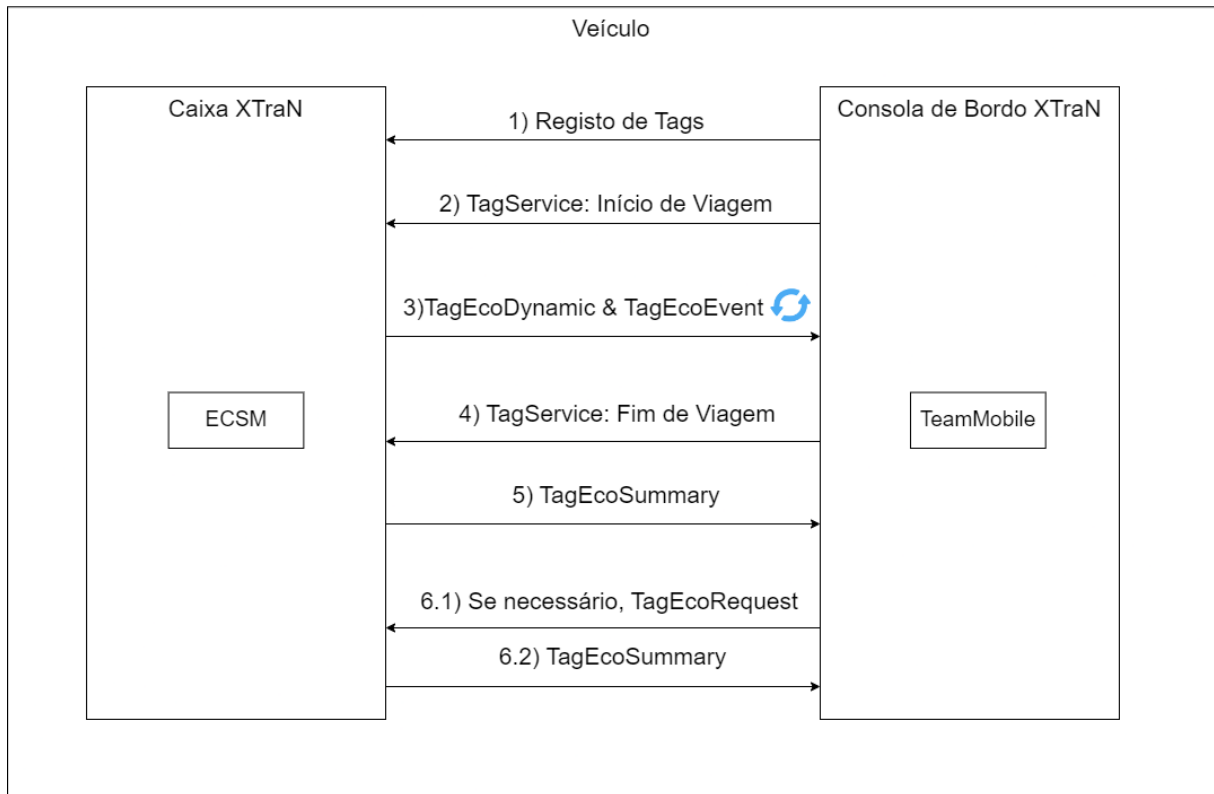


Figura 4.13: Diagrama de comunicação entre o módulo/aplicação e a caixa via *Tags*.

É de notar que neste momento o módulo não faz uso da informação disponibilizada pela *TagEcoEvent*. Esta *Tag* criada no contexto do módulo *Ecodriver* (secção 2.1.3) serve para alertar o sistema sobre os eventos de excesso ocorridos em tempo real. No entanto, seguindo os critérios definidos para este projeto de ter um módulo discreto e não intrusivo, nesta primeira versão, não será processada esta informação recebida por esta *Tag*, mas apenas mais tarde, quando recebida através da *TagEcoSummary*, representada no ecrã de resumo de viagens no quadro referente aos excessos.

Uma distinção importante na implementação destas *Tags*, está no facto de que as mais antigas, neste caso apenas a *TagService*, fazem uso de atributos normais como *strings* e *bytes* (antes de serem convertidas para *bytearrays*), enquanto que as quatro novas *Tags* utilizam o sistema de *Protocol Buffers* da *Google* [35]. *Protobuf* é um formato utilizado para criar, definir a estrutura e serializar mensagens de dados para sua transmissão ou armazenamento. É primeiro criado um ficheiro de definição das mensagens, similar por exemplo ao formato JavaScript Object Notation (JSON), e de seguida gerado automaticamente código para a linguagem desejada. Os objetos gerados nesse código podem depois ser utilizados no programa, juntamente com a biblioteca *Protobuf* para codificar e descodificar os dados enviados e recebidos. Desta forma, as quatro novas *Tags* deste projeto contêm apenas um único campo de *bytearray* para armazenar os dados serializados correspondentes a cada tipo de mensagem/*Tag*.

4.2.3.2 Via Webservice

Quando não é possível à aplicação comunicar diretamente com a caixa, é necessário obter os dados do veículo de outra maneira. Neste modelo, os dados do veículo tratados pela caixa e pelo módulo *ECSM* são enviados para a base de dados central da empresa através de um canal *TCP* e interpretados no *Gestor*, um componente do servidor que os coloca em base de dados. Para a aplicação aceder a esta base de dados irá comunicar com um novo *Webservice* que faz de intermediário entre os dois. Serão assim feitas periodicamente chamadas à API do serviço, para atualizar os ecrãs do módulo.

Este *Webservice* baseado numa arquitetura Representational State Transfer (*REST*) e criado com linguagem *C#* em *ASP.NET Core*, é assim responsável por aceder à base de dados, mais particularmente, ao *Gestor* que é o mediador do acesso à mesma e enviá-los de volta à aplicação. A solução inclui ainda referências a projetos da Tecmic para facilmente implementar mecanismos de segurança e autenticação neste serviço.

A API principal do *Webservice*, referente aos dados registados durante as viagens, está representada no diagrama da figura 4.14. O controlador, que recebe os pedidos da aplicação à API tem dois pontos de acesso principais, um que obtém todos os dados das viagens (mais recentes) de um condutor, com base no seu ID, e outro que obtém os dados de uma viagem específica, com base nos códigos da mesma. Este controlador ao receber as chamadas, utiliza o serviço *XTraNServiceBusiness* que acede à base de dados/*Gestor* para devolver os dados em formato *JSON*. As classes do modelo de dados são idênticas às utilizadas no contexto do módulo (secção 4.2.2.2), com os atributos identificadores das viagens, e a lista de variáveis recolhidas.

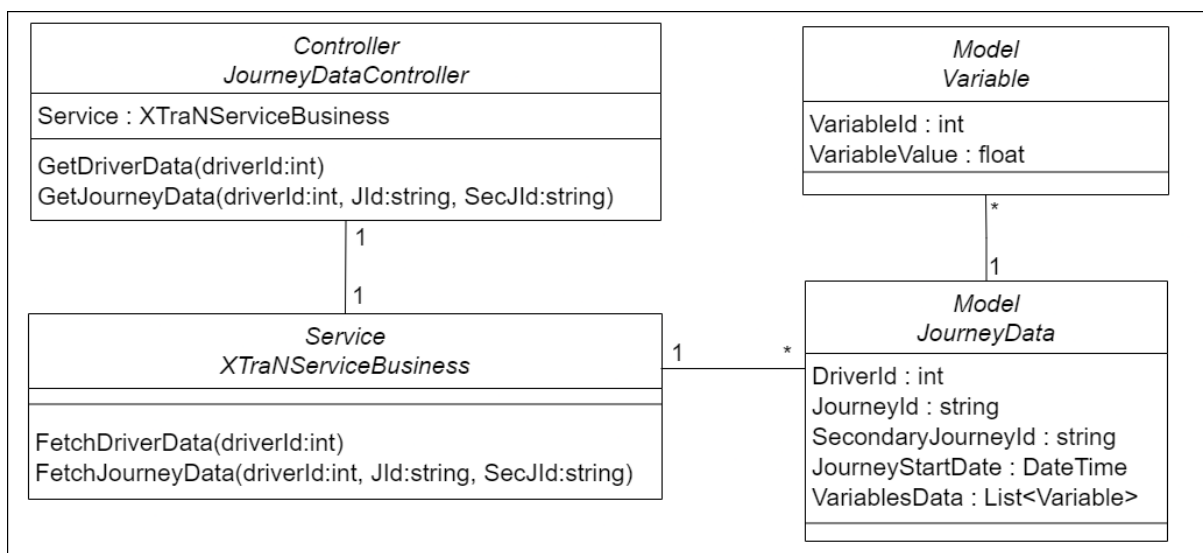


Figura 4.14: Diagrama de classes da API de dados das viagens do *Webservice*.

Já do lado da aplicação/módulo, foi criada a classe *DriveGreenApiService*, que contém os métodos

de acesso aos *endpoints* da API e atributos necessários (como Uniform Resource Locator (URL) base do serviço e *HTTPClient* para realizar chamadas assíncronas a este serviço, recebendo os dados em JSON e transformando-os nas classes presentes no módulo.

Relembremos o diagrama desta comunicação que pode ser visto na figura 4.15, esquema modificado do anteriormente apresentado na figura 3.5 da secção (3.1.3), de forma a isolar a comunicação desta segunda abordagem. Os dados da viagem são enviados pela caixa para a base de dados central, e quando a aplicação detetar que foi terminada uma viagem, manda ao *WebService* um pedido à sua API com parâmetros que definem o código do condutor e a viagem executada, para obter os dados necessários para atualização do ecrã de resumo. Do mesmo modo, durante uma viagem enviará pedidos similares de forma periódica para atualizar os gráficos em segundo plano. Isto implica que a informação dos histogramas em segundo plano sofre um impacto significativo, devido ao maior atraso nas comunicações. Assim, enquanto que na comunicação com a caixa, são recebidos dados, automaticamente, de minuto a minuto, na comunicação com o *WebService* estas chamadas serão feitas por exemplo de cinco em cinco minutos, devolvendo os dados da viagem gerados na janela de tempo anterior.

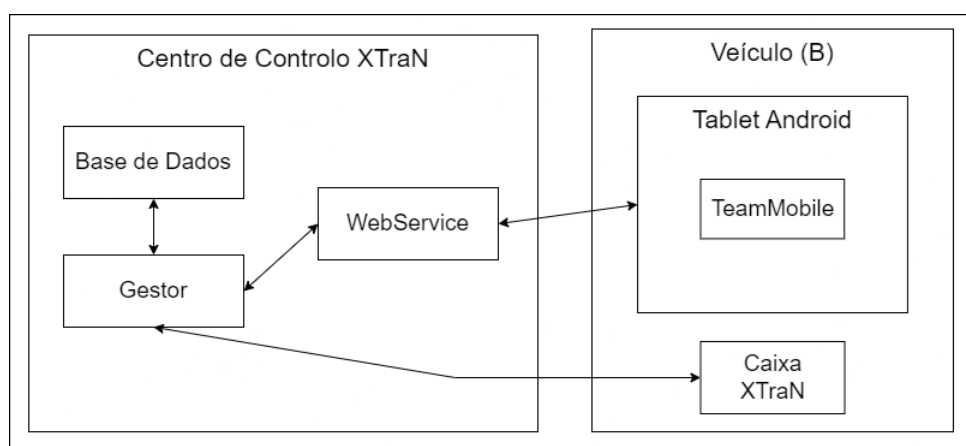


Figura 4.15: Diagrama de comunicação do módulo via *WebService*.

Este serviço tem ainda uma segunda API, associada à existência de um catálogo de variáveis. Este desenvolvimento faz com que o *WebService* mantenha a todo o momento uma tabela atualizada do tipo de variáveis presentes na base de dados central. No contexto deste módulo, as variáveis correspondem aos dados dos relatórios das viagens. Na figura 4.16 pode ser visto o diagrama desta segunda API. No modelo de dados, podemos ver que as variáveis estão associadas a um ID único e imutável, a um nome, descrição, e à respetiva unidade em que é medida, também esta associada a um ID único, nome e descrição. O serviço irá manter numa pequena base de dados em memória, uma tabela com estas variáveis.

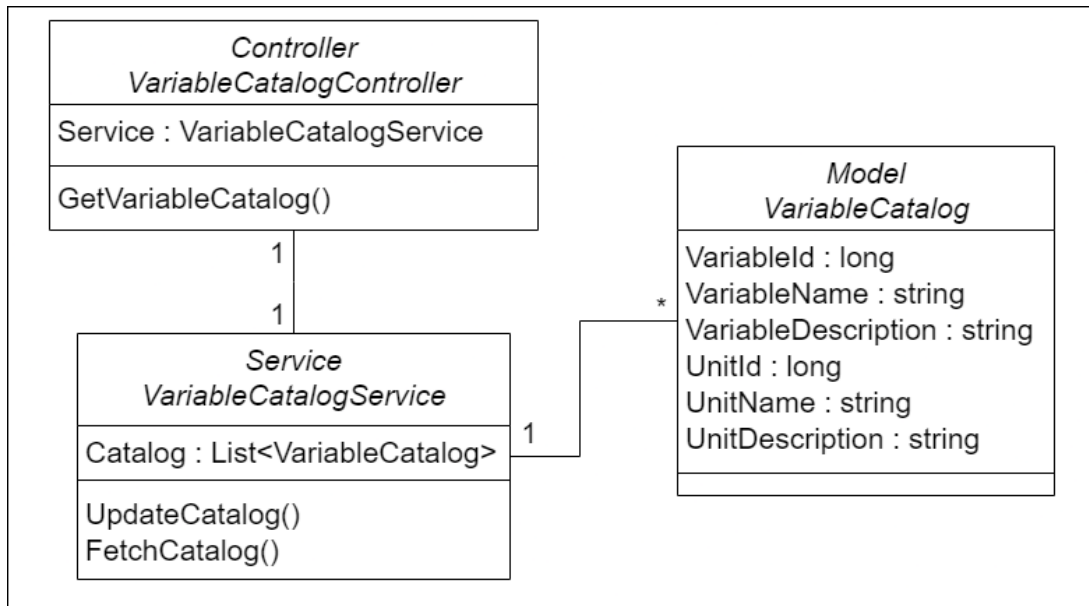


Figura 4.16: Diagrama de classes da API do catálogo de variáveis do *WebService*.

O *WebService*, através do serviço *VariableCatalogService*, atualiza diariamente o catálogo presente na sua tabela em memória com os dados obtidos da base de dados central/Gestor através do método *UpdateCatalog()*. O controlador da API, recebe os pedidos da aplicação, utilizando o serviço acima referido para enviar este catálogo para o módulo em formato JSON. Isto permite que a aplicação consulte este catálogo e receba atualizações dos parâmetros das variáveis e seja capaz de os mudar dinamicamente no ecrã do utilizador.

Observemos por exemplo a variável do catálogo especificada na listagem 4.2.

```

1 new VariableCatalog
2 {
3     VariableId = 2,
4     VariableName = "TOTAL_JOURNEY_DISTANCE",
5     VariableDescription = "Distancia total percorrida durante a viagem",
6     UnitId = 5,
7     UnitName = "KM",
8     UnitDescription = "Quilometros"
9 }
  
```

Listagem 4.2: Exemplo de variável do catálogo do *WebService*.

Esta variável, presente no catálogo do *WebService* com os parâmetros indicados, será armazenada pela aplicação quando feito o pedido a este API. A aplicação ao receber os dados da viagem, terá

um campo no ecrã para o valor da distância e outro para a unidade de medida. Se entretanto, a variável do catálogo for atualizada (método *UpdateCatalog()*) e mudar o parâmetro *UnitName* para *M* (metros), quando a aplicação fizer uma nova chamada a esta API, receberá esta atualização, podendo automaticamente atualizar a unidade de medida apresentada no ecrã do módulo. Isto facilita também a integração de novas variáveis, embora estas precisem necessariamente de algumas modificações no código da aplicação.

4.2.4 Uniformização dos Ecrãs da Aplicação TeamMobile

Este desenvolvimento consistiu na atualização dos ecrãs da aplicação de forma a cumprir as regras de uniformização visual estabelecidas pela empresa. Para tal foi preciso adquirir conhecimento sobre o funcionamento geral de toda a aplicação, e estar em contacto com colaboradores de outras áreas da empresa, como por exemplo da área do *Marketing*. Esta tarefa acabou por se tornar numa das mais longas, mas produzindo mudanças valiosas e significativas no *layout* da aplicação, incluindo nos novos ecrãs do módulo *DriveGreen*. No anexo A pode ser vista a evolução de alguns dos ecrãs da aplicação durante este desenvolvimento.

Aliado à modificação do *layout* dos ecrãs foram ao mesmo tempo feitas mudanças para a aplicação se melhor adaptar a dispositivos de diferentes dimensões ou resoluções. As duas estratégias mais úteis para esta implementação foram: (a) A criação de ficheiros de dimensões diferentes, por cada gama de densidade de pixels (*ldpi*, *mdpi*, *hdpi*, *xhdpi*). É usual em *Android* ter um ficheiro de dimensões (*dimens.xml*) onde estão definidos os tamanhos de vários elementos da aplicação, como botões, ícones e diferentes tamanhos de texto. Estes são normalmente definidos com base na visualização do ecrã no dispositivo ou emulador utilizado, com uma única densidade, usualmente a base, *mpdi*. Assim com este único ficheiro, um ecrã por exemplo com densidade alta (*hdpi*) não se vai ajustar às dimensões definidas. Mas na existência de vários ficheiros com as dimensões ajustadas, o sistema operativo escolhe o ficheiro adequado ao tipo de ecrã presente; (b) O uso da propriedade de *weight* (peso) para estabelecer a dimensão dos elementos com base no espaço disponível no ecrã. Este é um método mais usual, mas que não pode ser usado em qualquer situação. Como exemplo, não é útil definir que um ícone ocupe 1/17 do comprimento disponível no ecrã, mas será útil definir que duas colunas de texto lado a lado ocupem cada uma metade do espaço disponível;

Na figura 4.17 é possível ver um exemplo da comparação entre o ecrã de atividade antes e depois deste desenvolvimento, num ecrã de dimensões reduzidas.

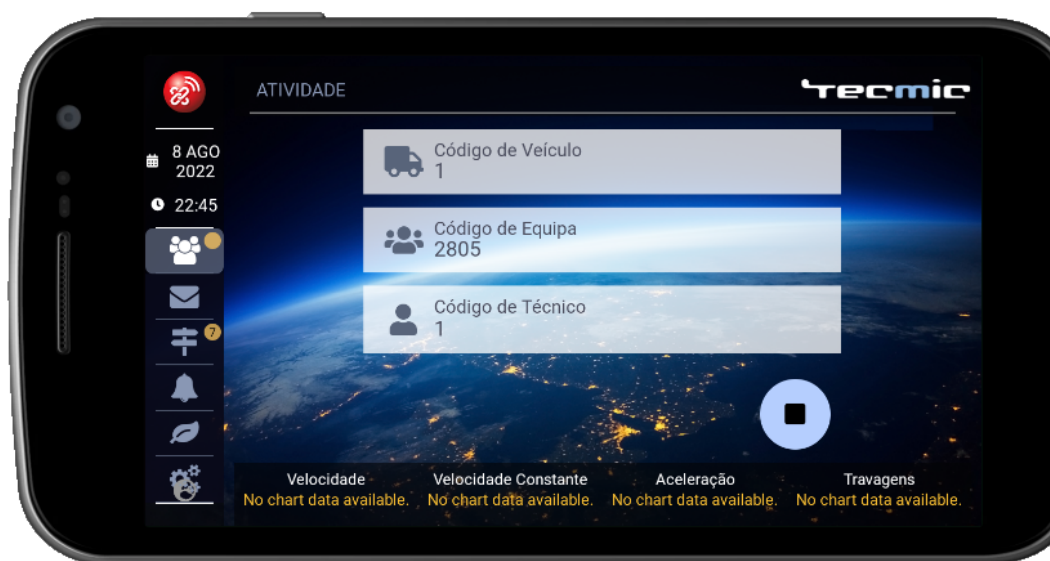
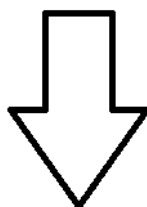


Figura 4.17: Evolução do ecrã de atividade da aplicação *TeamMobile* visto no emulador de um telemóvel.

5

Avaliação e Testes

Conteúdo

5.1 Testes Funcionais e de Usabilidade	63
5.2 Aprovação e Lançamento	64

Neste capítulo é apresentado o processo de testagem e metodologia de avaliação e aprovação durante o desenvolvimento deste projeto.

5.1 Testes Funcionais e de Usabilidade

Ao longo do desenvolvimento deste módulo foram efetuados testes de forma a assegurar a correção visual e funcional do trabalho desenvolvido. Estes testes foram realizados maioritariamente através de um emulador *Android*. De entre o código desenvolvido é de destacar a testagem de simulação de dados da viagem (periódicos ou de resumo) recebidos pela caixa/*WebService*. Este código é de grande importância pois permite avaliar a correção das principais funcionalidades do módulo. O *script* está presente no anexo B. Para além destes testes com um emulador, serão também realizados testes com uma consola e caixa *XTraN* para testar a parte relativa ao protocolo de comunicação via *Tags* e também testes à comunicação via *WebService*.

Adicionalmente, serão também feitos testes de usabilidade, tendo em conta a interface do módulo e cumprimento dos requisitos levantados (secção 1.2). Sendo um projeto cujo objetivo se depara em melhorar a condução sustentável dos seus utilizadores, seria útil obter *feedback* dos próprios, de forma a avaliar a sua eficácia. Para tal foi desenhado um pequeno questionário, que tendo em conta que o módulo não contém interações complexas na sua interface, este não é uma atividade guiada, com vários passos, mas apenas um conjunto de pequenas perguntas de modo a averiguar se os vários elementos presentes nos ecrãs são simples, intuitivos e ajudam o utilizador a melhorar a sua condução sustentável. O questionário pode ser visto no anexo C, e foi criado com base nas referências [36] e [37]. Este questionário seria integrado no próprio módulo *DriveGreen*.

Contudo, concluiu-se que não seria possível a aplicação deste questionário aos seus utilizadores alvo. Isto acontece porque os condutores, pertencentes a clientes da empresa, não iriam certamente ao fim de um longo dia de trabalho, e ao terminarem a última viagem, tirar o seu tempo para responder a estas perguntas, e a Tecmic não está numa posição em que seja possível criar incentivos.

Apesar disto, será recolhido *feedback* sobre o módulo de uma outra maneira, através de um teste de usabilidade interno, em que é disponibilizado à equipa a nova versão da aplicação e um questionário com base no apresentado anteriormente. Certamente não se poderá avaliar perguntas no âmbito do objetivo do módulo, como uma ferramenta de apoio a uma condução mais sustentável, mas sim aspetos relacionados com a *interface* e funcionamento do mesmo.

Mais tarde, haverá ainda uma oportunidade para obtenção de *feedback* pelos utilizadores, aquando da disponibilização da primeira versão da aplicação, assunto estudado na próxima secção (5.2).

5.2 Aprovação e Lançamento

Este módulo passou também por diversas fases de aceitação e aprovação do mesmo. Como é evidente, as várias etapas do projeto foram supervisionadas pelo coordenador e *Team Leader*, e para além disso, em tarefas como a de uniformização dos ecrãs da aplicação, e de criação dos ecrãs do presente módulo, estiveram também envolvidos elementos do departamento de *Marketing*, bem como o Chief Executive Officer (CEO) da empresa. Quando terminada esta última fase de testagem e desenvolvimento, será feito o *commit*, supervisionado, do código para o *master branch* do repositório, e gerada uma nova versão da aplicação *TeamMobile*. Esta nova versão será numa primeira fase disponibilizada a alguns dos clientes da empresa, e assim testada por algum tempo, altura em que poderá haver *feedback* sobre a mesma, indicado por responsáveis em cada cliente. No entanto, este tipo de *feedback* dado pelos responsáveis em cada cliente, trata-se frequentemente de erros mais graves que possam estar a acontecer, e não de opiniões diretamente dos utilizadores. Neste sentido, surgiu a ideia da criação de um pequeno módulo de *feedback* na *TeamMobile*, que os seus utilizadores possam usar para indicar sugestões tanto sobre o novo módulo *DriveGreen* como qualquer outra parte da aplicação. Apesar de os utilizadores não estarem disponíveis para responder a um questionário formal, é inevitável terem opiniões sobre a aplicação que usam todos os dias, e deste modo as suas sugestões podem ser recolhidas e periodicamente avaliadas pela equipa de desenvolvimento.

No fim desta fase, e com base no *feedback* obtido, serão feitos os últimos ajustes e correções até ao lançamento final do módulo e da nova versão da aplicação.

6

Conclusão

Conteúdo

6.1 Conclusões	67
6.2 Trabalho Futuro	68

Como referido anteriormente, o módulo *DriveGreen* ainda não foi lançado, estando nas últimas fases de testagem do protocolo entre consola e caixa. No entanto, isto não impossibilita que sejam retiradas ilações sobre todo o processo de desenvolvimento deste projeto, que serão depois confirmadas mais tarde. Neste capítulo final é ainda apresentado o trabalho futuro, no contexto do módulo desenvolvido, e da aplicação *TeamMobile*.

6.1 Conclusões

Este projeto teve como objetivo a integração de um módulo de auxílio à condução sustentável na aplicação *TeamMobile* da Tecmic S.A.. Esta aplicação é utilizada por clientes que efetuam prestação de serviços, fazendo múltiplas viagens por dia e registando os detalhes sobre as operações realizadas nesta aplicação. Desta maneira foram estudados sistemas existentes na empresa, bem como outras aplicações móveis disponíveis publicamente, de modo a melhor fixar e saber como cumprir os requisitos deste projeto. Após o desenho da solução, o seu desenvolvimento, e ao chegar a esta última fase de testes que ainda se encontra pendente, são antecipadas algumas conclusões.

Relativamente à experiência do utilizador, poderão seguir duas adversidades, a falta de envolvimento (*engagement*) no módulo, e a precisão do algoritmo de sugestões. Quanto à primeira, é fácil de perceber que o condutor pode simplesmente não querer fazer uso do módulo, nem dedicar o seu tempo no fim das viagens para aprender com o mesmo. Algumas características do módulo como a presença constante dos indicadores em segundo plano, e o aparecimento de uma janela que convida o utilizador a visitar o ecrã de resumo no fim das viagens, asseguram que o módulo não passará despercebido, mas isto pode não ser suficiente para manter o utilizador interessado. Já em relação à segunda questão, relacionada com o algoritmo de sugestões, é de esperar que após algumas viagens as sugestões recomendadas ao utilizador sejam vistas pelo próprio como repetitivas ou generalizadas. A confirmar-se esta situação, é preciso ter em conta as seguintes situações: (a) A existência de sugestões no módulo será sempre algo positivo, dado que identifica comportamentos errados do condutor, que este poderia não reconhecer de outra maneira, e indica maneira de os remediar; (b) A repetição de sugestões, associada a uma reduzida variedade das mesmas, é certamente um problema, mas não podemos esquecer que não é de todo exequível a disponibilização de sugestões que não pareçam repetitivas no contexto de um módulo de condução utilizado diariamente; (c) Também não é de esperar que, ao longo do tempo, sejam recomendadas toda a gama de sugestões disponíveis ao mesmo condutor. Em princípio, cada condutor pratica alguns comportamentos negativos, que fazem já parte do seu hábito de condução, e o objetivo é identificar e corrigir esses comportamentos até diminuírem substancialmente. As duas questões/adversidades apresentadas estão certamente relacionadas, já que a existência de um melhor algoritmo de sugestões e também de pontuação, poderá aumentar o envolvimento do utilizador no

módulo, como é discutido na próxima secção (6.2).

No entanto, e apesar dos problemas que possam surgir, os principais objetivos deste projeto foram alcançados. À aplicação *TeamMobile* foi adicionado um módulo de apoio à condução sustentável, que apresenta indicadores extremamente relevantes para este tema, gerando ainda uma pontuação e sugestões baseadas nos dados recolhidos em cada viagem. Os ideais de discrição e simplicidade dos ecrãs parecem também ter sido cumpridos, através de um ecrã secundário com pequenos gráficos que são ocultados sempre que necessário e de um ecrã detalhado, com pouca interação, que ajuda o utilizador a analisar a sua *performance* nas viagens passadas. Em termos tecnológicos foi ainda acrescentado um novo mecanismo de comunicação, os *Protocol Buffers*, que simplificam a troca de mensagens entre a consola e a caixa do veículo, bem como o desenvolvimento de um novo *WebService* que facilita a troca de informação entre a aplicação e o centro de controlo, sem passar pela caixa. A integração de um novo módulo nesta aplicação foi também um sucesso, dado que, e a título mais pessoal, foi feita uma aprendizagem célere da plataforma *Xamarin*, da linguagem *C#*, e da aplicação *TeamMobile* como um todo.

Assim sendo, com a integração de um novo módulo de apoio à condução sustentável e o *redesign* completo dos ecrãs da aplicação, a *TeamMobile* ganha verdadeiramente uma nova vida como resultado de todo este processo.

6.2 Trabalho Futuro

O trabalho futuro numa perspetiva mais imediata, consiste em terminar esta última fase de testagem do protocolo de comunicação com a caixa *XTraN* e o *WebService*, realizar os testes de usabilidade internos, e posteriormente, como explicado na secção 5.2, disponibilizar uma nova versão da aplicação a uma parte dos seus utilizadores. Com base no *feedback* obtido pelos responsáveis em cada cliente, serão feitos os últimos ajustes antes do seu lançamento final.

Para além disto, há outros aspetos planeados para as próximas versões da aplicação:

1. No contexto do *DriveGreen*, a melhoria dos algoritmos de pontuação e de sugestões das viagens. Para além da parte mais técnica, como o aperfeiçoamento da fórmula de cálculo da pontuação e aumento da variedade de sugestões disponíveis, seria também interessante aplicar conceitos relacionados com *gamification*, com o objetivo destes indicadores servirem como maneira de incentivar o condutor a melhorar a sua condução e tirar proveito do módulo. Para isso estes algoritmos deveriam ter em conta o progresso do utilizador nas suas várias viagens. Assim, a pontuação poderá ser baseada não apenas em valores constantes, mas na avaliação contínua do utilizador, e as sugestões poderão refletir o progresso do mesmo na correção dos comportamentos identificados. Contudo, é preciso não esquecer que estes elementos podem também provocar

o efeito contrário, o condutor pode sentir-se frustrado se repetidamente não conseguir melhorar a sua *performance* de condução, ou atingir níveis muito baixos de pontuação, quer por sua culpa (inevitável em viagens mais longas) ou por via das circunstâncias (em viagens com características anómalas);

2. No contexto do *DriveGreen*, a disponibilização dos relatórios de condução das viagens na plataforma *Web* da Tecmic, para condutores e supervisores;
3. No contexto da *TeamMobile*, o desenvolvimento do módulo para obtenção de *feedback* dos condutores. Um mecanismo simples, em que o utilizador pode seleccionar o módulo da aplicação que está a avaliar, e escrever um curto texto sobre o mesmo. Estas sugestões poderão ser enviadas por meio do *WebService* desenvolvido, e periodicamente discutidas dentro da empresa.

Bibliografia

- [1] Tecmic S.A., “FleetMobile na App Store,” Oct 2021, Last accessed 30 Oct 2022. [Online]. Available: <https://apps.apple.com/pt/app/fleetmobile/id1357748434>
- [2] INGTECH, “What is eco-driving? - INGTECH,” May 2020, Last accessed 30 Oct 2022. [Online]. Available: <https://www.ingtech.com/en/publications/what-is-eco-driving/>
- [3] J. Barkenbus, “Eco-driving: An overlooked climate change initiative,” *Energy Policy*, vol. 38, no. 2, pp. 762–769, 2010.
- [4] Y. Huang, E. Ng, J. Zhou, N. Surawski, E. Chan, and G. Hong, “Eco-driving technology for sustainable road transport: A review,” *Renewable and Sustainable Energy Reviews*, vol. 93, pp. 596–609, 2018.
- [5] S. Innamaa and M. Penttinen, “Impacts of a green-driving application in city buses on fuel consumption, speeding and passenger comfort,” *IET Intelligent Transport Systems*, vol. 8, no. 5, pp. 435–444, 2014.
- [6] J. Tulusan, T. Staake, and E. Fleisch, “Providing eco-driving feedback to corporate car drivers: what impact does a smartphone application have on their fuel efficiency?” in *Proceedings of the 2012 ACM conference on ubiquitous computing*, 2012, pp. 212–215.
- [7] B. Beusen, S. Broekx, T. Denys, C. Beckx, B. Degraeuwe, M. Gijssbers, K. Scheepers, L. Govaerts, R. Torfs, and L. Panis, “Using on-board logging devices to study the longer-term impact of an eco-driving course,” *Transportation research part D: transport and environment*, vol. 14, no. 7, pp. 514–520, 2009.
- [8] Tecmic S.A., “Gestão Profissional de Frotas e Localização GPS - TECMIC,” Apr 2021, Last accessed 30 Oct 2022. [Online]. Available: <https://www.tecmic.com/solucoes-tecmic/gestao-de-frotas/>

- [9] —, “Gestão Profissional de Transportes de Passageiros - TECMIC,” Apr 2021, Last accessed 30 Oct 2022. [Online]. Available: <https://www.tecmic.com/solucoes-tecmic/gestao-de-transporte-de-passageiros/>
- [10] L. Torres, “GreenDriving,” Master’s thesis, Politécnico de Leiria, Leiria, Portugal, Sep. 2018.
- [11] Samsara, “Eco Driving - Samsara,” Jan 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://www.samsara.com/fleet/ecodriving/>
- [12] INGTECH, “ECO DRIVING - Intelligent Fleet Management Solutions - INGTECH,” Jan 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://www.ingtech.com/en/solutions/ecodriving/>
- [13] ovZ, “Car Scanner ELM OBD2 – Apps on Google Play,” Jan 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=com.ovz.carscanner>
- [14] East Software Coders, “ECO-Driving Speedometer - Apps on Google Play,” Oct 2021, Last accessed 30 Oct 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=com.eastcoders.speedometer>
- [15] Hayandroid, “Drive Eco - Apps on Google Play,” Nov 2021, Last accessed 30 Oct 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=com.haytatu.roadslopemeter>
- [16] Ecologic.io, “Ecologic - Apps on Google Play,” Dec 2021, Last accessed 30 Oct 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=pl.netmart.ecologic>
- [17] I. Hawkins, “Torque Pro (OBD 2 & Car) - Apps on Google Play,” Jan 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=org.prowl.torque>
- [18] I. Hawkins, “Torque Lite (OBD2 & Car) - Apps on Google Play,” Jan 2018, Last accessed 30 Oct 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=org.prowl.torquefree>
- [19] Torque Pro, “Torque Lite Vs Torque Pro 2021- Torque Pro App OBD2 2021 Review,” Aug 2021, Last accessed 30 Oct 2022. [Online]. Available: https://torqueproapk.org/torque-lite-vs-torque-pro/#Torque_Lite_Vs_Torque_Pro
- [20] Palmer Performance Engineering, “DashCommand (OBD ELM App) - Apps on Google Play,” May 2021, Last accessed 30 Oct 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=com.palmerperformance.DashCommand>
- [21] Microsoft, “What is Xamarin? - Xamarin - Microsoft Learn,” Sep 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>

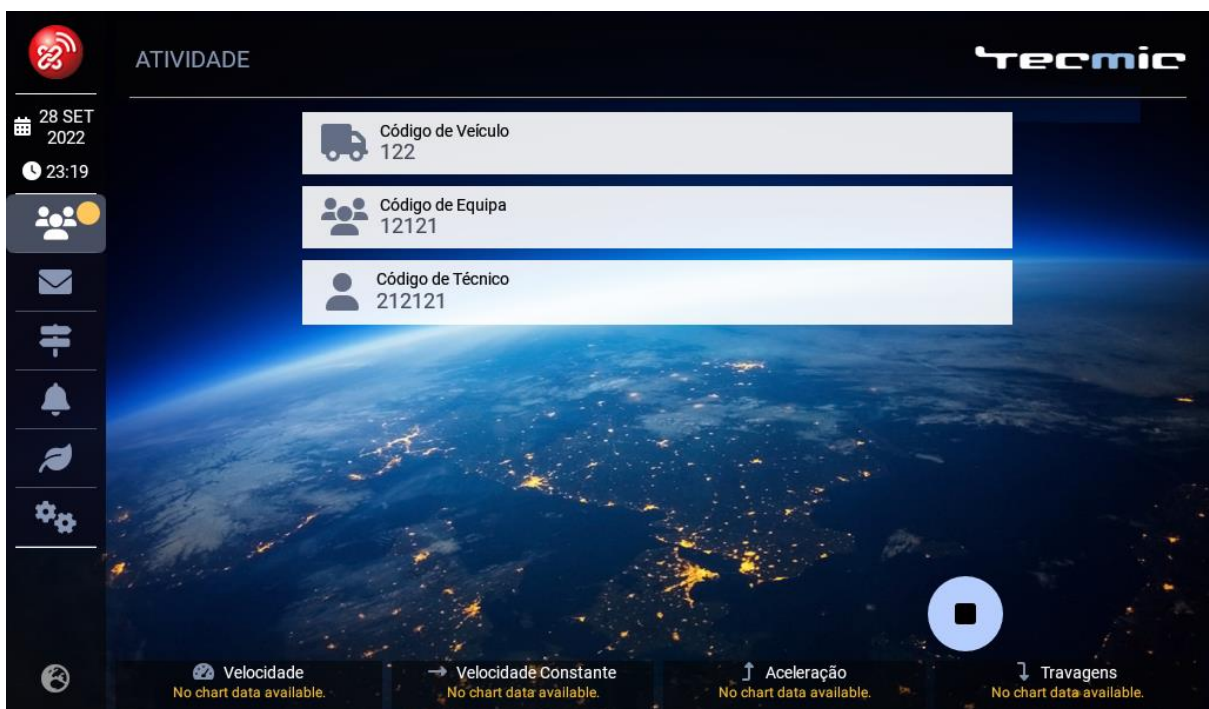
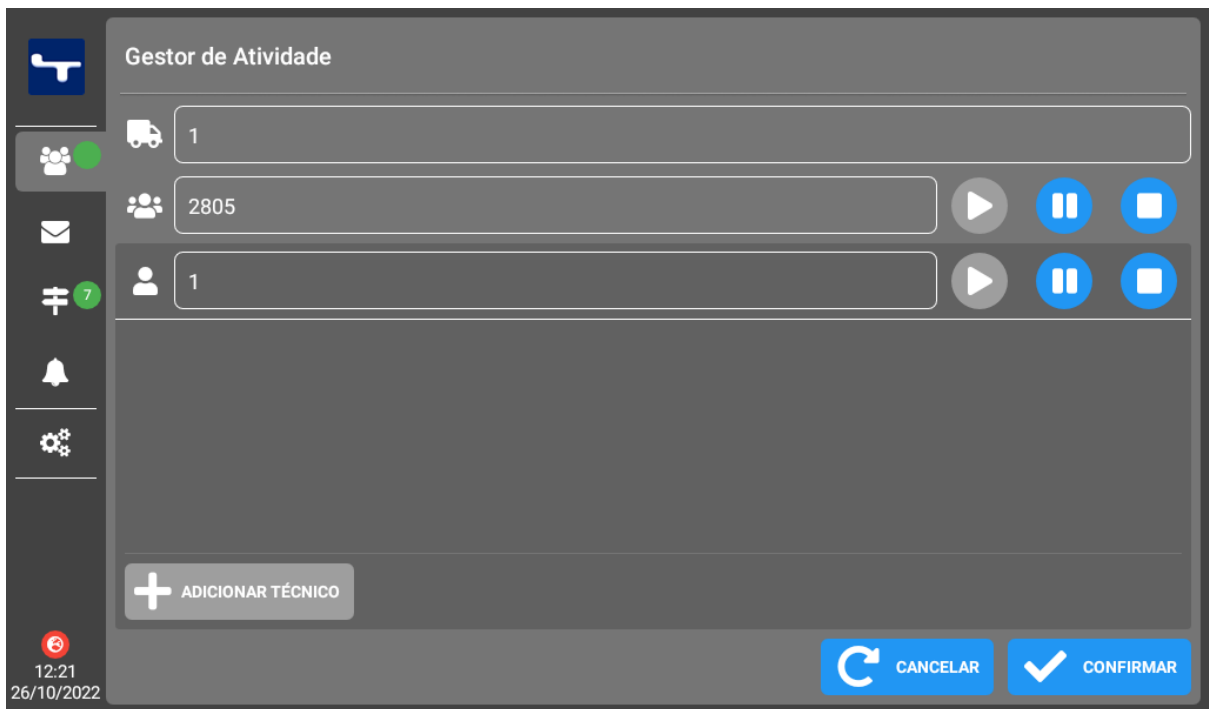
- [22] Pavel, "All Differences and Use Cases for Xamarin Native vs Xamarin.Forms," Sep 2020, Last accessed 30 Oct 2022. [Online]. Available: <https://jelvix.com/blog/xamarin-forms-vs-xamarin-native>
- [23] K. Beladiya, "Top 6 Features of Xamarin as Cross Platform Framework," Jan 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://theonetechnologies.com/blog/post/6-features-of-xamarin-as-cross-platform-framework>
- [24] Telerik, "When to Use Xamarin.Forms vs Xamarin Native?" Oct 2020, Last accessed 30 Oct 2022. [Online]. Available: <https://www.telerik.com/blogs/when-to-use-xamarin-forms-vs-xamarin-native>
- [25] TechTarget, "What is Model-View-ViewModel (MVVM)?" Feb 2019, Last accessed 30 Oct 2022. [Online]. Available: <https://www.techtarget.com/whatis/definition/Model-View-ViewModel>
- [26] MvvmCross, "Getting Started with MvvmCross - MvvmCross," Sep 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://www.mvvmcross.com/documentation/getting-started/getting-started>
- [27] Moqups, "Online Mockup, Wireframe & UI Prototyping Tool · Moqups," Jul 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://moqups.com/>
- [28] Syncfusion, "155+ Xamarin UI controls for iOS, Android & UWP apps - Syncfusion," Aug 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://www.syncfusion.com/xamarin-ui-controls>
- [29] SciChart, "SciChart - WPF Charts, iOS Charts, Android Charts, JavaScript Chart," Jul 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://www.scichart.com/>
- [30] OxyPlot, "GitHub - oxyplot/oxyplot: A cross-platform plotting library for .NET," Oct 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://github.com/oxyplot/oxyplot>
- [31] Microcharts, "GitHub - microcharts-dotnet/Microcharts: Create cross-platform simple charts," Feb 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://github.com/microcharts-dotnet/Microcharts>
- [32] P. Jahoda, "GitHub - PhilJay/MPAndroidChart: A powerful Android chart library," Jun 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://github.com/PhilJay/MPAndroidChart>
- [33] J. Cull, "Generate Color Gradient in C#," Jan 2013, Last accessed 30 Oct 2022. [Online]. Available: <https://stackoverflow.com/a/14246328>
- [34] V. Pecanac, "Events in C# - Code Maze," Mar 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://code-maze.com/csharp-events/>

- [35] Google, "Protocol Buffers - Google Developers," Oct 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://developers.google.com/protocol-buffers>
- [36] N. Babich, "User & Usability Testing Questions: Ultimate Guide - Adobe XD Ideas," Jan 2021, Last accessed 30 Oct 2022. [Online]. Available: <https://xd.adobe.com/ideas/process/user-testing/usability-testing-questions-tips-examples/>
- [37] A. Hinderks, M. Schrepp, J. Thomaschewski, "User Experience Questionnaire (UEQ)," Jan 2018, Last accessed 30 Oct 2022. [Online]. Available: <https://www.ueq-online.org/>
- [38] Google, "Google Forms: Online Form Creator - Google Workspace," Jun 2022, Last accessed 30 Oct 2022. [Online]. Available: <https://www.google.com/forms/about/>

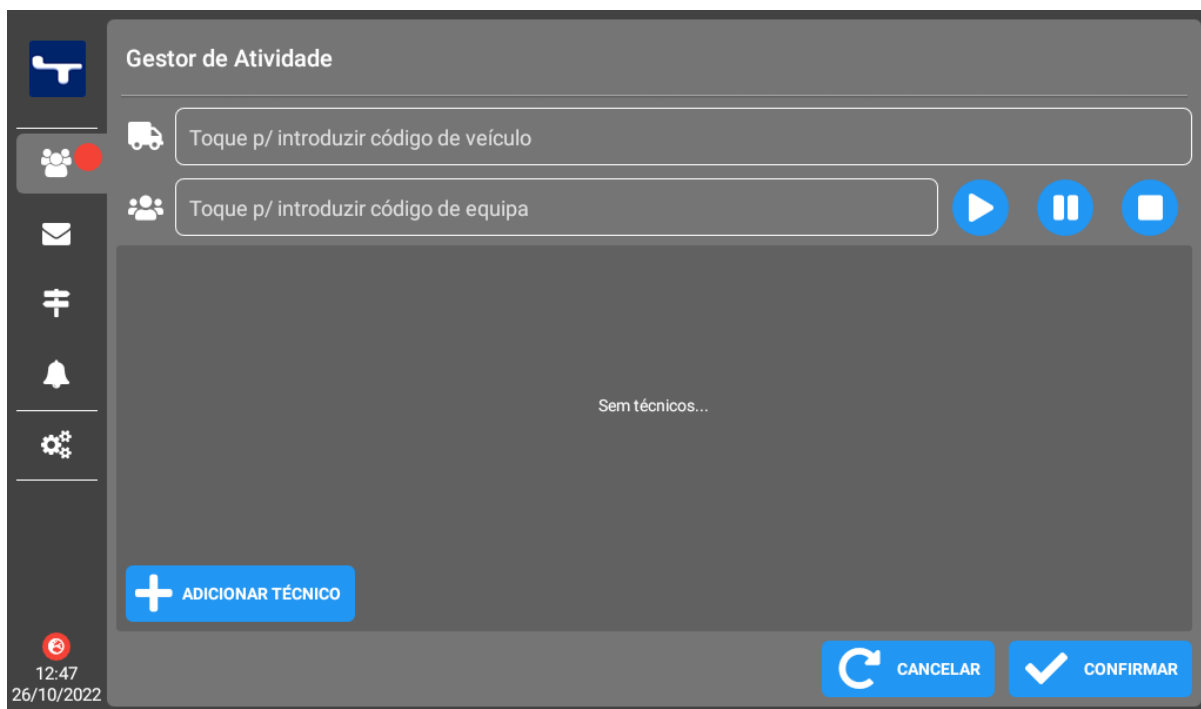


Anexo A

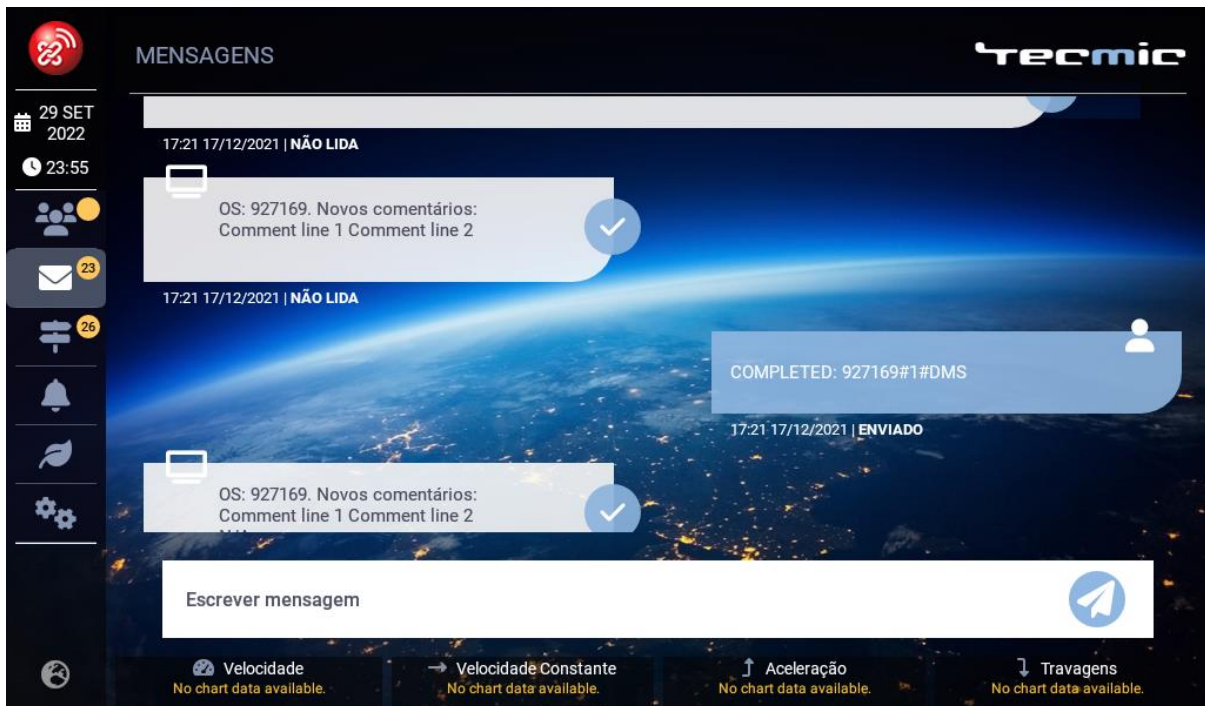
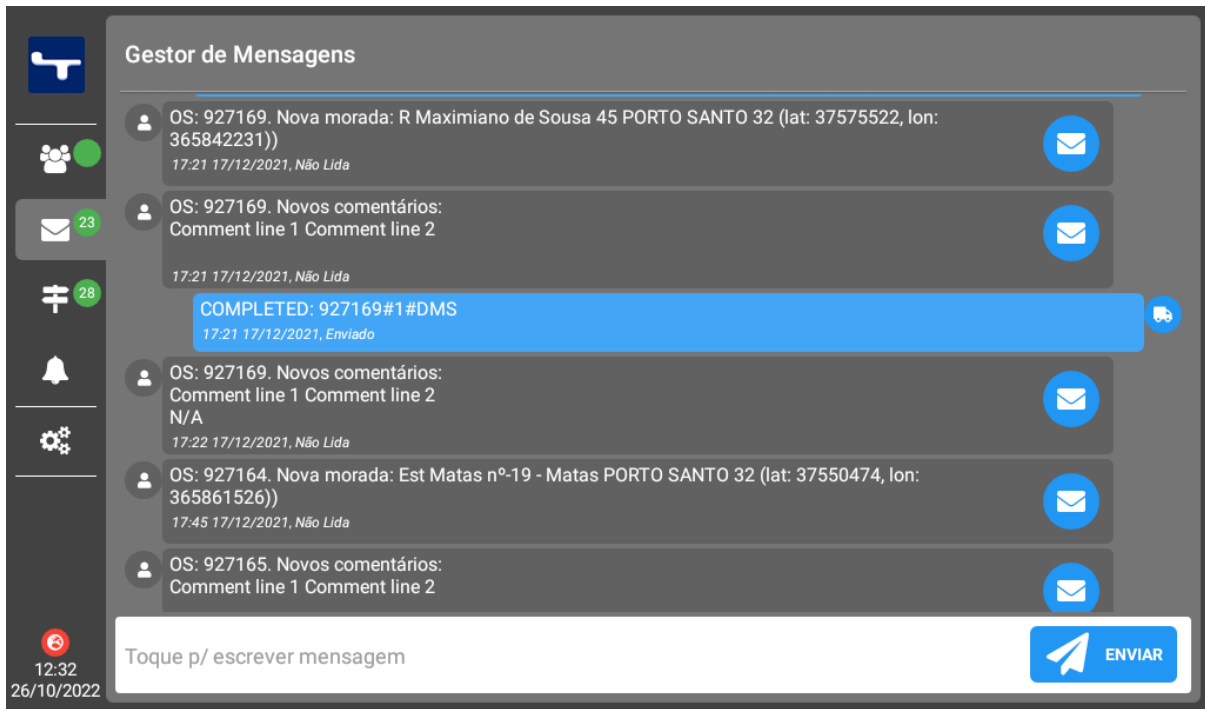
Neste anexo é apresentado o trabalho realizado no contexto da tarefa de uniformização dos ecrãs da *TeamMobile*. É mostrada uma comparação entre o antes-e-depois deste desenvolvimento para alguns dos ecrãs da aplicação.



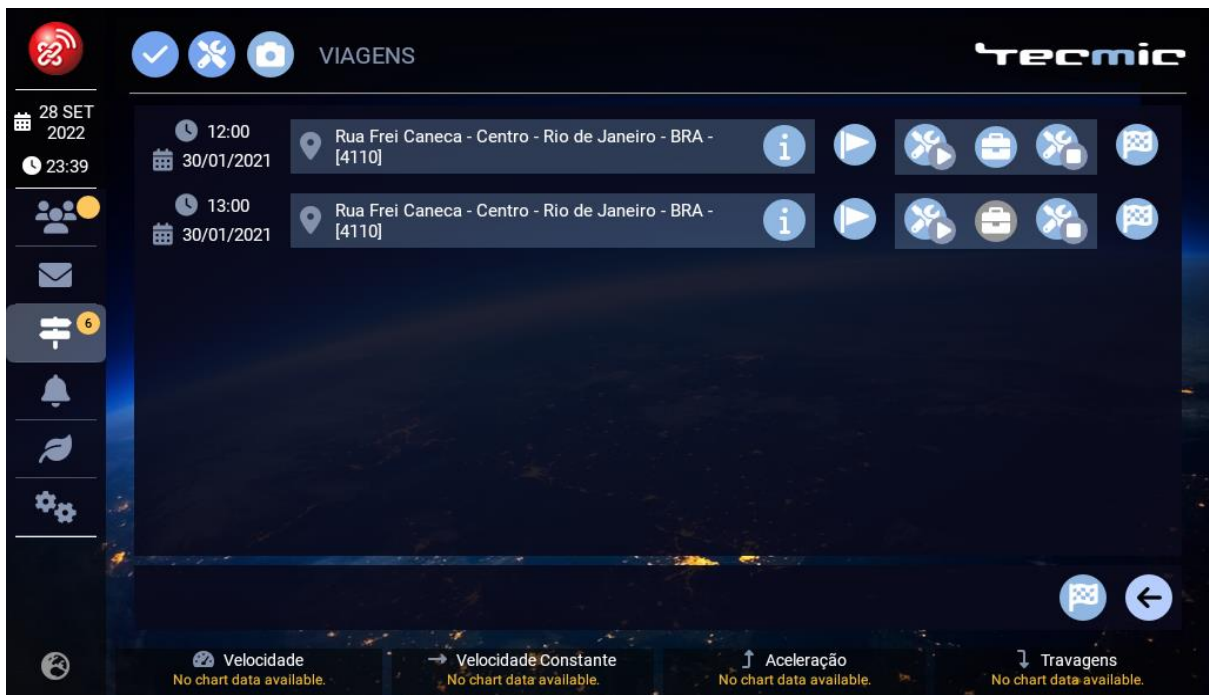
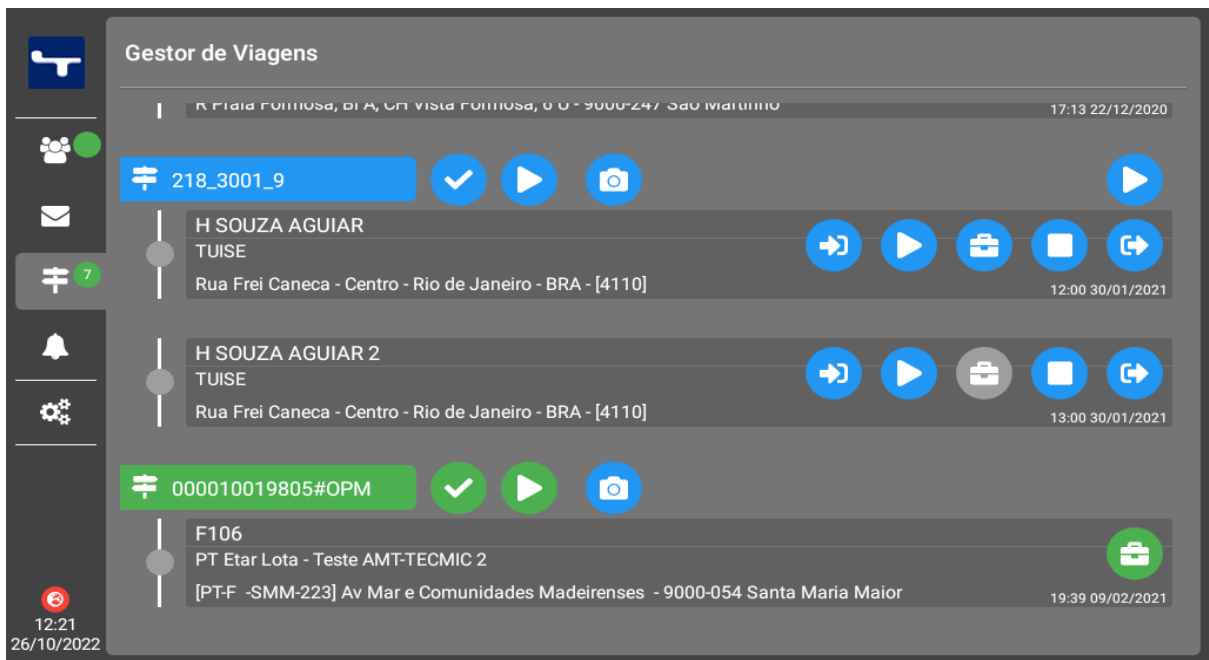
1. Ecrã de Atividade (em curso).



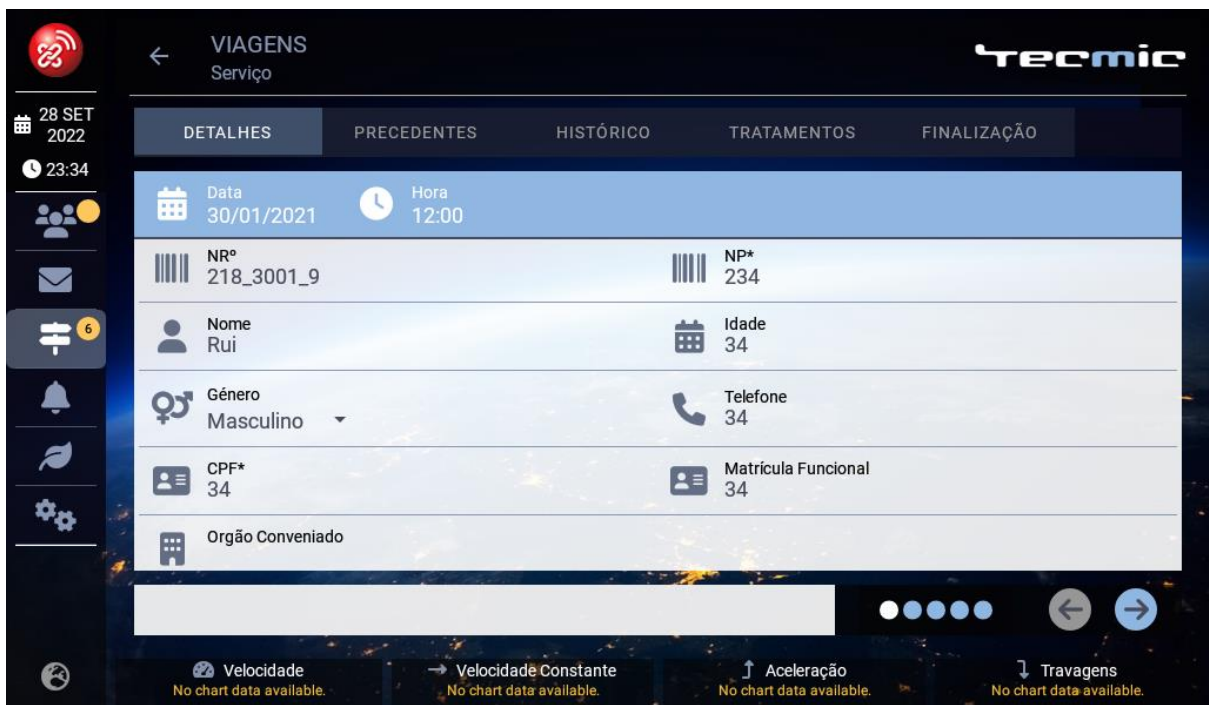
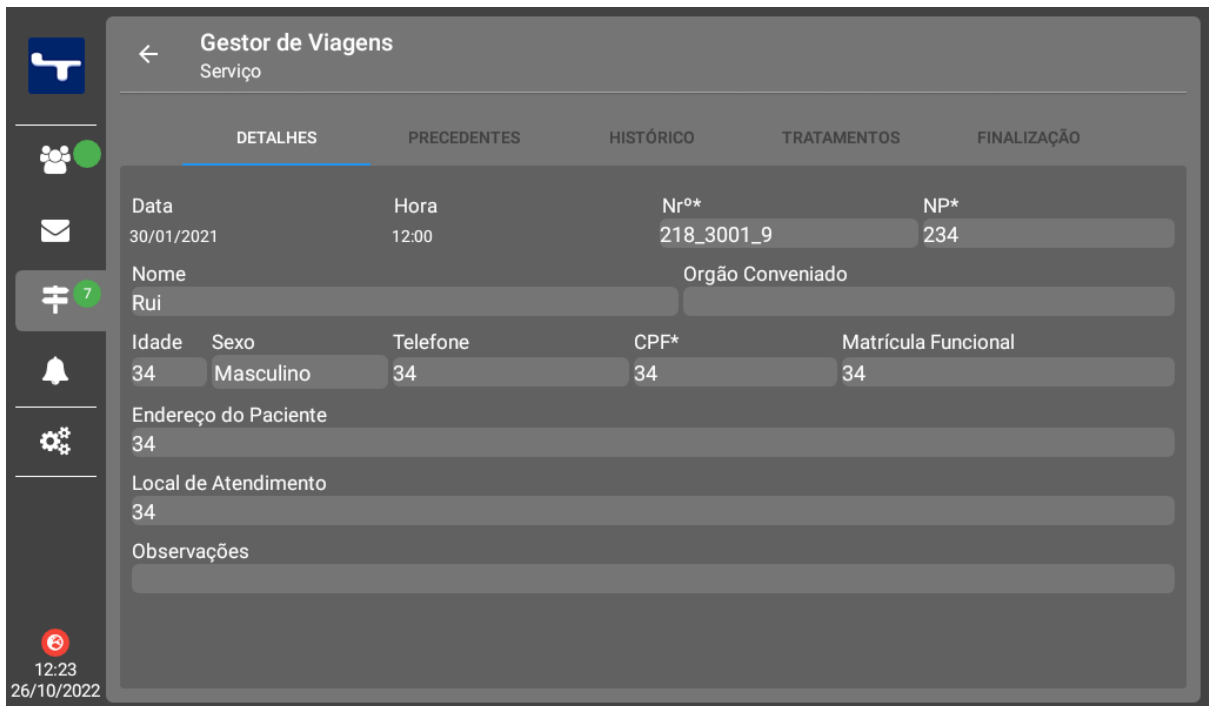
2. Ecrã de Atividade (a iniciar).



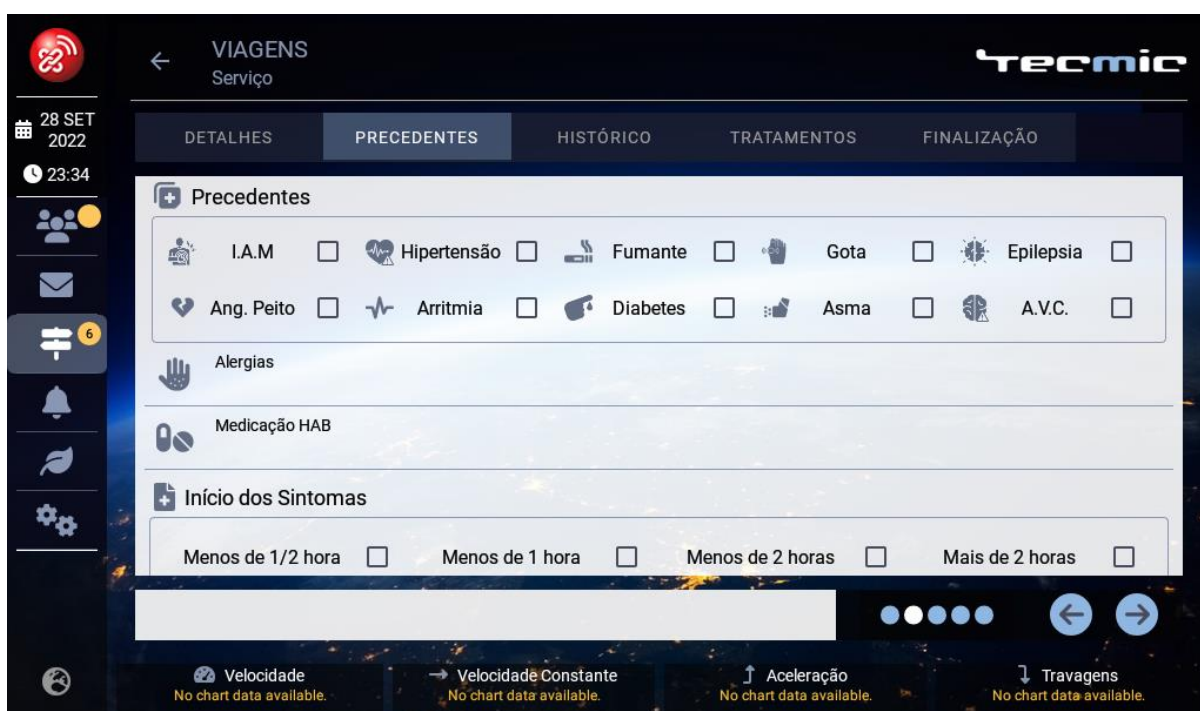
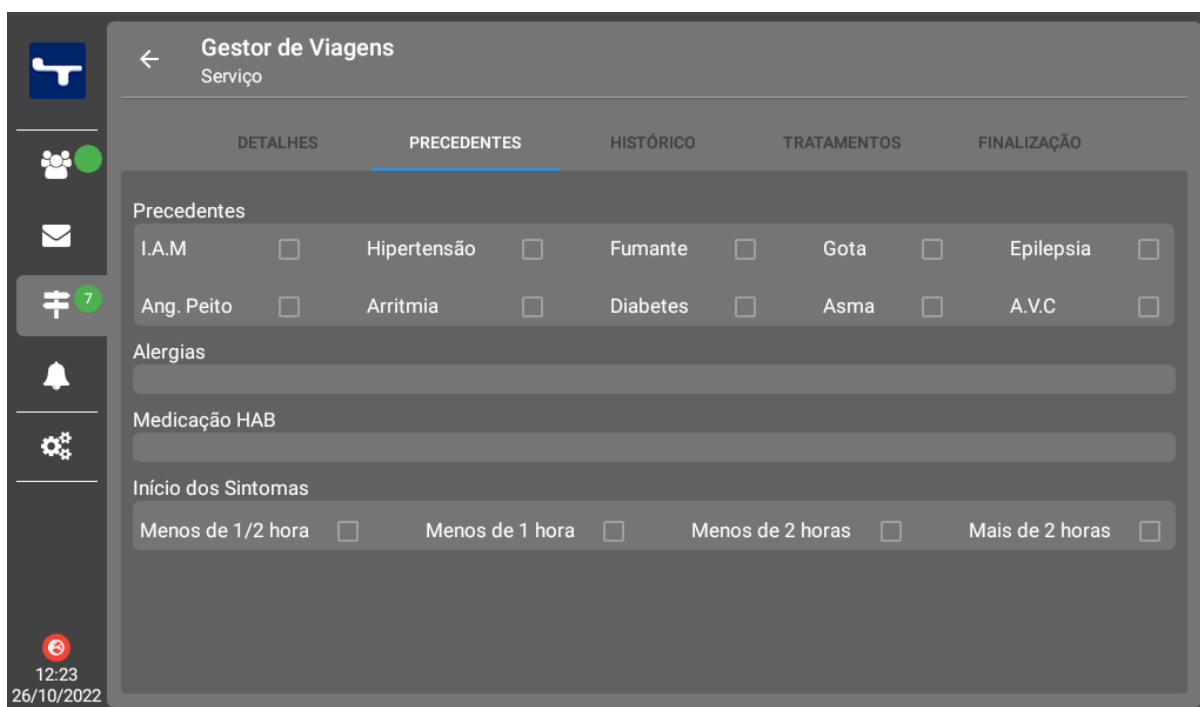
3. Ecrã de Mensagens.



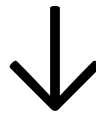
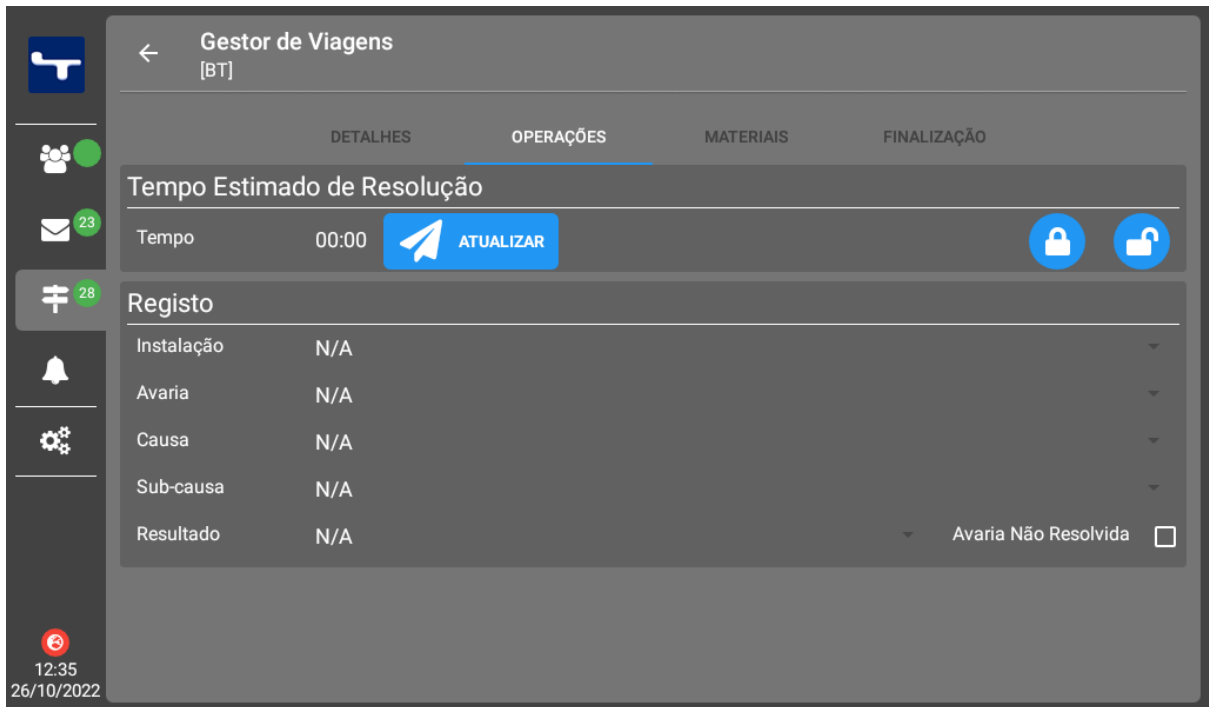
4. Ecrã de Viagens e Paragens, agora convertido em dois ecrãs.



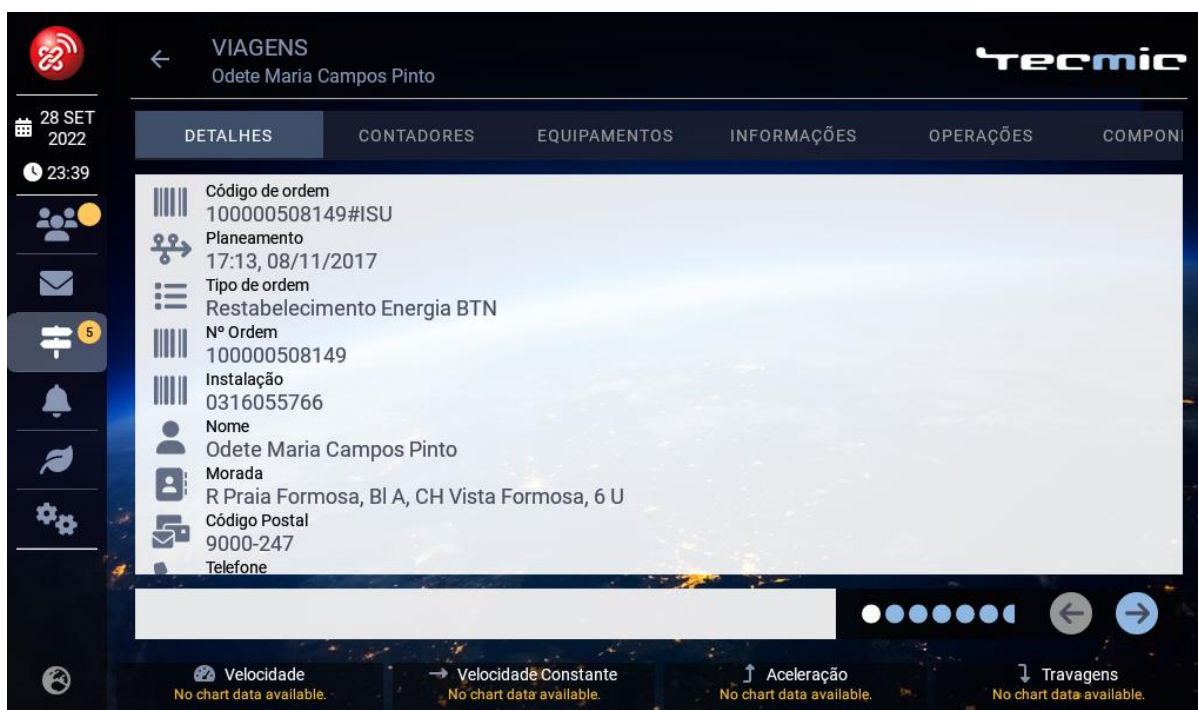
5. Ecrã de Formulário para Cliente Particular.



6. Ecrã de Formulário para Cliente Particular.



7. Ecrã de Formulário para Cliente Particular.



8. Ecrã de Formulário para Cliente Particular.

B

Anexo B

Neste anexo é apresentado o código de teste utilizado para simular a receção de dados das viagens. Na listagem B.1 é possível ver o fluxo simulado e a criação de dados em *Protobuf* mas omite, naturalmente, as funções de *parsing* e armazenamento dos dados recebidos, disparadas nas linhas 102 e 208: "*ConsoleBusiness.FireTagEcoDriveDynamicEvent(tagEcoDynamic);*". O código simula três viagens diferentes, em cada uma delas recebidas três vezes dados periódicos, para atualizar os gráficos em segundo plano, e uma vez os dados de resumo, para apresentar no ecrã principal. Os valores são gerados aleatoriamente.

```
1 // Some irrelevant code sections were omitted
2
3 // Core BusinessManager class
4 public partial class BusinessManager : IBusinessManager
5 {
6
```

```

7 // When the application starts, the StartCommunications() runs, which is now
  // also responsible for registering the new tags to the mobile unit and
  // running the testing method
8 public void StartCommunications(...)
9 {
10     //omitted
11
12     //register tags to the mobile unit
13     RegisterInTags();
14
15     //run testing code
16     if (tagsTest) TestProtoDataCycle();
17 }
18
19 }
20
21 // BusinessManagerDriveGreen class
22 public partial class BusinessManager
23 {
24     //test attributes
25     public int testDriverId = 1351;
26     int testIterationCounter = 0;
27     int testStep = 1;
28
29     public void TestProtoDataCycle()
30     {
31         TestJourney1();
32     }
33
34     private void TestJourney1()
35     {
36         LoggerPlugin.LogAsync(LogLevel.Debug, "DriveGreenBusinessManager::
           TestJourney1()", "JOURNEY1");
37         TestReceivedECSMPProtoPeriodic();
38     }
39     private void TestJourney2()
40     {
41         testStep = 2;

```



```

42     LoggerPlugin.LogAsync (LogLevel.Debug, "DriveGreenBusinessManager::
        TestJourney2()", "JOURNEY2");
43     TestReceivedECSMPProtoPeriodic();
44 }
45 private void TestJourney3()
46 {
47     testStep = 3;
48     LoggerPlugin.LogAsync (LogLevel.Debug, "DriveGreenBusinessManager::
        TestJourney3()", "JOURNEY3");
49     TestReceivedECSMPProtoPeriodic();
50 }
51
52 public void TestReceivedECSMPProtoPeriodic()
53 {
54     //Simulating received data
55     uint a = (uint)new Random().Next(1, 7);
56     uint b = (uint)new Random().Next(1, 7);
57     uint c = (uint)new Random().Next(1, 7);
58     uint d = (uint)new Random().Next(1, 7);
59     uint e = (uint)new Random().Next(1, 7);
60
61     //periodic data
62     TermEcoDataTag data = new TermEcoDataTag
63     {
64         EcoVallList = {
65             new TermEcoPieDataItem {
66                 ShowFlag=true,
67                 PieType= TermEcoPieType.TermUmEcoPieTypeSpeed,
68                 PieTimeVal=0,
69                 PieValues={a,a,b}
70             },
71             new TermEcoPieDataItem {
72                 ShowFlag=true,
73                 PieType= TermEcoPieType.TermUmEcoPieTypeAcceleration,
74                 PieTimeVal=0,
75                 PieValues={c,b,d}
76             },
77             new TermEcoPieDataItem {

```

```

78         ShowFlag=true,
79         PieType= TermEcoPieType.TermUmEcoPieTypeDeceleration,
80         PieTimeVal=0,
81         PieValues={a,e,e}
82     },
83     new TermEcoPieDataItem {
84         ShowFlag=true,
85         PieType= TermEcoPieType.TermUmEcoPieTypeMovement,
86         PieTimeVal=0,
87         PieValues={b,a}
88     },
89 },
90
91 };
92
93 byte[] protoData = Any.Pack(data).ToArray();
94
95 TagEcoDynamic tagEcoDynamic = new TagEcoDynamic
96 {
97     EcoDynamicProtoData = protoData
98 };
99
100 //simulating message sent, received, decoded, event fired
101 //this triggers the data parsing methods
102 ConsoleBusiness.FireTagEcoDriveDynamicEvent(tagEcoDynamic);
103
104 if (testIterationCounter < 2)
105 {
106     //send more periodic data for the same journey
107     testIterationCounter++;
108     RunActionTimerExtensions.StartRunActionAfter(
109         TestReceivedECSMProtoPeriodic, 2);
110 }
111 else
112 {
113     //send summary data
114     //stop journey

```

```

114         RunActionTimerExtensions.StartRunActionAfter(
            TestReceivedECSMProtoSummary, 2);
115         testIterationCounter = 0;
116     }
117 }
118 //test code
119 public void TestReceivedECSMProtoSummary()
120 {
121     uint a = (uint)new Random().Next(1, 7);
122     uint b = (uint)new Random().Next(1, 7);
123     uint c = (uint)new Random().Next(1, 7);
124     uint d = (uint)new Random().Next(1, 7);
125     uint e = (uint)new Random().Next(1, 7);
126     int totalKm = new Random().Next(30, 100);
127     //summary data
128     TermEcoGetCurDataConfTag data = new TermEcoGetCurDataConfTag
129     {
130         EcoGetCurDataConfMode = TermEcoGetCurDataConfMode.
            TermEcoGetCurDataConfSuccess,
131         DriverId = testDriverId.ToString(),
132         CircuitCode = "CID1"+ testStep.ToString(),
133         CircuitSecCode = "CID2" + testStep.ToString(),
134         EcoResumeData = new TermEcoResumeData
135         {
136             NumIgnitions = 1,
137             TotalDistance = (uint) totalKm * 1000,
138             FuelConsumption = 8 * 1000,
139             TotalIgnitionTime = 30,
140             TotalEngineTime = 27,
141             TotalEngineIdleTime = 3,
142         },
143         EventDist = {
144             new TermEcoExcEventCount {
145                 EventId=EcsmEventId.EcsmEventExcSpeed,
146                 ExcEvCount= (uint)new Random().Next(1, 4),
147             },
148             new TermEcoExcEventCount {
149                 EventId=EcsmEventId.EcsmEventExcAcc,

```

```

150         ExcEvCount= (uint)new Random().Next(1, 4),
151     },
152     new TermEcoExcEventCount {
153         EventId=EcsmEventId.EcsmEventExcBrk,
154         ExcEvCount= (uint)new Random().Next(1, 4),
155     },
156     new TermEcoExcEventCount {
157         EventId=EcsmEventId.EcsmEventExcRot,
158         ExcEvCount= (uint)new Random().Next(1, 4),
159     },
160     new TermEcoExcEventCount {
161         EventId=EcsmEventId.EcsmEventExcEngTemp,
162         ExcEvCount= (uint)new Random().Next(1, 4),
163     },
164     new TermEcoExcEventCount {
165         EventId=EcsmEventId.EcsmEventExcIdle,
166         ExcEvCount= (uint)new Random().Next(1, 4),
167     },
168 },
169 PieVoyageData = {
170     new TermEcoPieDataItem {
171         ShowFlag=true,
172         PieType= TermEcoPieType.TermUmEcoPieTypeSpeed,
173         PieTimeVal=0,
174         PieValues={a,a,b}
175     },
176     new TermEcoPieDataItem {
177         ShowFlag=true,
178         PieType= TermEcoPieType.TermUmEcoPieTypeAcceleration,
179         PieTimeVal=0,
180         PieValues={c,b,d}
181     },
182     new TermEcoPieDataItem {
183         ShowFlag=true,
184         PieType= TermEcoPieType.TermUmEcoPieTypeDeceleration,
185         PieTimeVal=0,
186         PieValues={a,e,e}
187     },

```

```

188         new TermEcoPieDataItem {
189             ShowFlag=true,
190             PieType= TermEcoPieType.TermUmEcoPieTypeMovement,
191             PieTimeVal=0,
192             PieValues={b,a}
193         },
194     },
195
196
197 };
198
199 byte[] protoData = Any.Pack(data).ToByteArray();
200
201 TagEcoReport tagEcoReport = new TagEcoReport
202 {
203     EcoReportProtoData = protoData
204 };
205
206 //simulating message sent, received, decoded, event fired
207 //this triggers the data parsing methods
208 ConsoleBusiness.FireTagEcoDriveReportEvent(tagEcoReport);
209
210 if (testStep == 1) RunActionTimerExtensions.StartRunActionAfter(
211     TestJourney2, 2);
212 else if (testStep == 2) RunActionTimerExtensions.StartRunActionAfter(
213     TestJourney3, 2);
214 else
215 {
216     //stop activity
217     LoggerPlugin.LogAsync(LogLevel.Debug, "DriveGreenBusinessManager::
218         TestCode()", "FINISHED TEST");
219 }
220 }

```

Listagem B.1: Código de teste da recepção de dados das viagens.

C

Anexo C

Neste anexo é apresentado o questionário de usabilidade a disponibilizar aos utilizadores do novo módulo. O questionário foi criado em *Google Forms* [38] e através das referências [36] e [37] como já mencionado.

Avaliação do Módulo de Condução Sustentável da TeamMobile

Neste curto questionário são feitas algumas perguntas sobre a sua experiência com o novo módulo de condução sustentável da aplicação.

O questionário é anónimo. Agradecemos a sua participação!

Tempo Estimado de Resposta: 3 minutos.

*Obrigatório

1. Teve alguma dificuldade em utilizar/entender o novo módulo? Qual? *

Marcar apenas uma oval.

Não

Outra: _____

2. Achou o módulo intrusivo ou distrativo? Porquê? *

Marcar apenas uma oval.

Não

Outra: _____

3. Avalie a seguinte frase: *

"Este módulo disponibiliza informação valiosa para me ajudar na prática de uma condução melhor/mais sustentável."

Marcar apenas uma oval.

1 - Discordo

2 - Discordo parcialmente

3 - Neutro

4 - Concordo parcialmente

5 - Concordo

4. Avalie a seguinte frase:

*

"Este módulo **motiva-me** a praticar uma condução melhor/mais sustentável."

Marcar apenas uma oval.

- 1 - Discordo
- 2 - Discordo parcialmente
- 3 - Neutro
- 4 - Concordo parcialmente
- 5 - Concordo

5. O que **mais** gostou neste módulo?

6. O que **menos** gostou neste módulo?

7. Se pudesse **alterar** algo neste módulo, o que seria?

8. Se pudesse **adicionar** algo a este módulo, o que seria?

9. Avalie a sua experiência global face a integração deste novo módulo. *

Marcar apenas uma oval.

- 1 - Muito má
- 2 - Má
- 3 - Neutro
- 4 - Boa
- 5 - Muito boa

10. Outros comentários e observações.

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários

