

Mestrado em Engenharia Informática
Estágio
Relatório Final

MoCaS – Mobile Carpooling System

Álvaro Jorge Queirós Ribeiro
2011139150@student.dei.uc.pt

Orientadores:

Prof. Daniel Castro Silva

Prof. Pedro Henriques Abreu

Data: 2 de Setembro de 2014



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

Resumo

O *carpooling* é uma prática de partilha de automóvel que começou por ser adoptada nos Estados Unidos da América (EUA) durante os anos 70 quando surgiu a crise dos combustíveis. Desde então, depois de alguns altos e baixos, nestes últimos anos esta prática tem vindo a crescer e a transformar-se, sendo actualmente usada por todo o mundo.

Com a evolução das tecnologias móveis o *carpooling* teve a oportunidade de se expandir como nunca, principalmente através de aplicações móveis e páginas *web*. Com estas tecnologias é possível a qualquer pessoa em qualquer parte do globo procurar outras pessoas que pretendam ir para o mesmo local e queiram partilhar o automóvel. Com isto, pretende-se poupar dinheiro, poupar o ambiente, diminuir o congestionamento nas cidades, aumentar o número de lugares disponíveis para estacionar e conhecer pessoas novas.

O sistema MoCaS (*Mobile Carpooling System*) disponibiliza serviços de partilha de boleia em que cada utilizador terá de estar devidamente registado, de forma a garantir uma maior segurança. Neste sistema, cada utilizador pode inserir as suas viagens e fazer a sua marcação, atribuir avaliações, inserir os seus veículos (no caso dos condutores) e adicionar as suas preferências de viagem. Tudo isto é possível através de uma interface *web* e uma aplicação móvel que juntas dão um maior suporte a quem procura este tipo de serviços.

Os grandes pontos fortes deste sistema são precisamente na componente móvel em que através dos serviços de localização permite marcações de viagens em tempo real, ou seja, não só de viagens que ainda não iniciaram, mas de viagens que já estão a decorrer e que de alguma forma intersectam a posição do utilizador. Além desta novidade este sistema tem disponível um mapa em tempo real onde são visíveis todos os pontos de paragem previstos bem como a localização dos *carpoolers* que estão em viagem.

Através deste sistema pretende-se otimizar o uso desta prática tornando assim mais fácil e útil o seu uso no dia-a-dia das pessoas.

Palavras-Chave

Carpooling, *vanpooling*, *carsharing*, partilha de boleia, aplicação móvel, Android, ASP.NET, aplicação *web*.

Aos meus pais e avós.

Agradecimentos

Nesta fase final do meu percurso académico quero deixar um agradecimento a todos aqueles que de uma forma ou de outra me ajudaram a atingir este patamar.

Aos meus orientadores, pela sua disponibilidade e apoio, em especial ao Professor Daniel Castro Silva por todo o seu empenho, dedicação e esforço que contribuíram para o sucesso deste estágio.

À Estudantina Universitária de Coimbra, por tanto me ensinar e formar pessoalmente.

Aos meus amigos, por toda a amizade e apoio que ao longo destes anos me foram transmitindo. Em especial aos meus amigos Carlos da Silva, Pedro Laranjeiro, Jorge Vicente e Miguel Silva pelo apoio e disponibilidade demonstrada ao longo do estágio.

Ao Engenheiro Luís Nuno Silva, por todo o incentivo e conhecimento transmitidos.

Aos pais da minha namorada, por todo apoio, amizade e incentivo transmitidos ao longo dos anos.

À minha família, por todos os valores, apoio, ajuda e humildade que sempre me transmitiram. Por me acreditarem sempre que é possível ir mais além.

À minha irmã pelo carinho, apoio e confiança que me transmite.

À minha namorada, por toda a sua ajuda, incentivo, carinho, compreensão e dedicação. Pela paciência, que ao longo destes anos precisou. Por estar sempre presente nos bons e maus momentos.

Índice

Capítulo 1 Introdução	1
1.1. Motivação.....	1
1.2. Objectivos.....	2
1.3. Estrutura.....	2
Capítulo 2 Enquadramento e Trabalhos Relacionados	3
2.1. O <i>Carpooling</i>	3
2.2. Estudo sobre Sistemas de <i>Carpooling</i> Existentes	9
2.3. Comparação dos Sistemas Estudados	31
2.4. Evolução da Tecnologia Móvel.....	33
Capítulo 3 Requisitos e Arquitectura da Solução	37
3.1. <i>User Stories</i>	37
3.2. Visão Geral da Arquitectura	37
3.3. Diagramas de Casos de Uso.....	38
3.4. Diagrama de Componentes.....	39
3.5. Modelo de Base de Dados.....	40
3.6. Requisitos Funcionais	42
3.7. Requisitos não funcionais.....	45
Capítulo 4 Plano de Trabalho e Implicações	46
4.1. Planeamento	46
4.2. Riscos e Estratégias de Mitigação	48
4.3. Caminhos Críticos e Desvios.....	49
Capítulo 5 Escolhas Tecnológicas.....	51
5.1. Desenvolvimento da Aplicação Móvel	51
5.2. Desenvolvimento do <i>Website</i>	55
5.3. Base de Dados do Sistema	56
5.4. Comunicação Aplicação-Servidor	59
Capítulo 6 Implementação	62
6.1. MoCaS <i>Web</i>	62
6.2. MoCaS <i>Web Service</i>	75
6.3. MoCaS <i>Mobile</i>	77
Capítulo 7 Conclusão e Trabalho Futuro.....	86

7.1. Conclusão.....	86
7.2. Trabalho Futuro.....	87
Referências	88
Anexos.....	92
Anexo A – Descrição dos Casos de Uso.....	93
Anexo B – Protótipos das Aplicações Web e Móvel do Sistema MoCaS.....	94
Anexo C – Simbologia do Sistema MoCaS.....	96
Anexo D – Listagem dos Serviços Disponíveis no <i>Web Service</i>	97

Lista de Acrónimos

API	<i>Application Programming Interface</i>
BD	Base de Dados
DCPP	<i>Daily Car Pooling Problem</i>
EUA	Estados Unidos da América
GPS	<i>Global Positioning System</i>
HOV	<i>Highly Occupancy Vehicle</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDC	<i>International Data Corporation</i>
IIS	<i>Internet Information Services</i>
JS	Javascript
LTCPP	<i>Long-term Car Pooling Problem</i>
MoCaS	<i>Mobile Carpooling System</i>
MVCC	<i>Multi-Version Concurrency Control</i>
OPEC	Organização dos Países Exportadores de Petróleo
SGBD	Sistema de Gestão de Base de Dados
SMTP	<i>Simple Mail Transfer Protocol</i>
SOAP	<i>Simple Object Access Protocol</i>
SQL	<i>Structured Query Language</i>
REST	<i>Representational State Transfer</i>
TCP	<i>Transmission Control Protocol</i>
UI	<i>User Interface</i>

XML *eXtensible Markup Language*

WP8 *Windows Phone 8*

Lista de Figuras

Figura 1 – Propaganda sobre <i>Carpool</i> na década de 1940. Fonte: U.S. National Archives and Records Administration.	3
Figura 2 - Gráfico dos preços do petróleo entre 1861 e 2008. A linha laranja é ajustada pela inflação. Fonte: Wikimedia.org.	4
Figura 3 - Sinal de via exclusiva ao <i>carpooling</i> durante a crise do petróleo de 1973. Fonte: Wikimedia.org.	4
Figura 4 – Gráfico da percentagem de trabalhadores norte-americanos que partilham o carro até 2011. Fonte: Wikimedia.org.	5
Figura 5 - Processo de optimização de planeamento de viagem. Fonte: [22].	9
Figura 6 - Ecrã inicial do <i>website</i> <i>deboleia.com</i>	10
Figura 7 - Anúncios para partilha de boleira para o distrito de Coimbra do <i>website</i> <i>deboleia.com</i>	10
Figura 8 - Calculadora de preço médio de viagem por pessoa do <i>website</i> <i>deboleia.com</i>	11
Figura 9 - Ecrã inicial do <i>website</i> <i>pendura.pt</i>	11
Figura 10 - Listagem de ofertas disponíveis no <i>pendura.pt</i>	11
Figura 11 - Detalhe de oferta de boleia Sesimbra - Oeiras.	12
Figura 12 - Ecrã de pesquisa de viagens do <i>RoadSharing.com</i>	13
Figura 13 - Ecrã de pesquisa através do mapa do <i>RoadSharing.com</i>	13
Figura 14 - Ecrã de detalhe de viagem do <i>RoadSharing.com</i>	14
Figura 15 - Ecrã de eventos disponíveis do <i>RoadSharing.com</i>	14
Figura 16 - Ecrã de viagens associadas ao evento do <i>RoadSharing.com</i>	15
Figura 17 - Ecrã inicial da aplicação com pesquisa de início e fim de viagem do <i>BlaBlaCar.pt</i>	15
Figura 18 - Listagem de viagens Coimbra – Lisboa do <i>BlaBlaCar.pt</i>	16
Figura 19 - Informação detalhada de viagem do <i>BlaBlaCar.pt</i>	16
Figura 20 - Perfil de condutor do <i>BlaBlaCar.pt</i>	17
Figura 21 - Ecrãs de apresentação, filtragem e listagem da aplicação móvel <i>Toogethr</i>	17
Figura 22 – Grupos na aplicação móvel <i>Toogethr</i>	18
Figura 23 - Ecrã de detalhe de viagem da aplicação móvel <i>Toogethr</i>	18
Figura 24 - Filtros avançados da aplicação <i>Carpooling.co.uk</i>	19
Figura 25 - Listagem e detalhe de viagem da aplicação <i>Carpooling.co.uk</i>	20

Figura 26 - Marcação e confirmação do serviço da aplicação Carpooling.co.uk.....	20
Figura 27 - Formulário de Viagem do serviço de <i>carpool</i> da Universidade de Princeton, EUA.	21
Figura 28 - Página principal do serviço de <i>carpooling</i> da Universidade de Princeton, EUA.....	21
Figura 29 - Formulário destinado ao <i>carpooling</i> comercial do Jayride.com.au.....	22
Figura 30 - Ecrã inicial do <i>website</i> com a listagem das viagens com partida em Sydney na Austrália do Jayride.com.au.....	22
Figura 31 - Detalhe de uma viagem no Jayride.com.au.....	23
Figura 32 - Opção de destaque da viagem criada do Jayride.com.au.....	23
Figura 33 - Ecrã inicial do Boleia.net.....	24
Figura 34 - Informação de detalhe de viagem do Boleia.net.....	24
Figura 35 - Ecrã inicial do CarpoolWorld.com com a listagem das viagens recentemente adicionadas do CarpoolWorld.com.....	25
Figura 36 - Viagem a aguardar correspondências de viagens semelhantes de outros utilizadores do CarpoolWorld.com.....	26
Figura 37 - Correspondência encontrada com toda a informação detalhada do CarpoolWorld.com.....	26
Figura 38 - Funcionalidades de cada tipo de serviço e respectivos custos do CarpoolWorld.com.....	27
Figura 39 - Ecrã inicial do Zimride.com.....	28
Figura 40 - Detalhes da viagem seleccionada no Zimride.com.....	28
Figura 41 - Ecrã de filtragem da aplicação móvel Carma.....	29
Figura 42 - Perfil de utilizador (esquerda) e correspondência de viagem respectiva (direita) da aplicação móvel Carma.....	30
Figura 43 - Chat (esquerda) e confirmação de viagem com alguma informação da mesma da aplicação móvel Carma.....	30
Figura 44 - Número de vendas mundiais por dispositivos em 2013 com previsão para 2017, em milhões. Fonte: IDC.....	33
Figura 45 - Número de vendas mundiais por Sistemas Operativos Móveis do 3º trimestre de 2013 em comparação com o mesmo trimestre em 2012, em milhões. Fonte: Strategy Analytics.	35
Figura 46 - Quota de mercado mundial dos Sistemas Operativos Móveis do 3º trimestre de 2013 em comparação com o mesmo trimestre em 2012, em percentagem. Fonte: Strategy Analytics.....	36
Figura 47 – Diagrama com uma visão geral da arquitectura do sistema.....	38
Figura 48 - Diagrama de casos de uso para os utilizadores autenticados.....	39

Figura 49 - Diagrama de componentes do MoCaS.....	40
Figura 50 - Diagrama conceptual do sistema MoCaS.....	41
Figura 51 - Diagrama físico do sistema MoCaS.	42
Figura 52 - Mapa Gantt do 1º semestre.....	47
Figura 53 - Mapa Gantt do 2º semestre.....	48
Figura 54 - Mapa Gantt do planeamento real do 2º semestre.....	50
Figura 55 - Comparação de estratégias de desenvolvimento móvel. Fonte: [56].	51
Figura 56 - Vantagens e desvantagens dos tipos de desenvolvimento móvel. Fonte: [56].	52
Figura 57 - Arquitectura multiplataforma. Fonte: [56].	53
Figura 58 - Página inicial do <i>website</i> MoCaS.	63
Figura 59 - Listagem de viagens entre Lisboa e Porto do <i>website</i> MoCaS.....	63
Figura 60 - Detalhe de viagem do <i>website</i> MoCaS.....	64
Figura 61 - <i>Login</i> e registo do <i>website</i> MoCaS.....	64
Figura 62 - Perfil do utilizador do <i>website</i> MoCaS.	65
Figura 63 - Preferências de viagem do utilizador do <i>website</i> MoCaS.....	65
Figura 64 - Gestão de viagens do utilizador do <i>website</i> MoCaS.	66
Figura 65 - Definição do trajecto de viagem do <i>website</i> MoCaS.	66
Figura 66 - Definição do horário e data da viagem no <i>website</i> MoCaS.	67
Figura 67 - Detalhes da viagem no <i>website</i> MoCaS.....	67
Figura 68 - Veículos do utilizador no <i>website</i> MoCaS.....	68
Figura 69 - Formulário de inserção/edição de veículo no <i>website</i> MoCaS.	68
Figura 70 - Classificações recebidas no <i>website</i> MoCaS.....	69
Figura 71 - Detalhe de classificação recebida do <i>website</i> MoCaS.....	69
Figura 72 - Gestão de conta do utilizador no <i>website</i> MoCaS.	70
Figura 73 - Listagem de novas mensagens do utilizador no <i>website</i> MoCaS.....	70
Figura 74 - Mensagem de pedido de associação a viagem no <i>website</i> MoCaS.....	70
Figura 75 - Detalhe de uma mensagem recebida no <i>website</i> MoCaS.....	71
Figura 76 - Validadores no formulário de pesquisa no <i>website</i> MoCaS.....	71
Figura 77 - Os três indicadores de preço verde, amarelo e vermelho que significam preço baixo, moderado e alto respectivamente. Retirado do <i>website</i> MoCaS.....	72
Figura 78 - Inserção da localização "Lisboa" com suporte do Google Place Autocomplete no <i>website</i> MoCaS.....	72

Figura 79 - Percentagem de utilização de <i>web browsers</i> a nível mundial. Fonte: [70].	73
Figura 80 – Teste na ferramenta Screenfly [71] à resolução 1280x800 do <i>website</i> MoCaS.	73
Figura 81 - Resultados sobre os conteúdos carregados no 1º acesso ao <i>website</i> MoCaS.	74
Figura 82 - Resultados sobre os conteúdos carregados no 2º acesso ao <i>website</i> MoCaS.	75
Figura 83 – Pedido (esquerda) e resposta (direita) SOAP do serviço de pesquisa de viagens do utilizador no sistema MoCaS.	76
Figura 84 - Testes de performance ao servidor.	77
Figura 85 - Ecrã de apresentação da aplicação móvel MoCaS.	78
Figura 86 - Ecrã de <i>login</i> da aplicação móvel MoCaS.	78
Figura 87 - Ecrã de listagem de viagens do utilizador no sistema MoCaS.	79
Figura 88 - Ecrã de início de viagem em tempo real na aplicação móvel MoCaS.	79
Figura 89 - Mapa com as viagens a decorrer em tempo real na aplicação móvel MoCaS.	80
Figura 90 - Listagem de viagens por perto na aplicação móvel MoCaS.	80
Figura 91 - Listagem das próximas viagens na aplicação móvel MoCaS.	81
Figura 92 - Ecrã de parametrização de pesquisa na aplicação móvel MoCaS.	81
Figura 93 - Listagem de resultados de pesquisa na aplicação móvel MoCaS.	82
Figura 94 - Ecrã de detalhe de viagem da aplicação móvel MoCaS.	82
Figura 95 - Ecrã de notificação e resposta a um pedido de associação de viagem na aplicação móvel MoCaS.	83
Figura 96 - Barra de acção da aplicação móvel MoCaS.	83
Figura 97 - Percentagem de mercado de várias versões do Android. Fonte: [69].	83
Figura 98 - Protótipo do ecrã inicial do <i>website</i> MoCaS.	94
Figura 99 - Protótipo da listagem de viagens do <i>website</i> MoCaS.	94
Figura 100 - Protótipo do ecrã de detalhe de viagem do <i>website</i> MoCaS.	95
Figura 101 - Protótipos do ecrã de entrada (à esquerda) e <i>login</i> (à direita) da aplicação móvel MoCaS.	95
Figura 102 - Protótipos do ecrã de gestão (à esquerda) e pesquisa de viagens (à direita) da aplicação móvel MoCaS.	95
Figura 103 - Logotipo do sistema MoCaS	96
Figura 104 - Ícone da definição de condutor na aplicação móvel MoCaS.	96
Figura 105 - Ícone da definição de passageiro na aplicação móvel MoCaS.	96
Figura 106 - Marcador de posição do veículo no mapa de viagens em tempo-real na aplicação móvel MoCaS.	96

Figura 107 - Marcador de posição de pontos de paragem no mapa de viagens em tempo-real na aplicação móvel MoCaS.....	96
Figura 108 - Botão de pesquisa de viagens na aplicação móvel MoCaS.....	96
Figura 109 - Botão de actualização de viagens na aplicação móvel MoCaS.....	96
Figura 110 - Botão de <i>Logout</i> na aplicação móvel MoCaS.	96

Lista de Tabelas

Tabela 1 - Matriz comparativa dos sistemas de <i>carpooling</i> estudados (continua).	32
Tabela 2 - Funcionalidades do Sistema MoCaS.	44
Tabela 3 - Planeamento de actividades do primeiro semestre.	46
Tabela 4 - Planeamento de actividades do segundo semestre.	47
Tabela 5 - Planeamento real de actividades do segundo semestre.	49
Tabela 6 - Vantagens e desvantagens das tecnologias PHP, Java, ASP .NET.	56
Tabela 7 - Resultados resumidos do teste de performance ao <i>website</i> MoCaS.	74
Tabela 8 - Testes de performance ao servidor com número de clientes variável.....	77
Tabela 9 - Listagem de resultados aos acessos de <i>Login</i> na plataforma <i>web</i> e móvel do sistema MoCaS.	84
Tabela 10 - Descrição dos casos de uso do MoCaS.	93
Tabela 11 - Listagem e descrição dos serviços disponíveis no <i>web service</i> do sistema MoCaS.	100

Anexos

Anexo A – Descrição dos Casos de Uso

Anexo B – Protótipos das Aplicações Web e Móvel do Sistema MoCaS

Anexo C – Simbologia do Sistema MoCaS

Anexo D – Listagem dos Serviços Disponíveis no *Web Service*

Capítulo 1

Introdução

Em tempos de crise, o *carpooling* é cada vez mais indicado em situações que envolvem a condução diária para o trabalho ou escola. As pessoas que consideram usar o *carpooling* são principalmente aquelas que trabalham ou estudam na mesma área ou numa área próxima. Esta opção é especialmente uma boa ideia para as pessoas que não têm viatura própria ou as pessoas que não querem experimentar problemas associados às viagens nos transportes públicos. Além disso, é uma questão de escolha pessoal e de conforto a escolha do *carpooling*.

O *carpooling*, que também é conhecido como *ridesharing*, é composto por duas ou mais pessoas que viajam num veículo particular para o trabalho ou para outros destinos juntos. A ideia por trás deste conceito é que as pessoas que participam mais, mais dinheiro economizam.

A abordagem *carpool* inclui a decisão de quem vai estar a conduzir, combinando horários e dividindo as despesas de viagem como combustível e portagens. No entanto, a abordagem comum consiste apenas na partilha de boleia sem troca de dinheiro.

O *carpooling* é normalmente usado quando as pessoas têm origens e destinos similares, tornando mais fácil para os *carpoolers* chegar onde precisam, sem perder muito tempo e dinheiro. Com o uso desta prática cria-se uma ótima maneira de conhecer melhor os colegas de trabalho mantendo um ambiente confortável durante a viagem. No *carpooling* as viagens podem ser agendadas em uma base regular ou com pouca frequência dependendo das necessidades das pessoas. Entre outras vantagens, os condutores por vezes podem trocar com os passageiros, revezando-se durante viagens mais longas.

Hoje em dia, os congestionamentos e o aumento dos custos de combustível tornaram-se um problema mundial, o que resultou em mais pessoas a aderir a esta prática. Optar pelo *carpooling* traz uma série de benefícios, como a redução dos custos de viagem, o aumento do espaço de estacionamento, reduzindo a poluição do ar ajudando o meio ambiente. Embora o *carpooling* tenha grandes vantagens, ainda existem algumas preocupações como a dependência de terceiros e a fiabilidade.

1.1. Motivação

Num mundo em que a economia atravessa dias difíceis cada vez se torna mais importante a palavra “Poupar”. Portanto, escolher o *carpooling* é uma ótima forma de economizar algum dinheiro em tempos de crise. Esta prática cada vez mais popular nas grandes cidades tem ainda outras vantagens principalmente nas áreas urbanas onde diminui a poluição no ar e reduz os congestionamentos. Como esta prática implica menos carros a circular, também haverá um aumento nos lugares de estacionamento disponíveis.

No entanto o *carpooling* apresente ainda alguns problemas como a dependência de terceiros. Pois neste ponto tanto o condutor como os passageiros estão sujeitos a atrasos devido a desvios para apanhar passageiros e isto pode causar uma longa espera a todos os intervenientes. Outro problema desta prática é o facto de se viajar com desconhecidos, este problema revela-se como o principal, visto que a base para que haja mais pessoas a praticar o *carpooling* é a confiança.

Por estas razões propomos implementar um novo sistema com base em tecnologias móveis que incentive mais pessoas a esta prática, sobretudo em Portugal onde o *carpooling* é ainda pouco praticado.

1.2. Objectivos

Um dos objectivos do sistema proposto é resolver alguns dos problemas do *carpooling*, como viajar com desconhecidos e dependência de terceiros. Para tentar eliminar estes problemas o sistema a criar terá um registo obrigatório e é opcional a sua integração com uma rede social, como o Facebook. Através desta funcionalidade os utilizadores podem consultar informações como, dados pessoais, informação sobre o veículo, preferências de viagem e avaliações de qualquer utilizador registado no sistema. Quanto ao problema de depender de terceiros, será mitigado com a implementação de um sistema em tempo real que permite ao condutor poder aceitar ou não fazer um desvio para ir buscar um novo passageiro que precise de viagem no momento.

Outro objectivo é criar um sistema inovador em que as pessoas confiem para aumentar a prática do *carpooling* a nível nacional e internacional.

1.3. Estrutura

Este relatório encontra-se dividido em sete Capítulos, que começa pela introdução ao projecto e termina nas conclusões.

No capítulo 1 é feita uma introdução ao projecto, o seu conceito, a motivação, objectivos e estrutura do documento produzido.

No capítulo 2 são apresentados o conceito de *carpooling*, uma análise a alguns sistemas de *carpooling* existentes e uma análise às tecnologias móveis actuais que servirão de estudo ao sistema proposto.

O capítulo 3 mostra a arquitectura proposta, com uma descrição geral da arquitectura, descrevendo o sistema com diagramas de casos de uso, diagramas de componentes e *User Stories*, bem como os requisitos funcionais e não funcionais do sistema.

No capítulo 4, planeamento do trabalho e implicações, é apresentado o planeamento do trabalho realizado ao longo do ano através de uma distribuição de actividades de trabalho. Neste capítulo foram também analisados alguns riscos, bem como os processos de mitigação e desvios ao planeamento.

No capítulo 5 são apresentadas as escolhas tecnológicas para a implementação do sistema, tendo em conta toda a informação adquirida nos capítulos anteriores, com especial foco no capítulo 3.

O capítulo 6 revela toda a implementação do sistema proposto com maior ênfase nas inovações mais marcantes do sistema MoCaS. Neste capítulo estão também representados os testes ao sistema e respectivos resultados.

No último capítulo apresentam-se as considerações finais sobre o trabalho efectuado e sobre trabalho futuro.

Capítulo 2

Enquadramento e Trabalhos Relacionados

O objectivo deste capítulo é abordar o conceito de *carpooling*, falando um pouco da sua história e factores associados, analisar alguns sistemas de *carpooling* existentes, e analisar as tecnologias móveis actuais que servirão de estudo à aplicação proposta. Num primeiro momento serão analisados alguns processos de *carpooling*, conceitos, vantagens e problemas, comportamentos e práticas. Posteriormente faremos o estudo de vários sistemas existentes de *carpooling* e uma comparação entre eles identificando as suas potencialidades e lacunas.

Por fim será abordada a tecnologia móvel; esta análise permitirá conhecer possíveis plataformas para desenvolvimento bem como o seu mercado e rentabilidade. Será ainda abordado o conceito de multiplataforma para avaliar a possibilidade da aplicação a desenvolver poder estar disponível em mais do que um sistema operativo móvel. Por fim, será apresentada uma análise ao sistema de localização, um componente fundamental para o bom funcionamento do sistema a desenvolver.

2.1. O *Carpooling*

Nesta secção será introduzido o conceito do *carpooling*, com um breve contexto histórico. Serão abordadas também algumas características, problemas, vantagens, o contexto português e algumas práticas deste conceito.

2.1.1. Evolução Histórica

O conceito de *carpooling* surge pela primeira vez nos EUA durante a II Guerra Mundial, quando a escassez no fornecimento de combustíveis obrigou a uma utilização mais racional dos automóveis. Nessa altura optou-se por recorrer principalmente a campanhas de publicidade apelando ao sentido patriótico dos norte-americanos (Figura 1) [15].



Figura 1 – Propaganda sobre *Carpool* na década de 1940. Fonte: U.S. National Archives and Records Administration.

No final da II Guerra, o *carpooling* é abandonado e em apenas 10 anos a utilização dos transportes colectivos caiu 50% e a utilização do automóvel particular teve um crescimento muito rápido. Este termo só viria a ser falado novamente quando a 17 de Outubro de 1973 os membros Árabes da OPEC (Organização dos Países Exportadores de Petróleo) decidem embargar a exportação de petróleo para todos os países que ajudaram Israel na sua guerra com a Síria e o Egipto [12] e novamente em 1979 com a crise energética. Como se pode ver na Figura 2, nota-se um aumento acentuado no preço do petróleo em 1973, e novamente durante a crise energética de 1979 [13].

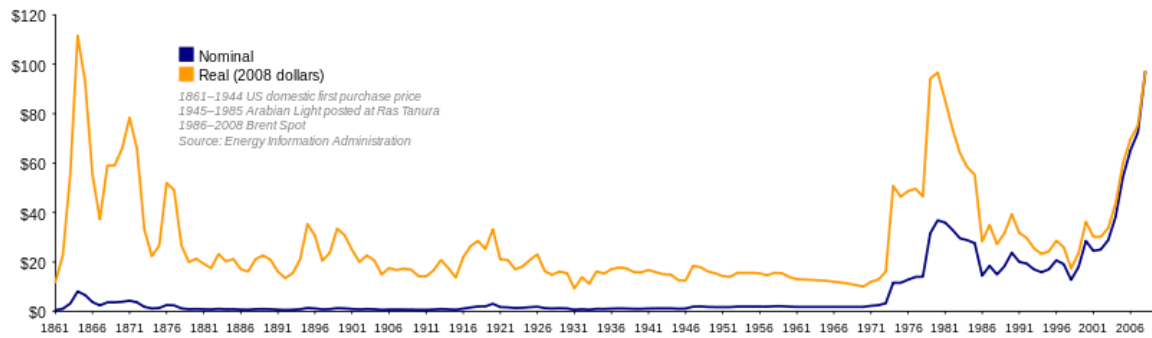


Figura 2 - Gráfico dos preços do petróleo entre 1861 e 2008. A linha laranja é ajustada pela inflação. Fonte: Wikimedia.org.

Com todos estes acontecimentos foi necessário poupar novamente na utilização dos combustíveis. No entanto, desta vez eram necessárias políticas concretas de incentivo à utilização da partilha do automóvel, as campanhas já não resolviam a situação. Foi então que em apenas dois meses surge a primeira medida concreta na região de Washington D.C, a faixa que era anteriormente utilizada apenas por autocarros na principal Auto-Estrada que liga o estado da Virgínia a Washington é aberta também a veículos com ocupação maior ou igual a 4 pessoas. Como se pode ver na Figura 3 o um sinal colocado na Auto-Estrada que liga o estado da Virgínia a Washington é um meio encorajador de apoio *carpooling* [13].



Figura 3 - Sinal de via exclusiva ao *carpooling* durante a crise do petróleo de 1973. Fonte: Wikimedia.org.

Reconhecendo a grande vantagem de juntar pessoas com destinos muito próximos, o governo norte-americano lança programas de partilha do automóvel nas Empresas e Universidades, onde obtém alguns resultados positivos como a Chrysler e a 3M que organizaram os primeiros *vanpools* de funcionários [13].

Apesar do *carpooling* em 1970 atingir o seu pico nos EUA, chegando ao valor de 20,4% de utilização, o *carpooling* acabou por cair abruptamente entre os anos 70 e 2000, tendo a sua utilização descido para 9,7% em 2011 [15].

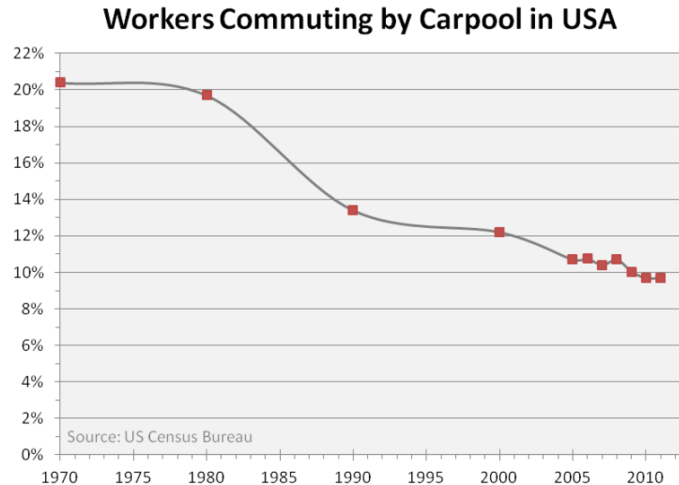


Figura 4 – Gráfico da percentagem de trabalhadores norte-americanos que partilham o carro até 2011. Fonte: Wikimedia.org.

Esta situação deveu-se essencialmente à grande queda nos preços dos combustíveis (45%) durante a década de 80 [16]. Além disso, o conceito de *Carpooling* foi mudando a partir do conceito "Dagwood Bumstead", em que cada condutor ia buscar os passageiros em sequência, num conceito "park and ride", onde todos os passageiros se encontravam num local comum [17].

Actualmente, com as novas tecnologias, como a internet e os sistemas de comunicação móvel foi possível às pessoas oferecer e encontrar viagens graças a vários *websites* e aplicações móveis criados para a prática do *carpooling*, o que levou a um crescimento significativo desta prática [15].

2.1.2. Em Portugal [44]

Um estudo feito pela Cetelem (Caderno Automóvel do Observador Cetelem) revela que a utilização conjunta da viatura desperta o interesse de 43% dos portugueses, este valor está acima da média dos oito países avaliados, em que a média era de 33%. O estudo revela ainda que esta prática de partilha já é utilizada por cerca de 6% da população nacional.

Este estudo baseou-se em inquéritos realizados nos meses de Maio e Junho de 2012, na Alemanha, Bélgica, Espanha, França, Itália, Portugal, Reino Unido e Turquia, tendo sido inquiridos um total de 4.830 indivíduos. Este estudo revela ainda que outra das opções de mobilidade que desperta o interesse dos portugueses é o *carsharing*, que correspondente a um serviço pago que permite ter acesso a uma viatura em regime self-service 24/24h, para utilizações ocasionais ou de curta duração, uma opção que se revela bastante útil para 36% dos portugueses, também acima da média europeia de 30%.

2.1.3. Vantagens e Desvantagens [19]

Uma melhor compreensão das motivações pessoais para a prática do *carpooling* pode ajudar muito para seu sucesso. Por exemplo, esforços promocionais de incentivo ao *carpooling* podem ser mais bem conseguidos se a motivação para esta prática for bem conhecida. Os factores motivacionais mais comuns são: a poupança económica, pois com a partilha de boleia haverá uma repartição de custos; a protecção do ambiente, que está directamente ligada a uma menor afluência de veículos a circular com a aplicação do *carpooling*; a socialização, em que as pessoas ao viajar com desconhecidos criam novas relações; a redução de congestionamentos

e o aumento de espaços livres para estacionamento, pois com menos carros a circular haverá mais espaço nas estradas e mais estacionamentos livres.

No entanto, também compreender as preocupações e as limitações em relação ao *carpooling* pode ajudar na formação de um sistema bem-sucedido. Por exemplo, se existir um caso em que um indivíduo que não fuma tenha de partilhar boleia com um fumador, este problema poderia ser resolvida com um processo de registo em que se incluía a preferência fumador ou não fumador nos critérios utilizados para fazer coincidir os *carpoolers*. Portanto, apesar de todas as vantagens, esta prática ainda apresenta algumas preocupações: o medo de viajar com estranhos, pois trata-se de uma prática que envolve vários tipos de pessoas; a perda de privacidade, que se reflecte quando uma pessoa não viaja sozinha; a dependência de terceiros, em que no caso do passageiro tem de esperar pelo condutor, e no caso do condutor que tem de ir buscar o passageiro criando aqui uma dependência.

2.1.4. Factores de Incentivo

Existem vários factores para que as pessoas escolham o *carpooling*. O primeiro factor está relacionado com as pessoas que escolhem o *carpooling* com base no seu rendimento pessoal, ou seja, as pessoas que recebem menos decidem recorrer mais a esta prática do que as pessoas que recebem mais; aqui os jovens são os mais propensos ao *carpooling*, assim como aqueles que vivem em grandes famílias e aqueles que são casados. Outro factor importante é a língua, uma vez que pode ser uma barreira ao *carpooling* porque poderá ser difícil para as pessoas que falam uma certa língua interagir socialmente com outras pessoas que falam outra [20].

A criação de grupos de *carpooling* também é um factor que se pode revelar importante e é relativamente fácil nos dias de hoje, devido à Internet. Existem actualmente vários *websites* disponíveis para corresponder os passageiros com grupos de *carpooling* - geograficamente, cronologicamente e por preferências pessoais (sexo, não-fumadores, etc) [21]. O aparecimento de vários recursos para a prática do *carpooling* também é um factor de incentivo pela facilidade com que aplicam a esta prática, exemplos disso são [22]:

- *Websites* públicos;
- *Websites* fechados (por exemplo para empregados);
- *Software* de *carpooling* (Aplicações móveis);
- Agências de *carpooling*;
- Pontos de recolha (não pré-arranjado).

A abordagem ao planeamento de viagem é um factor útil na sua gestão. A abordagem *carpooling* deve facilitar de forma fácil e rápida o agendamento das viagens com base nas preferências e necessidades de um viajante [22,23].

2.1.5. Tipos de Funcionamento

Segundo Calvo [24] o *carpooling* pode funcionar de duas maneiras distintas:

- *Daily Car Pooling Problem (DCPP)*: a cada dia um conjunto de utilizadores está disponível para ir buscar ou trazer de volta os seus colegas naquele dia em particular. Estes utilizadores são considerados “servidores”. O problema aqui consiste em atribuir “clientes” a “servidores” para identificar os itinerários a ser percorridos pelos “servidores” com o objectivo de minimizar os custos do serviço [24].

- *Long-term Car Pooling Problem (LTCPP)*: cada utilizador tem que agir como “servidor” e “cliente”, o objectivo é definir grupos de utilizadores, onde cada um, por sua vez, em dias diferentes, vai buscar os restantes membros do grupo. O objectivo aqui é minimizar a quantidade de veículos utilizados assim como reduzir a distância total percorrida por todos os utilizadores quando actuando como “servidores” [24].

De acordo com Buliung [25], Teal [47] identificou três tipos de *carpoolers*: domésticos, externos e passageiros. O *carpooler* doméstico consiste num cenário em que uma pessoa partilha o carro com pelo menos uma outra pessoa da mesma família, ao contrário do que acontece com *os externos* que partilham responsabilidades de condução com desconhecidos. Por fim, o *carpooler* passageiro partilha boleia com outros trabalhadores desconhecidos, mas só utiliza o serviço, pois nunca fornece um veículo. Em relação de todos estes tipos de *carpoolers*, o *carpooler* doméstico é o mais bem conseguido, uma vez que é mais fácil de formar e não é demorado nas paragens durante a viagem [25].

2.1.6. Tipos de Práticas

Existem três grupos principais de práticas de *carpooling* que são os seguintes [18]:

- ***Carpooling Informal (Slugging)***: Praticado em áreas que têm faixas HOV (*Highly Occupancy Vehicle*), estas faixas são vias exclusivas para veículos com mais de um ocupante. Geralmente os condutores não ficam com quaisquer contribuições dos passageiros pois passam a ter a vantagem de usar as faixas de HOV e chegam ao seu destino mais rápido. Contudo alguns passageiros por vezes contribuem para os custos da viagem.
- ***Carpooling Flexível***: Esta é a forma mais comum de *carpooling*, uma vez que os serviços baseados na Internet são relativamente fáceis de proporcionar uma viagem planeada com alguma antecedência. Estes sistemas são baseados em bases de dados que relacionam as preferências de viagem de condutores e passageiros para que os mesmos possam entrar em contacto e marcar um ponto de encontro com antecedência. Normalmente estes tipos de serviço exigem o registo no *website* para garantir a segurança dos condutores e passageiros.
- ***Carpooling em tempo real***: Existem vários serviços de internet e de comunicações móveis que suportam a correspondência de trajectos de viagem em tempo real. Nestes sistemas a correspondência de um pedido de viagem compatível com uma oferta deve ser de imediato combinada, a fim de manter o tempo de espera o mais curto possível.

Para se determinar a flexibilidade na abordagem ao *carpooling*, as formas de organização anteriormente mencionadas (informal, flexível e tempo-real) podem ser especificadas pelas seguintes características [18]:

- **Flexibilidade Espacial**: *carpooling* em tempo real num determinado ponto de busca de pessoas e *carpooling* informal que parte da presença simultânea de condutor e passageiro no local. Ambos podem determinar um ponto de encontro por diversos meios de comunicação.
- **Flexibilidade Temporal**: *carpooling* baseado na Internet que necessita de tempo para arranjar local e horário para o encontro. No entanto, os serviços baseados na internet também podem integrar a comunicação em tempo real através de

telefonemas já que envolve algum tempo de espera para uma correspondência correcta.

- **Flexibilidade de Função:** A flexibilidade de função é um papel essencial para a redução do tráfego. Há formas *carpooling* que usam os estacionamento como pontos de paragem, bem como no *carpooling* pré-combinado que dá oportunidade de negociar quem irá assumir a função de motorista.
- **Flexibilidade de Rota:** Com poucas excepções, os sistemas de *carpooling* assumem uma rota fixa a partir de um ponto de partida para um determinado destino. A maioria dos serviços de *carpooling* usam pontos de paragem fixos nas rotas. Os serviços baseados na Internet oferecem mais flexibilidade de rota desde os pontos de encontro que podem variar de acordo com o combinado entre motoristas e passageiros.

2.1.7. Convergência de Percurso vs. Convergência de Tempo

- **Convergência de Percurso** – Se condutor e passageiro têm o mesmo ponto de partida e o mesmo destino, então é dada uma correspondência de percursos. Os condutores e passageiros que partilham o mesmo destino geralmente pertencem à mesma escola ou local de trabalho, portanto o desafio é combinar o seu ponto de partida. Os condutores podem ir buscar os passageiros e deixá-los sair durante o percurso de viagem, neste caso o destino do condutor e do passageiro não tem de ser o mesmo. O objectivo principal para esta convergência é identificar que condutores e passageiros partilham o mesmo percurso de viagem; no entanto, muitas vezes não haverá uma correspondência perfeita de percurso. É pois importante que condutores e passageiros determinem uma distância aceitável do percurso para o ponto de encontro. Isto é fundamental para que o passageiro chegar ao ponto de encontro com facilidade, enquanto que o condutor terá a decisão de fazer um desvio de percurso para o ir buscar [18].
- **Convergência de Tempo** – Os condutores e passageiros não podem partilhar a viagem se o tempo desta não corresponder, mesmo que vivam na mesma casa ou trabalhem no mesmo local. Contudo, este tipo de convergência oferece uma certa tolerância desde que os parceiros de viagem nem sempre cheguem a horas certas ao seu trabalho ou escola. Enquanto que a convergência abordada anteriormente é determinada pelo percurso de casa para o trabalho ou escola, a convergência de tempo depende dos padrões de mobilidade e preferências individuais. É importante determinar uma tolerância de tempo com o objectivo de aumentar a probabilidade de uma correspondência [18].

Resumindo, a convergência de tempo é mais difícil de determinar do que a convergência de percurso. O percurso das viagens partilhadas pode ser flexível quando um condutor faz um desvio para ir buscar um passageiro, ou quando um passageiro caminha até um certo ponto de passagem. A solução para a correspondência de viagem está em permitir a condutores e passageiros negociarem directamente o ponto de encontro e o horário de partida que seja conveniente a ambas as partes [18].

2.1.8. Optimização de Planeamento de Viagem

A Figura 5 representa o processo de optimização de planeamento de viagem composto por dois motores [22]:

1. **Motor de encaminhamento baseado no tempo;**
2. **Motor de cálculo de consumo de energia;**

O motor baseado no tempo calcula rotas que visitam o máximo de lugares de interesse sob uma restrição específica de tempo. As rotas devem ser terminadas no mesmo ponto em que começaram. Os algoritmos deste motor podem ser divididos em três passos [22]:

1. **Geração de Rota Completa:** visita todos os locais de interesse ou locais turísticos;
2. **Geração de Rota Parcial:** para cada rota completa da primeira etapa, esta etapa irá calcular todas as rotas que alguns visitam ou todos os lugares turísticos;
3. **Geração de Rota Progressivo:** Numa selecção de rota exaustiva, este simplesmente procura por rotas que têm o máximo de locais sob um período específico de tempo. O período de tempo inclui tanto o tempo de viagem como o tempo gasto em cada local turístico

O motor de cálculo de consumo de energia selecciona a melhor rota que tem o menor consumo de energia. Cada carro tem diferentes consumos de energia (na cidade e nas auto-estradas). Existem três processos para calcular o consumo de energia [22]:

1. **Cálculo de Velocidade Média:** recolhe informações de cada rota, como a distância de viagem e o tempo entre dois pontos, a partir da API (*Application Programming Interface*) do *Google Maps* e usa essas informações para calcular a velocidade média das rotas;
2. **Cálculo do Consumo de Energia:** calcula o consumo de energia para um carro específico (marca e modelo) numa rota. O cálculo é realizado através da classificação do percurso em cidade ou em auto-estrada com base na velocidade média obtida na primeira etapa. De seguida, o cálculo utiliza os dados de fábrica do consumo de combustível (em km/l ou mpg) do carro de um viajante para calcular o consumo de combustível (em litros ou galões) da rota;
3. **Seleccção da Rota Designada:** selecciona a rota (chamada rota designada), que tem o menor consumo de energia de todas as rotas candidatas. Por fim, a rota designada é passada através do módulo de explicação de planeamento de viagem para gerar um plano de viagem detalhado.

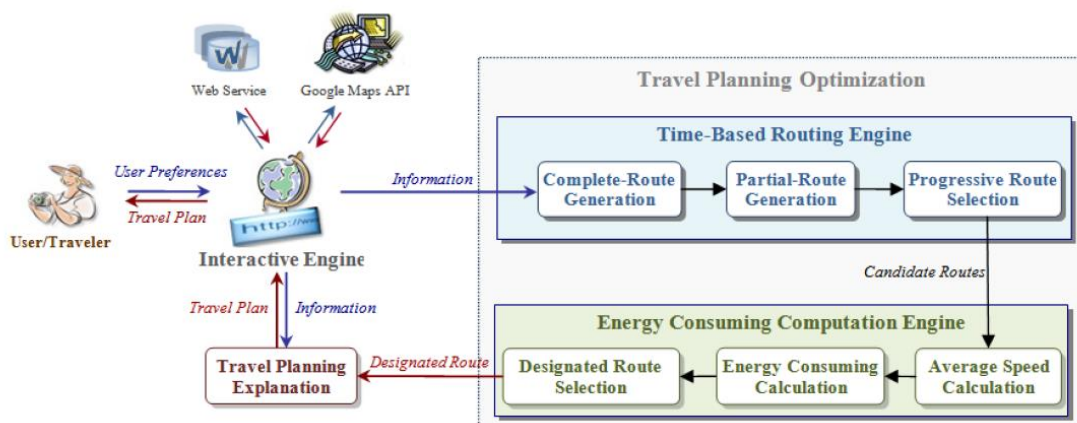


Figura 5 - Processo de otimização de planeamento de viagem. Fonte: [22].

2.2. Estudo sobre Sistemas de *Carpooling* Existentes

Nesta secção serão analisados vários sistemas de *carpooling* com intuito de recolher informação sobre potencialidades e problemas neste tipo de sistemas.

2.2.1. Deboleia.com [1]

O deboleia.com é um *website* português sem fins lucrativos e foi criado por dois licenciados da universidade do Minho em 2007. Este sistema foi baseado num semelhante existente na Bélgica chamado *Taxi Stop* e tem como principais objectivos minimizar os custos para quem viaja de automóvel, diminuir os níveis de dióxido de carbono, aumentar o número de estacionamentos livres nos locais para onde se viaja e promover a socialização.

Partidas de:	Tipos de Boleias
Aveiro	Conductor (25) Passageiro (13)
Beja	Conductor (8) Passageiro (7)
Braga	Conductor (28) Passageiro (24)
Bragança	Conductor (2) Passageiro (1)
Castelo Branco	Conductor (7) Passageiro (4)
Coimbra	Conductor (9) Passageiro (26)
Évora	Conductor (4) Passageiro (4)
Faro	Conductor (36) Passageiro (30)
Guarda	Conductor (5) Passageiro (3)
Leiria	Conductor (15) Passageiro (19)
Lisboa	Conductor (182) Passageiro (239)
Madeira	Passageiro (1)
Portalegre	Conductor (1) Passageiro (1)
Porto	Conductor (81) Passageiro (122)
Santarém	Conductor (15) Passageiro (3)
Setúbal	Conductor (30) Passageiro (14)
Viana do Castelo	Conductor (13) Passageiro (8)
Vila Real	Conductor (6) Passageiro (4)
Viseu	Conductor (24) Passageiro (13)
Outro País	Conductor (24) Passageiro (25)
Santiago de Compostela	Conductor (1)

Figura 6 - Ecrã inicial do *website* deboleia.com.

Figura 7 - Anúncios para partilha de boleia para o distrito de Coimbra do *website* deboleia.com.

É um sistema de *carpooling* que assenta numa página *web* onde quem quer pedir/dar boleia pode fazê-lo em forma de anúncio (ver Figura 7). Cada anúncio está dividido por distrito como se pode ver na Figura 6, podendo o utilizador descrever o tipo de serviço que irá prestar/solicitar. Para contactar quem publica estão disponíveis o nome, telefone (opcional) e *e-mail* como se pode verificar pela Figura 7. Para além deste serviço ainda existe a possibilidade de fazer uma estimativa do custo de viagem por pessoa através de uma calculadora fornecida no *website* (ver Figura 8).

The screenshot shows the website interface for deboleia.com. On the left is a navigation menu with links like 'Pesquisar Boleias', 'Pedir/Oferecer Boleia', and 'Calculadora'. The main content area features a 'Calculadora' section with a table titled 'Calculadora de valores da viagem'. The table has columns for 'Consumo L/100 km', 'Preço Combustível', 'Distância', 'Nº Pessoas', 'Portagens?', and 'Valor por Pessoa'. The input values are 4, 1.39, 100, 1, and 7, resulting in a value of 12.5599999. Below the table is a 'Calcular Limpar' button and a '[Voltar]' link. At the bottom right, the website name 'deboleia.com' is visible.

Figura 8 - Calculadora de preço médio de viagem por pessoa do website deboleia.com.

2.2.2. Pendura.pt [2]

O Pendura.pt foi criado por Carlos Pereira, Humberto Gonçalves e Diogo Castro quando estes se encontravam a trabalhar fora dos seus locais de residência. A aplicação foi construída com objectivo de promover a poupança de combustível, poupança de tempo, ajuda ao ambiente e a interacção com novas pessoas. Apesar deste *website* não ter como objectivo obter lucros através da promoção deste serviço, é utilizada a publicidade para colmatar custos de manutenção do *website*.

The screenshot shows the home page of Pendura.pt. It features a search bar at the top, a navigation menu on the left, and a main content area with a 'Começa Aqui!' button and several news items. At the bottom, there is a 'Notícias Trânsito' section with a map of Lisbon showing traffic conditions.

Figura 9 - Ecrã inicial do website pendura.pt.

The screenshot shows a list of carpooling offers on the Pendura.pt website. The list is titled 'Todas as boleias em oferta' and contains 20 entries. Each entry includes a number, type, cost, departure, destination, start time, and end time. The offers are listed in a table format.

#	Tipo	Custo	Partida	Destino	Início	Hora	Volta / Fim Hora
1	Reg.	Pergunta	Sesimbra	Oeiras	15.10.2013	Diferença	31.12.2013 Diferença
2	Reg.	Pergunta	Vila Nova de Famalicão	Ferrelros	08.10.2013	Diferença	08.04.2014 Diferença
3	Reg.	45 EUR	Porto	Lisboa	14.10.2013	Diferença	06.04.2014 Diferença
4	Reg.	Pergunta	Braga	Vila Nova de Famalicão	03.10.2013	Diferença	31.03.2014 Diferença
5	Reg.	Pergunta	Viseu	Tábua	26.09.2013	Diferença	28.02.2014 Diferença
6	Reg.	Pergunta	Caldas da Rainha	Torres Vedras	24.09.2013	Diferença	21.03.2014 Diferença
7	Reg.	Pergunta	São Domingos de Rana	Massamá	19.09.2013	Diferença	28.02.2014 Diferença
8	Reg.	Pergunta	Podence, 5340	Coimbra	20.09.2013	Diferença	20.12.2013 Diferença
9	Reg.	Pergunta	Porto	Vila Verde	19.09.2013	Diferença	19.12.2013 Diferença
10	Reg.	8 EUR	Lisboa	Leiria	20.09.2013	Diferença	14.03.2014 Diferença
11	Reg.	12 EUR	Lisboa	Portimão	13.09.2013	Diferença	20.12.2013 Diferença
12	Reg.	Pergunta	6000 Castelo Branco	6060 Idanha a Nova	16.09.2013	Diferença	25.10.2013 Diferença
13	Reg.	12 EUR	Portimão	Lisboa	09.09.2013	Diferença	23.12.2013 Diferença
14	Reg.	Pergunta	Porto Alto, 2135	Amadora	02.09.2013	Diferença	02.03.2014 Diferença
15	Reg.	7 EUR	6000 Castelo Branco	Alvalázere	02.09.2013	Diferença	01.03.2014 Diferença
16	Reg.	Pergunta	4755 Carvalhos	4760 Ribeira	29.08.2013	Diferença	28.02.2014 Diferença
17	Reg.	Pergunta	4740 Esposende	Braga	20.08.2013	Diferença	20.02.2014 Diferença
18	Reg.	Pergunta	4750 Galegos (Santa Maria)	4755 Cristelo	11.08.2013	Diferença	11.02.2014 Diferença
19	Reg.	Pergunta	Leja de Palmela	4900 Ponte de Barca	08.08.2013	Diferença	31.12.2013 Diferença
20	Reg.	Pergunta	Bombarral	Peniche	06.08.2013	Diferença	06.02.2014 Diferença

Figura 10 - Listagem de ofertas disponíveis no pendura.pt.

Este *website* apresenta uma listagem por ofertas/pedidos de boleia, notícias sobre o tráfego em algumas estradas nacionais e outras notícias ligadas ao trânsito (ver Figura 9 e Figura 10).

pendura.pt

segunda, outubro 14, 2013

Está em... Home / Ofertas

Menu

- Pendura
- Adicionar Boleia/Pedido
- Ofertas**
- Pedidos
- Pesquisa
- Notificação por email
- Remover subscrição
- Contactos

Penduras Online
Temos 3 visitantes em linha

Detalhes da oferta de boleia

Imprimir - Diz aos teus amigos! - Código QR - Reporta este anúncio aos administradores

Oferta regular de boleia oferecida de Sesimbra até Oeiras

Tipo e regularidade

Tipo	Regular - Semanalmente
Via e volta?	Sim, ida e volta!

Local e hora

Viagem de	Sesimbra até Oeiras
Data início	15.10.2013 até 31.12.2013
Segunda-feira	Viagem até 07:30 Viagem de volta 18:15
Terça-feira	Viagem até 07:30 Viagem de volta 18:15
Quarta-feira	Viagem até 07:30 Viagem de volta 18:15
Quinta-feira	Viagem até 07:30 Viagem de volta 18:15
Sexta-feira	Viagem até 07:30 Viagem de volta 18:15
A caminho passo por	Cotovia, 2970

Informações adicionais

Lugares disponíveis	3	Fumador: Sim
Tipo veículo	Automóvel	
Custo para cada viagem	Pergunta	

Informação do contato

Nome / Apelido	pedro baeta
Telefone	O número de telefone/telemóvel não foi publicado
Email	pbbaeta@live.com.pt

[Abrir formulário de contacto - clique aqui!](#)

Distância total: 50.9 km

Figura 11 - Detalhe de oferta de boleia Sesimbra - Oeiras.

Em cada inserção de pedido/oferta de boleia é possível ao utilizador indicar o tipo de regularidade da viagem, local e hora do ponto de partida, pontos de passagem durante a viagem, se é permitido fumar a bordo, tipo de veículo, lugares disponíveis e preço a pagar por cada passageiro. Para além de todas estas funcionalidades é ainda possível subscrever o serviço de notificações, que envia um *e-mail* ao utilizador com a informação de viagens disponíveis semelhantes à que procura ou pedidos com características semelhantes à que oferece.

2.2.3. RoadSharing.com [3]

O *website* RoadSharing.com foi projectado e construído pela empresa de Florença WebDev.it em 2008, uma empresa *web* fundada por um grupo de programadores, atendendo ao impacto social dos transportes no aquecimento global. Com o acumular de veículos a circular com apenas uma pessoa, testaram a sua ideia pela primeira vez com o *website* local, Viavai.it. Actualmente, este sistema conta já com cerca de 20.000 utilizadores e 35.000 viagens partilhadas, o que contribuiu para que este projecto tenha crescido e evoluído até aos dias de hoje.

The screenshot shows the RoadSharing website's search interface. At the top, there is a navigation bar with 'Home', 'Sign Up', 'Search Rides', 'Add Ride', 'Events', 'Contact', 'EcoPartner', and 'Press'. Below this is a search form with fields for 'Departure' and 'Arrival', both with '+/- km [?]' indicators. There are also checkboxes for 'Commuters', 'Type' (Offers, Requests, Offers and requests), and 'Cost contribution'. A 'Search' button and a 'Close advanced search' link are present. A 'Tip' box suggests searching by country or city. Below the search form is a table of rides with columns for 'Departure', 'Arrival', and 'Validity'. The table lists various routes, including those from Orbasano TO, Italia, Rivarolo Canavese TO, Italia, Pisa PI, Italia, Frankfurt Airport (FRA), 60547 Frankfurt, Gerns, Torino, Italia, Milan, Italy, Gorizia GO, Italia, and others.

Departure	Arrival	Validity
Orbasano TO, Italia	Rivarolo Canavese TO, Italia	15/10/13 - 01/10/14
Rivarolo Canavese TO, Italia	Orbasano TO, Italia	15/10/13 - 01/10/14
Pisa PI, Italia	Lecce LE, Italia	31/10/13
Frankfurt Airport (FRA), 60547 Frankfurt, Gerns	Rotterdam, The Netherlands	20/10/13
Torino, Italia	Granada, Provincia di Granada, Spagna	30/10/13 - 06/11/13
Milan, Italy	Marseille, France	18/10/13 - 20/10/13
Gorizia GO, Italia	8212 Čatež, Slovenia	30/10/13

Figura 12 - Ecrã de pesquisa de viagens do RoadSharing.com.

Este sistema de *carpooling* possui um filtro avançado de pesquisa com opções de viagens regulares, datas de partida e chegada e custos inerentes à viagem (ver Figura 12). Além deste filtro é ainda possível procurar viagens utilizando um mapa com pontos de concentração de oferta/pedidos de boleia (ver Figura 13).



Figura 13 - Ecrã de pesquisa através do mapa do RoadSharing.com.

Para além das funcionalidades mais comuns de indicação de locais, horas e percurso de viagem este *website* permite ainda atribuir um *rating* a cada utilizador, indicar a língua que fala, indicar se é um pedido ou uma oferta de boleia e ainda indicar se a viagem tem custos (ver Figura 14).

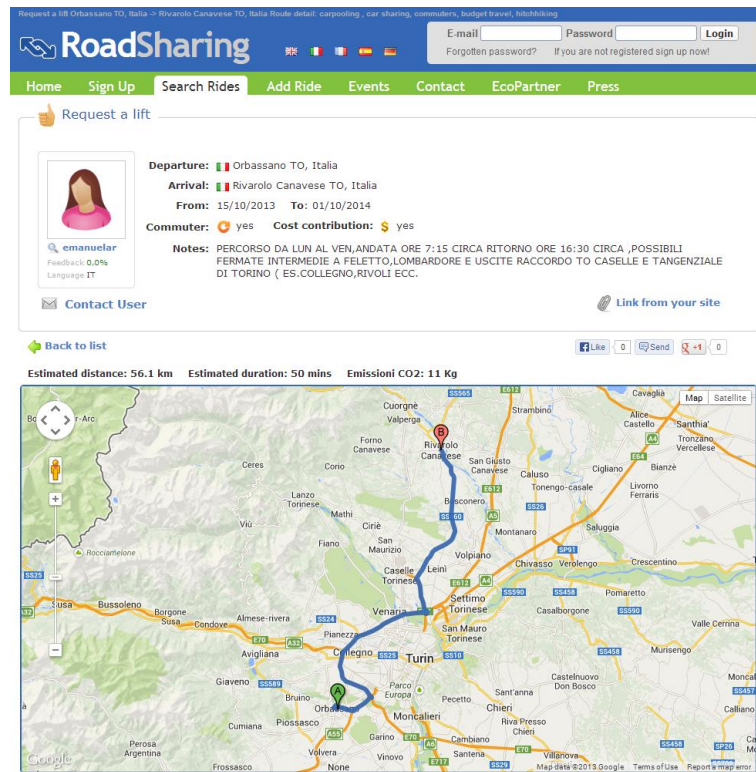


Figura 14 - Ecrã de detalhe de viagem do RoadSharing.com.

Deve ainda ser salientada a possibilidade de pesquisa de viagens por eventos importantes como por exemplo, festivais de Verão, concertos, feiras, entre outros (ver Figura 15).



Figura 15 - Ecrã de eventos disponíveis do RoadSharing.com.

Cada evento listado tem associados os itinerários que passam ou terminam no local onde se realiza o mesmo (ver Figura 16).

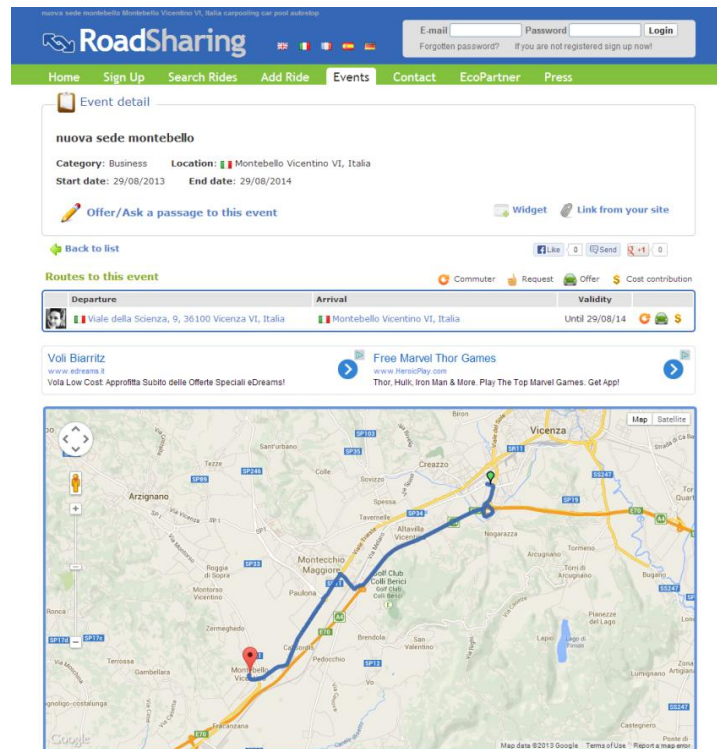


Figura 16 - Ecrã de viagens associadas ao evento do RoadSharing.com.

2.2.4. Blablacar.pt [4]

O BlaBlaCar foi idealizado por um dos fundadores da empresa, Frédéric Mazzella, um estudante na Universidade de Stanford. Este sistema de partilha de boleia assenta num *website* que mensalmente “transporta” mais de 550 mil passageiros, sendo hoje a maior empresa de *carpooling* da Europa. Tem como principais objectivos criar uma rede de transporte alternativa mais eficiente, amiga do ambiente, social e económica para milhões de pessoas em toda a Europa.

A aplicação baseia-se num *website* com um filtro simples de pesquisa com os pontos inicial e final da viagem pretendida (ver Figura 17).



Figura 17 - Ecrã inicial da aplicação com pesquisa de início e fim de viagem do BlaBlaCar.pt.

Ao pesquisar um determinado percurso são listadas as viagens disponíveis para o mesmo, são apresentadas informações resumidas de cada viagem listada com possibilidades de ordenação por preço e horário de viagem, e filtragem por preço, hora e data de partida, tipos de veículos e visualização de perfis com foto (ver Figura 18).

The screenshot shows the BlaBlaCar search results for the Coimbra to Lisbon route. The search parameters are Coimbra (origin) and Lisboa (destination). The results are sorted by price and time. Three results are visible:

Conductor	Time	Route	Price	Seats	Car
Sergio C (29 anos)	Domingo 20 Outubro - 13:00	Coimbra → Lisboa	10 € por passageiro	3 lugares livres	HONDA ACCORD ★★★
Vera C (32 anos)	Domingo 20 Outubro - 19:00	Coimbra → Lisboa	10 € por passageiro	4 lugares livres	
Rui M (29 anos)	Domingo 20 Outubro - 20:05	Porto → Coimbra → Lisboa	12 € por passageiro	4 lugares livres	VOLKSWAGEN GOLF V ★★

Figura 18 - Listagem de viagens Coimbra – Lisboa do BlaBlaCar.pt.

Para quem quer usufruir de uma das viagens listadas pode consultar a sua informação em detalhe. É fornecida informação do tipo de viatura, do tipo de bagagem que se pode transportar, se a viagem é só para mulheres, se há flexibilidade na hora de partida bem como no desvio da rota para ir buscar alguém, mapa da viagem com o trajecto e respectivos locais por onde passa o condutor, preço por pessoa, número de lugares livres e o perfil do condutor (ver Figura 19).

The screenshot shows the detailed view of a BlaBlaCar trip. The route is Santa Clara to Lisbon. The price is 12 € per passenger, and there are 3 free seats. The conductor is Rui D, 33 years old. The departure is on Friday, October 24, at 18:40. The trip details include a maximum deviation of 15 minutes, departure on time, medium baggage, and a BMW 320 vehicle.

Figura 19 - Informação detalhada de viagem do BlaBlaCar.pt.

Um aspecto muito importante em relação a esta aplicação é o perfil do condutor, onde está definido se o condutor conversa nas viagens, se gosta de ouvir musica, se permite transportar animais ou se permite fumar durante as viagens. Para além disto é visível a classificação atribuída pelos passageiros, o tipo de carro que utiliza e as suas últimas actividades na aplicação (ver Figura 20).



Figura 20 - Perfil de condutor do BlaBlaCar.pt.

2.2.5. Toogethr [5]

Toogethr é o nome de uma *startup* holandesa que criou uma aplicação móvel de *carpooling* com o mesmo nome. Esta aplicação nasceu com os objectivos: tornar o *carpooling* no método de transporte mais utilizado por todos os holandeses, baixar custos de viagem e a criação de novas amizades conhecendo pessoas novas.

Esta aplicação está disponível para os sistemas operativos Google Android e Apple iOS. Nesta aplicação cada utilizador tem de estar registado para usufruir das suas funcionalidades; é possível filtrar viagens por locais de partida e chegada, ver trajectos por perto da localização actual da pessoa que utilizada a aplicação, e eventos próximos da data actual da pesquisa (ver Figura 21).



Figura 21 - Ecrãs de apresentação, filtragem e listagem da aplicação móvel Toogethr.

Esta aplicação acrescenta uma nova funcionalidade em relação às anteriormente referidas, que é a possibilidade de criar grupos em que pessoas com procuras de trajectos comuns podem seguir para assim estarem sempre informadas caso existam viagens do seu interesse, como por exemplo grupos de festivais de Verão, feiras gastronómicas, etc. (ver Figura 22).

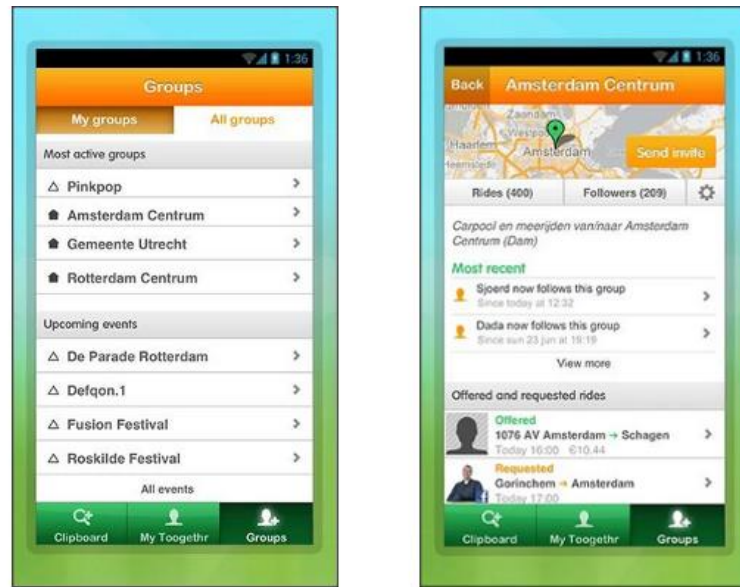


Figura 22 – Grupos na aplicação móvel Togethr.

Ao escolher uma viagem é possível ver informação mais detalhada acerca da mesma e contactar a pessoa que oferece o serviço. As informações apresentadas são muito simplificadas, apenas são apresentados os custos da viagem, hora da partida, locais de partida e chegada, mapa e duração da viagem. Para confirmar a viagem é necessário apenas utilizar o botão *“join ride”* que por sua vez envia uma mensagem formatada ao prestador do serviço que posteriormente irá confirmar ou não a viagem (ver Figura 23).

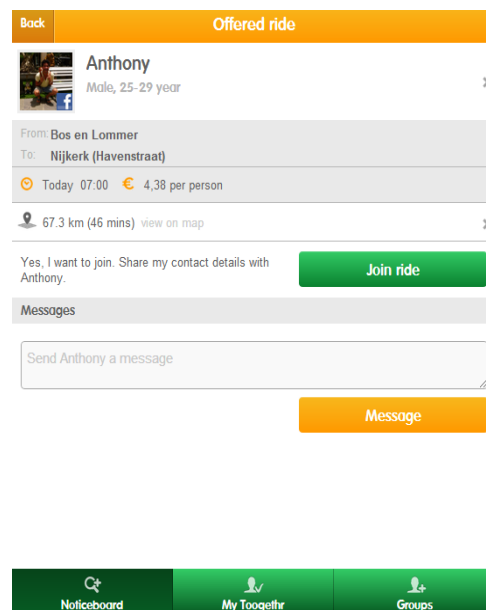


Figura 23 - Ecrã de detalhe de viagem da aplicação móvel Togethr.

2.2.6. Carpooling.co.uk [6]

O Carpooling.co.uk é o *website* do Reino Unido da Carpooling.com, a rede de partilha de boleias líder no mundo, movimentando mais de 1 milhão de pessoas por mês em 5000 cidades, em 45 países. O Carpooling.co.uk pretende uma partilha de viagens de forma segura e de fácil acesso através da sua aplicação *web* e móvel com integração com Facebook. Está disponível em 7 línguas e em mais de 40 países (incluindo Alemanha, França, Itália, Grécia, Polónia, Reino Unido, Áustria, Suíça e Espanha), o *website* ajuda mais de 4 milhões de utilizadores registados a conectarem-se com mais de 750 mil oportunidades de transporte. Para este estudo será abordada a aplicação móvel disponível para Google Android e Apple iOS.

Esta aplicação além de um filtro básico de pesquisa possui ainda opções avançadas de pesquisa, como o tipo de serviço (pedido/oferta de boleia), se é permitido fumar se é permitida a reserva prévia do lugar e se a viagem é partilhada apenas com mulheres (ver Figura 6).



Figura 24 - Filtros avançados da aplicação Carpooling.co.uk.

Ao inserir os dados de pesquisa, são listadas as viagens disponíveis com a informação relativa aos lugares disponíveis e se é permitida a pré-reserva de lugar.

Depois de seleccionar a viagem pretendida é apresentada a informação detalhada da viagem. Através desta selecção é possível visualizar informação sobre o perfil do condutor, o seu *rating* (avaliação atribuída ao condutor pelas pessoas que já viajaram com o mesmo), o seu nível de actividade na aplicação, o veículo que possui e o custo por pessoa (ver Figura 25).

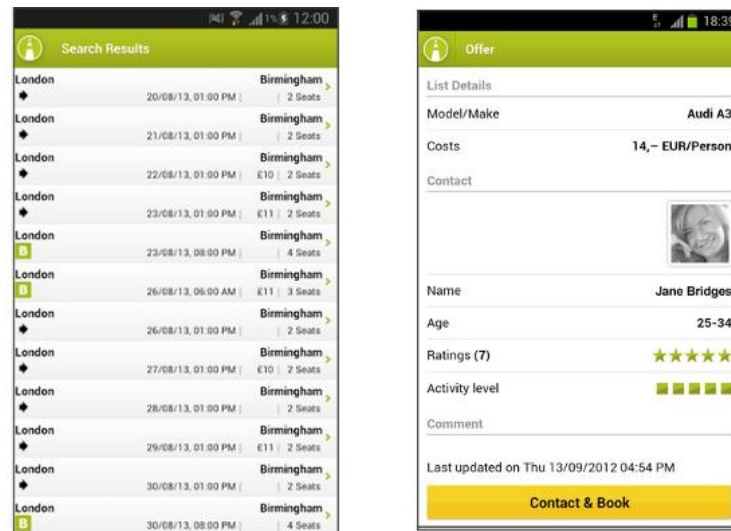


Figura 25 - Listagem e detalhe de viagem da aplicação Carpooling.co.uk.

Para contactar o condutor cada utilizador pode contactar através de mensagem privada da aplicação, ou caso seja possível a pré-reserva de lugar, o lugar é marcado tendo em conta o número de lugares livres podendo adicionar mais do que um passageiro indicando o nome de cada um (ver Figura 26).

Depois de marcados os lugares, o condutor recebe uma notificação através de uma mensagem privada. O condutor terá de dar resposta afirmativa à mensagem para que o passageiro veja a viagem como confirmada e planeada.

O pagamento dos custos de viagem é efectuado presencialmente antes do início da viagem (ver Figura 26).

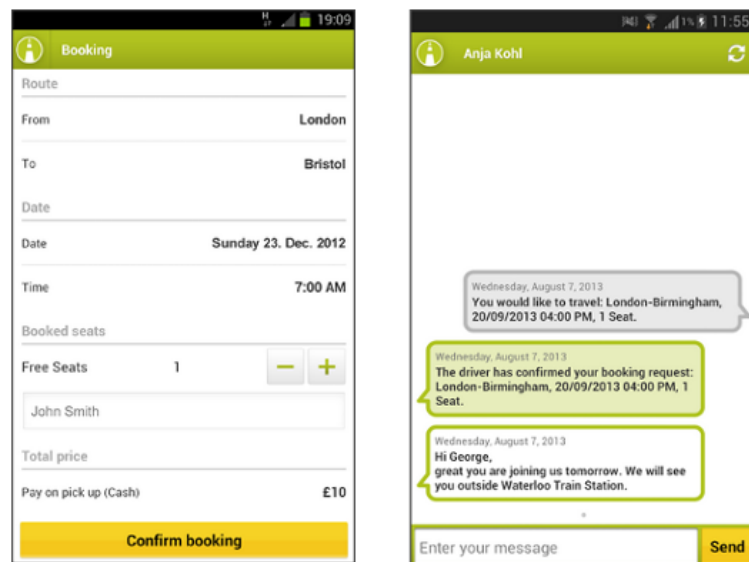


Figura 26 - Marcação e confirmação do serviço da aplicação Carpooling.co.uk.

2.2.7. Princeton University Carpool Service [7]

O serviço de *carpool* da Universidade de Princeton nos Estados Unidos é um serviço de combinações de boleias gratuito para toda comunidade desta universidade e a sua participação é voluntária. O programa fornece aos candidatos interessados uma lista com as correspondências de outros membros da comunidade da universitária que vivem e trabalham perto deles e querem partilhar as suas boleias. Os viajantes podem assim escolher as suas

próprias boleias para atender às suas necessidades de deslocação. Este serviço está disponível apenas para o corpo docente e pessoal da Universidade de Princeton.

Para utilizar este serviço é necessário estar registado na aplicação web fornecendo o endereço de *email* da Universidade de Princeton para o efeito. Posteriormente é necessário preencher os critérios de viagem (destino, partida, hora de partida, se é permitido fumar, se há musica durante a viagem, se o condutor partilha a condução, se a viagem é frequente) (Figura 27).

Figura 27 - Formulário de Viagem do serviço de *carpool* da Universidade de Princeton, EUA.

A informação é inserida na base de dados, e a partir desse momento é só esperar que alguém entre em contacto ou então pode ser consultada a lista de viagens disponíveis através do botão "Show Detailed List of Rides" para ver se alguém corresponde aos critérios pretendidos. (Figura 28)

Figura 28 - Página principal do serviço de *carpooling* da Universidade de Princeton, EUA.

Para incentivar esta prática a universidade atribui aos funcionários que se deslocam através deste meio um cartão de 50 dólares, a cada três meses enquanto o *carpool* permanece em funcionamento. Para os novos participantes a universidade destina um certificado de 25 dólares para mudança de óleo se estes se inscreverem no programa. Para além destes benefícios os *carpoolers* têm locais de estacionamento reservados.

Para os alunos a Universidade de Princeton oferece um serviço de correspondência de viagens semelhante ao anterior para os alunos que procuram uma boleia para casa ou estão dispostos a dar uma boleia durante as paragens escolares ou viagens ao fim de semana. O modo de funcionamento é semelhante ao anterior direccionado para os funcionários, no entanto este serviço está disponível apenas para estudantes da Universidade de Princeton.

2.2.8. Jayride.com.au [8]

O Jayride.com é um *website* australiano de partilha de boleia que une serviços grátis e pagos na mesma plataforma. Os serviços grátis são para uso particular, em que uma pessoa quer viajar e pode oferecer ou pedir uma boleia, enquanto que os serviços pagos vão desde transportes de autocarro a transportes de empresas privadas (por exemplo Limusines). Os serviços pagos têm percursos predefinidos tipicamente ligando cidades entre si ou aeroportos e cidades. Todos os serviços comerciais, ou seja, os que são pagos por quem anuncia, não podem ser criados através do *website*. Para anunciar um serviço comercial a entidade tem de entrar em contacto com a Jayride através do seu formulário (Figura 29) de contacto e posteriormente é colocado o percurso com as respectivas condições.

Join Jayride today
Jayride listings are designed to make you extra sales.

- ✓ Be found by people searching online
- ✓ Take bookings by phone or email
- ✓ List any transport
- ✓ Link to your website
- ✓ Add full business details
- ✓ Logos, timetables, maps and more
- ✓ Be found by locals and travelers

Create listings for your transport services, an online profile for your transport business, fill seats, reach new customers and more.

Contact us to learn more

Company Transport type

Your name Your position

Your phone number Your email address

Route, region or other comments

SEND

Figura 29 - Formulário destinado ao *carpooling* comercial do Jayride.com.au.

Para um utilizador banal que pretenda a prática de *carpool* o *website* apresenta-se com uma listagem de todas as viagens disponíveis para todos os serviços anteriormente falados. Pode ser feita uma pesquisa com os locais de início e fim de viagem, bem como filtrar o tipo de serviço desejado (Figura 30).

JAYRIDE Search List Your Ride

From Map to Map

Home / Search results

Sydney NSW → Anywhere

Search Results

Show: Airport Transfer Rideshare Other Transport

	<p>Herston, QLD 4006 ... Marion Bay, TAS 7175</p> <p>Rideshare with AdrianaWebster</p> <p>Runs once only (Thursday 26 December)</p>	\$100.00 per seat
	<p>Sydney, NSW 2000 ... Lorne, VIC 3221</p> <p>Rideshare with MarcCurrie-Nguyen</p> <p>Runs once only (from Sunday 29 December)</p>	\$100.00 per seat
	<p>Sydney, NSW 2000 Melbourne, VIC 3000</p> <p>Rideshare with Petershidler111</p> <p>Runs regularly (5 days per week)</p>	Free per seat

Figura 30 - Ecrã inicial do *website* com a listagem das viagens com partida em Sydney na Austrália do Jayride.com.au.

Após a selecção da viagem é possível ver todo o detalhe da viagem, desde o condutor, ao trajecto que será percorrido, horas de viagem, locais, se é permitido fumar, os lugares disponíveis, o custo da viagem, se a viagem é frequente ou não e o comentário deixado pelo condutor caso deseje (Figura 31).

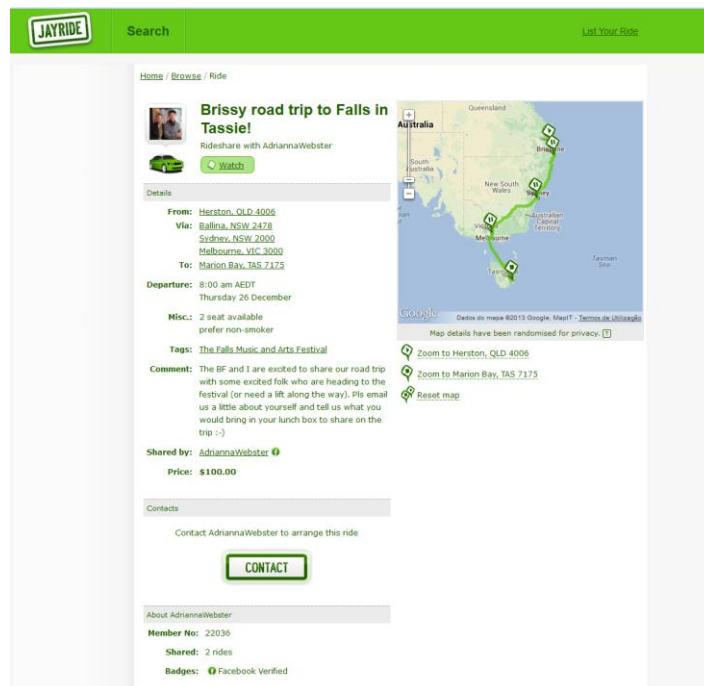


Figura 31 - Detalhe de uma viagem no Jayride.com.au.

Cada utilizador não comercial também pode promover a sua viagem pagando 3,5 dólares ao Jayride para colocar o seu anúncio no topo da página por 14 dias e assim atrair mais possíveis *carpoolers*.

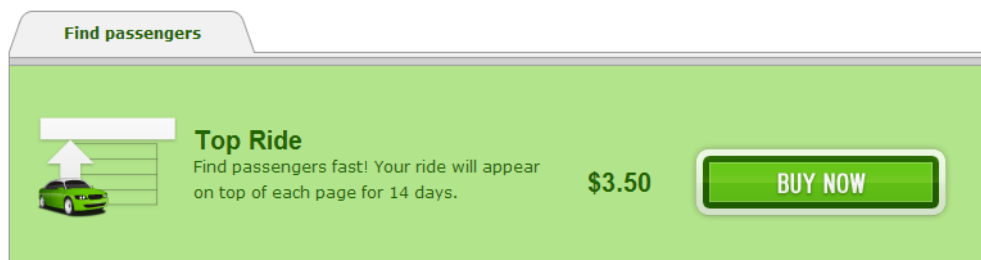


Figura 32 - Opção de destaque da viagem criada do Jayride.com.au.

2.2.9. Boleia.net [9]

Este *website* português foi criado em Julho de 2013 por Toni Jorge e tem como principais objectivos a poupança económica, a protecção ambiental e o conhecimento de novas pessoas. A forma de subsistência desta plataforma é conseguida através da publicidade.

Funcionalmente, o *website* apresenta no ecrã inicial uma listagem de viagens recentes com a informação do número de lugares, preço, horas e datas de partida e chegada. Existe também a possibilidade de filtrar as viagens por local de partida e destino, por tipo de viagem (oferta ou pedido) (Figura 33).

Além das listagens de viagens é possível efectuar uma listagem de avaliações de membros registados no sistema (Figura 33).



Figura 33 - Ecrã inicial do Boleia.net.

Após a selecção da viagem pretendida podemos ver mais informação sobre a mesma como comentários adicionais, regularidade de viagem, duração, distância percorrida e o mapa da viagem (Figura 34). O contacto feito com o condutor é feito por troca de mensagem através da plataforma (Figura 34).

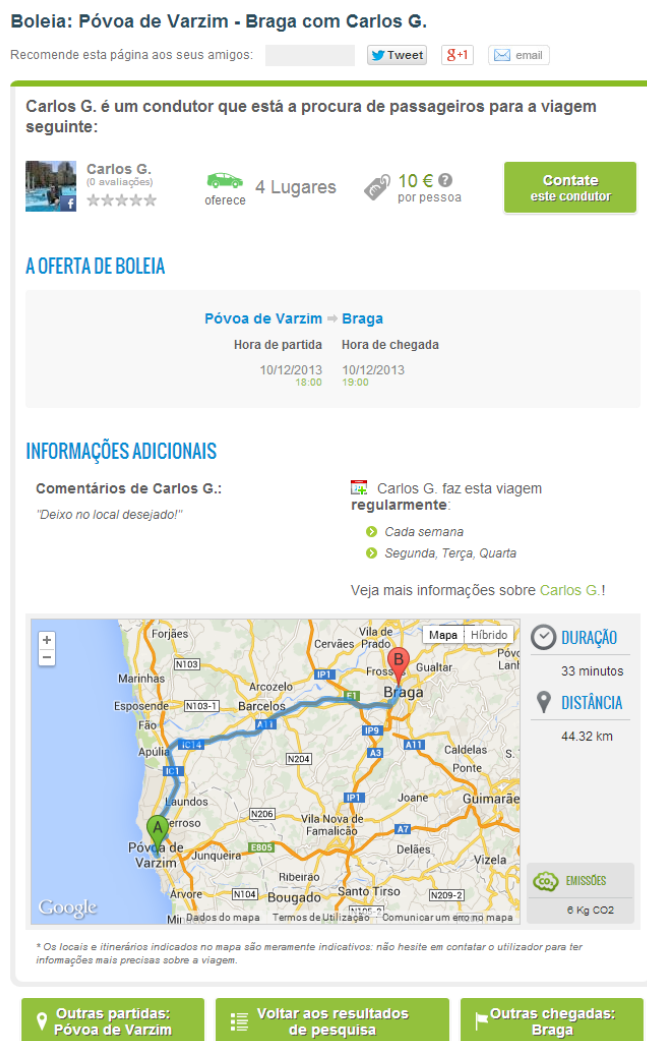


Figura 34 - Informação de detalhe de viagem do Boleia.net.

Além de todas estas funcionalidades, é ainda possível ver o membro que fornece ou pede o serviço, como a possibilidade de ver toda a informação do seu perfil (avaliação, idiomas falados, cidade de residência, ultimas viagens).

2.2.10. CarpoolWorld.com [10]

O Carpoolworld.com é um *website* criado em 2000 por Max Fox e Isabelle Boulard. Esta aplicação pretende criar ao utilizador benefícios económicos, ambientais e sociais. A forma de subsidiar a aplicação é feita através de anúncios publicitários e de pagamentos do serviço de grupos de *carpooling premium* (descritos abaixo).

Ao aceder ao *website* é mostrada uma listagem de pedidos e ofertas de viagens por todo o mundo. É possível filtrar essa informação por local de início e fim de viagem, distância desses locais, e se se pretende procurar pedidos e/ou ofertas de boleia (Figura 35).

The screenshot displays the CarpoolWorld.com website interface. At the top, there is a navigation bar with 'Home' and 'Help' links, and a search bar. Below the navigation bar, there are three main sections: 'Free carpool matching', 'Free carpool groups', and 'Premium carpool groups'. The 'Free carpool matching' section includes a text input for trip origin and destination, and a 'Join now' button. The 'Free carpool groups' section includes a text input for organizing a carpool club, and a 'Start here' button. The 'Premium carpool groups' section includes a text input for providing a complete carpool matching service, and a 'Learn more' button.

Below these sections, there is a section titled 'Most recent carpool listings around the world shown below.' This section contains a list of recent listings, including:

- Carpooling Amsterdam → London:** Passenger, 2 Seats required, December 28 9am, split costs, United Kingdom, One-Time Trip. Description: Two friendly, educated males looking for a ride from Amsterdam to London and / or back around the holidays. We are spending the holidays in Europe and are more than happy to split costs.
- Carpooling Peterborough → Lindsay:** Passenger, 1 Seat required, Wed Thu Fri Sat, split costs, Canada, Weekly Commute.
- Carpooling Montréal Lionel Groulx → Burlington:** Passenger, 1 Seat required, December 13 9am, split costs, United States, One-Time Trip.

On the left side of the dashboard, there is a 'Register for free!' button and a list of steps: 1. Register your trip or commute, 2. Confirm your email, 3. Contact your matches. Below this, there is a 'Carpoolworld market' section with a link to 'Carpoolworld decals \$2-\$5'. At the bottom left, there is a 'Quick search' form with fields for 'From' and 'To' (both set to '+/-90km (56mi)'), and a radio button for 'I am a:' with options for 'Passenger only', 'Driver only', and 'Driver or Passenger'. A 'Search' button is located below the form.

Figura 35 - Ecrã inicial do CarpoolWorld.com com a listagem das viagens recentemente adicionadas do CarpoolWorld.com.

Neste *website* o processo de correspondência de viagens é automático, ou seja, os utilizadores têm de se registar e no registo indicam a origem e o destino da viagem, a data e hora, regularidade, comentários adicionais, se permite fumar durante a viagem, se pretende viajar só com homens ou mulheres, o custo, se está interessado em *carpooling/vanpooling* ou outro serviços e se pretende pedir ou dar boleia. Depois de inserida toda essa informação, o

sistema activa de imediato a pesquisa de viagens semelhantes e a viagem criada fica em estado activo até ser fechada pelo utilizador ou pelo sistema (Figura 36).

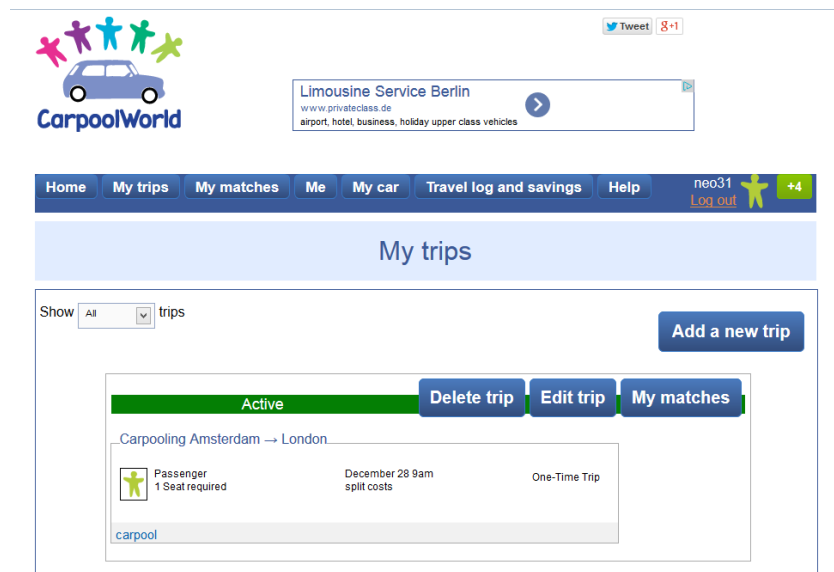


Figura 36 - Viagem a aguardar correspondências de viagens semelhantes de outros utilizadores do CarpoolWorld.com.

Posteriormente o sistema apresenta uma lista de viagens correspondentes, onde o utilizador pode escolher a viagem pretendida e entrar em contacto (através do telefone, e-mail ou SMS) com o utilizador que fornece o serviço para planear a viagem (Figura 37).

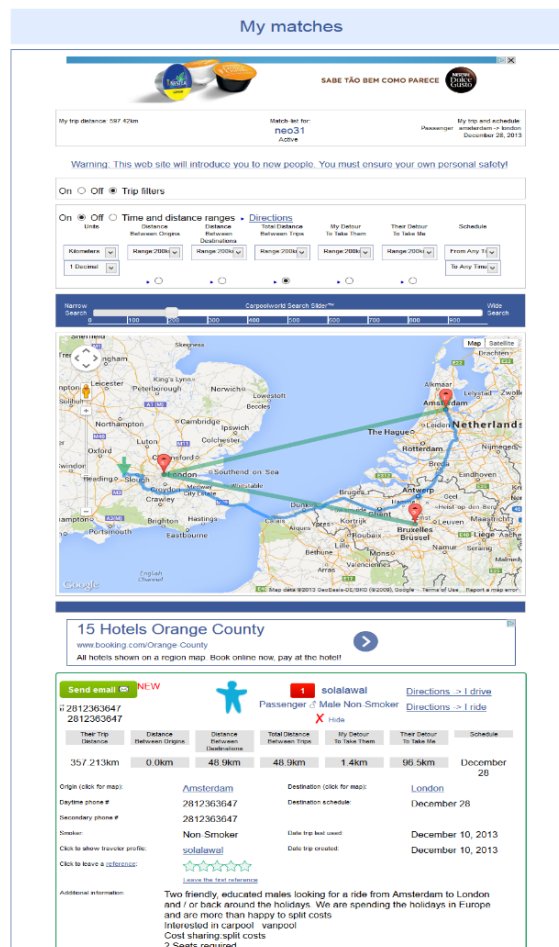


Figura 37 - Correspondência encontrada com toda a informação detalhada do CarpoolWorld.com.

O Carpoolworld.com também fornece serviços de grupo para qualquer negócio, município, escola ou outra instituição que queira administrar o seu próprio *website* de *carpooling*. Carpoolworld.com oferece grupos livres com características básicas, e grupos *premium* com recursos avançados que permitem a personalização de uma página *web* com as funcionalidades do CarpoolWorld.com de acordo com a instituição que contrata este serviço (Figura 38).

The screenshot shows the CarpoolWorld.com website interface. At the top, there is a navigation bar with 'Home' and 'Help' links. Below this, three service options are presented in boxes: 'Free carpool matching', 'Free carpool groups', and 'Premium carpool groups'. Each box contains a brief description and a 'Join now' or 'Start here' button. The main content area is titled 'Carpool matching software for all' and features a comparison table for the three service types.

	Individuals	Free carpool groups	Premium carpool groups
Pricing	Free	Free	\$25/month per 1,000 users. \$5000/year per 100,000 users. Cancel anytime.
Features	<ul style="list-style-type: none"> Register your daily commute or one-time trip as a driver or passenger Check your matchlist and contact other travelers to make your ridesharing arrangements Track your commuting habits and calculate your savings 	<ul style="list-style-type: none"> Choose a group template Use a generated link to tell your community about the group that you just created so that they can register for free to find matches and start carpooling Add or change destinations for work, school, events Non-intrusive advertisements are displayed to your users <p>Find out more</p>	<ul style="list-style-type: none"> Customize your carpool group home page and upload your logo Customize your carpool group preferences Customize the look and feel of your page and banner Promote your carpool group to your community using the Carpoolworld invitation module You can restrict registration to users that received printed invitations or email invitations Track your carpool group's efficiency using the Carpoolworld statistics module Import and export membership data No third-party advertising Flexible billing options <p>Find out more</p>

Figura 38 - Funcionalidades de cada tipo de serviço e respectivos custos do CarpoolWorld.com.

2.2.11. Zimride.com [11]

Este *website* foi criado em 2007 por John Zimmer e Logan Green, inicialmente construído com o objectivo de ajudar nas viagens de estudantes da Universidade de Cornell nos Estados Unidos e pouco depois expandindo-se para mais de 30 campus universitários. Posteriormente o Zimride também se virou para empresas criando serviços de *carpooling* no Wal-Mart Stores Inc. e na Cigna Corp. A característica principal deste serviço é a ligação entre condutores e passageiros que procurem o *carpool* para ir para o trabalho ou aulas.

O Zimride é actualmente o maior programa de partilha de boleia nos Estados Unidos, este serviço conta com mais de 350.000 utilizadores, está activo em 125 campus universitários. A forma de subsistência desta empresa é feita através dos serviços de *carpool* universitário e empresarial.

A página inicial do *website* apresenta as opções de pesquisa por local de partida, local de destino e data para a viagem pretendida. Além destas opções é apresentada uma listagem das últimas viagem inseridas pelos utilizadores da aplicação e os destinos mais procurados (Figura 39).

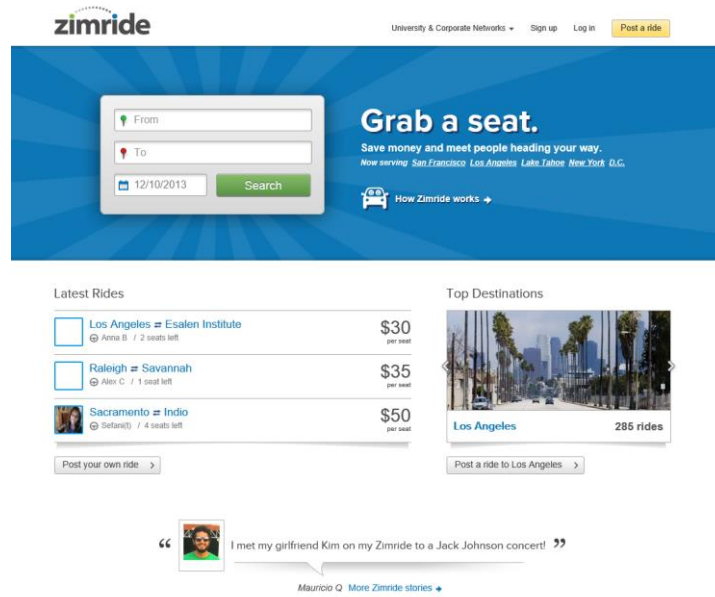


Figura 39 - Ecrã inicial do Zimride.com.

Ao seleccionar uma viagem é possível ver a informação relativa ao veículo, data da viagem, comentários sobre a viagem, custos, regularidade, perfil do condutor (integrado no Facebook), lugares disponíveis, itinerário ou *feedbacks* de pessoas que partilharam boleia com a pessoa em questão (Figura 40). É possível reservar o lugar através da aplicação, sendo a reserva paga de imediato através dos serviços PayPal ou cartão de crédito. Caso a boleia não se concretize por culpa do condutor é garantido o reembolso total do valor pago pelo passageiro.



Figura 40 - Detalhes da viagem seleccionada no Zimride.com.

Deve ser referido que a obrigatoriedade do registo na aplicação através do Facebook é importante para eliminar o problema de não conhecer as pessoas com quem é partilhada a viagem, pois a aplicação ao inserir um perfil de viagem irá fazer correspondências com locais em comum e amigos em comum. Para o Co-fundador Zimmer este factor era o principal responsável para o *carpooling* ter falhado no passado, referindo que a razão principal para o seu funcionamento era a confiança.

2.2.12. Carma [12]

Anteriormente conhecida por Averno, foi fundada em 2007 por Sean O'Sullivan na divisão de pesquisa e desenvolvimento da empresa Mapflow, em Dublin na Irlanda. Em Agosto de 2013 a empresa mudou de nome para Carma.

Actualmente a Carma é um fornecedor global de *software*, *hardware* e serviços profissionais para a melhoria da eficiência do transporte de passageiros. O sistema Carma é um sistema de informação em tempo real que possui a capacidade de conhecimento de lugares disponíveis para os passageiros e que fornece opções convenientes de compra de bilhetes, permitindo assim relatórios mais amplos para a melhoria na eficiência do transporte de passageiros.

Os conhecimentos de transporte da Carma abrangem clientes tão diversos como a Transport for London, os ministérios dos transportes do Reino Unido e Holanda, a Agência Espacial Europeia, as cidades e os operadores de transporte público de Dublin para Madrid, e milhares de consumidores individuais, bem como operadores de transportes privados.

A Carma para além de todos os serviços referidos acima possui uma aplicação móvel para a prática do *carpooling*. Esta aplicação permite efectuar uma filtragem de viagens por origem e destino, data, e tipo de viagem (pedido ou oferta) (Figura 41).

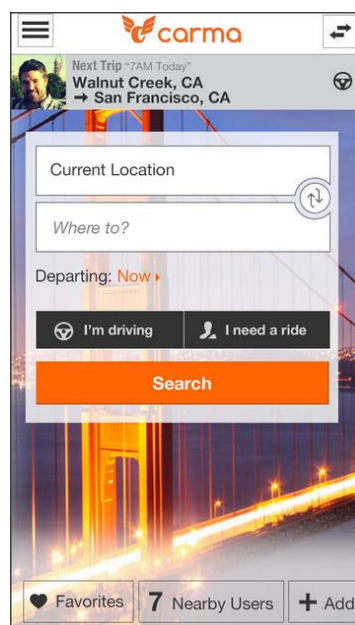


Figura 41 - Ecrã de filtragem da aplicação móvel Carma.

Além desta possibilidade manual, a aplicação permite o cruzamento de informação de perfil com as viagens inseridas no sistema, podendo encontrar correspondências automaticamente. O perfil de cada utilizador possui informação sobre o seu histórico de viagens, avaliação atribuída, grupos em que está inserido, os seus utilizadores favoritos e o balanço monetário da sua conta na aplicação Carma (Figura 42).

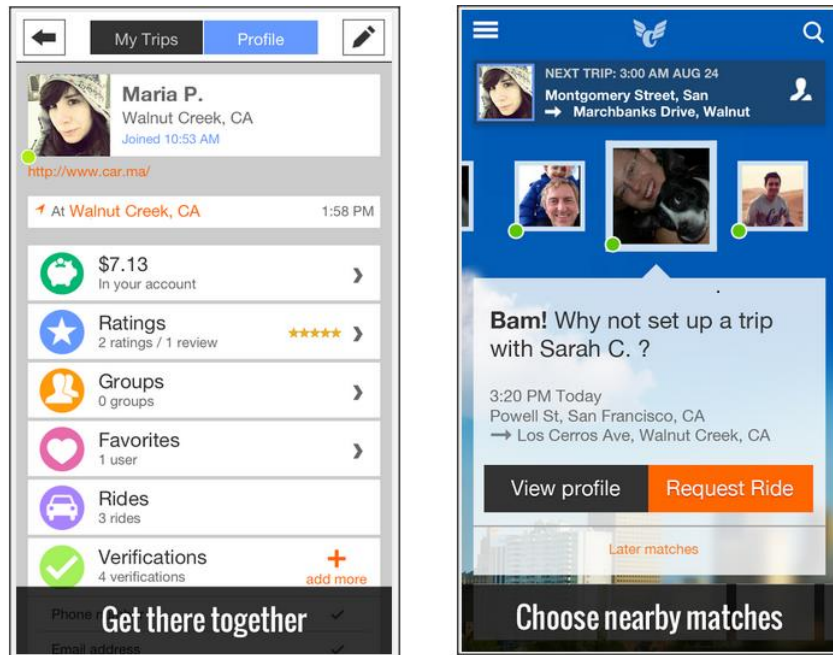


Figura 42 - Perfil de utilizador (esquerda) e correspondência de viagem respectiva (direita) da aplicação móvel Carma.

Depois de encontrada uma correspondência é iniciado o contacto através do *chat* criado para o efeito. Após negociar e chegar ao consenso dos moldes da viagem é mostrada a informação sobre o dinheiro ganho com a partilha de boleia, bem com a distância e tempo de viagem (Figura 43). Cada passageiro tem de associar o cartão de crédito válido para o pagamento da viagem, sendo que do valor ganho com a viagem, 15% é para a Carma, para cobrir custos de licenciamento, hospedagem, suporte e comunicação.

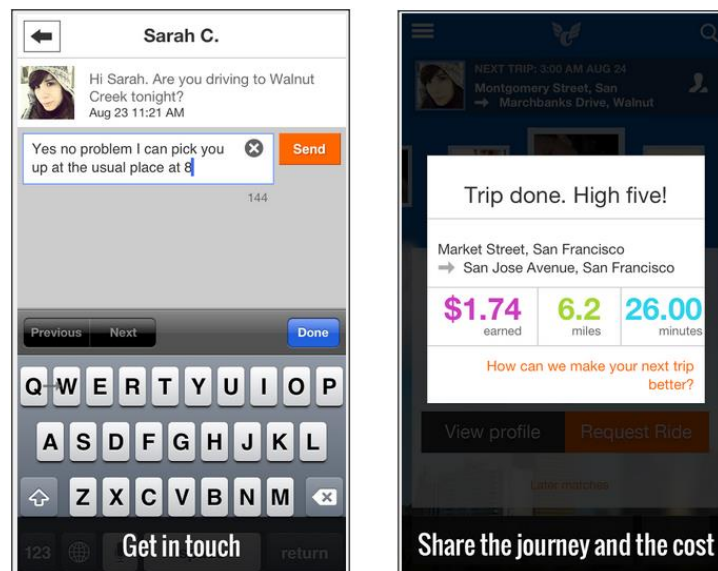


Figura 43 - Chat (esquerda) e confirmação de viagem com alguma informação da mesma da aplicação móvel Carma.

É ainda de salientar que esta aplicação tem uma funcionalidade em tempo real chamada informação do passageiro em tempo real, que consiste na possibilidade do condutor dar início à viagem na aplicação móvel para ir vendo todos os locais próximos que tem de percorrer para ir buscar os passageiros.

2.3. Comparação dos Sistemas Estudados

Comparando os vários sistemas abordados na secção anterior podemos concluir que o *website* blablacar.pt é o mais estruturado abrangendo quase por completo todas as funcionalidades básicas e avançadas. Na área móvel destaca-se a aplicação Carpooling.co.uk, que abrange grande parte das funcionalidades da aplicação anterior e acrescenta ainda a componente de criação de grupos ou eventos para agrupar boleias para destinos específicos. Há ainda que destacar a aplicação móvel Carma que tem uma componente de localização em tempo real que torna mais fácil a busca de cada passageiro através dos serviços de localização do telemóvel.

Deste estudo feito aos vários sistemas analisados conclui-se que há vários pontos que ainda não são abordados por estas aplicações, como as viagens em tempo real onde não há associações de novos passageiros durante as viagens, ou o controlo dos custos automáticos que são pouco fiéis às médias de custo total. Apesar disso há pontos positivos a reter como o sistema de avaliação que pode ajudar a cativar mais pessoas para o uso do *carpooling*, ou as preferências associadas a cada utilizador que servem para uma melhor socialização entre passageiros e condutor.

Através das ilações retiradas deste estudo é possível estar ciente do mercado em questão e tornar melhor o sistema MoCaS.

		Deboleia.com	Pendura.pt	Roadsharing.com	Blablacar.pt	Together	Carpooling.co.uk
Filtros Básicos	Locais de viagem	✓		✓	✓	✓	✓
	Data			✓	✓	✓	✓
Filtros Avançados	Hora de partida			✓	✓	✓	
	Raio de Distância						
	Fumador						✓
	Só Mulheres						✓
	Pré-reserva de lugar						✓
	Tipo de Pedido	✓	✓	✓		✓	✓
	Custos			✓	✓		
	Mapa			✓			
	Tipo de Veículo				✓		
	Perfis com Foto				✓		
Funcionalidades Básicas	Locais de viagem	✓	✓	✓	✓	✓	✓
	Data e hora de viagem	✓	✓	✓	✓	✓	✓
	Descrição	✓	✓	✓	✓	✓	✓
	Custo definido manualmente	✓	✓	✓	✓	✓	✓
	Pré-reserva de lugar				✓		✓
Funcionalidades Avançadas	Perfil de condutor			✓	✓	✓	✓
	Informação do veículo		✓		✓		✓
	Rating do condutor			✓	✓		✓
	Grupos / Eventos			✓			✓
	Custo automático	✓		✓	✓	✓	✓
	Mapa		✓	✓	✓	✓	✓
	Flexibilidade de horário				✓		

	Desvio máximo				✓		
	Fumador		✓		✓		✓
	Só mulheres				✓		✓
	Animais				✓		
	Regularidade		✓		✓		✓
	Chat/Mensagens Privadas					✓	✓
Plataformas	Web	✓	✓	✓	✓	✓	✓
	Móvel				✓	✓	✓
Registo	Obrigatório				✓	✓	✓
	Facebook				✓	✓	✓
Restrições de Uso	País/Organização					✓	
Financiamento	Publicidade	✓	✓	✓			✓
	Serviços			✓		✓	✓
	Privados				✓	✓	

Tabela 1 - Matriz comparativa dos sistemas de *carpooling* estudados (continua).

		Princeton University	Jayride.com	Boleia.net	Carpoolworld.com	Zimride.com	Carma
Filtros Básicos	Locais de viagem		✓	✓	✓	✓	✓
	Data					✓	✓
Filtros Avançados	Hora de partida						✓
	Raio de Distância				✓		
	Fumador						
	Só Mulheres						
	Pré-reserva de lugar						
	Tipo de Pedido			✓	✓		✓
	Custos						
	Mapa		✓				
	Tipo de Veículo		✓				
	Perfis com Foto						
Funcionalidades Básicas	Locais de viagem	✓	✓	✓	✓	✓	✓
	Data e hora de viagem	✓	✓	✓	✓	✓	✓
	Descrição	✓	✓	✓	✓	✓	✓
	Custo definido manualmente	✓	✓	✓		✓	✓
Funcionalidades Avançadas	Pré-reserva de lugar		✓			✓	✓
	Perfil de condutor		✓	✓	✓	✓	✓
	Informação do veículo			✓	✓	✓	
	Rating do condutor			✓	✓	✓	✓
	Grupos / Eventos				✓	✓	✓
	Custo automático					✓	✓
	Mapa		✓	✓	✓	✓	
	Flexibilidade de horário						
	Desvio máximo				✓		
	Fumador		✓		✓		
Só mulheres		✓		✓			
Animais				✓			

	Regularidade		✓	✓	✓	✓	✓
	Chat/Mensagens Privadas						✓
Plataformas	Web	✓	✓	✓	✓	✓	✓
	Móvel						✓
Registo	Obrigatório	✓		✓	✓	✓	✓
	Facebook		✓	✓	✓	✓	✓
Restrições de Uso	País/Organização	✓	✓			✓	
Financiamento	Publicidade			✓	✓		
	Serviços		✓	✓	✓	✓	✓
	Privados	✓					

Tabela 1 - Matriz comparativa dos sistemas de *carpooling* estudados (continuação).

2.4. Evolução da Tecnologia Móvel

2.4.1. Os Smartphones

Os telemóveis tornaram-se uma parte integrante do nosso dia-a-dia. Todas as pessoas, desde adolescentes até aos mais velhos, têm um telemóvel pessoal [26]. A evolução das tecnologias móveis contribuiu para um grande “salto” na história dos telemóveis, transformando os enormes telemóveis de 1995 para os actuais *smartphones* elegantes e modernos [26]. Além do serviço de voz digital, os *smartphones* actuais fornecem o serviço de mensagens de texto, *email*, navegação na Web, navegação GPS (*global positioning system*), ecrã táctil, câmaras de vídeo, leitor de MP3, reprodução de vídeo e chamadas. Além das suas funções internas, podem executar um vasto número de aplicações, transformando um simples telemóvel num computador móvel [28].

Segundo a empresa de análise de mercado, International Data Corporation (IDC), as encomendas globais de *smartphones* atravessaram os mil milhões pela primeira vez no ano de 2013, crescendo mais de 40% em comparação com o ano de 2012, e mais que o dobro de há três anos atrás (ver Figura 44). Isso comparado com apenas 315 milhões de computadores e 227 milhões de *tablets*. De acordo com a empresa de pesquisa, este crescimento é impulsionado por aparelhos de baixo custo e ganhos em mercados (ver Figura 44). A IDC prevê que "Os *Smartphones* representarão praticamente todo o mercado de telemóveis em muitas das economias mais desenvolvidas do mundo até o final de 2017" [27].



Figura 44 - Número de vendas mundiais por dispositivos em 2013 com previsão para 2017, em milhões. Fonte: IDC.

2.4.2. Sistemas Operativos Móveis

Os telemóveis têm vindo a transformar-se gradualmente a partir de simples telemóveis que forneciam apenas serviços de chamadas até aos *smartphones* actuais com poderosas capacidades multimédia e Internet de banda larga. Embora os dispositivos de computação portátil tivessem sistemas operacionais desde o início da década de 1980, o seu papel tornou-se mais relevante nos últimos anos com o lançamento de novos tipos de *smartphones* [29]. Igualmente importantes são as previsões de que os telemóveis em breve irão ultrapassar os computadores como os dispositivos mais comuns de acesso à internet. Em consequência de tudo isto, os sistemas operativos móveis em *smartphones* e outros dispositivos móveis tornaram-se plataformas cada vez mais importantes para a criação de ecologias de serviços, funcionando como um alvo para o desenvolvimento de aplicações e novos serviços em proporções cada vez maiores da população mundial [29].

Actualmente os sistemas operativos mais utilizados a nível mundial são [32]:

- **Android** - baseado no *kernel* do Linux [36], e projectado principalmente para dispositivos móveis de ecrã táctil como *smartphones* e *tablets*. Foi inicialmente desenvolvido pela empresa Android Inc., que era financiada pela Google que em 2005 acabou por adquirir toda a Android Inc. [37]. Este sistema operativo móvel é do tipo aberto, a Google abre o código sob a licença Apache [36]. Este código-fonte aberto e licenciamento permissivo permitem que o *software* seja livremente modificado e distribuído por fabricantes de dispositivos, operadoras móveis e por qualquer pessoa que o deseje fazer [38].
- **iOS** - desenvolvido e distribuído pela Apple Inc., foi originalmente lançado para o iPhone em 2007 e foi estendido para suportar outros dispositivos da Apple como o iPod Touch, iPad, iPad Mini e a Apple TV de segunda geração. Ao contrário do Windows Phone, da Microsoft e do Google Android, a Apple não licencia o iOS para instalação em equipamentos que não sejam produzidos pela mesma. O iOS é a versão móvel do sistema operativo OS X usado nos computadores pessoais da Apple [39].
- **Windows Phone** - desenvolvido pela Microsoft, este sistema operativo é o sucessor da plataforma Windows Mobile [30], contudo o Windows Phone e o Windows Mobile são totalmente incompatíveis [31]. Ao contrário do antecessor, este é focado no mercado consumidor, em vez do mercado empresarial [33]. Com o Windows Phone, a Microsoft criou uma nova interface, com uma linguagem de *design* designada de "Modern" [que antigamente era conhecida como "Metro"] [34]. Além disso, o *software* é integrado com serviços da Microsoft e de terceiros, e define requisitos mínimos para o hardware em que é executado [35].
- **BlackBerry OS** – desenvolvido pela empresa canadiana BlackBerry (antiga Research In Motion), este sistema de código fechado foi construído especialmente para os modelos empresariais. BlackBerry OS oferece suporte nativo para *email* corporativo via MIDP, que permite a sincronização sem fio fácil com o Microsoft Exchange, Lotus Domino e e-mail, contactos, calendário, notas e assim por diante, enquanto utilizado juntamente com o BlackBerry Enterprise Server [41]. Actualmente o BlackBerry OS tem uma nova plataforma totalmente refeita, passou a dar suporte a aparelhos com ecrãs tácteis, além de fazer uma interessante divisão: tanto dá acesso a uma interface "pessoal", com tarefas mais cotidianas e divertidas, como a uma empresarial, para ser acedida durante o período de trabalho [40].

2.4.3. Estudo de Mercado

De acordo com o último relatório da empresa Strategy Analytics, no terceiro trimestre de 2013 foram vendidos cerca de 251 milhões de *smartphones*. A plataforma Android bateu todos os recordes até ao momento, tendo conseguido conquistar 81,3% do mercado de vendas de *smartphones*, muito por culpa da Apple e da Blackberry [32]. Já o Windows Phone da Microsoft duplicou sua presença no mercado e é actualmente o crescimento mais rápido de plataforma de *smartphones* do mundo [32].

Ao nível global, as vendas de cresceram cerca de 45%, tendo sido registadas 172.8 milhões de unidades vendidas no terceiro trimestre de 2012 e, em igual período do ano de 2013, o valor subiu para as 251 milhões de unidades vendidas. Este crescimento foi impulsionado principalmente pela grande procura dos modelos Android e Windows Phone em mercados desenvolvidos e em desenvolvimento, nomeadamente na Europa e na Ásia [32].

Como se pode ver na Figura 45, no terceiro trimestre, em 251 milhões de *smartphones* vendidos, 204.4 são Android o que significa que 4 em cada 5 *smartphones* vendidos são dispositivos Android. Esta subida deveu-se principalmente à BlackBerry, que teve uma queda de 4 para 1 por cento no ano de 2012 na sua quota de mercado mundial de *smartphones*, devido a uma fraca adesão ao seu sistema operativo Blackberry 10. Este estudo também reflecte a perda de terreno da Apple em relação ao Android devido à sua presença limitada em mercados de *smartphones* de gamas inferiores. O Windows Phone da Microsoft tem vindo a aumentar as vendas, e durante o 3º trimestre de 2013 vendeu mais de 10 milhões de *smartphones* em todo o mundo [32].

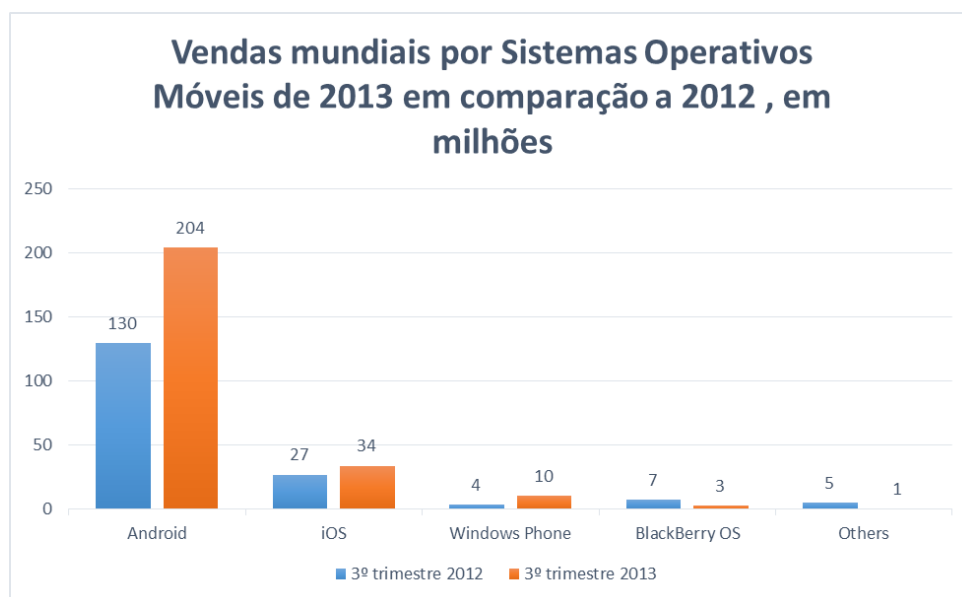


Figura 45 - Número de vendas mundiais por Sistemas Operativos Móveis do 3º trimestre de 2013 em comparação com o mesmo trimestre em 2012, em milhões. Fonte: Strategy Analytics.

Traduzindo os números anteriores em percentagem de mercado, podemos ver na Figura 46 que o Android lidera o mercado com 81%, seguido de muito longe pela Apple com apenas 13,4% e com o Windows Phone ainda em crescimento para tentar, no mínimo, ultrapassar a Apple. O Windows Phone subiu 178% na sua participação de mercado no 3º trimestre de 2013. O crescimento do Windows Phone deveu-se quase inteiramente à Nokia e ao seu portfólio de melhoria constante da série Lumia em toda a Europa, Ásia e Estados Unidos. No entanto, a Microsoft está claramente ainda a um baixo nível de participação em

todo o mundo, tentando a entrada em vários mercados importantes como Japão, Coreia do Sul e África [32].

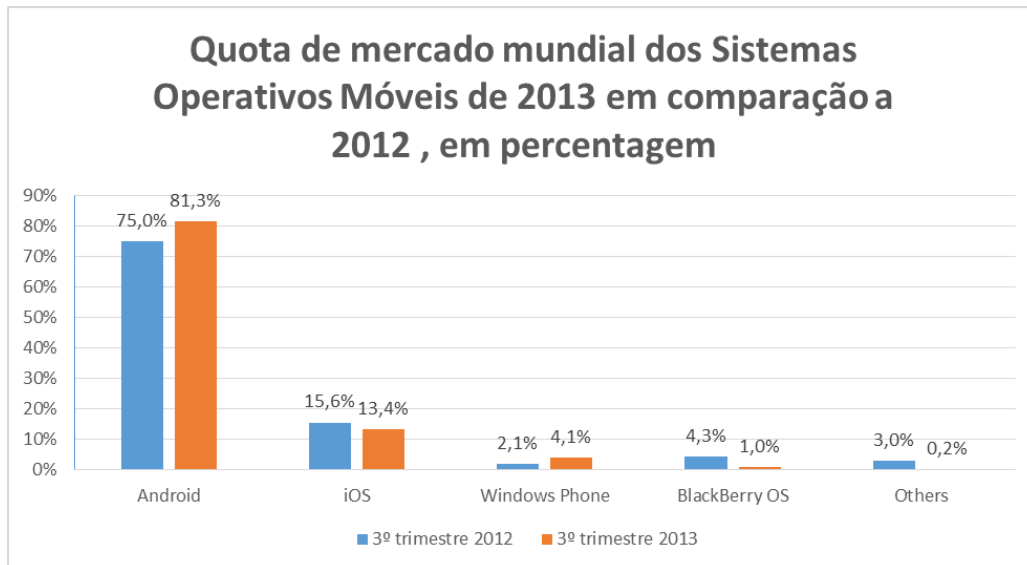


Figura 46 - Quota de mercado mundial dos Sistemas Operativos Móveis do 3º trimestre de 2013 em comparação com o mesmo trimestre em 2012, em percentagem. Fonte: Strategy Analytics.

2.4.4. Localização e Comunicação

Com o crescimento da popularidade dos telemóveis os fabricantes têm vindo a adicionar cada vez mais tecnologia aos mesmos, como o acesso a internet, câmaras digitais e GPS, o que os torna cada vez mais atraentes para os consumidores. A adição de *hardware* de GPS para telemóveis criou muitas vantagens, juntamente com alguns problemas. Com as capacidades de GPS vem um enorme potencial de mercado [42].

Uma grande vantagem é o uso de um sistema semelhante ao encontrado nos dispositivos de navegação automóvel. Quando um indivíduo precisa de se deslocar de um local para outro e necessita de orientação, pode utilizar o *software* do seu telemóvel com sistema GPS, para indicar o melhor percurso para chegar ao destino que pretende. Este sistema também é vantajoso para uma situação em que alguém se perde, neste caso os telemóveis poderiam direccionar essa pessoa de volta para um lugar onde a mesma estava familiarizada.

O sistema GPS gera ainda alguma preocupação a muitas pessoas devido à perda de privacidade que esta tecnologia pode trazer. Mas apesar dessa preocupação este sistema tem ajudado inúmeras pessoas, e alguns exemplos abordados acima são boa prova disso [42].

Capítulo 3

Requisitos e Arquitectura da Solução

Este capítulo abrange toda a arquitectura e requisitos do sistema, explicando de uma forma geral todo o seu funcionamento. Além disso, irá aprofundar todas as interacções dos utilizadores com o sistema, representando através de diagramas todo o fluxo de processos do sistema.

3.1. *User Stories*

As *User Stories* são pequenas frases e de simples descrição de um recurso contadas a partir da perspectiva da pessoa que deseja um novo recurso, geralmente um utilizador ou cliente do sistema [51]. As *User Stories* definidas para este sistema são as seguintes:

1. Como utilizador anónimo quero efectuar o registo no sistema para que possa autenticar-me no sistema.
2. Como utilizador anónimo quero efectuar a autenticação no sistema para que possa ter acesso à marcação de viagens.
3. Como utilizador autenticado quero efectuar uma inserção de viagem para que possa partilhar os custos da viagem.
4. Como utilizador autenticado quero efectuar uma reserva de lugar para que possa fazer uma viagem para o destino que pretendo.
5. Como utilizador autenticado quero ver a lista de utilizadores para que possa atribuir uma avaliação.
6. Como utilizador autenticado quero aceder ao meu perfil para que possa modificar os meus dados.
7. Como utilizador autenticado quero aceder às mensagens para que possa entrar em contacto com um utilizador.
8. Como utilizador autenticado quero criar um grupo para que possa adicionar viagens cujo percurso é semelhante ao que uso diariamente para o trabalho.
9. Como utilizador autenticado quero pedir novamente as minhas credenciais de acesso para que possa continuar a utilizar o sistema.
10. Como utilizador autenticado quero inserir o meu veículo para que possa partilhar uma viagem como condutor.
11. Como utilizador autenticado quero pesquisar viagens para poder viajar.
12. Como utilizador autenticado quero ver a informação relativa a outro utilizador para ver se me identifico com o seu perfil de viagem.

3.2. Visão Geral da Arquitectura

Para a arquitectura deste sistema irá ser utilizado um servidor que irá armazenar toda a informação dos clientes do serviço numa base de dados (BD) criada para o efeito. Esse servidor irá também suportar um *website* com algumas funcionalidades da aplicação móvel (Figura 47).

A aplicação móvel irá consultar a informação armazenada no servidor, bem como actualizar a sua informação sempre que sejam inseridas novas viagens por exemplo. Esta aplicação apenas irá armazenar na base de dados local do telemóvel as suas credenciais de acesso, para que em cada sessão não seja necessário inserir as credenciais novamente (Figura 47).

É ainda de salientar que todos os dispositivos incluídos nesta arquitectura têm de conter ligação à internet para que o sistema funcione.



Figura 47 – Diagrama com uma visão geral da arquitectura do sistema.

3.3. Diagramas de Casos de Uso

Depois de identificadas as *user stories* do sistema na secção 3.1. *User Stories* é possível identificar uma série de casos de uso. Através de diagramas gráficos, é mais fácil visualizar as ideias e soluções para o projecto [46].

Os diagramas de casos de usos apresentam uma linguagem simples e compreensível, de modo que o utilizador possa ter uma ideia geral de funcionamento de todo o sistema. Estes dividem o sistema em actores e casos de uso. Os actores podem ser pessoas, computadores ou sistemas de *software*. Os casos de uso descrevem o comportamento do sistema. Estes diagramas são conhecidos por serem extremamente simples e fáceis de entender [46].

No sistema *MoCaS* forma identificados três tipos de actores, o utilizador não autenticado, o utilizador de sistema e o utilizador autenticado.

O actor utilizador anónimo apenas pode consultar as viagens disponíveis no sistema sem ter acesso a funcionalidades como dados de contacto, opções de pré-reserva e visualização de perfil, além disso pode efectuar o registo no sistema utilizando o seu *email* ou a sua conta do Facebook caso possua uma (utilizador de sistema). A partir do momento em que passa a ser um utilizador autenticado no sistema passa a ter acesso a todas as funcionalidades como utilizador autenticado.

Um utilizador autenticado tem acesso total às funcionalidades do sistema podendo este inserir e pesquisar viagens, atribuir e receber avaliações, alterar o seu perfil, criar grupos para viagens comuns, enviar e receber mensagens privadas, reservar lugares, visualizar perfis de outros utilizadores, alterar dados referentes ao seu veiculo e as suas preferências para cada viagem em que participa e, sempre que se esqueça das suas credenciais de acesso, pode solicitá-las através da sua conta de *email*.

No diagrama de casos de uso da Figura 48 estão representados os de uma forma geral os casos de uso para os actores acima identificados. Todos os casos de uso estão descritos de forma detalhada no Anexo A – Descrição dos Casos de Uso, com uma relação relativamente à secção 3.1. *User Stories*

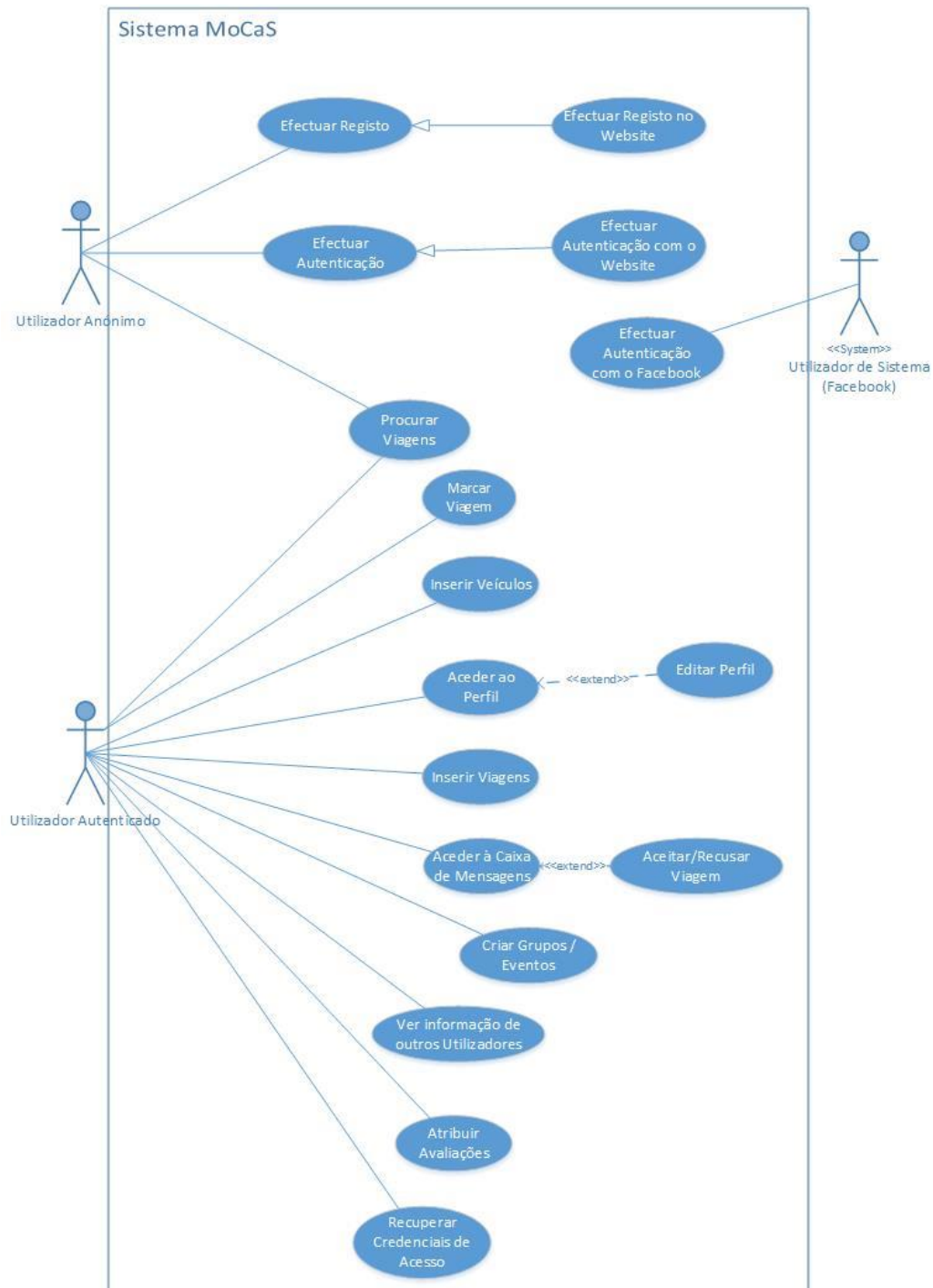


Figura 48 - Diagrama de casos de uso para os utilizadores autenticados.

3.4. Diagrama de Componentes

Os diagramas de componentes são representações gráficas da estrutura física da implementação do sistema, constituindo parte da especificação da arquitectura. Estes diagramas têm como objectivos organizar o código fonte, construir uma *release* executável e especificar uma base de dados física [52].

O diagrama de componentes é composto por componentes, interfaces e relações entre componentes. Um componente representa a parte física de um sistema e permite a realização

de um conjunto de interfaces que são um conjunto de operações usadas para especificar os serviços de um componente [52].

Na Figura 49 está representado de uma forma geral o diagrama de componentes do sistema MoCaS que demonstra a interligação de todos os competentes do sistema. Este sistema funcionará na base de comunicação cliente-servidor, em que os clientes (*web browser* e aplicação móvel) trocam informação com a BD através de um *web service* que possibilita a interligação destas plataformas tão diferentes. Como o *website* e o *webservice* fazem parte da plataforma .NET para serem executadas têm de estar suportadas pelo Internet Information Services (IIS).

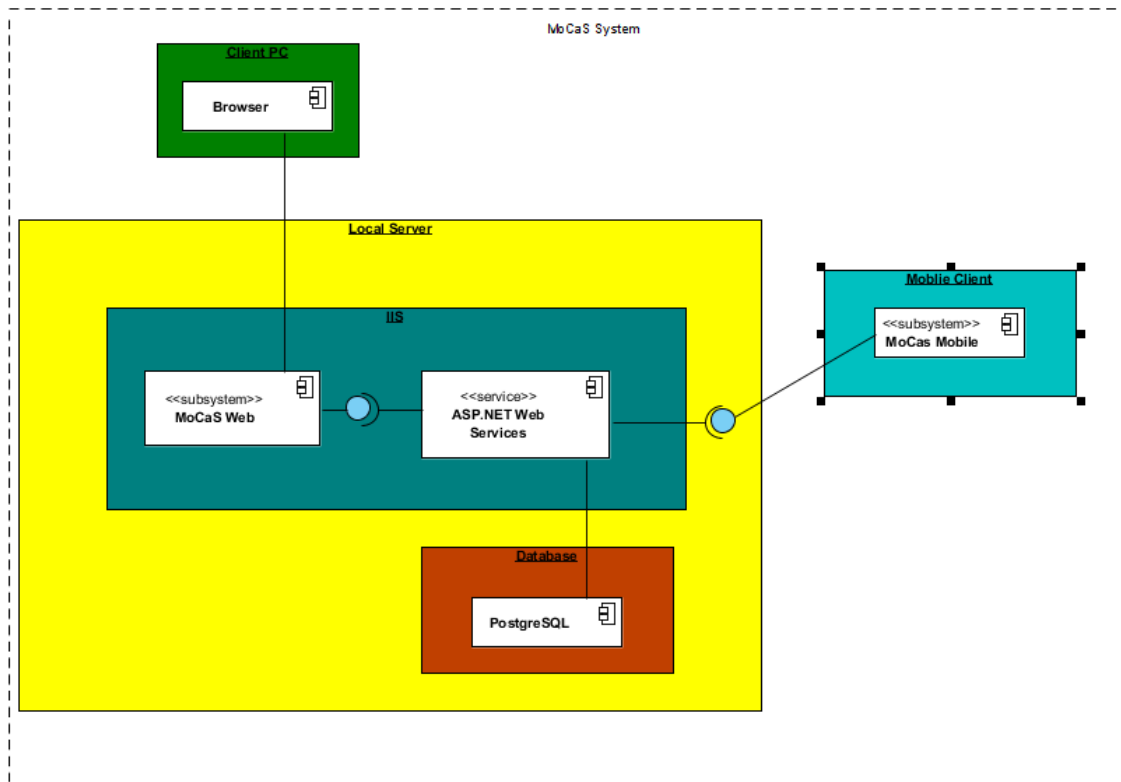


Figura 49 - Diagrama de componentes do MoCaS.

3.5. Modelo de Base de Dados

Nesta secção, será abordada toda a estrutura de BD através de diagramas e tabelas descritivas com informação detalhada de cada tabela.

3.5.1. Diagrama Conceptual

Como a BD é um componente essencial neste sistema, pois é através dela que se faz a troca de dados entre aplicações, é importante que tenha uma estrutura simples e ao mesmo tempo robusta. Para projectar a BD foi criado um modelo conceptual representado na Figura 50. Este descreve de maneira concisa as necessidades dos utilizadores e é de fácil compreensão.

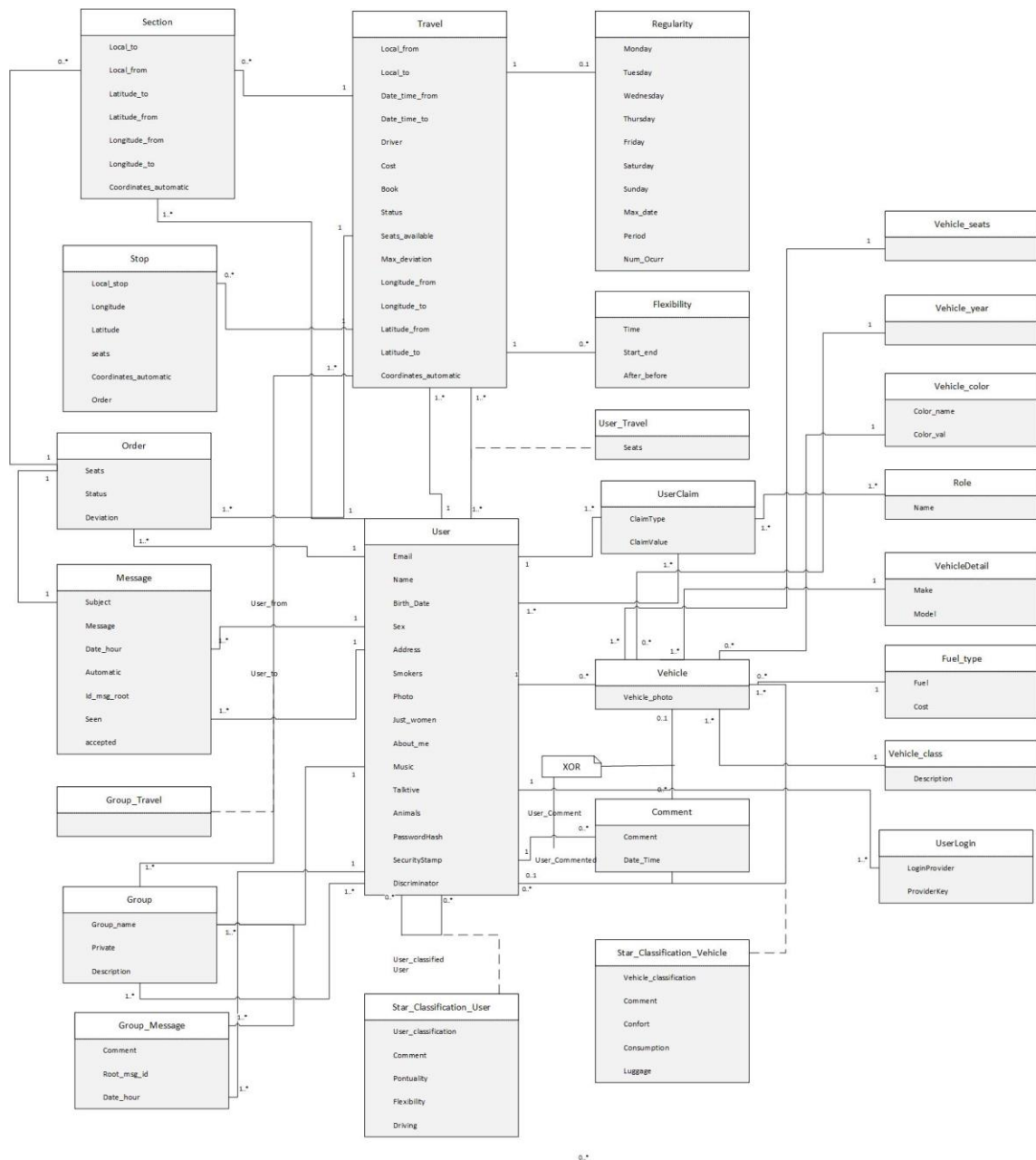


Figura 50 - Diagrama conceptual do sistema MoCaS.

3.5.2. Diagrama Físico

Depois de analisado o diagrama conceptual foi feito um mapeamento para uma abordagem de implementação que implica especificar as estruturas internas de armazenamento bem como as formas dos dados que serão armazenados na BD. O diagrama físico da Figura 51 representa isso mesmo.

O sistema MoCas é suportado por 34 tabelas, estando estas maioritariamente divididas em 7 módulos, viagens, veículos, utilizadores, grupos, mensagens, avaliações e configurações. Nestes módulos é de destacar o módulo de utilizadores, que é constituído por 5 tabelas e é responsável pelos dados relativos a cada utilizador, este módulo é muito importante pois é nele que reside informação sobre autenticação no sistema. Outro módulo a destacar é o módulo de viagens que engloba 10 tabelas e é responsável pela grande maioria das transacções feitas no sistema, pois é aqui que se registam as viagens e se faz a interacção de partilha das

mesmas. Cada parametrização de cada viagem é aqui guardada, assim como os dados sobre as localizações de viagens em tempo real que estão em constante actualização.

Nos restantes módulos encontra-se informação relativa à configuração do sistema, dados sobre os veículos de cada condutor inserido no sistema MoCaS, dados com mensagens privadas trocadas entre utilizadores, informação relativa a grupos *carpool* e por fim informação sobre a avaliação de utilizadores e veículos.

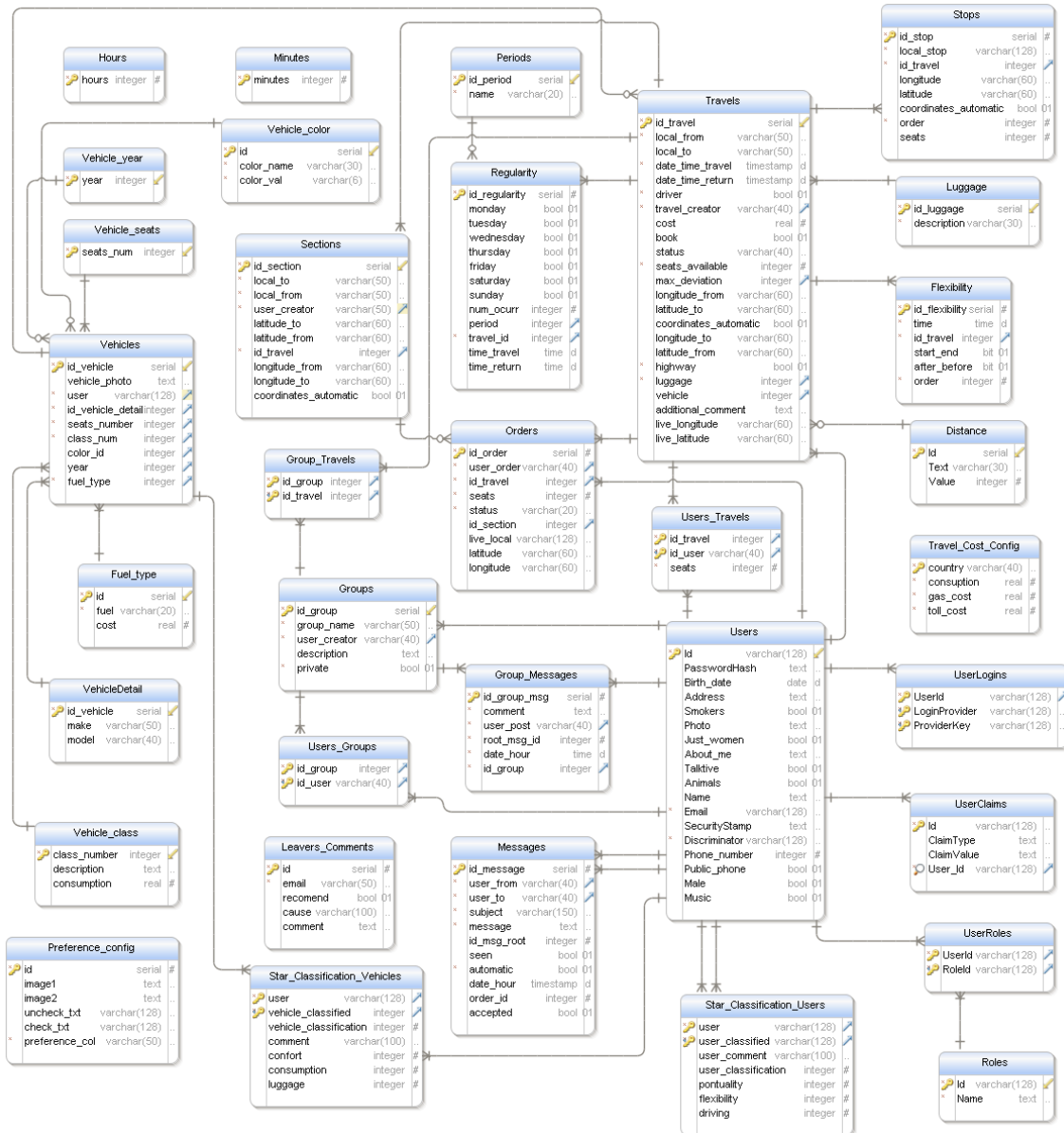


Figura 51 - Diagrama físico do sistema MoCaS.

3.6. Requisitos Funcionais

Através da abordagem aos vários sistemas de *carpooling* do Capítulo 2 Enquadramento e Trabalhos Relacionados extraímos algumas conclusões positivas e negativas. Esta informação é extremamente importante para definir o sistema pois por um lado podemos usar algumas potencialidades encontradas e por outro, podemos evitar certos problemas encontrados em alguns desses sistemas.

Como um dos principais problemas na prática do *carpooling* é o facto de se viajar com desconhecidos, a aplicação irá obrigar ao registo de cada utilizador, criando assim um perfil, também haverá a possibilidade de tornar este perfil mais confiável através da integração com redes sociais mais comuns como o Facebook.

Outra importante funcionalidade será a de marcação de viagem em tempo real, ou seja, durante uma viagem que já esteja a decorrer, uma pessoa que necessite de boleia para o mesmo destino ou pontos de paragem antes do destino e esteja nas imediações do trajecto que está a ser percorrido ou num ponto de paragem do mesmo, pode usufruir desta viagem. Isto irá permitir dar respostas rápidas a casos onde é preciso uma deslocação imediata. Deve ainda ser referido que para utilizar este serviço é necessário que os dispositivos móveis possuam GPS integrado. Directamente ligada a esta funcionalidade está também a resposta de confirmação de viagem em tempo real, ou seja, o condutor recebe uma notificação de pedido de boleia durante a viagem e pode aceitar ou não através de um comando de voz para que não se distraia manuseando o equipamento durante a condução.

Como o objectivo desta prática não é obter benefícios lucrativos das viagens, o sistema irá calcular o gasto médio de viagem e irá definir um preço automático. No entanto, este custo poderá ser alterado até um valor máximo definido pela aplicação, sendo o valor mínimo livre de qualquer restrição pois os utilizadores podem optar por não cobrar qualquer valor.

Para que o utilizador conheça melhor a pessoa com quem vai viajar e o seu veículo, o sistema MoCaS irá possuir um sistema de avaliação que além de avaliar o utilizador que fornece o serviço, também vai dar a possibilidade de avaliar o seu veículo. Este sistema vai contribuir para que os utilizadores possuam mais informação fidedigna tanto do utilizador (passageiro ou condutor), como do veículo (no caso em que o utilizador avaliado seja condutor).

Uma outra funcionalidade inovadora a implementar é a possibilidade dos utilizadores que são passageiros poderem inserir as viagens que pretendem fazer e os condutores poderem associar estes às suas viagens desde que os critérios das suas viagens correspondam.

Por fim, a Tabela 2 apresenta as restantes funcionalidades do sistema (incluindo as já abordadas acima) e respectivas prioridades de implementação. Podem ainda ser consultados no Anexo B alguns protótipos iniciais das aplicações móvel e web com base em alguns dos requisitos aqui apresentados.

		MoCaS	Prioridade
Filtros Básicos	Locais de viagem	✓	Alta
	Data	✓	Alta
Filtros Avançados	Hora de partida	✓	Alta
	Raio de Distância	✓	Média
	Fumador	✓	Média
	Só Mulheres	✓	Baixa
	Pré-reserva de lugar	✓	Baixa
	Tipo de Pedido	✓	Média
	Custos	✓	Baixa
	Mapa	✓	Baixa
	Tipo de Veículo	✓	Baixa
	Perfis com Foto	✓	Baixa
Funcionalidades Básicas	Locais de viagem	✓	Alta

	Data e hora de viagem	✓	Alta	
	Descrição	✓	Alta	
	Custo definido manualmente	✓	Baixa	
Funcionalidades Avançadas	Pré-reserva de lugar	✓	Alta	
	Perfil de condutor	✓	Alta	
	Informação do veículo	✓	Alta	
	Rating do condutor	✓	Alta	
	Grupos / Eventos	✓	Baixa	
	Custo automático	✓	Alta	
	Mapa	✓	Alta	
	Flexibilidade de horário	✓	Média	
	Desvio máximo	✓	Alta	
	Fumador	✓	Média	
	Só mulheres	✓	Média	
	Animais	✓	Média	
	Regularidade	✓	Alta	
	Ouvir música	✓	Média	
	Chat/Mensagens Privadas	✓	Alta	
	Reserva em tempo-real	✓	Alta	
	Informação Meteorológica	✓	Baixa	
	Resposta em Tempo-Real através da Voz	✓	Alta	
	Plataformas	Web	✓	Alta
		Móvel	✓	Alta
Registo	Obrigatório	✓	Alta	
	Facebook	✓	Alta	
Restrições de Uso	País/Organização			

Tabela 2 - Funcionalidades do Sistema MoCaS.

Nesta Tabela 2 é visível que o sistema MoCaS será constituído por várias funcionalidades já existentes nas aplicações estudadas na secção 2.2. Estudo sobre Sistemas de *Carpooling* Existentes e por outras que foram abordadas em destaque na secção anterior. As prioridades que foram atribuídas reflectem as escolhas de funcionalidades fundamentais para que a aplicação possa ser utilizada, no entanto o objectivo será sempre inclui-las todas no sistema. É de salientar também que neste sistema não há restrições de uso, no entanto este está optimizado para funcionar em Portugal.

3.7. Requisitos não funcionais

Os requisitos não funcionais são aqueles que não afectando directamente as funcionalidades do sistema são responsáveis por garantir o seu bom funcionamento com qualidade [83].

A usabilidade é um factor que deve ser tido em consideração em qualquer sistema que possua um interface para uso de um utilizador [83]. Este factor permite entender se a interface com o utilizador é fácil e agradável de usar. Algumas das métricas usadas para avaliar este factor serão a aprendizagem, ou seja, se não é complicado para o utilizador num contacto inicial, concluir as suas tarefas.

Outro factor a ter em conta é a eficiência, este factor diz respeito ao tempo que o utilizador perde para fazer uma tarefa ou realizar um objectivo (tudo isto depois de se tornarem familiarizados com o sistema) [83]. A memorização precede em parte a eficiência, ou seja, após um período de ausência, perceber se se torna complexa uma nova habituação aos processos do sistema.

Em termos de factor de confiabilidade, deve ser tido em conta que a base de dados local deverá estar sempre disponível em qualquer momento para que o sistema possa ser utilizado, só assim se evitarão erros e outro tipo de problemas.

Por fim, a nível de segurança o sistema deverá adoptar medidas que incidam sobre a protecção de dados. Para garantir confidencialidade dos dados, é necessário fazer login no início da aplicação (seja na aplicação móvel ou *web*) garantindo assim que a informação apenas é fornecida às entidades autorizadas, sendo que esta autenticação precisa de ser verificada a com o servidor. No controlo de acesso, deve ser garantido que os pedidos são feitos por um utilizador devidamente autenticado e com sessão iniciada, impedindo assim que qualquer utilizador anónimo tente replicar um pedido como se de um utilizador autenticado se tratasse.

Capítulo 4

Plano de Trabalho e Implicações

Neste capítulo será descrito detalhadamente todo o planeamento deste projecto. Além do planeamento serão também apresentados alguns riscos que levaram a desvios no planeamento do projecto, serão também apresentados alguns caminhos críticos de algumas actividades chave na construção deste sistema e formas de mitigação.

4.1. Planeamento

A tabela abaixo apresenta uma listagem das actividades efectuadas ao longo do primeiro semestre. Este planeamento está de acordo com tudo o que foi abordado no primeiro semestre sem haver qualquer atraso ou desvio no plano.

Actividade	Início	Fim	Detalhes
1	18-09-2013	03-10-2013	Sendo esta a primeira actividade, foram abordados os conceitos em que o projecto se inseria e foi iniciado o processo de investigação sobre o <i>carpooling</i> .
2	04-10-2013	17-10-2013	Análise de resultados da investigação e definição das próximas etapas na investigação. Início da escrita do capítulo 2.
3	18-10-2013	31-10-2013	Investigação sobre factores de incentivo, vantagens e problemas do <i>carpooling</i> .
4	01-11-2013	14-11-2013	Análise aos tipos de prática do <i>carpooling</i> com suas componentes de percurso e tempo.
5	15-11-2013	28-11-2013	Estudo de concorrência em sistemas de <i>carpooling</i> nacionais e internacionais. Comparação entre os sistemas analisados.
6	29-11-2013	12-12-2013	Investigação sobre as tecnologias móveis e suas componentes. Estudo de mercado sobre sistemas operativos móveis e equipamentos tecnológicos.
7	13-12-2013	09-01-2014	Conclusão do capítulo 2 e início da escrita dos capítulos 3 e 4. Aplicação dos conhecimentos do capítulo 2 para a arquitectura da aplicação. Definição de requisitos e orientações de trabalho.
8	10-01-2014	23-01-2014	Conclusão dos capítulos 3 e 4. Escrita do capítulo 5, 7 e 1, bem como do resumo do projecto. Discussão sobre o planeamento, riscos e conclusões.
9	24-01-2014	27-01-2014	Finalização e revisão da dissertação.

Tabela 3 - Planeamento de actividades do primeiro semestre.

Além da Tabela 3, foi criado um mapa Gantt (ver Figura 52) para uma melhor percepção do planeamento do primeiro semestre.

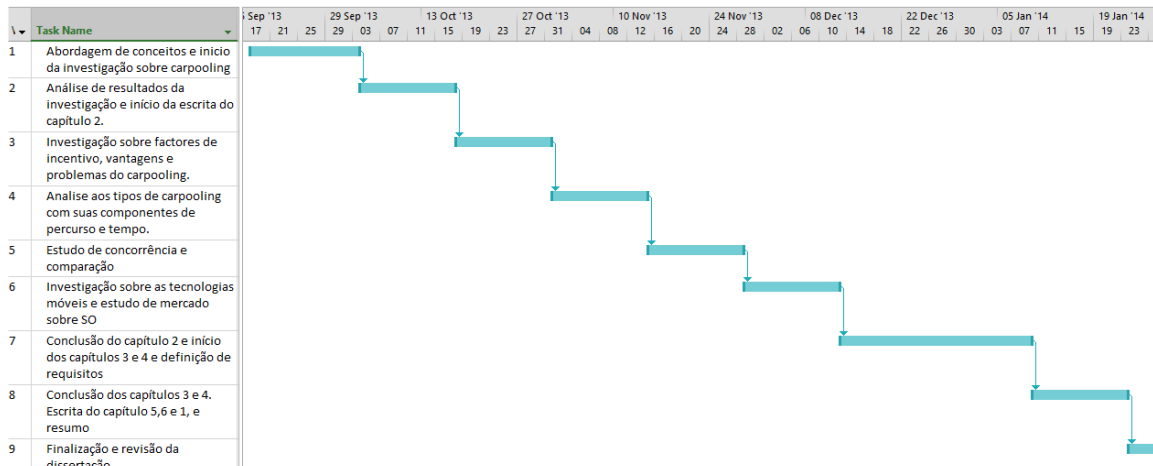


Figura 52 - Mapa Gantt do 1º semestre.

Para o segundo semestre foram planeadas sete actividades que visam essencialmente a implementação do sistema MoCaS. A Tabela 4 representa em detalhe as descrições das actividades para o segundo semestre bem como os períodos de cada uma.

Actividade	Início	Fim	Detalhes
10	05-02-2014	12-02-2014	Configuração do Servidor do Ambiente de Desenvolvimento
11	13-02-2014	17-02-2014	Implementação da Base de Dados
12	18-02-2014	17-03-2014	Implementação do <i>WebService</i>
13	25-02-2014	07-04-2014	Implementação do <i>WebSite</i>
14	25-03-2014	16-06-2014	Implementação da Aplicação Móvel
15	17-06-2014	30-06-2014	Testes ao Sistema e Correções
16	01-07-2014	08-07-2014	Conclusão do Relatório

Tabela 4 - Planeamento de actividades do segundo semestre.

Na Figura 53 está representado o planeamento do segundo semestre de uma forma muito detalhada com as subactividades representadas através de um mapa Gantt. Nesta figura estão assinalados a vermelho os caminhos críticos do planeamento que serão abordados na secção 4.3. Caminhos Críticos com o intuito de explicar o impacto destes caminhos no planeamento do projecto.

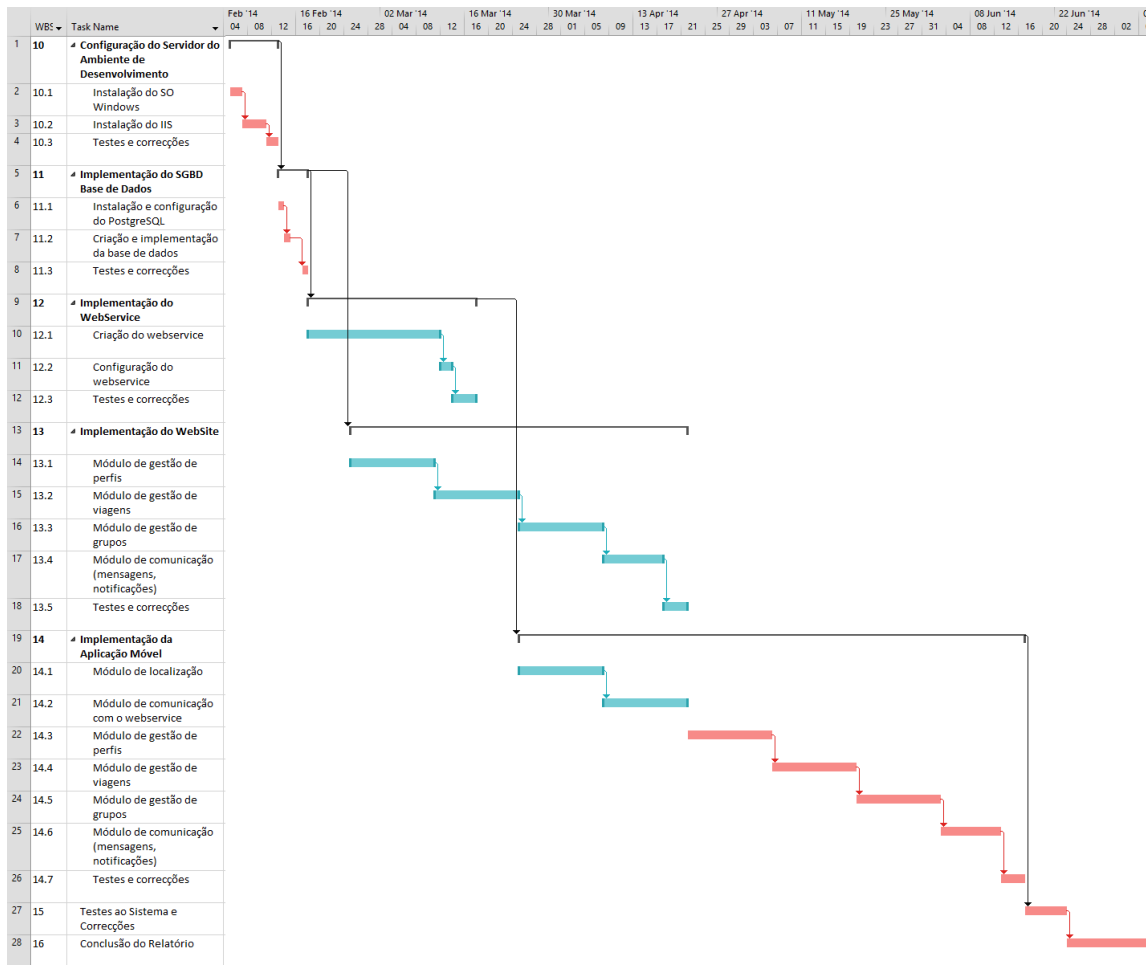


Figura 53 - Mapa Gantt do 2º semestre.

4.2. Riscos e Estratégias de Mitigação

Nesta secção serão abordados alguns riscos tomados em conta na criação deste projecto, bem como algumas estratégias para reduzir esses riscos.

4.2.1. Riscos

Para a implementação deste sistema devemos ter em conta os seguintes factores de risco:

1. O primeiro risco a ter em conta é a programação em Android que será usada na implementação da aplicação móvel. Pois apesar da programação nesta plataforma ser feita em JAVA, existem algumas especificidades do Android que precisam ser tidas em conta, como integração com API's de navegação.
2. A pouca experiência na programação em Android pode levar a maior tempo de adaptação e conseqüentemente mais tempo nas actividades de implementação.
3. Outro risco apontado foi o facto de não existir arquitectura definida. Como este sistema pretende ser criado de raiz, a sua especificidade elevada pode levar a alguma demora na investigação e planeamento da arquitectura pois têm de ser analisadas várias componentes relacionadas com a prática do *carpooling*.

4.2.1. Estratégias de Mitigação

Para tentar reduzir os riscos referidos na subsecção anterior foram definidas as seguintes estratégias de mitigação (a numeração indica o risco a mitigar):

1. Efectuar pesquisas em toda a documentação Android e APIs que serão usadas no sistema.
2. Aceder a tutorias e exemplos sobre programação em Android. Fazer um controlo rigoroso na implementação de cada componente da aplicação de forma a prevenir atrasos.
3. Neste caso a estratégia de mitigação passa por planear bem cada fase na criação da aplicação, para que o tempo disponível para investigação e planeamento da arquitectura seja adequado às necessidades destas tarefas.

4.3. Caminhos Críticos e Desvios

Tendo em conta o planeamento do projecto, deve ser referido ainda uma sequência de actividades que requerem atenção adicional por representarem pontos cruciais do desenvolvimento. Essa sequência de actividades é considerada um caminho crítico pois estas devem ser concluídas nas datas programadas para que o projecto possa ser concluído dentro do prazo final, caso contrário, no mínimo uma das actividades do caminho crítico não será concluída na data planeada [64].

Como se pode verificar na Figura 53 no segundo semestre havia três actividades cujo planeamento, ao falhar, punham em causa o cumprimento dos prazos finais do projecto. Essas actividades eram a configuração do servidor do ambiente de desenvolvimento, a criação e implementação da base de dados do sistema e por fim a aplicação móvel. Destas três actividades só a implementação da aplicação móvel apresentou alguns receios que foram abordados na secção anterior com as respectivas formas de mitigação. No entanto, apesar dos cuidados nesta fase do planeamento não foi possível cumprir alguns dos prazos estipulados no plano, o que levou a alguns desvios no planeamento do projecto.

Foi então feito um novo planeamento para o segundo semestre que representamos detalhadamente na tabela abaixo:

Actividade	Início	Fim	Detalhes
10	05-02-2014	12-02-2014	Configuração do Servidor do Ambiente de Desenvolvimento
11	13-02-2014	17-02-2014	Implementação da Base de Dados
12	18-02-2014	17-03-2014	Implementação do <i>WebService</i>
13	25-02-2014	07-04-2014	Implementação do <i>WebSite</i>
14	25-03-2014	30-07-2014	Implementação da Aplicação Móvel
15	01-08-2014	14-08-2014	Conclusão do Relatório
16	31-07-2014	28-08-2014	Testes ao Sistema e Correções
17	01-09-2014	02-09-2014	Entrega e Ultimas Correções

Tabela 5 - Planeamento real de actividades do segundo semestre.

A Figura 54 representa o mapa Gantt com o planeamento real do projecto após os desvios da actividade de implementação da aplicação móvel. Como se pode verificar no mapa Gantt houve algumas alterações ao plano inicial do segundo semestre pois o desvio no tempo de desenvolvimento no módulo de viagens veio alterar a tática de abordagem a outras actividades.

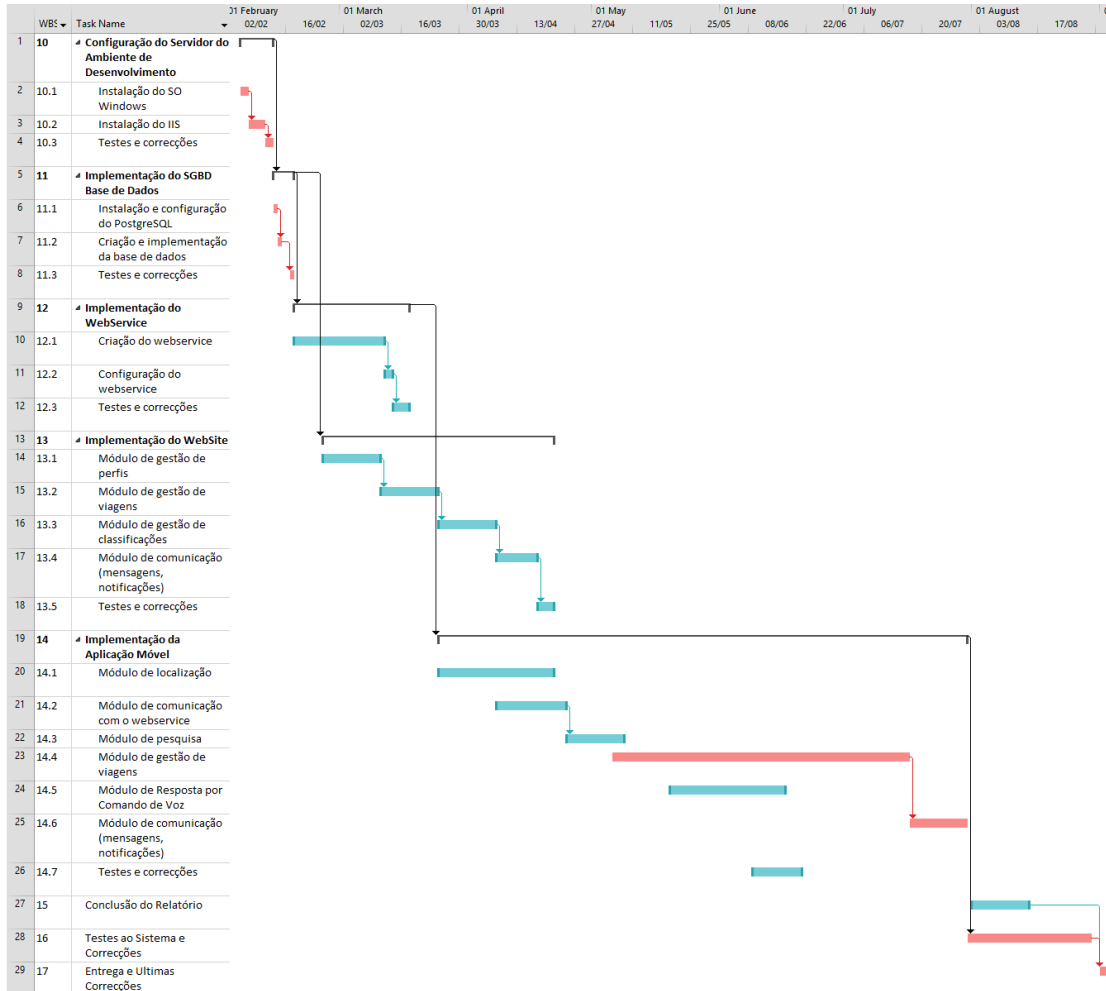


Figura 54 - Mapa Gantt do planeamento real do 2º semestre.

Capítulo 5

Escolhas Tecnológicas

Neste capítulo é feita a análise de nível mais elevado em que se encontram as várias temáticas, tanto a nível de ferramentas como de linguagens de programação, que possam auxiliar na construção do sistema a implementar.

5.1. Desenvolvimento da Aplicação Móvel

No desenvolvimento de uma aplicação móvel temos de abordar questões como a de arquitectura multiplataforma. Para compreendermos esta arquitectura teremos antes de analisar tipos de estratégias usadas no desenvolvimento de aplicações para dispositivos móveis. Existem 3 conceitos conhecidos deste tipo de estratégias de desenvolvimento, como segue [56]:

- **Aplicações Nativas:** São desenvolvidas especificamente para uma determinada plataforma. Estas aplicações são desenvolvidas usando código nativo “sdks” e terão um código muito relacionado com a plataforma sobre a qual foram escritas.
- **Aplicações Web:** Aplicações desenvolvidas usando tecnologias web padrão: HTML5, JavaScript e Css. São executadas em qualquer dispositivo com acesso à internet através de um *browser*.
- **Aplicações híbridas:** Estas aplicações, de uma forma geral são desenvolvidas usando padrões *web* e envolvidas numa “Shell” nativa. As “shells” possibilitam o acesso a características específicas das plataformas “mobile” tais como GPS, câmara, ou à aparência fornecida pelo UI (*User Interface*) da plataforma, etc.

A Figura 55 apresenta uma comparação entre os 3 tipos de abordagens de desenvolvimento móvel referidos acima com a indicação da sua aptidão multiplataforma.

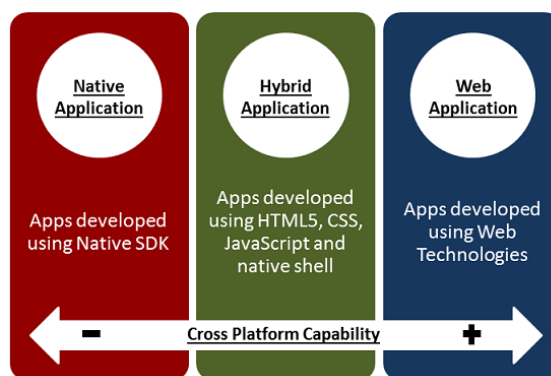


Figura 55 - Comparação de estratégias de desenvolvimento móvel. Fonte: [56].

5.1.1. Vantagens e Desvantagens dos Tipos de Desenvolvimento [56]

As aplicações nativas oferecem muito poucas ou nenhuma vantagens multiplataforma, e portanto não são aconselháveis para uma arquitectura multiplataforma. No entanto, no desenvolvimento de aplicações em Windows Phone 8 (WP8) existe uma margem para a partilha de código. A reutilização de código C# para outras plataformas de desenvolvimento pode ser conseguido através de ferramentas de desenvolvimento como o Xamarin. Para além disto pode-se partilhar código para o Windows 8 usando técnicas como *Portable Class Library* e *Windows Runtime API*.

As aplicações *web* permitem uma aptidão multiplataforma com algumas limitações. No desenvolvimento de aplicações deste tipo não é possível tirar partido de um UI nativo, para além disso o processamento local não será uma boa opção para aplicações cuja performance seja crítica. Nos casos em que o desempenho não seja um problema e apenas se precisa de hospedar uma aplicação *web*, então a componente multiplataforma fornecida é suficiente, sem colocar em causa o uso da Framework.

Combinando as vantagens do desenvolvimento de ambos os tipos de aplicações, as aplicações do tipo híbridas, são a melhor escolha para criar aplicações multiplataforma. Algumas das vantagens do desenvolvimento híbrido são a partilha de código que é desenvolvido apenas uma vez e é usado para fazer o “deploy” em várias plataformas de dispositivos móveis; Usa o mesmo código de interface “UI” HTML / “Third Party Library Code” para múltiplos dispositivos; Chamadas nativas usando a “Native Shell”; O modo *offline* permite ter acesso à aplicação mesmo que a internet não se encontre disponível; Permite que um grande número de utilizadores possa obter a aplicação devido ao seu carácter multiplataforma.

A Figura 56 representa uma comparação das vantagens e desvantagens destes tipos de desenvolvimento.

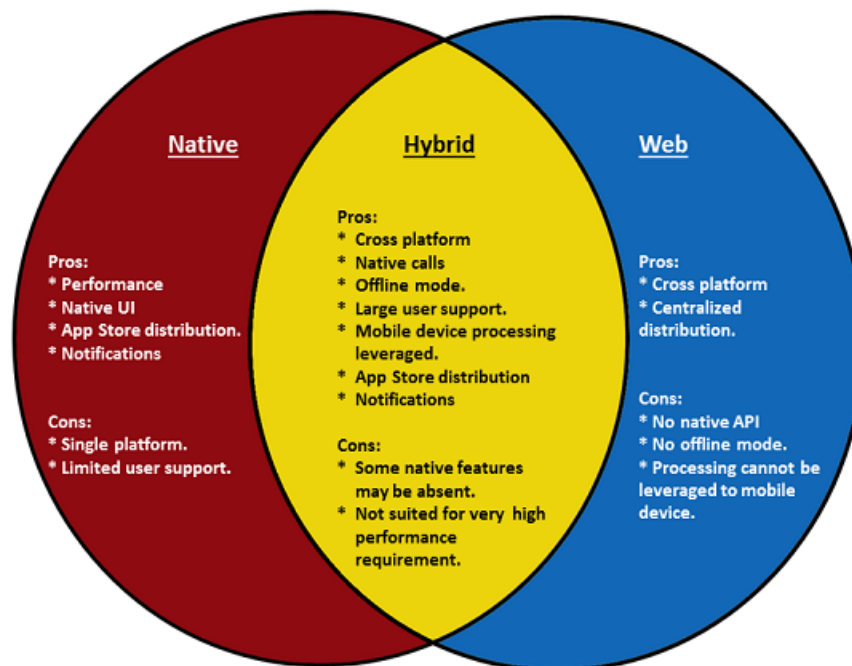


Figura 56 - Vantagens e desvantagens dos tipos de desenvolvimento móvel. Fonte: [56].

5.1.2. Arquitectura Multiplataforma [56]

Numa arquitectura multiplataforma em que o desenvolvimento híbrido é utilizado, as tecnologias Web como HTML5, Javascript e CSS são utilizadas dentro da “Shell” nativa da plataforma Mobile. Estas aplicações usam um tipo de controlo *web* para apresentar o Web UI de maneira a que seja possível “codificá-lo” em tecnologias *web*. Graças à grande evolução de ferramentas multiplataforma, tanto ao nível do desempenho como do aspecto, usar este tipo de aplicação dá uma sensação de que se está a utilizar um UI nativo. A Figura 57 representa a arquitectura multiplataforma de um sistema móvel.

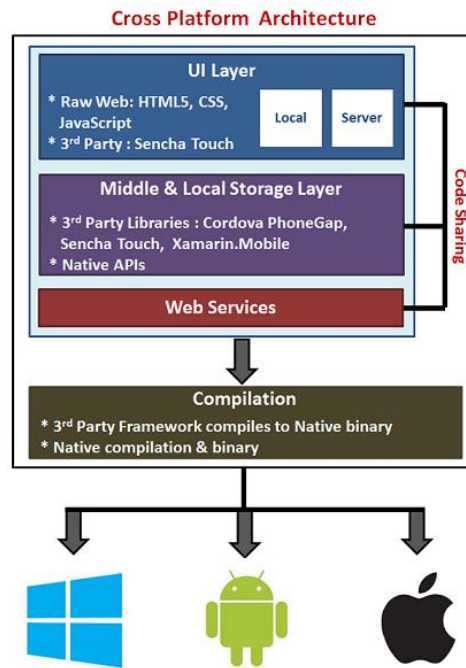


Figura 57 - Arquitectura multiplataforma. Fonte: [56].

5.1.3. Métodos de Implementação Multiplataforma [56]

Para implementação multiplataforma são utilizados os seguintes métodos:

- **Aplicação Híbrida Local:** Esta arquitectura faz uso do poder de processamento do dispositivo levando a um melhor desempenho, apesar de não ser tão bom como numa nativa. Para além disso as chamadas para os servidores serão evitadas, o que funcionará como um intensificador de performance. O “*deploying*” irá necessitar de uma notificação de actualização para o utilizar da mesma forma que acontece para aplicações nativas.
- **Aplicação Híbrida Servidor:** Nesta abordagem o processamento é todo feito no servidor que pode melhorar ao fazer um “*local caching*”, apesar de não ser o ideal para aplicações cujo desempenho é crítico. Mudanças no “*deploying*” serão fáceis pois basta mudar o código e por no servidor.

5.1.4. Ferramentas/API's para desenvolvimento Híbrido [56]

O PhoneGap (Apache Cordova), Sencha Touch, Xamarin e jQuery Mobile são alguns exemplos de ferramentas multiplataforma que fornecem uma grande ajuda na compatibilidade multiplataforma. O código é escrito numa linguagem e depois é compilado para as plataformas alvo como WP8, iOS, Android entre outros. Numa aplicação híbrida típica, apenas o código reutilizável e a aptidão para multiplataforma são algo limitados pela tecnologia web utilizada. Mas agora a funcionalidade nativa pode ser utilizada usando estes *frameworks* criando assim aplicações híbridas que sejam verdadeiramente multiplataforma e muito mais reutilizáveis.

- **PhoneGap (Apache Cordova):** Contém algumas das bibliotecas JavaScript que podem ser invocadas com código específico do dispositivo para as respectivas bibliotecas JavaScript. Isto combinado com um UI Framework como por exemplo JQuery ou Sencha Touch cria uma grande aplicação multiplataforma. É usado o JavaScript para aceder ao API nativo, como localização, câmara, armazenamento, etc. usando uma única chamada da API da plataforma mobile. No entanto, existe uma

desagregação do conceito do desenvolvimento, pois mesmo sendo o código apenas desenvolvido uma vez, é necessário uma máquina com o sistema operativo da Apple para desenvolvimento iOS, ou no caso do Android os requisitos são similares aos do Android, o Eclipse juntamente com o JDT, o Java JDK 6+, o plugin ADT (Android Development Tools) e por último o SDK (Software Development Kit) do Android.

- **Sench Touch:** Proporciona importantes “frameworks” HTML5 e API *wrappers* nativos. É codificado apenas uma vez e depois compilado para uma diferente plataforma mobile fazendo dela uma aplicação multiplataforma. Sendo a aplicação desenvolvida em HTML5 possui limitações nas possibilidades de interacção com o dispositivo, não conseguindo aceder à larga maioria das funcionalidades do mesmo (excepção da localização e do acelerómetro), assim como a integração com as aplicações nativas (ex.: calendário, maps). Outro problema é o facto de se pagar para ter acesso a todas as opções e funcionalidades desta ferramenta [73].
- **Xamarin:** Tira partido de uma lógica já existente em aplicações WP8 ou Windows 8 que está em C# e pode partilhar este código em várias outras plataformas. O compilador do Xamarin em .Net cria um executável nativo, um pacote de instalação de uma aplicação IOS ou Android. O Xamarin estende a partilha de código não só para várias plataformas mobile mas também com código para Windows 8. Apesar de muito completo esta ferramenta, ao contrário das outras aqui abordadas, não é gratuita [74].
- **jQuery Mobile:** É uma “framework” otimizada para o toque baseado em jQuery como linguagem de programação e HTML5 para a configuração das páginas. É possível a criação de aplicações multiplataforma usando esta Framework. Possui no entanto limitações nos dispositivos em que funciona, criando ai um grande entrave na sua utilização, pois surgiria uma limitação muito grande ao cliente final, obrigando-o a ter de adquirir determinados dispositivos em vez dos que já possui (caso estes não sejam compatíveis) [72].

5.1.5. Conclusão

Nos pontos acima foram analisadas as vantagens da arquitectura multiplataforma e a sua implementação usando varias estratégias de desenvolvimento, no entanto não existe uma metodologia que se possa classificar como perfeita, portanto deve ser utilizada aquela que melhor serve os nossos propósitos.

Portanto, como nas ferramentas híbridas existem ainda algumas limitações nas funcionalidades nativas que se conseguem obter, mais precisamente no acesso ao acelerómetro, câmara, contactos, ficheiros, localização, ficheiros media, rede móvel, alertas sonoros, vibração e possibilidade de usar os sistemas de armazenamento de dados do dispositivo. Outro problema prende-se ao facto de sistemas mais completos e com menos limitações como o Xamarin não serem gratuitos. Por todas estas razões iremos optar por uma implementação nativa na implementação da aplicação móvel MoCaS.

Depois de analisados os principais sistemas no capítulo anterior concluímos que o sistema operativo a adoptar no nosso projecto é o Android. Esta escolha deve-se essencialmente ao elevado número de pessoas que utilizam *smartphones* com este sistema operativo, levando este a ser líder actual no mercado dos *smartphones*, destacando-se assim ainda mais dos rivais mais directos (Windows Phone 8 da Microsoft e o iOS da Apple) [32].

Isto trará grandes vantagens à aplicação pois o leque de utilizadores que possivelmente irão utilizar a aplicação é maior e assim é esperado que a sua utilização seja mais alargada.

A plataforma de desenvolvimento seleccionada para a criação da aplicação móvel é o Android Studio. Esta nova plataforma que foi criada pela Google, oferece as melhores ferramentas e funcionalidades aos programadores. Segundo a empresa, o Android Studio é a ferramenta de programação para Android mais simples e rápida [45].

5.2. Desenvolvimento do *Website*

Outra componente importante deste sistema é o *website*, este irá abrir portas ao uso do sistema em qualquer dispositivo computacional com ligação à internet. O *website* irá servir os propósitos fundamentais do sistema MoCaS, ou seja, oferecer e pedir viagens. Para além destas funcionalidades também será responsável por todo o registo dos utilizadores, atribuição de classificações e inserção de veículos.

5.2.1. Tecnologias de Desenvolvimento *Web*

Para ter uma melhor percepção das tecnologias *web* existentes no mercado serão analisadas algumas delas com intuito de escolher a que melhor se adapta ao sistema MoCaS.

As tecnologias escolhidas para esta análise foram as seguintes:

- **ASP .NET:** É a plataforma de desenvolvimento de aplicações web da Microsoft. O ASP.NET é baseado na Framework .NET herdando todas as suas características, portanto, qualquer aplicação para esta plataforma pode ser escrita em linguagens como o C# e Visual Basic .NET. Para serem executadas, as aplicações Web ASP.NET necessitam da Framework .NET e do servidor IIS (Internet Information Services) para a plataforma Windows [49].
- **PHP (*Hypertext Preprocessor*):** Criado por Rasmus Lerdorf na Dinamarca, posteriormente começou a ser desenvolvido como código livre e actualmente é desenvolvido pelo PHP Group. O PHP não é um padrão *web* mas sim uma tecnologia de código aberto. No entanto, não é uma linguagem de programação no sentido da palavra, mas sim uma tecnologia que permite a colocar *scripts* embutidos no HTML delimitados por *tags* de início e fim [67].
- **Java:** Refere-se a um conjunto de programas e padrões inicialmente criados pela Sun Microsystems, agora propriedade da Oracle Corporation. A plataforma Java inclui o Java Development Kit (JDK), que inclui o compilador Java e outras utilidades, e Java Runtime Environment (JRE), que inclui a Máquina Virtual Java (JVM). Esta plataforma também inclui o Java Class Libraries, um conjunto de bibliotecas de código reutilizável que contém a maioria das APIs e funções que seriam fornecidos pelo sistema operacional. Esta plataforma utiliza a linguagem de programação Java, com a qual os programadores podem escrever as suas aplicações [66].

5.2.2. Comparação PHP vs. ASP.NET vs. JAVA

Para uma melhor análise a cada uma destas tecnologias iremos comparar algumas vantagens e desvantagens de cada um deles através da Tabela 6 abaixo.

	PHP [68,67]	ASP .NET [68,66,49]	Java [68,66]
Vantagens	<ul style="list-style-type: none"> • Fácil aprendizagem; • Conecta-se facilmente a sistemas Sybase, MySQL, MS-SQL, Oracle e muitos outros compatíveis com o padrão ODBC; 	<ul style="list-style-type: none"> • Suporte multilinguagem (suporta linguagens como C++, C#, Visual Basic.NET, Jscript.NET, Python, entre outras); 	<ul style="list-style-type: none"> • O JDK é uma ferramenta gratuita. • Suportado por um grande número de fornecedores; • Permite aos programadores

	<ul style="list-style-type: none"> • É multiplataforma pois funciona em Linux, Windows, entre outros; • É gratuito, por isso é apresentado como uma alternativa de fácil acesso para todos. 	<ul style="list-style-type: none"> • As aplicações .NET são compiladas o que as torna mais rápidas; • Facilidade de aprendizagem, uma vez que o programador pode escolher entre muitas linguagens de programação; • Facilidade de se conectar a qualquer base de dados. 	<ul style="list-style-type: none"> • aproveitar a flexibilidade de programação orientada a objectos no desenho de suas aplicações; • É independente da plataforma de desenvolvimento.
Desvantagens	<ul style="list-style-type: none"> • A leitura do código pode ser afectada pela mistura de declarações HTML e PHP; • Programação orientada a objectos ainda é muito pobre para grandes aplicações; 	<ul style="list-style-type: none"> • Para <i>websites</i> mais complexos a versão do grátis do Visual Studio não será a mais indicada tendo de se pagar pela versão avançada. • Só pode ser instalado em servidores com sistema operativo Windows. 	<ul style="list-style-type: none"> • Existem diferentes tipos de suporte para a mesma ferramenta, o que torna análise da melhor escolha difícil; • Algumas ferramentas têm um custo adicional.

Tabela 6 - Vantagens e desvantagens das tecnologias PHP, Java, ASP .NET.

5.2.3. Conclusão

Depois de analisados alguns pontos importantes das tecnologias web acima mencionadas, podemos concluir que qualquer uma delas seria uma boa opção para a implementação da aplicação *web* do sistema MoCaS.

No entanto, tendo em conta a experiência profissional que o autor possui na tecnologia ASP .NET será feita a escolha desta tecnologia com o argumento de que poderá contribuir para uma melhor prestação não só ao nível técnico mas também ao nível temporal do projecto, que poderá ver reduzido o seu tempo de implementação.

O desenvolvimento em ASP.NET será feito através da ferramenta da Microsoft, o Visual Studio .NET. Esta ferramenta de desenvolvimento já possui algumas características que facilitam o trabalho de quem programa, como os componentes visuais para criação de formulários de páginas *web* [49].

5.3. Base de Dados do Sistema

Como em qualquer sistema de informação actual, este terá de armazenar informação de diversas naturezas, por exemplo a informação relativa ao perfil dos *carpoolers*, veículos, avaliações, etc. Esta informação terá de estar sempre disponível e actualizada para que todo o sistema funcione como pretendido. Como a base de dados (BD) terá de interagir com a aplicação web e móvel em simultâneo, esta terá de ser suficientemente robusta para que não existam erros de sincronização.

5.3.1. Tipos de Base de Dados [50]

Para esta escolha tecnológica tivemos de ter em conta o tipo base de dados, ou seja, se pretendemos um sistema relacional baseado SQL (*Structured Query Language*) ou um não relacional (NoSQL).

- **Modelo Relacional:** Oferece uma forma de adaptação muito matemática de estruturar, manter e utilizar os dados. Este modelo amplia o *design* de anteriores modelos como o horizontal, o de rede, etc., introduzindo o conceito de relações. As relações vêm trazer benefícios de agrupamento de dados, como colecções restritas de tabelas de dados, que contêm as informações de forma estruturada (por exemplo, nome e morada de uma pessoa), relacionam todas as entradas, atribuindo valores aos atributos (por exemplo, número de bilhete de identidade de uma pessoa). A linguagem padrão de BD relacionais é o SQL.
- **Modelo não Relacional:** A forma de estruturação de dados NoSQL consiste em livrar-se das restrições do modelo relacional, ou seja, libertar os meios de manter, consultar e utilizar informações. Estas bases de dados utilizam uma abordagem não estruturada, que procura eliminar as limitações de relações estritas, e oferecem muitas formas diferentes de manter e trabalhar com os dados para um uso específico de casos de forma eficiente (por exemplo, o armazenamento de documentos de texto).

5.3.2. Comparação de sistemas baseados em SQL e NoSQL [50]

Para perceber melhor as diferenças destes dois tipos modelo de base de dados vamos analisar algumas características comuns às bases de dados:

- **Estrutura e tipo de dados a ser mantido:** As BD baseadas em SQL exigem uma estrutura definida com atributos para armazenar os dados, ao contrário das baseadas em NoSQL, que permitem normalmente as operações livres de estrutura.
- **Consultas:** Independentemente das suas licenças, todas as BD relacionais implementam o padrão SQL e assim, podem ser feitas consultas usando o SQL. As BD NoSQL, por outro lado, implementam um modo único para trabalhar com os dados que gerem.
- **Escalonamento:** Ambas as soluções são fáceis de escalar verticalmente (ou seja, aumentar os recursos do sistema). No entanto, sendo as aplicações cada vez mais modernas (e mais simples), as soluções NoSQL costumam oferecer meios muito mais simples de escalonamento horizontal (por exemplo, pela criação de um cluster de várias máquinas).
- **Confiabilidade:** Quando se trata da fiabilidade dos dados e da garantia segura de transacções realizadas, as BD SQL ainda são a melhor aposta.
- **Suporte:** Os Sistemas de Gestão de Base de Dados (SGBD) relacionais têm décadas de história. Estes sistemas são extremamente populares e é muito fácil encontrar suporte de forma paga e gratuita para ambos. Portanto, se surgir um problema, é muito mais fácil de resolver comparando com SGBD baseados em NOSQL que são mais recentes, especialmente se a dita solução é de natureza complexa (por exemplo, MongoDB).
- **Manutenção de dados complexos e necessidades de consulta:** Por natureza, as BD relacionais são a melhor solução para consultas complexas e manutenção de dados, pois são mais eficientes e têm mais sucesso neste domínio.

Depois de esta análise aprofundada às características destes dois modelos de BD podemos concluir que os sistemas baseados em SQL ainda oferecem maior confiabilidade dos

dados e garantia segura de transacções, um maior suporte para resoluções de problemas e mais eficiência em consultas complexas [50]. Portanto, tendo em conta a complexidade do sistema MoCaS iremos optar por este tipo de SGBD.

5.3.3. Sistemas de Gestão de Base de Dados

Depois de optar por um modelo de BD relacional iremos abordar a escolha do SGBD que irá suportar toda a informação do sistema MoCaS. Para isso serão analisados os dois SGBD mais populares do mercado, o PostgreSQL e o MySQL [65]. Outros SGBD, como o Oracle e o SQL Server foram desconsiderados para esta comparação por não oferecerem opções gratuitas viáveis para aplicação do SGBD do sistema MoCaS.

- **PostgreSQL:** É baseado no Postgres desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. O Postgres foi o fundador de alguns conceitos que só se tornaram disponíveis mais tarde em alguns SGBD comerciais. Este SGBD é de código fonte aberto e suporta grande parte do padrão SQL 2003 [48].
- **MySQL:** O nome deste SGBD vem do nome de uma das filhas (My) de um dos co-fundadores, Michael Widenius. O código fonte está disponível de forma aberta e é baseado em SQL. Este SGBD foi criado pela empresa sueca MySQLAB no entanto agora é propriedade da Oracle Corporation. O MySQL é amplamente utilizado em vários *websites* que requerem um SGBD com todos os recursos de BD que o MySQL costuma usar [65].

5.3.4. Comparação PostgreSQL vs. MySQL

Para esta comparação serão abordadas vantagens e desvantagens destes dois SGBD. O MySQL é o mais utilizado no desenvolvimento de aplicações onde se dá grande prioridade à velocidade (no entanto, isto tem mudado com as recentes versões PostgreSQL), enquanto que o PostgreSQL se destaca por ser ter mais robustez e possuir muito mais recursos que o MySQL [48].

Nas últimas versões do MySQL, os programadores adicionaram já alguns recursos que existiam no PostgreSQL como transacções (confirmação e cancelamento de operações realizadas COMMIT e ROLLBACK), *Triggers*, *Stored Procedures*, *Views*, *Lock Line* e *Constraints* [48].

Apesar de tudo isso o PostgreSQL continua a ser o mais eficiente em vários aspectos pois possui um mecanismo inovador de bloqueio chamado *Multi-Version Concurrency Control* (MVCC) que, ao contrário de outros SGBD que utilizam *Locks* para controlo de concorrência, o PostgreSQL mantém a consistência dos dados usando um modelo multiversão. Além disso o PostgreSQL suporta tamanhos ilimitados de linhas, bancos de dados e tabelas (até 16TB), aceita vários tipos de subconsultas, tem mais tipos de dados e conta com um bom mecanismo de segurança contra falhas (por exemplo na falha repentina do sistema) [48].

5.3.5. Conclusão

Portanto, como analisado ao longo da secção 5.3. Base de Dados do Sistema, o MySQL destaca-se pela velocidade de acesso a dados (que por vezes nem é notada porque está num escala de milésimos de segundo), no entanto para bases de dados muito grandes, complexas e que exigem fiabilidade e escalabilidade, vale a pena usar o PostgreSQL. Como a

BD do sistema MoCaS terá várias transacções a decorrer em simultâneo (principalmente na componente de tempo real), deve ser garantida a sua robustez e consistência das transacções, portanto é importante usar mecanismos como os que possui o PostgreSQL [48]. Por isso pensamos ser o PostgreSQL o SGBD mais indicado para o sistema MoCaS.

5.4. Comunicação Aplicação-Servidor

Para integração dos sistemas e comunicação entre as diferentes aplicações é necessário recorrer a um *web service* que realize esta tarefa. Com este *web service* é possível a novas aplicações interagir com aquelas que já existem e a que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

5.4.1. Arquitecturas de Comunicação

Para fazer a comunicação entre o *web service* e a aplicação móvel e web terá de ser usado um mecanismo de comunicação entre estes sistemas distintos. Neste caso, foram analisadas duas opções, o REST (*Representational State Transfer*) e o SOAP (*Simple Object Access Protocol*).

- **REST:** Consiste numa arquitectura simples sem estado, geralmente usada no sentido de descrever qualquer interface web simples, sendo um dos principais pontos da arquitectura os *Web Services* [59]. Utiliza preferencialmente o protocolo funcional HTTP (*Hypertext Transfer Protocol*) e o formato de dados XML (*eXtensible Markup Language*), sem as abstracções adicionais dos protocolos baseados em padrões de trocas de mensagem como o protocolo de serviços web SOAP. Por omissão o REST usa os métodos do HTTP, GET, POST, PUT e DELETE facilitando assim a sua interoperabilidade e facilidade de comunicação com um servidor [58].
- **SOAP:** Este protocolo é usado na troca de informação estruturada aquando da implementação de *Web Services*. O seu formato de dados adoptado na troca de mensagens é o XML sendo que necessita de outros protocolos, como os de comunicação, sendo eles o HTTP, TCP (*Transmission Control Protocol*) ou SMTP (*Simple Mail Transfer Protocol*) para negociação e troca de mensagens e faz uso do método POST do HTTP devido à necessidade de inclusão da mensagem SOAP. Este protocolo baseado em XML consiste em três partes, um envelope que define o que é a mensagem e como a processar, um conjunto de regras codificadas para indicar instâncias dos tipos de dados definidos na aplicação e, por último, uma convenção para representar chamadas e respostas aos serviços [55].

5.4.2. Comparação REST vs. SOAP [63]

Para compararmos estas duas opções vamos analisar algumas das principais vantagens de cada uma delas. Começando pelo REST, este é mais simples de entender pois usa o padrão HTTP, permite formatos de dados diferentes enquanto que o SOAP só permite o XML, o REST também tem um melhor desempenho e escalabilidade e permite guardar os dados lidos em “*cache*”, ao contrário do SOAP.

No entanto há também algumas razões para o uso do SOAP como o suporte ao *WS-Security* que acrescenta alguns recursos de segurança, para além do SSL. O SOAP também suporta transacções ACID (*Atomicity, Consistency, Isolation, and Durability*) ao contrário do REST que não é tão abrangente pois é limitado pelo HTTP em si, que não pode fornecer transacções de duas fases através de recursos transaccionais distribuídos. O SOAP possui mecanismo para lidar com falhas de comunicação embutido o que oferece uma maior fiabilidade.

5.4.3. Representação de dados

Em relação ao formato dos dados que se podem transferir entre o servidor e o dispositivo móvel serão analisados dois casos, o JSON (*JavaScript Object Notation*) e o XML. Como ambos são os formatos mais utilizados [57], estes foram alvo de uma análise.

- **XML:** Foi criado pelo World Wide Web Consortium (W3C) com o intuito de ser um standard, ou uma linguagem que fosse flexível e ao mesmo tempo simples para facilitar a partilha de informação através da Internet. O XML é uma linguagem de marcação (*Markup language*) como o HTML, em que as marcações são *tags*, que descrevem conteúdo no documento. No entanto, ao contrário do HTML, o XML foi desenhado para transportar informação e não para a partilhar. Uma outra diferença tem a ver com as *tags*, que neste caso não são predefinidas, permitindo ao programador definir a arquitectura do documento. O XML é baseado em texto simples e portanto é de fácil percepção e leitura [61].
- **JSON:** É uma sintaxe para transferência ou partilha de dados em formato de texto, parecido com o XML, no entanto o JSON é mais leve a nível sintáctico e mais fácil de interpretar. Tal como o XML também é auto descritivo e segue também uma estrutura hierárquica, ou seja valores dentro de valores, e acaba por poder ser "transportado" através de HTTP. Apesar destas semelhanças também existem algumas diferenças sobre o XML, como o facto do JSON não ter uma *tag* final, poupando assim caracteres, diminuindo o tamanho do pacote a enviar/receber e tem uma melhor performance ao nível da interpretação, tanto a ler como a escrever, não possuindo palavras reservadas, dando a possibilidade de usar estruturas de dados (*arrays*) [62].

5.4.4. Comparação XML vs. JSON

Para uma melhor análise destes formatos de dados iremos comparar as vantagens de cada um. O XML permite um uso dinâmico na medida em que cada ficheiro pode ser criado para corresponder às necessidades de quem o cria, definindo assim as *tags* a usar. Outra vantagem é o facto de ser baseado em texto simples e suportar Unicode. Suporta a representação para estruturas de dados, como listas, registos, árvores e possibilita a utilização de XML *Schemas* para validar tipos de dados e estrutura dos mesmos [61].

No caso do JSON, este possui uma sintaxe mais simples, o que resulta em mais facilidade na sua leitura. O JSON tem também uma maior velocidade na execução e transporte de dados do que o XML, além disso necessita de menos espaço para guardar os mesmos dados, facilitando as transferências de informação [62].

5.4.5. Conclusão

Depois de analisadas as arquitecturas de comunicação e a representação de troca de dados entre cliente e servidor, podemos concluir que há vantagens em todas elas, portanto as escolhas devem ser feitas conforme as necessidades do sistema.

Como o website está assente numa plataforma .NET será utilizado o ASP.NET Web Service que contém objectos e métodos que estão expostos através da internet usando uma simples pilha de protocolo de mensagens. Assim, em qualquer aplicação cliente é possível invocar um pedido ao *web service* através de HTTP utilizando um dos seus métodos [54].

Este tipo de *web service* para fazer a comunicação com a aplicação utiliza o protocolo de mensagens padrão SOAP, portanto é natural a nossa escolha por esta arquitectura de comunicação. Esta escolha implica desde logo que o formato de dados adoptado na troca de

mensagens seja o XML, pois é o único formato suportado pelo SOAP, tal como foi analisado no ponto 5.4.1. Architecturas de Comunicação [55].

Capítulo 6

Implementação

Este capítulo tem como objectivo descrever as principais implementações na criação do sistema MoCaS. Serão abordadas as construções de toda a estrutura do sistema com maior foco na implementação do serviço em tempo real de marcação de viagens.

Numa primeira fase serão abordadas as componentes abrangidas pelo *website* através da análise aos módulos de gestão de perfil, viagem, mensagens, no fundo toda a área pessoal, posteriormente será abrangida toda a área de pesquisa e marcação de viagens.

Depois será abordado o *webservice* que faz a interligação entre as plataformas distintas (aplicação móvel, *website* e base de dados) do sistema MoCaS.

Por fim, será dada uma explicação da componente mobile, não só as suas funcionalidades de marcação temporal e espacial mas também quais as versões do Android que suportam a aplicação.

É ainda de referir que em cada uma das estruturas do sistema serão apresentados os respectivos testes de performance de forma a tirar conclusões quanto ao desempenho geral do sistema MoCaS.

6.1. MoCaS *Web*

Nesta secção serão apresentados os interfaces da aplicação *web* do sistema MoCaS, sendo abordado em detalhe cada um dos módulos (página inicial, pesquisa e marcação de viagens, área de registo e *login*, gestão de perfil e preferências, gestão de veículos, gestão de viagens, gestão de classificações e área de notificações/mensagens).

6.1.1. Página inicial

A aplicação *web* do sistema MoCaS pretende ser simples e agradável de usar sendo que a apresentação da página inicial terá de ter cuidados nesse sentido. Como se pode verificar na Figura 58 a página inicial apresenta-se de uma forma limpa e agradável com os filtros de pesquisa de início e fim da viagem e adicionalmente data, tipo de viagem, se deseja ou não viajar com fumadores e qual a distância a que os utilizadores estão dispostos a fazer um desvio em cada viagem.

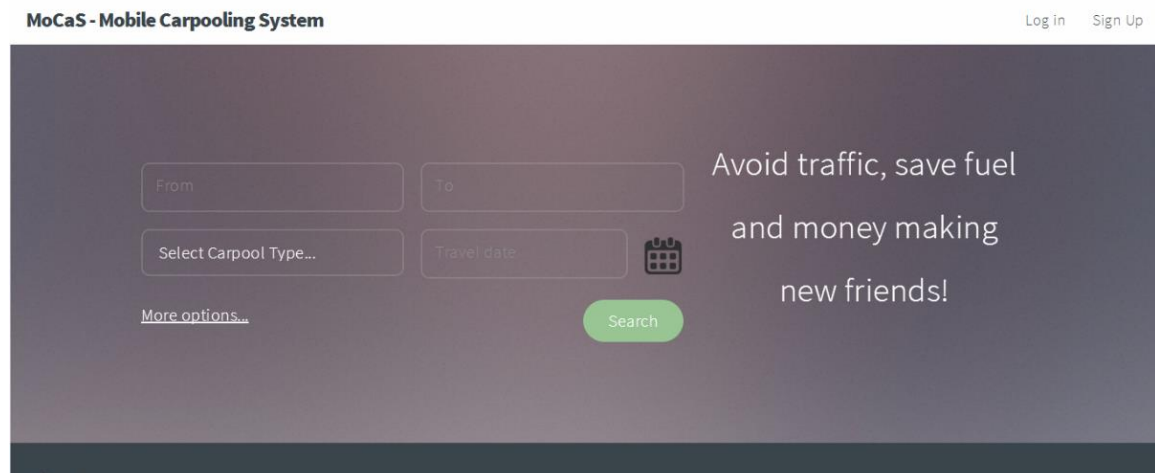


Figura 58 - Página inicial do *website* MoCaS.

6.1.1. Pesquisa e Marcação de Viagens

Após preenchidos os filtros de origem e destino e feita a pesquisa pelos parâmetros inseridos é mostrada uma nova página com os resultados da pesquisa. Os resultados são apresentados em forma de grelha, em que cada linha corresponde a uma viagem. Cada linha tem a informação sobre início e fim da viagem, quem a criou, hora e data da partida, preço, número de lugares e tipo de utilizador (condutor ou passageiro). Na Figura 59 pode verificar-se uma listagem de viagens entre Lisboa e Porto com um resultado obtido.

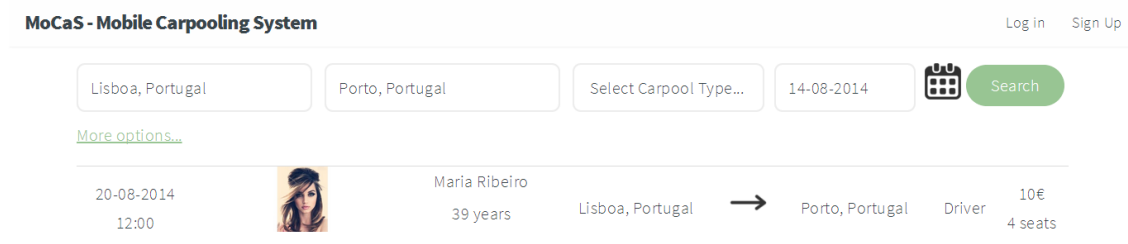


Figura 59 - Listagem de viagens entre Lisboa e Porto do *website* MoCaS.

Depois de encontrado o resultado pretendido é seleccionada a viagem que se pretende e irá aparecer todo o seu detalhe. Neste ecrã, que nos mostra a Figura 60, é possível ver os locais de paragem desta viagem (caso existam), a indicação de auto-estrada, alguns comentários sobre esta viagem, possíveis desvios espaciais e temporais para tornar as viagens mais flexíveis para quem se pretende associar à viagem, informação sobre o condutor e sobre o veículo, preferências de viagem do condutor, mapa com o percurso da viagem e indicações, classificação do veículo e do utilizador, preço e número de lugares disponíveis.

MoCaS - Mobile Carpooling System Log in Sign Up

Lisboa, Portugal → Porto, Portugal Show Map

Stops: Coimbra, Portugal

Departure Date: 20-08-2014 at 12:00

Travel Details

Deviation: I can move me up to 3 km from the travel route.
"Faço um pequeno desvio para apanhar alguém."

Vehicle Details

Vehicle: BMW 5 Series

Luggage: Small

Classification: ★★★★★

Driver Profile: Maria Ribeiro, 39 years, ★★★★★,

10€ per passenger
4 seats available
[Log in to contact](#)

Figura 60 - Detalhe de viagem do *website* MoCaS.

Para marcação de viagem, caso seja um passageiro à procura de um condutor para se associar, é possível contactar o mesmo através do telefone (caso esteja disponível), mensagem privada ou ainda, se o condutor permitir, a pré reserva de lugar. Caso o utilizador que procura viagens seja um condutor à procura de passageiros só é possível associar estes à viagem caso os percursos coincidam com as viagens criadas por este.

6.1.2. Registo e *Login*

Para usufruir de todas as potencialidades da aplicação é necessário que cada utilizador se registe no sistema. O registo no sistema pode ser feito através da criação de uma conta MoCaS através da opção de registo ou pode ser feita através da associação de uma conta Google ou Facebook (ver Figura 61). Este registo garante uma maior confiança no sistema MoCaS pois cada utilizador terá de possuir uma conta num dos sistemas mencionados para se autenticar e assim mais garantias dão dadas a quem partilha os serviços *carpooling*.

Após a autenticação o utilizador terá disponível toda uma área de gestão pessoal que será abordada nos pontos abaixo. É ainda de referir que caso o utilizador se esqueça da sua *password* pode pedir uma novas credenciais de acesso através da inserção do seu endereço email, posteriormente será enviado um email com o endereço encriptado para inserção de uma nova *password*.

Login

Use a local account to log in

Email

Password

Remember me?

[Log In](#)

Forgot your password? Click [Here](#).

[Register](#) if you don't have a local account.

Use another service to log in

[SIGN IN WITH GOOGLE+](#)

[SIGN IN WITH FACEBOOK](#)

Register

Create a new account

Name

Email

Password

Confirm password

Figura 61 - *Login* e registo do *website* MoCaS.

6.1.3. Perfil e Preferências

A gestão de dados pessoais e preferências de viagem de cada utilizador são informações fundamentais para a partilha de serviços *carpooling*. É através de um perfil mais completo e de preferências comuns de quem oferece e de quem procura viagem que pode haver ou não correspondências nas viagens partilhadas.

Como se pode ver na Figura 62, a área de perfil do utilizador apresenta uma série de campos editáveis para inserir ou editar dados pessoais relativos ao utilizador, são eles o nome, *email*, sexo, número de telefone, data de nascimento, morada, comentários adicionais, fotografia e se deseja que o seu número de telefone esteja disponível para contacto de partilha de viagem.

Figura 62 - Perfil do utilizador do *website* MoCaS.

A indicação das preferências de viagem como se pode ver na Figura 63 está dividida em cinco componentes. É possível indicar se é permitido transportar animais, se é permitido fumar, se o utilizador conversa durante as viagens, se pretende que a viagem seja feita só com mulheres e se o utilizador gosta de ouvir musica durante as viagens.

Figura 63 - Preferências de viagem do utilizador do *website* MoCaS.

6.1.4. Gestão de Viagens

Este módulo é fundamental para todo o sistema, pois é aqui que se definem as viagens a oferecer ou a procurar. Nesta área representada na Figura 64 é possível ao utilizador ver as suas viagens criadas e a que este se associou, editar viagens ou cancelar, ver as viagens passadas e criar novos pedidos ou ofertas de viagem.

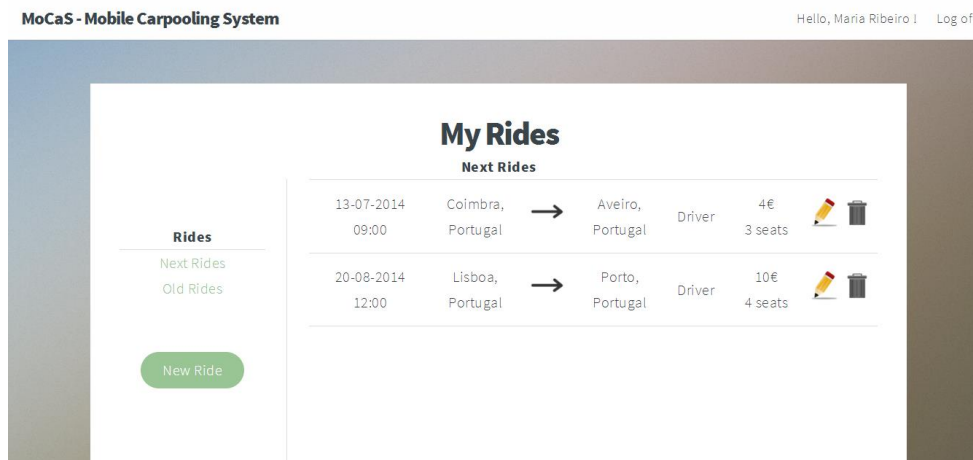


Figura 64 - Gestão de viagens do utilizador do *website* MoCaS.

A criação da viagem está dividida em três passos distintos, numa primeira fase o utilizador define a rota através de um ponto de início e um de fim em que pode adicionar pontos de paragem pelo meio. Como se pode verificar na Figura 65 o trajecto é apresentado no mapa através do serviço *Google Maps* juntamente com a informação de distância percorrida e tempo aproximado de viagem. Além disso é possível indicar se é pretendido viajar por auto-estradas ou estradas nacionais apenas. Neste formulário é ainda possível ordenar os pontos de paragem arrastando as caixas de texto na ordem que se pretenda.

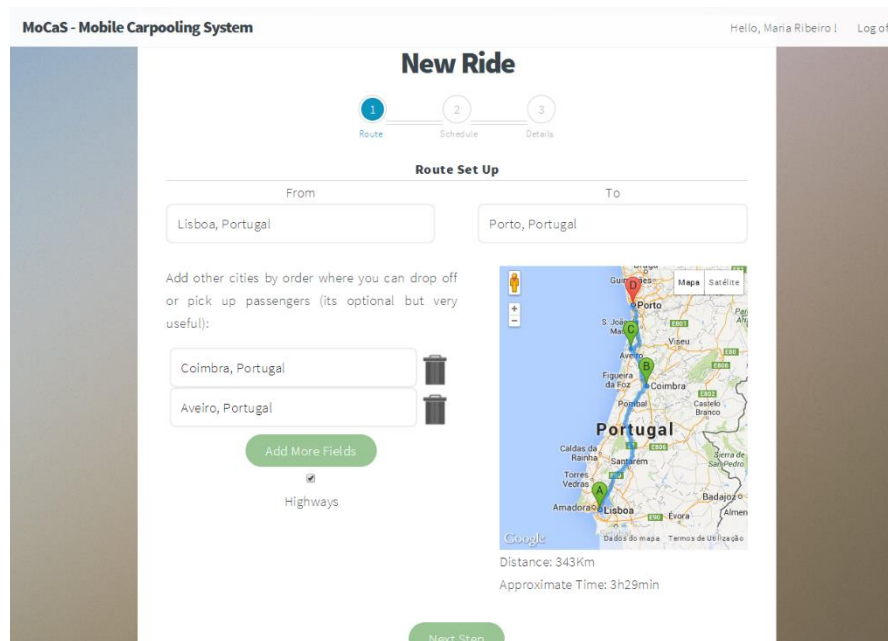


Figura 65 - Definição do trajecto de viagem do *website* MoCaS.

A segunda fase da criação de viagem assenta na base temporal, ou seja, na definição da data de partida, horário e, caso exista uma frequência, definir a mesma. Como apresenta a Figura 66, nesta fase é permitido definir se a viagem é de ida e volta, se é regular ou frequente, caso seja frequente qual é o período em que faz as viagens, dias da semana em que viaja, data de início e fim da frequência ou caso assim o deseje definir o número de ocorrências da frequência.

Figura 66 - Definição do horário e data da viagem no *website* MoCaS.

Por último, serão definidos os detalhes da viagem, ou seja, se o utilizador é passageiro ou condutor, caso seja condutor terá de indicar o veículo a usar, se permite reservas de lugar, o número de lugares disponíveis, o tipo de espaço na bagageira, comentários adicionais, os quilómetros de desvio da rota que está disposto a fazer para ir buscar alguém, pode definir o custo manualmente (o custo é calculado automaticamente com base nos dados da rota e será abordado em detalhe no ponto 6.1.10. Custo Automático), e definir a flexibilidade temporal para indicar se pode sair mais cedo ou mais tarde e chegar mais cedo ou mais tarde (ver Figura 67). É de referir que caso o utilizador opte por ser passageiro, só terá acesso a indicar a necessidade de número de lugares, bagagem, desvio, flexibilidade e informação adicional.

Figura 67 - Detalhes da viagem no *website* MoCaS.

6.1.5. Gestão de Veículos

Neste módulo é possível a cada utilizador inserir, editar e remover os seus veículos para posterior partilha de viagem. Como se pode ver na Figura 68 a área de gestão de veículos apresenta uma listagem dos veículos inseridos com informação resumida de cada um indicando a marca, modelo e tipo de combustível.

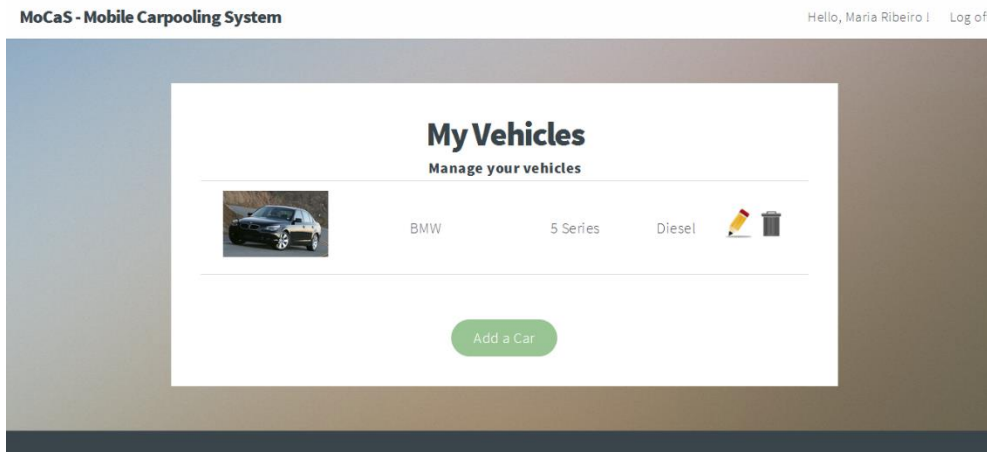


Figura 68 - Veículos do utilizador no website MoCaS.

Para adicionar um veículo é necessário inserir informação relativa à marca e modelo, ano de matrícula, número de lugares totais, cor, tipo de combustível a classe de auto-estrada e uma fotografia. Esta informação é extremamente importante pois servirão para garantir aos passageiros uma maior informação sobre o veículo em que vão viajar.

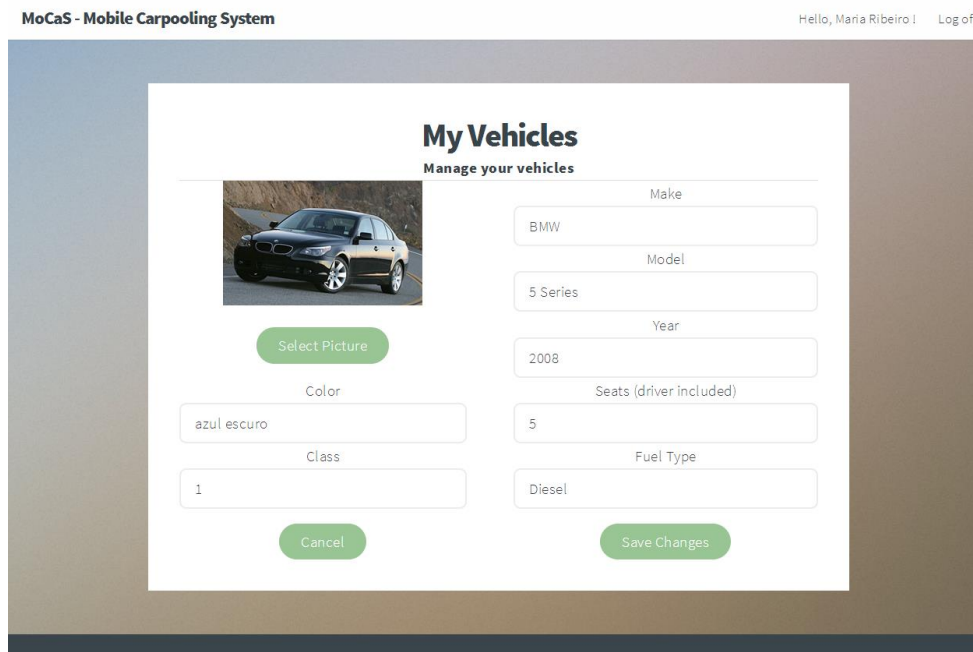


Figura 69 - Formulário de inserção/edição de veículo no website MoCaS.

6.1.6. Gestão de Classificações dos Utilizadores

Este módulo é extremamente importante em todo o sistema MoCaS, pois é através das classificações atribuídas por cada utilizador que é possível garantir um melhor e mais fiável serviço a cada passageiro ou condutor. Na gestão de classificações é possível a cada utilizador ver as classificações por atribuir, as classificações enviadas e recebidas e a qualquer momento

tem ainda a possibilidade de alterar classificações já atribuídas. A Figura 70 apresenta a listagem de classificações recebidas de um determinado utilizador. Nesta listagem é dada a informação de que utilizador veio a classificação e a avaliação média que atribuiu ao utilizador avaliado bem como ao seu veículo (no caso do utilizador avaliado ser condutor em determinada viagem).

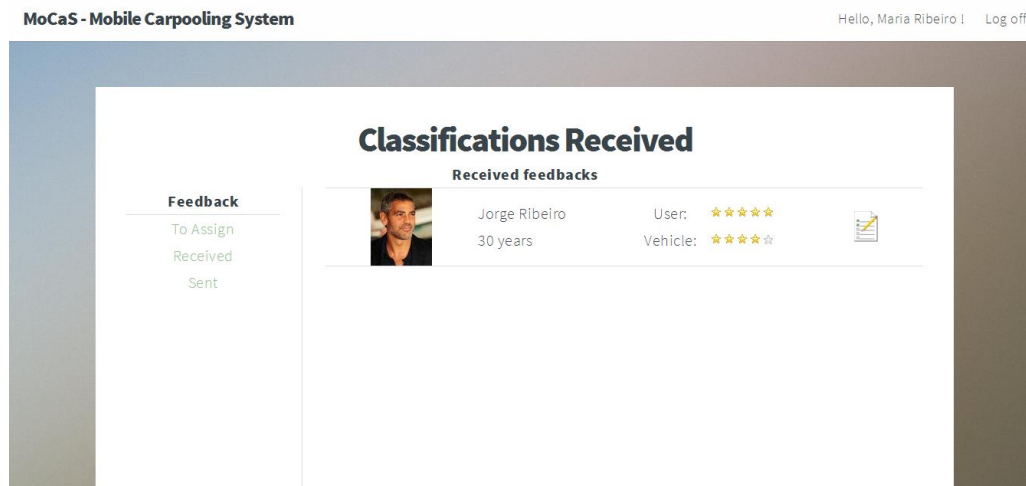


Figura 70 - Classificações recebidas no website MoCaS.

É possível também ver todo o detalhe da avaliação feita, a Figura 71 mostra isso mesmo, revelando para o utilizador três níveis de classificação avaliando a sua pontualidade, flexibilidade e condução, e outros três níveis para o veículo avaliando o seu conforto, consumo e bagageira, havendo ainda espaço para um comentário adicional à avaliação, para um maior detalhe na sua experiência nos serviços partilhados.

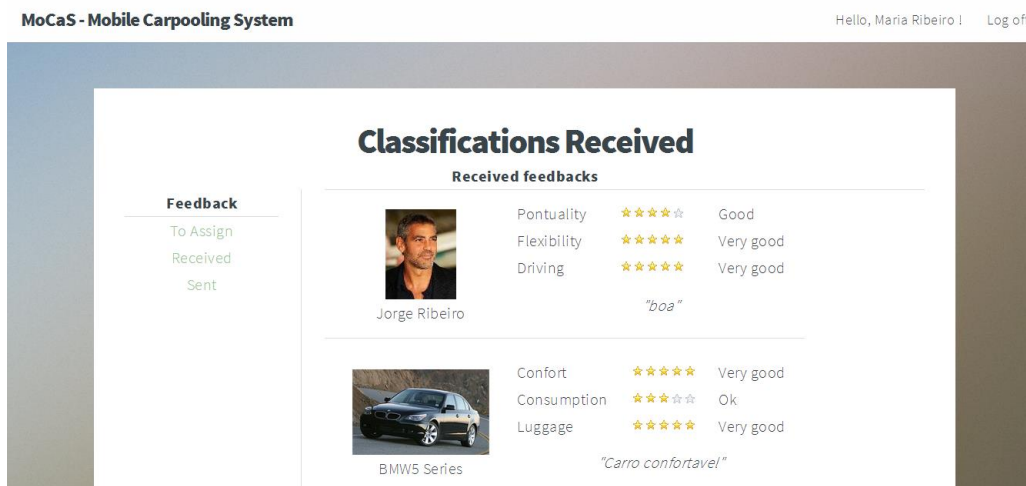


Figura 71 - Detalhe de classificação recebida do website MoCaS.

6.1.7. Gestão de Conta

A gestão de conta permite ao utilizador associar ou remover contas Facebook e/ou Google associadas a conta MoCaS, permite alterar a sua *password* e por fim permite também remover a sua conta indicando as razões desta opção para uma recolha de dados sobre o que leva as pessoas a deixar o serviço e assim tentar melhorá-lo (ver Figura 72).

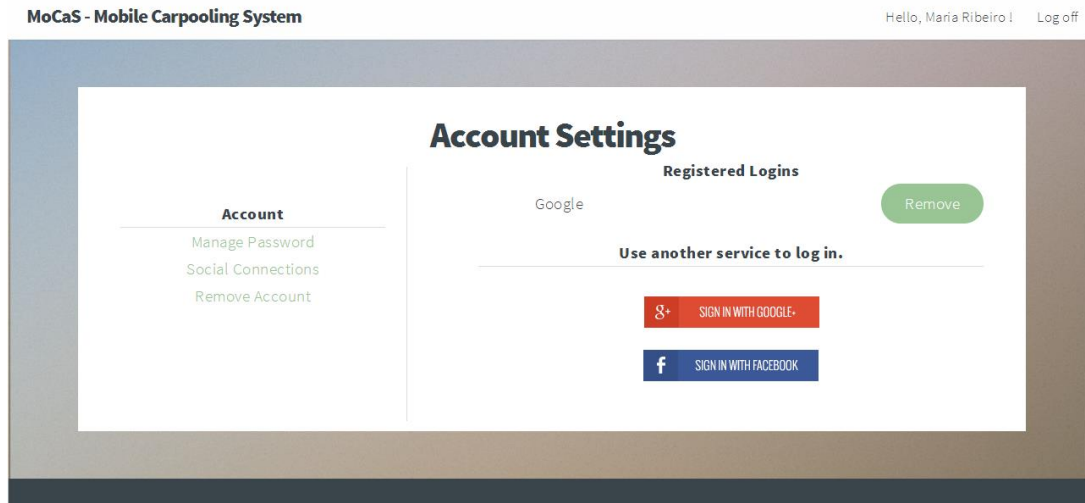


Figura 72 - Gestão de conta do utilizador no *website* MoCaS.

6.1.8. Gestão de Mensagens

O módulo de gestão de mensagens apresenta uma listagem de mensagens privadas trocadas por dois utilizadores. Como se pode verificar na Figura 73 é possível visualizar mensagens novas e antigas, assim como apagar essas mensagens.

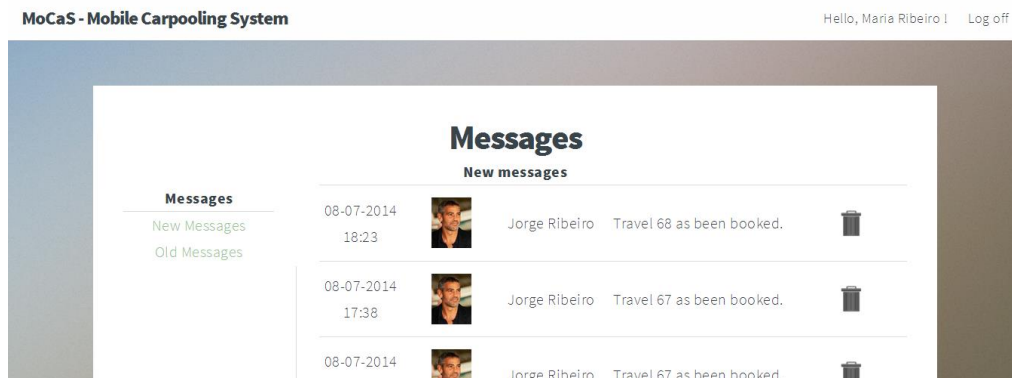


Figura 73 - Listagem de novas mensagens do utilizador no *website* MoCaS.

A Figura 74 apresenta um exemplo de uma mensagem de pedido de adesão a uma viagem, em que o utilizador tem de aceitar ou não o pedido.

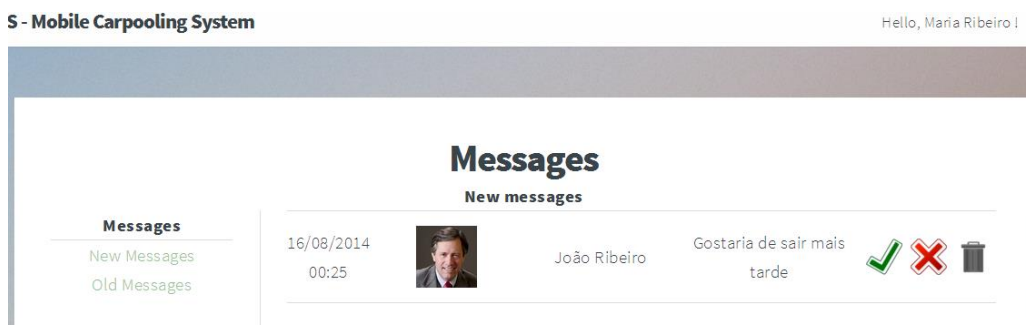


Figura 74 - Mensagem de pedido de associação a viagem no *website* MoCaS.

Caso aceite, os dois utilizadores em questão recebem uma notificação com a informação do pedido aceite e os respectivos contactos de telefone e email para troca de informação futura como se pode ver na Figura 75.

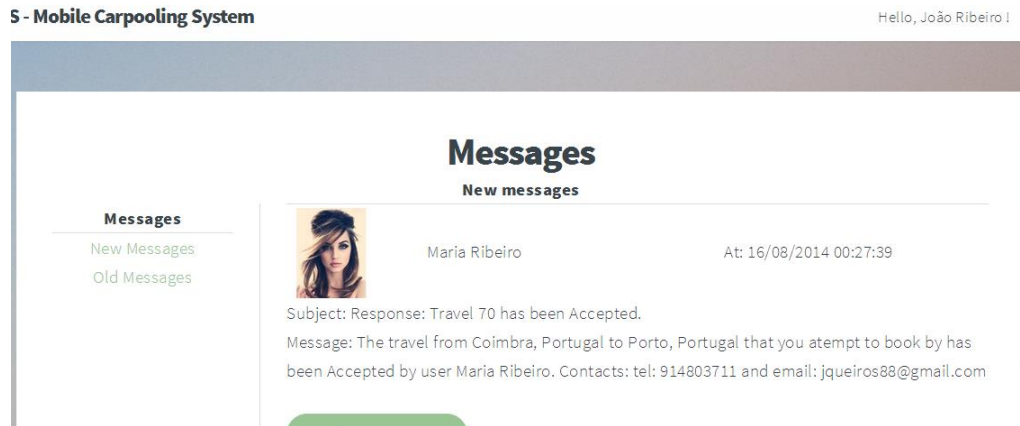


Figura 75 - Detalhe de uma mensagem recebida no *website* MoCaS.

6.1.9. Validadores

Os validadores são controlos fundamentais para o bom uso do *website*, são eles que permitem ou não a inserção de determinados tipos de dados ou preenchimentos de campos nos formulários. A Figura 76 apresenta um exemplo de validação de campos obrigatórios, neste caso é necessário inserir um local de partida e um de chegada para efectuar uma pesquisa no sistema MoCaS. Além deste tipo de validadores existem outros que validam expressões regulares para a inserção do endereço de *email*, validadores de tamanho numérico para números de telefone e validadores de preenchimento de regularidade no modulo de inserção de viagens.

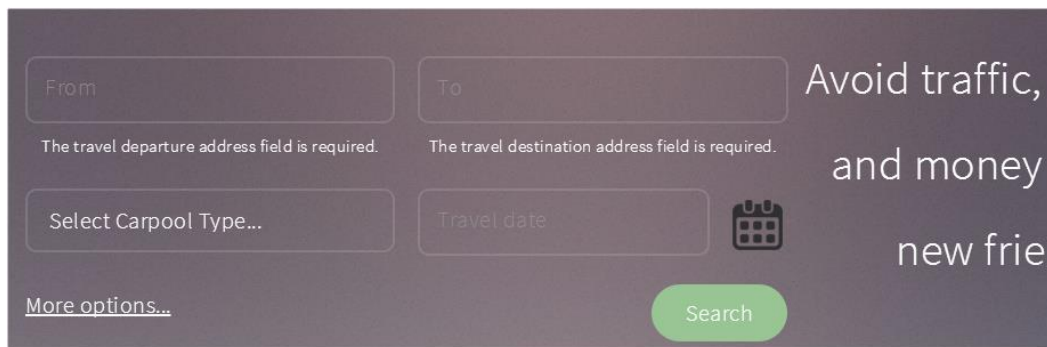


Figura 76 - Validadores no formulário de pesquisa no *website* MoCaS.

6.1.10. Custo Automático

No módulo de viagens existe a componente de custo a cobrar a cada passageiro que deseje associar-se a uma determinada viagem. Como o conceito de *carpooling* não pretende obter lucros de uma viagem, mas sim partilhar apenas custos, o *website* MoCaS calcula de forma automática o preço médio para cada viagem baseado em dados de combustível e portagens portuguesas. Para o cálculo do custo por pessoa foi criada a seguinte fórmula:

$$\text{custoPorPessoa} = \frac{\left(\frac{(\text{consumo} \times \text{distanciaTotal})}{\text{precoMedioCombustivel}} + (\text{distanciaPortagens} \times \text{custoPortagem}) \right)}{\text{numeroLugares}}$$

Os dados sobre o preço de combustível, consumo médio do veículo e custo de portagens estão armazenados numa tabela de configuração deste cálculo na base de dados e poderá ser alterada sempre que for necessário com vista a otimizar o valor calculado. Os restantes valores (distância total, distância em portagens e número de lugares) são obtidos através do *website*.

É ainda de salientar que no custo automático é definido um valor máximo a aplicar a cada passageiro nunca permitindo ultrapassar o custo total da viagem. Quanto ao valor mínimo, esse pode ser o que o utilizador desejar pois pode nem sequer querer cobrar qualquer custo para usufruir apenas de companhia. Para ajustar o preço o controlo de custo indica através de 3 cores o tipo de preço praticado, verde para custo baixo, amarelo para custo moderado e vermelho para custo elevado (ver Figura 77).

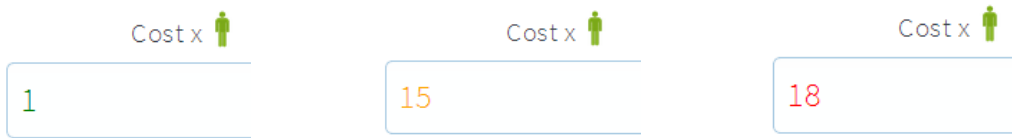


Figura 77 - Os três indicadores de preço verde, amarelo e vermelho que significam preço baixo, moderado e alto respectivamente. Retirado do *website* MoCaS.

6.1.11. Locais Automáticos

Em todo o sistema MoCaS é fundamental para a definição de um trajecto definir os locais de partida e destino, bem como de possíveis pontos de paragem entre eles. Para apoiar os utilizadores na inserção desses dados tanto nas pesquisas como nas criações de novas viagens foi implementado um controlo de apoio do Google Maps, o Place Autocomplete. A Figura 78 mostra o preenchimento de uma localização com o suporte do Google Place Autocomplete que fornece localizações semelhantes ao que se está a inserir.

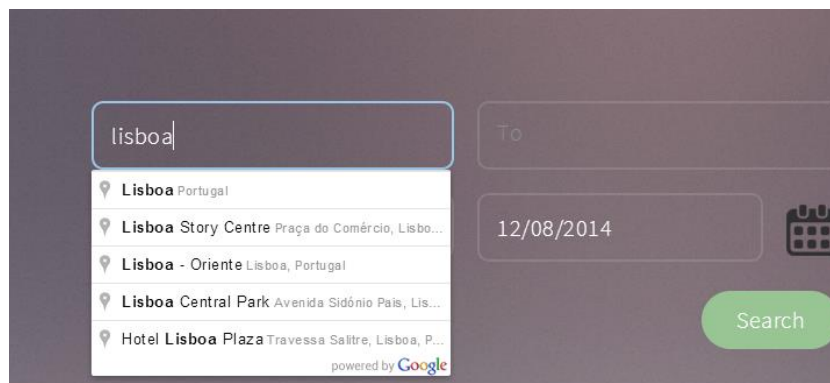


Figura 78 - Inserção da localização "Lisboa" com suporte do Google Place Autocomplete no *website* MoCaS.

6.1.12. Suporte

Segundo um estudo da NetMarketShare (Market Share Statistics for Internet Technologies) o *browser* mais utilizado desde Setembro de 2013 até Junho de 2014 foi o Microsoft Internet Explorer com 58,8% do mercado seguido pelo Google Chrome com 19,3% como se pode ver na Figura 79 [70]. Em resultado deste estudo o *website* MoCaS tem uma construção otimizada para ser executado nestes mesmos *browsers*.

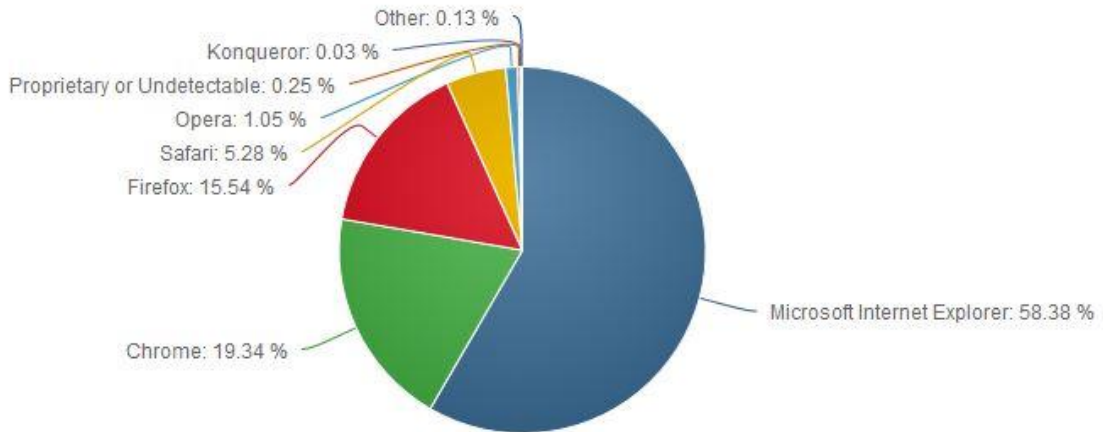


Figura 79 - Percentagem de utilização de *web browsers* a nível mundial. Fonte: [70].

Outro dado importante no acesso a páginas web é a resolução de ecrã suportada, pois é através da resolução que podemos tirar um maior partido da interface da aplicação em vários dispositivos. O *website* MoCaS é compatível com resoluções desde 1280x800. Para verificar esta condição utilizámos a ferramenta *online* Screenfly [71] que permite simular várias resoluções de ecrã de vários dispositivos. A Figura 80 representa um dos testes feitos ao *website* MoCaS para simulação da resolução 1280x800.

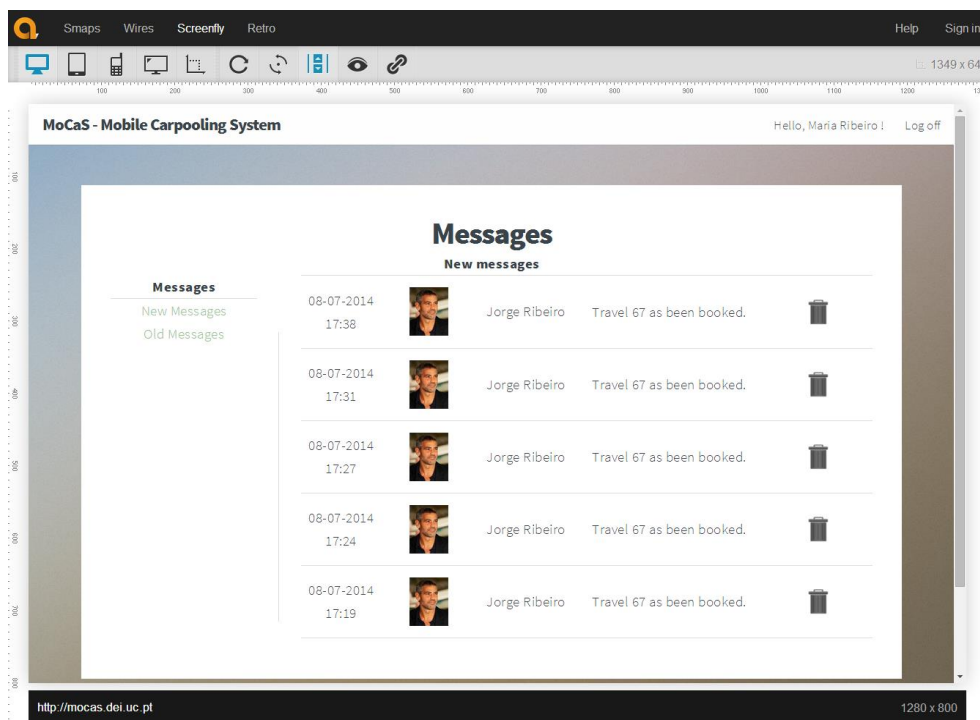


Figura 80 – Teste na ferramenta Screenfly [71] à resolução 1280x800 do *website* MoCaS.

6.1.13. Testes de Performance ao *Website* MoCaS

Os testes realizados ao *website* tiveram como plataformas de teste:

- Plataforma A – Portátil Toshiba Satellite L50-A-18T, com um processador Intel Core i7 de 2.4 GHZ e 8GB de RAM [78];
- Ferramenta – WebPageTest [76]

Objectivo:

- Perceber a performance que o *website* consegue obter.

Descrição do teste:

Foram realizados testes de performance ao *website* através da ferramenta online WebPageTest que comparou o primeiro acesso ao *website* com o segundo e comparou algumas métricas de tempos de acesso e de descarregamento de conteúdo. A Tabela 7 representa a apresentação resumida de resultados do teste efectuado.

	Tempos de Carregamento	1º Byte	Início da Renderização	Documento Completo			Completamente Carregada		
				Tempo	Pedidos	Bytes de Entrada	Tempo	Pedidos	Bytes de Entrada
1º Acesso	5,199s	0,332s	4,392s	5,199s	198	1.297 KB	5,354s	198	1.328 KB
2º Acesso	2,296s	0,238s	1,790s	2,296s	2	41 KB	2,296s	2	41 KB

Tabela 7 - Resultados resumidos do teste de performance ao *website* MoCaS.

Para uma interpretação mais aprofundada dos resultados obtidos, apresentamos nas figuras (Figura 81, Figura 82) abaixo as percentagens e valores sobre os conteúdos carregados no primeiro e segundo acesso respectivamente.

1º Acesso



MIME Type	Requests
js	181
css	6
image	5
font	3
html	2
flash	0
other	0

MIME Type	Bytes
js	970412
font	109574
image	105667
html	36995
css	17086
flash	0
other	0

Figura 81 - Resultados sobre os conteúdos carregados no 1º acesso ao *website* MoCaS.

2º Acesso

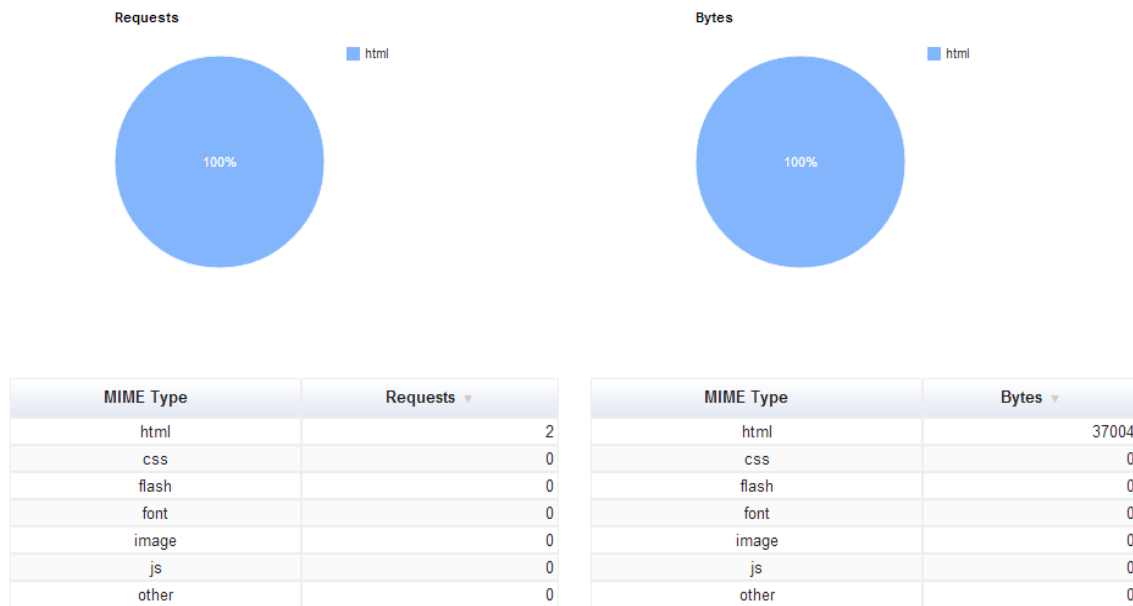


Figura 82 - Resultados sobre os conteúdos carregados no 2º acesso ao *website* MoCaS.

Conclusão

Através dos resultados obtidos podemos concluir que existe uma grande diferença entre a primeira vez em que a página é acedida para a segunda em menos de metade do tempo de carregamento. A justificar estes valores estão os conteúdos carregados pela primeira vez representados na Figura 81 que nesta execução são carregados os conteúdos Javascript (js), CSS, imagens, fontes e HTML, ao contrário do segundo acesso representado na Figura 82 indicando que nesta fase só é necessário carregar o HTML da página.

Os tempos e acesso ao *website* são muito bons considerando que em média um *website* nos dias de hoje demora em média 7,25 segundos a carregar na totalidade [77]. Este valor é superior ao valor do primeiro acesso em 2,051 segundos e ao segundo acesso em 4,954 segundos, isto significa que o acesso ao *website* apresenta óptimos indicadores de performance acima da média.

6.2. MoCaS Web Service

O *web service* do sistema MoCaS é uma parte fundamental e central no sistema, pois é aqui que é feita toda a transacção da informação entre as várias plataformas que compõem este sistema. A implementação deste *web service* foi feita na plataforma .NET, usando a linguagem C# na criação de 58 serviços que apoiam as mais diversas áreas do sistema. Desses serviços destacam-se os serviços de actualização das posições em tempo real, os serviços de actualização de dados de viagem, perfil de utilizador, envio de mensagens privadas, inserção de veículos, obtenção de informação de viagens e veículos, entre outros. Para consultar em detalhe cada um dos serviços disponíveis no sistema MoCaS deverá ser consultado o Anexo D – Listagem dos Serviços Disponíveis no *Web Service*.

Como foi abordado na secção 5.4. Comunicação Aplicação-Servidor o *web service* recorre ao protocolo SOAP para troca de dados entre as várias plataformas. A Figura 83 representa um exemplo de um pedido SOAP (à esquerda) efectuado pelo *web service* que recebe

o identificador único de um utilizador e à direita a estrutura da resposta que é devolvida pelo serviço através de um *schema* representando a lista das viagens.

<pre> POST /MoCaSWebService.asmx HTTP/1.1 Host: localhost Content-Type: text/xml; charset=utf-8 Content-Length: length SOAPAction: "http://tempuri.org/GetUserTravels" <?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <GetUserTravels xmlns="http://tempuri.org/"> <id>string</id> </GetUserTravels> </soap:Body> </soap:Envelope> </pre>	<pre> HTTP/1.1 200 OK Content-Type: text/xml; charset=utf-8 Content-Length: length <?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <GetUserTravelResponse xmlns="http://tempuri.org/"> <GetUserTravelResult> <xsd:schema>schema</xsd:schema>xml</GetUserTravelResult> </GetUserTravelResponse> </soap:Body> </soap:Envelope> </pre>
--	--

Figura 83 – Pedido (esquerda) e resposta (direita) SOAP do serviço de pesquisa de viagens do utilizador no sistema MoCaS.

Sobre o *web service* deve ainda ser referido que este permite fornecer uma API aberta à comunidade, que irá permitir a integração e utilização do sistema MoCaS por terceiros. Esta particularidade para além de ser também algo característico de aplicações da *web 2.0*, também contribui para um aumento da base de potenciais utilizadores do sistema MoCaS.

6.2.1. Testes de Performance ao Servidor

Os testes realizados ao servidor tiveram como plataformas de teste:

- Plataforma A – Portátil Toshiba Satellite L50-A-18T, com um processador Intel Core i7 de 2.4 GHZ e 8GB de RAM [78];
- Servidor – CPU: Intel Xeon E5-2650, 2,0 GHz com 2GB de RAM e o Windows Server 2008 R2 Standard como sistema operativo;
- Ferramenta – SoapUI Pro [79]

Objectivo:

- Perceber a performance que o servidor consegue atingir.

Descrição do teste:

Foram realizados testes de carga ao servidor usando a ferramenta SoapUI Pro, esta permite fazer testes aos serviços SOAP como os implementados. Estes testes consistiram em correr um conjunto de 10 serviços que devolviam dados distintos como acção. A Figura 84 representa um dos testes aplicados, neste teste os 10 serviços estavam a ser executados em simultâneo durante 5 iterações

Test Step	min	max	avg	last	cnt	tps	bytes	bps	err	rat
GetStops - Request 1	49	117	63,4	49	5	1,64	8775	2893	0	0
GetAllTravelVehicleDetail - Request 1	50	56	52,8	54	5	1,64	8375	2761	0	0
GetUserTravel - Request 1	54	56	55	54	5	1,64	18925	6239	0	0
GetLiveTravels - Request 1	54	72	57,8	72	5	1,64	17495	5768	0	0
GetUpcommingTravels - Request 1	78	88	80,8	79	5	1,64	58825	19394	0	0
GetCostVariables - Request 1	49	58	51,6	49	5	1,64	6915	2279	0	0
GetUserInfo - Request 1	51	54	53,2	54	5	1,64	14130	4658	0	0
GetRegularityWithStatus - Request 1	49	53	51,4	53	5	1,64	16905	5573	0	0
GetFlexibilityWithStatus - Request 1	47	55	51,8	54	5	1,64	10645	3509	0	0
GetUserOrderDetail - Request 1	50	63	53,4	63	5	1,64	9265	3054	0	0
TestCase:	531	672	571,2	581	5	1,64	170255	56134	0	0

Figura 84 - Testes de performance ao servidor.

Através do teste acima descrito foi possível obter o tempo médio dos serviços para completar o teste com o valor de 571,2 milissegundos. Com este valor resolvemos definir como um erro todas as execuções que demorassem o dobro, ou seja 1142 milissegundos. Foram então executados alguns testes com um número de clientes variável, tentando assim perceber qual o número máximo de clientes que o servidor aguenta, antes de começar a produzir erros. Utilizando o método de teste acima mencionado foram obtidos os seguintes resultados:

Nº de Clientes	Nº de Pedidos	Nº de Erros
5	100	0
10	100	0
20	100	0
30	100	0
40	100	1
50	100	2
100	100	19

Tabela 8 - Testes de performance ao servidor com número de clientes variável.

Conclusão

Através dos resultados obtidos na Tabela 8 podemos verificar que o número de clientes com o qual o servidor actual consegue garantir uma resposta em cerca de 1 segundo é cerca de 30. Contudo, este valor deverá servir ser apenas como estimativa, pois existem aqui outros factores que não estão abrangidos, como a rede e a carga a que esta se sujeita no momento, assim como o uso do servidor para outras funções por parte de outras pessoas da instituição.

6.3. MoCaS Mobile

A aplicação móvel do sistema MoCaS servirá essencialmente para a marcação de viagens. Com esta aplicação pretende-se o uso útil e rápido dos serviços *carpool* disponíveis no momento. Nesta secção serão apresentadas as funcionalidades desta aplicação com especial foco no modo em tempo real.

6.3.1. Inico e Login

A aplicação móvel MoCaS possui um ecrã de apresentação ao abrir como se pode ver pela Figura 85. Apesar de parecer meramente decorativa a aplicação durante este curto período de três segundos avalia se o dispositivo móvel tem ligação a internet, caso contrário a aplicação fecha automaticamente alertando o utilizador para a situação.



Figura 85 - Ecrã de apresentação da aplicação móvel MoCaS.

Depois do ecrã de apresentação o utilizador deve inserir as suas credenciais para se autenticar na aplicação. A Figura 86 apresenta o ecrã de *login* usado para a autenticação, este ecrã depois de o utilizador se autenticar a primeira vez com sucesso passará a deixar de aparecer, pois o utilizador fica sempre com sessão iniciada cada vez que entre na aplicação. Esta funcionalidade tornará mais eficiente o uso da aplicação, pois para cada vez que o utilizador utilize a aplicação não terá de colocar sempre as suas credências.

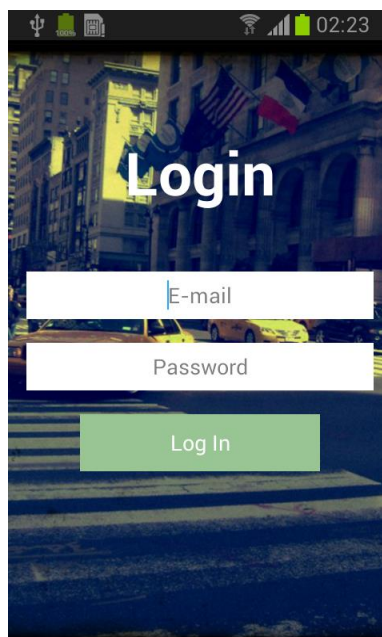


Figura 86 - Ecrã de *login* da aplicação móvel MoCaS.

6.3.2. Gestão de Viagens

A área de gestão de viagens apresenta uma listagem de todas as viagens a que o utilizador com sessão iniciada na aplicação está associado. Como se pode ver pela Figura 87 este ecrã apresenta uma listagem das viagens distinguindo o tipo de cada uma por oferta ou pedido de boleia através das imagens de carro e pessoa respectivamente. Além disto é mostrada informação do trajecto bem com a data e hora da partida. Os símbolos da aplicação encontram-se para consulta no Anexo C – Simbologia do Sistema MoCaS.

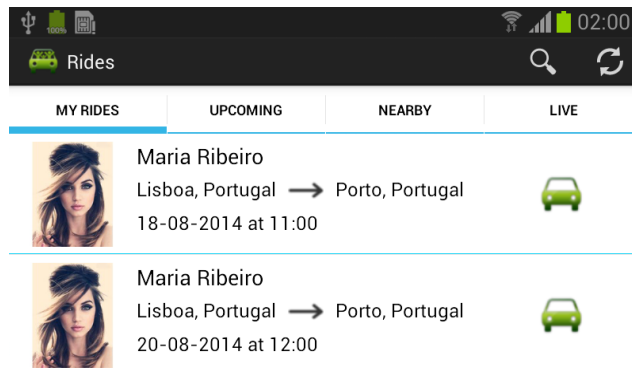


Figura 87 - Ecrã de listagem de viagens do utilizador no sistema MoCaS.

A funcionalidade em tempo real para iniciar a viagem também é feita neste ecrã através da selecção de uma viagem. Após a selecção da viagem que se quer iniciar é mostrado ao utilizador uma segunda janela com a indicação do trajecto e locais intermédios de paragem. A Figura 88 mostra essa janela com o mapa do trajecto com a distância e duração aproximada da viagem.

O utilizador ao iniciar a viagem, caso pretenda utilizar a componente em tempo real deverá seleccionar a opção “Start” e a partir desse momento a viagem está pronta a receber pedidos de associação em tempo real (caso haja lugares disponíveis).

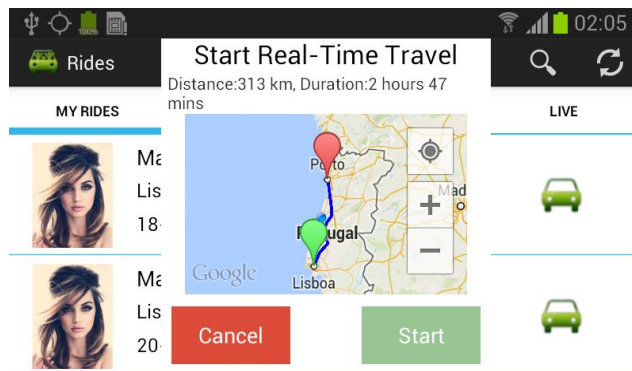


Figura 88 - Ecrã de início de viagem em tempo real na aplicação móvel MoCaS.

6.3.3. Viagens em Tempo-Real

Este módulo de viagens em tempo-real representa todas as viagens no sistema MoCaS a decorrer em tempo-real. A Figura 89 representa o mapa com as viagens a decorrer em tempo-real indicando através de dois símbolos distintos as paragens (assinalados por uma bandeira) e os veículos (assinalados com um veículo com passageiros). Os símbolos da aplicação encontram-se para consulta no Anexo C – Simbologia do Sistema MoCaS.

É a partir deste mapa que os utilizadores se poderão associar às viagens em tempo real através de um ponto de paragem pré-estabelecido ou através da sua própria localização, deixando a decisão de aceitação dessas condições para o condutor.

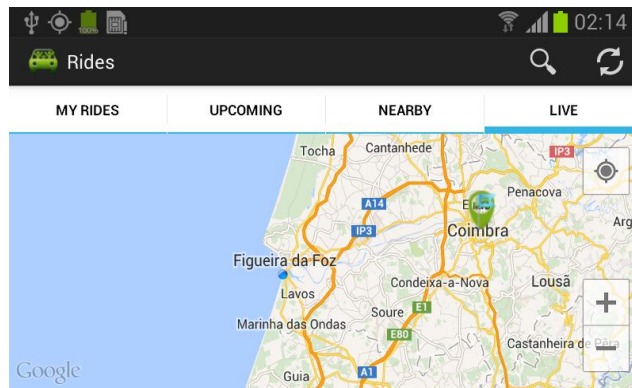


Figura 89 - Mapa com as viagens a decorrer em tempo real na aplicação móvel MoCaS.

6.3.4. Viagens por Perto (Espacial)

A componente de viagens por perto representa a listagem de todas as viagens no sistema MoCaS até a uma distância máxima de 50 quilómetros, ordenadas pela distância mais curta desde o utilizador até ao ponto de partida das viagens. A Figura 90 apresenta essa listagem com a respectiva informação dos pontos de partida e chegada, que utilizador partilha o serviço, data e hora da partida, distância a que o utilizador está do ponto de partida da viagem listada e o tipo de partilha (oferta ou pedido de boleia).

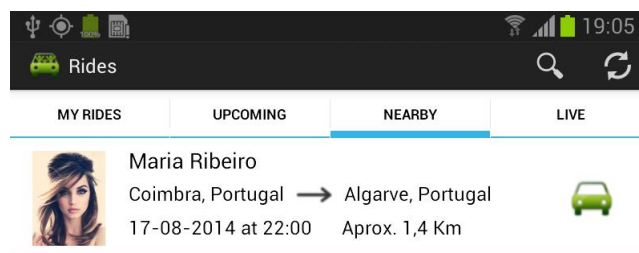


Figura 90 - Listagem de viagens por perto na aplicação móvel MoCaS.

6.3.5. Próximas Viagens (Temporal)

Esta área de próximas viagens a decorrer representada na Figura 91 mostra uma listagem de todas as viagens que estão para decorrer no dia em que é feita a consulta. Nesta listagem é apresentada a hora da partida, os pontos de origem e destino, o utilizador que partilha o serviço e o tipo de partilha (oferta ou pedido de boleia).

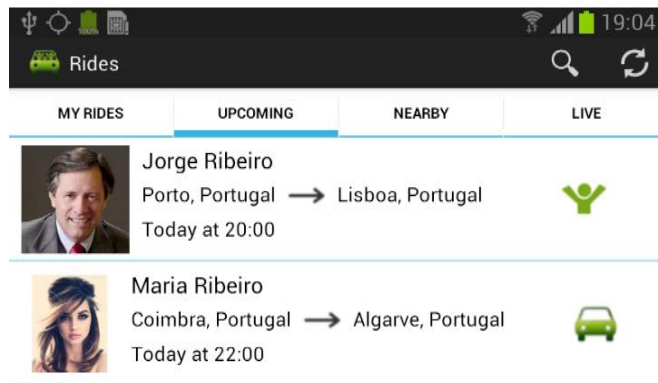


Figura 91 - Listagem das próximas viagens na aplicação móvel MoCaS.

6.3.6. Pesquisa de Viagens e Resultados

Para uma pesquisa parametrizada de viagens o utilizador terá de aceder ao ecrã de pesquisa através da opção presente na barra de acção (ver Figura 108 do Anexo C – Simbologia do Sistema MoCaS). A Figura 92 apresenta a parametrização da pesquisa de uma viagem dando a possibilidade ao utilizador de inserir o ponto de partida e chegada, data e hora de partida.

Como se pode verificar na Figura 92 está representado o ecrã de pesquisa da aplicação móvel, no ponto de partida é possível obter a posição actual do utilizador que pretende fazer a pesquisa para uma pesquisa mais rápida não tendo de preencher o ponto de partida caso esse seja o local onde esse utilizador se encontra.

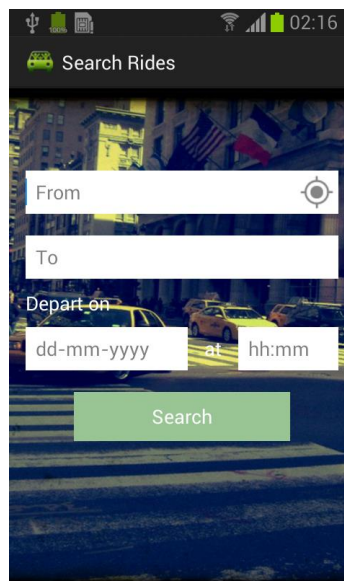


Figura 92 - Ecrã de parametrização de pesquisa na aplicação móvel MoCaS.

Depois de fazer a pesquisa e caso existam viagens com a parametrização correspondente à indicada pelo utilizador será apresentada uma listagem das viagens encontradas (ver Figura 93).



Figura 93 - Listagem de resultados de pesquisa na aplicação móvel MoCaS.

6.3.7. Marcação de Viagem

Para marcação de uma viagem é necessário que o utilizador seleccione uma viagem através dos diversos meios referidos nos pontos anteriores (pesquisa, viagens por perto, próximas viagens e mapa de viagens em tempo real). A Figura 94 mostra o detalhe da viagem seleccionada, aqui é apresentada informação sobre o trajecto, data e hora, flexibilidade temporal e espacial, veículo, tamanho da bagagem, classificação do utilizador e veículo e comentários adicionais.

A marcação da viagem poderá ser efectuada através de uma chamada (caso o número esteja disponível), ou mensagem privada (sujeita a aprovação), ou através de pré-reserva de lugar (caso o utilizador que cria a viagem o permita na criação da viagem).

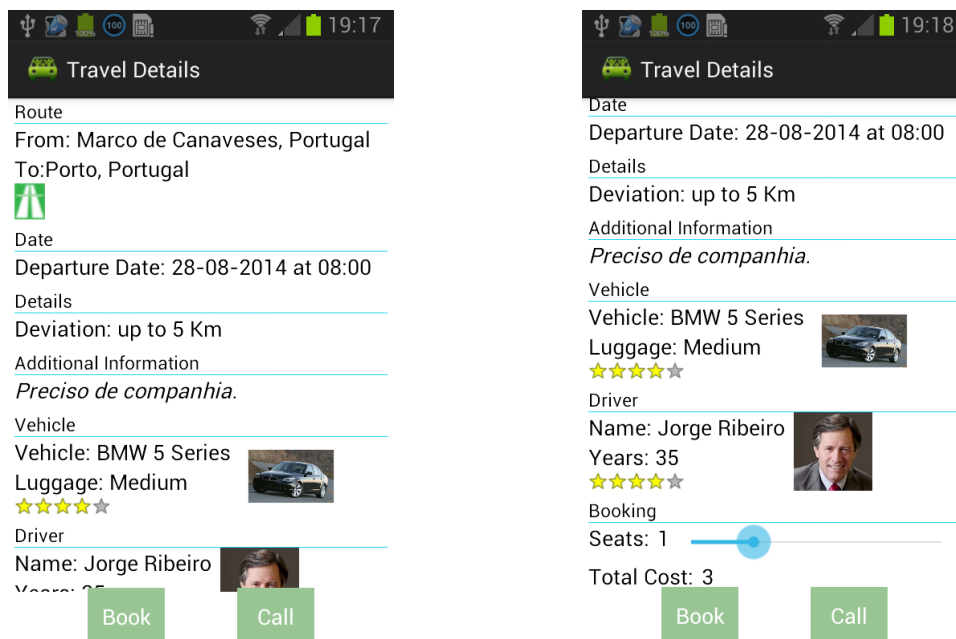


Figura 94 - Ecrã de detalhe de viagem da aplicação móvel MoCaS.

6.3.8. Resposta em Tempo-Real

O módulo de resposta em tempo real é de grande importância pois é através dele que serão aceites os pedidos de associação em tempo real sem manusear o dispositivo móvel, será apenas transmitida a resposta ao aparelho por comando de voz. A Figura 95 representa um pedido de associação à viagem que está a decorrer no momento, este pedido apresenta a

localização do utilizador que se pretende associar deixando o condutor aceitar ou não o seu pedido através duma resposta por comando de voz (“Sim” ou “Não”).

Após essa resposta é enviada uma mensagem privada de confirmação ao passageiro com os dados do condutor, é de referir que o passageiro pode consultar a posição do condutor em tempo-real através da função *Live* referida na subsecção 6.3.3. Viagens em Tempo-Real

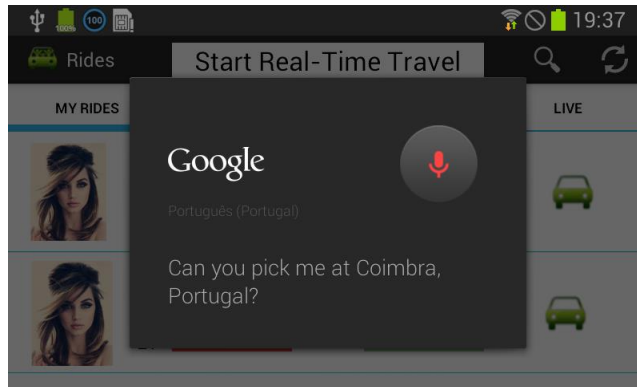


Figura 95 - Ecrã de notificação e resposta a um pedido de associação de viagem na aplicação móvel MoCaS.

6.3.9. Opções da Barra de Acção

A barra de acções da aplicação móvel está disponível no menu inicial de gestão de viagens. A Figura 96 mostra a barra de acção da aplicação móvel, aqui é possível ir para o ecrã de pesquisa de viagens, actualizar todas as viagens listadas no sistema de forma manual e também efectuar o *logout* da aplicação para que possa ser acedida com outras credenciais.

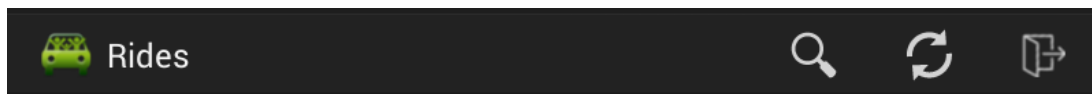


Figura 96 - Barra de acção da aplicação móvel MoCaS.

6.3.10. Suporte

A componente móvel da aplicação MoCaS será suportada desde a versão Android 2.3.3 até à actual 4.4. Esta escolha deve-se ao facto do mercado actual ter uma grande percentagem de equipamentos Android com a versão 2.3.3-2.3.7 (16,2%), sendo a maior “fatia” para as versões 4.1.x com cerca de 33,5% de mercado [69]. A Figura 97 apresenta os resultados de um estudo recente sobre as versões Android presentes no mercado actual [69].

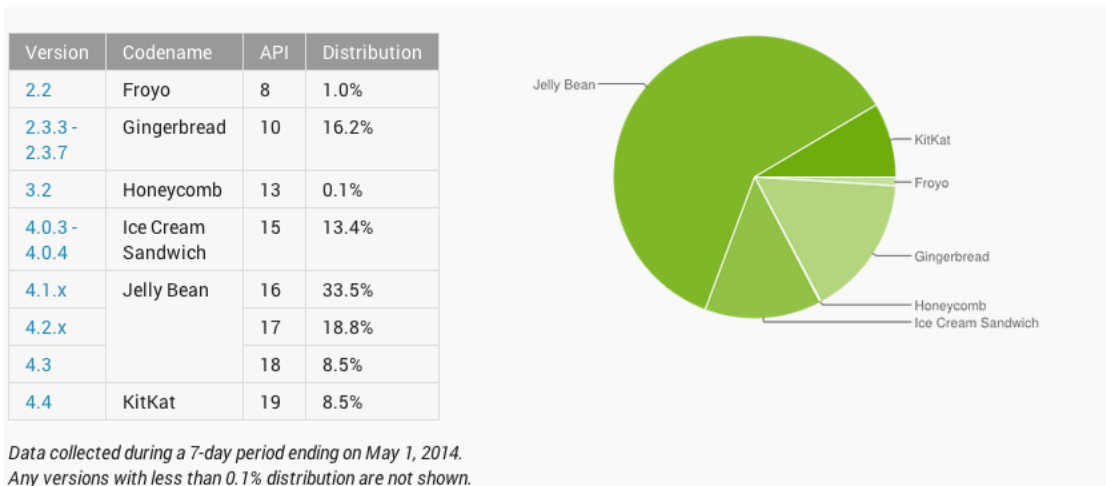


Figura 97 - Percentagem de mercado de várias versões do Android. Fonte: [69].

6.3.11. Testes de Performance à Aplicação Móvel MoCaS

Os testes realizados ao servidor tiveram como plataformas de teste:

- Plataforma A – *Smartphone* Samsung Galaxy Ace 2 I8160, com um processador *dual-core* de 800 MHz e 768 MB RAM [80];
- Plataforma B – Portátil Toshiba Satellite L50-A-18T, com um processador Intel Core i7 de 2.4 GHz e 8GB de RAM [78];

Objectivo:

- Perceber se existem ganhos de performance de aplicação móvel em relação ao *website*.

Descrição do teste:

Este teste consistiu em realizar a operação de *Login* em ambos os sistemas, móvel e *web*, de modo a perceber o tempo que demora em cada um dos sistemas. Foram efectuadas 5 iterações à operação de *Login* tendo sido registados os tempos de cada um, a média, valor máximo e mínimo do conjunto das iterações para uma melhor comparação.

Foram ainda realizados dois testes diferentes no dispositivo móvel, em que se alternou entre a ligação via Wi-Fi e ligação de dados da rede móvel.

A execução deste teste foi iniciada após a colocação dos dados de *login*, começando o tempo a contar aquando do “click” para início da autenticação e parando a contagem quando fosse alcançada a autenticação.

A Tabela 9 apresenta os resultados obtidos através dos testes efectuados às duas plataformas.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Min	Max	Média
Plataforma A	8,5s	2,7s	0,2,5s	2,4s	2,0s	2,0s	8,5s	3,6s
Plataforma B – Wi-Fi	1,5s	1,3s	1,6s	1,3s	1,3s	1,3s	1,6s	1,4s
Plataforma B – Dados Móveis	12,9s	3,9s	10,5s	4,0s	3,0s	3,0s	12,9s	6,9s

Tabela 9 - Listagem de resultados aos acessos de *Login* na plataforma *web* e móvel do sistema MoCaS.

Conclusão

Através dos resultados obtidos é possível verificar que existe um ganho de performance da versão móvel em relação à versão *web*. No entanto, os piores resultados também foram obtidos com a aplicação móvel ligada aos dados móveis, sendo que a sua velocidade na altura dos testes era E (EDGE) que pode atingir até 236 Kbps [81], o que é bastante pouco tendo em conta as velocidades da internet hoje em dia.

Portanto, a plataforma de teste A com ligação Wi-Fi é a que obtém melhor performance tendo conseguido em média um ganho de cerca de 2,2 segundos em relação à versão *web* e cerca de 5,5 segundos com ligação de dados móveis.

A melhor performance na aplicação móvel deve-se ao facto desta ser uma aplicação nativa, pois numa aplicação deste tipo não existe nenhuma camada extra entre o sistema e a

aplicação, isto permite a que toda a performance que o dispositivo tem para dar seja mais bem aproveitada [82].

Capítulo 7

Conclusão e Trabalho Futuro

7.1. Conclusão

O *carpooling* é uma prática que está em crescimento, através dela são vários os benefícios dados a quem utiliza esta forma de viajar. O objectivo deste projecto foi criar um sistema de *carpooling* inovador que possibilitasse uma maior flexibilidade, comodidade, segurança e mobilidade aos utilizadores que utilizam esta prática.

Fazendo uma análise retrospectiva ao trabalho desenvolvido, começamos por estudar o conceito de *carpooling*, a sua história, as suas vantagens e alguns dos seus problemas, tipos de funcionamento e métodos de optimização. Depois de interiorizado o conceito, partimos para uma análise profunda a 12 sistemas de *carpooling* (nacionais e internacionais) comparando estes para assim termos uma melhor percepção da concorrência de mercado e também de pontos positivos e negativos a reter para a construção do sistema MoCaS. Posteriormente foram definidos os requisitos do sistema MoCaS e sua arquitectura, foram também analisadas várias opções tecnológicas para a melhor implementação deste novo sistema de *carpooling*. No final desta primeira fase, que durou todo o primeiro semestre, foram definidas metas para a implementação dos requisitos estipulados.

No decorrer da fase de implementação fomos nos deparando com alguns problemas na construção da implementação da aplicação móvel devido a alguma falta de conhecimento nesta área, o que levou a um aumento no tempo de implementação da componente móvel. Este contratempo levou também a um replaneamento da implementação do projecto e à não implementação de alguns requisitos de prioridade baixa (filtros avançados, informação meteorológica e grupos). No entanto, foram cumpridos grande parte dos requisitos, todos os de prioridade alta e média, bem como a maior parte dos requisitos de baixa prioridade.

No final do projecto podemos verificar que este sistema é inovador pois não há nenhum semelhante no mercado, possui um *website* simples e fácil de utilizar, tem uma grande abrangência na preparação de uma viagem, pois tem várias parametrizações que podem fazer a diferença na correspondência de condutores e passageiros, como as preferências de permissão de fumar, se o utilizador conversa durante a viagem, se há alguma flexibilidade temporal e espacial, entre outras. A componente de marcação em tempo-real revela um grande “passo” para o constante preenchimento de lugares durante alguns pontos de paragem onde saem alguns passageiros e também dá uma grande utilidade à aplicação móvel pois um utilizador pode resolver contratemplos de deslocação com a utilização deste serviço inovador. A resposta a marcações em tempo-real por comando de voz também se revela inovadora e extremamente importante para a segurança do condutor e ocupantes, visto que não é necessário manusear o equipamento para confirmar os pedidos de adesão.

Este sistema poderá contribuir não só para um aumento da prática do *carpooling* mas também para uma optimização no seu uso, pois este sistema é bastante completo na medida em que é suportado por um *website* e uma aplicação móvel permitindo assim ser utilizado em todo o lado desde que se possua um dispositivo com ligação à internet, e também pelo uso da marcação em tempo-real que irá permitir uma maior oferta de serviços visto que os pontos de paragem do condutor para ir buscar um passageiro podem ser dinâmicos.

7.2. Trabalho Futuro

Para trabalho futuro fica a implementação dos requisitos de baixa prioridade que não foram implementados, como alguns filtros avançados, como os de custo em que se poderão filtrar viagens por valor. Outra funcionalidade que poderá ser aplicada é a criação de grupos que incluem pessoas que diariamente têm viagens em comum (exemplos: ida para o trabalho, escola, os regressos a casa ao fim-de-semana, etc) e a criação de grupos com eventos de grande dimensão (exemplos: concertos, festivais de Verão, etc). Outro requisito para implementar futuramente seria o de informação meteorológica no planeamento das viagens, para tornar mais vasta a informação e posteriormente para tomar precauções para cada estado climatérico, por exemplo caso chova intensamente os passageiros e condutor poderão querer combinar nova hora ou data para a fazer a viagem.

Outras evoluções no sistema poderiam incluir um motor autónomo de correspondência de viagens, que permitisse ao utilizador, ao inserir o seu pedido ou oferta de viagem, encontrar viagens correspondentes de forma automática evitando as pesquisas manuais.

A implementação de um sistema de pagamento automático na aplicação para pré-pagamento da viagem traria vantagens, pois um utilizador que fosse lesado num serviço (no caso de um passageiro ou condutor não aparecer) seria sempre reembolsado dando assim garantia no serviço. Neste serviço também poderia ser implementado um sistema de financiamento do MoCaS através de uma percentagem de comissão no valor cobrado em cada viagem.

Também deveria ser feito um melhoramento à interface móvel, particularmente no módulo de marcação de viagens em que se poderia personalizar o ecrã de pedido de adesão a viagens com mais informação sobre que necessita do serviço e, no módulo de iniciação de uma viagem em que se poderia incluir no mapa indicações sobre o percurso que se está a percorrer.

Referências

1. “Deboleia” [Online]. Disponível: <http://www.deboleia.com>. [Acedido: 13-Out-2013].
2. “Pendura” [Online]. Disponível: <http://www.pendura.pt>. [Acedido: 13-Out-2013].
3. “RoadSharing” [Online]. Disponível: <http://www.roadsharing.com>. [Acedido: 13-Out-2013].
4. “BlaBlaCar” [Online]. Disponível: <http://www.blablacar.com>. [Acedido: 13-Out-2013].
5. “Toogether” [Online]. Disponível: <http://www.toogether.com>. [Acedido: 13-Out-2013].
6. “Carpooling.co.uk” [Online]. Disponível: <http://www.carpooling.co.uk>. [Acedido: 13-Out-2013].
7. “Princeton University Carpooling Service” [Online]. Disponível: <http://www.princeton.edu/parking/rideshare.html>. [Acedido: 11-Nov-2013].
8. “Jayride.com.au” [Online]. Disponível: <http://www.jayride.com.au>. [Acedido: 11-Nov-2013].
9. “Boleia.net” [Online]. Disponível: <http://www.boleia.net>. [Acedido: 11-Nov-2013].
10. “CarpoolWorld.com” [Online]. Disponível: <https://www.carpoolworld.com>. [Acedido: 11-Nov-2013].
11. “Zimride.com” [Online]. Disponível: <http://www.zimride.com>. [Acedido: 11-Nov-2013].
12. “Carma” [Online]. Disponível: <https://car.ma>. [Acedido: 11-Nov-2013].
13. “1973 oil crisis” [Online]. Disponível: http://en.wikipedia.org/wiki/1973_oil_crisis. [Acedido: 11-Nov-2013].
14. Marc Oliphant & Andrew Amey [2010]. “Dynamic Ridesharing: Carpooling Meets the Informaion Age”.
15. Chan, Nelson D. & Susan A. Shaheen [2012]. “Ridesharing in North America: Past, Present, and Future. Transport Reviews”. Vol. 32, nº 1. p. 93–112.
16. Erik Ferguson [1997]. “The rise and fall of the American carpool: 1970–1990. Transportation” Vol. 4, nº 4. p. 349-376
17. Stephen DeLoach and Thomas Tiemann [2010]. “Not driving alone: Commuting in the Twenty-first century”. Elon University Department of Economics.
18. Handke, V., & Jonuschat, H. [2013]. “Flexible Ridesharing: New Opportunities and Service Concepts for Sustainable Mobility”. Berlin, Germany: Springer.
19. DeGruyter, C. [2006]. “Investigating a CBD-wide carpooling scheme for Melbourne”. Department of Infrastructure, Melbourne VIC, Australia. p. 14.
20. Charles, K. K., & Kline, P. [2001]. “Love Thy Neighbor? – Carpooling, Relational Costs, and the Production of Social Capital”. p. 54.
21. Westerbeck, G. J., & Evans, J. R. [1981]. “Car-Pooling and Regional Ridesharing Programs”. Vol. 27, nº4, p. 489-490
22. Angskun, T., & Angskun, J. [2009]. “A Travel Planning Optimization under Energy and Time Constraint”. International Conference on Information and Multimedia Technology. p. 131 - 134.
23. Skov-Peterson, H., Zachariassen, M., & Kefaloukos, P. K. [2010]. “Have a nice trip: an algorithm for identifying excess routes under satisfaction constraints”. International Journal of Geographical Information Science, Vol. 24, nº11. p. 1745-1758.
24. Calvo, R. W., Luigi, F. d., Haastrup, P., & Maniezzo, V. [2003]. “A distributed geographic information system for the daily car pooling system: Computers & operations research”. Vol. 31, nº 13. p. 2263–2278.
25. Buliung, R., Soltys, K., & Bui, R. [2009]. “Understanding Carpool Formation and Use: A Study of the Carpooling Behaviour of Smart Commute Respondents”. p. 56.
26. Rahul Chowdhury, “Evolution Of Mobile Phones: 1995 – 2012” [Online]. Disponível: <http://www.hongkiat.com/blog/evolution-of-mobile-phones/>. [Acedido: 05-Jan-2014].

27. Eglantine Dever, “Global 2013 smartphone sales to hit 1 billion” [Online]. Disponível: <http://blog.himediagroup.com/global-2013-smartphone-sales-to-hit-1-billion/>. [Acedido: 05-Jan-2014].
28. “Smartphone definition. PC Magazine Encyclopedia.” [Online]. Disponível: <http://www.pcmag.com/encyclopedia/term/51537/smartphone>. [Acedido: 05-Jan-2014].
29. Tilson, David. Sorensen, Carsten. Lyytinen, Kalle. [2011]. “The Paradoxes of Change and Control in Digital Infrastructures: The Mobile Operating Systems Case”. p. 26 – 35.
30. Koh, Damian [2010]. “Q&A: Microsoft on Windows Phone 7”. CNET Asia. CBS Interactive.
31. Ziegler, Chris, “Microsoft talks Windows Phone 7 Series development ahead of GDC: Silverlight, XNA, and no backward compatibility” [Online]. Disponível: <http://www.engadget.com/2010/03/04/microsoft-talks-windows-phone-7-series-development-ahead-of-gdc/>. [Acedido: 05-Jan-2014].
32. Strategy Analytics, “Android Captures Record 81 Percent Share of Global Smartphone Shipments in Q3 2013” [Online]. Disponível: <http://blogs.strategyanalytics.com/WSS/post/2013/10/31/Android-Captures-Record-81-Percent-Share-of-Global-Smartphone-Shipments-in-Q3-2013.aspx>. [Acedido: 05-Jan-2014].
33. Bright, Peter. “Windows Phone 7 Series in the Enterprise: not all good news” [Online]. Disponível: <http://arstechnica.com/information-technology/2010/03/windows-phone-7-series-in-the-enterprise-not-all-good-news/>. [Acedido: 05-Jan-2014].
34. Chacos, Brad. “Microsoft Now Calling It’s Windows 8 Metro Interface “Modern UI”” [Online]. Disponível: <http://blog.laptopmag.com/microsoft-now-calling-its-windows-8-metro-interface-modern-ui>. [Acedido: 05-Jan-2014].
35. Buchanan, Matt. “Windows Phone 7 Series: Everything Is Different Now” [Online]. Disponível: <http://gizmodo.com/5471805/windows-phone-7-series-everything-is-different-now>. [Acedido: 05-Jan-2014].
36. “Android Overview” [Online]. Disponível: http://www.openhandsetalliance.com/android_overview.html. [Acedido: 05-Jan-2014].
37. Elgin, Ben. “Google Buys Android for Its Mobile Arsenal” [Online]. Disponível: <http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>. [Acedido: 05-Jan-2014].
38. Amadeo, Ron. “Google’s iron grip on Android: Controlling open source by any means necessary” [Online]. Disponível: <http://arstechnica.com/gadgets/2013/10/googles-iron-grip-on-android-controlling-open-source-by-any-means-necessary/>. [Acedido: 05-Jan-2014].
39. “iOS: A visual history. The Verge” [Online]. Disponível: <http://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad>. [Acedido: 05-Jan-2014].
40. Coutinho, Dário. “Conheça a história da RIM, a fabricante do BlackBerry” [Online]. Disponível: <http://www.techtudo.com.br/artigos/noticia/2013/01/conheca-historia-da-rim-fabricante-do-blackberry.html>. [Acedido: 05-Jan-2014].
41. Viswanathan, Priya. “What is the BlackBerry OS?. Mobile Devices” [Online]. Disponível: <http://mobiledevices.about.com/od/glossary/g/What-Is-The-Blackberry-Os.htm>. [Acedido: 05-Jan-2014].
42. Aber, Jeremy. Harvey, Jerry. Inmon, Jonh. “GPS and its Emerging Role in Cellular Phones and Car Navigation Systems” [Online]. Disponível: <http://academic.emporium.edu/aberjame/student/aber1/>. [Acedido: 05-Jan-2014].
43. Raj, Rahul. Tolety, Seshu. [2012]. “A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach”. India Conference (INDICON). p.625 - 629.

44. “Noticias ao Minuto” [Online]. Disponível: <http://www.noticiasao minuto.com/pais/46645/portugueses-s%C3%A3o-os-que-mais-querem-partilhar-o-carro>. [Acedido: 05-Jan-2014].
45. “Getting Started with Android Studio” [Online]. Disponível: <http://developer.android.com/sdk/installing/studio.html>. [Acedido: 05-Jan-2014].
46. Tacla, C. A. [2008]. “Análise e Projeto OO & UML 2.0”. Departamento Académico de Informática, Panamá. p. 97.
47. Teal, R. [1987]. “Carpooling: who, how and why. Transportation Research Part A”. Vol. 21A, nº3. p. 203-214.
48. “PostgreSQL x MySQL. Qual Escolher?” [Online]. Disponível: <http://www.devmedia.com.br/postgresql-x-mysql-qual-escolher/3923>. [Acedido: 05-Jan-2014].
49. “ASP.NET” [Online]. Disponível: <http://pt.wikipedia.org/wiki/ASP.NET>. [Acedido: 05-Jan-2014].
50. “Understanding SQL And NoSQL Databases And Different Database Models” [Online]. Disponível: <https://www.digitalocean.com/community/tutorials/understanding-sql-and-nosql-databases-and-different-database-models>. [Acedido: 05-Jan-2014].
51. “User Stories” [Online]. Disponível: <http://www.mountaingoatsoftware.com/agile/user-stories>. [Acedido: 05-Jan-2014].
52. Faria, João. 2001. UML – diagramas de componentes, v.1.0.
53. “Web Service” [Online]. Disponível: http://pt.wikipedia.org/wiki/Web_service. [Acedido: 20-Jun-2014].
54. “ASP.NET Web Services Overview” [Online]. Disponível: http://www.elementswiki.com/en/ASP.NET_Web_Services_Overview. [Acedido: 20-Jun-2014].
55. “SOAP” [Online]. Disponível: <http://pt.wikipedia.org/wiki/SOAP>. [Acedido: 01-Ago-2014].
56. “Arquitetura de aplicações multiplataforma para dispositivos móveis” [Online]. Disponível: http://developer.nokia.com/community/wiki/Arquitetura_de_aplica%C3%A7%C3%B5es_multiplataforma_para_dispositivos_m%C3%B3veis. [Acedido: 01-Ago-2014].
57. Heltonbiker, “What are popular/good file formats used by dataloggers and data acquisition systems/software? - Stack Overflow.” [Online]. Disponível: <http://stackoverflow.com/questions/5033544/what-are-popular-good-file-formats-used-by-dataloggers-and-data-acquisition-syst/13040116#13040116>. [Acedido: 01-Ago-2014].
58. “REST” [Online]. Disponível: <http://searchsoa.techtarget.com/definition/REST>. [Acedido: 01-Ago-2014].
59. “Learn REST: A Tutorial.” [Online]. Disponível: <http://rest.elkstein.org/>. [Acedido: 01-Ago-2014].
60. “Extensible Markup Language (XML).” [Online]. Disponível: <http://www.w3.org/XML/>. [Acedido: 01-Ago-2014].
61. “XML Advantages & Disadvantages.” [Online]. Disponível: <http://www.theukwebdesigncompany.com/articles/xml-advantages-disadvantages.php>. [Acedido: 01-Ago-2014].
62. “JSON: The Fat-Free Alternative to XML.” [Online]. Disponível: <http://www.json.org/xml.html>. [Acedido: 01-Ago-2014].
63. “REST Vs SOAP, The Difference Between Soap And Rest” [Online]. Disponível: <http://spf13.com/post/soap-vs-rest>. [Acedido: 10-Ago-2014].
64. “Caminho crítico” [Online]. Disponível: http://pt.wikipedia.org/wiki/Caminho_cr%C3%ADtico. [Acedido: 10-Ago-2014].

65. “MySQL vs Postgres” [Online]. Disponível: <http://www.scriptrock.com/articles/postgres-vs-mysql>. [Acedido: 10-Ago-2014].
66. “.NET vs Java: How to Make Your Pick” [Online]. Disponível: <http://www.seguetech.com/blog/2013/06/03/dotnet-vs-java-how-to-pick>. [Acedido: 10-Ago-2014].
67. “PHP” [Online]. Disponível: <http://en.wikipedia.org/wiki/PHP>. [Acedido: 10-Ago-2014].
68. “Estudio comparativo de PHP, ASP.NET Y JAVA” [Online]. Disponível: <http://www.slideshare.net/Helmilpa/estudio-comparativo-de-php-aspnet-y-java#>. [Acedido: 10-Ago-2014].
69. “Worldwide Android Adoption: 80% Smartphone Market” [Online]. Disponível: <http://www.dazeinfo.com/2014/05/06/google-inc-goog-android-market-oems-monopoly/>. [Acedido: 10-Ago-2014].
70. “Desktop Top Browser Share Trend” [Online]. Disponível: <http://www.netmarketshare.com/>. [Acedido: 10-Ago-2014].
71. “Screenfly” [Online]. Disponível: <http://quirktools.com/screenfly/>. [Acedido: 22-Ago-2014].
72. “jQuery Mobile Docs - Supported platforms.” [Online]. Disponível: <http://jquerymobile.com/demos/1.2.0/docs/about/platforms.html>. [Acedido: 22-Ago-2014].
73. “A Sencha Touch MVC application with PhoneGap | Learn | Sencha.” [Online]. Disponível: <http://www.sencha.com/learn/a-sencha-touch-mvc-application-with-phonegap/>. [Acedido: 22-Ago-2014].
74. “Xamarin” [Online]. Disponível: <http://xamarin.com/>. [Acedido: 22-Ago-2014].
75. “PhoneGap | Home.” [Online]. Disponível: <http://phonegap.com/>. [Acedido: 22-Ago-2014].
76. “WebPageTest” [Online]. Disponível: <http://www.webpagetest.org/>. [Acedido: 25-Ago-2014].
77. “Top Retail Websites Not Getting Faster: Average Web Page Load Time Is 7.25 Seconds [Report]” [Online]. Disponível: <http://marketingland.com/retail-website-load-times-continue-to-decline-with-a-22-decrease-during-the-last-year-37604>. [Acedido: 25-Ago-2014].
78. “SATELLITE L50-A-18T” [Online]. Disponível: <http://www.toshiba.pt/discontinued-products/satellite-l50-a-18t/>. [Acedido: 25-Ago-2014].
79. “SoapUI - The Home of Functional Testing.” [Online]. Disponível: <http://www.soapui.org/>. [Acedido: 25-Ago-2014].
80. “Samsung Galaxy Ace 2 I8160” [Online]. Disponível: http://www.gsmarena.com/samsung_galaxy_ace_2_i8160-4559.php. [Acedido: 25-Ago-2014].
81. “GSM, 3G, EDGE, HPSA, 4G e LTE: entenda as siglas de conexão mobile” [Online]. Disponível: <http://canaltech.com.br/o-que-e/telecom/GSM-EDGE-HPSA-LTE-Entenda-as-siglas-de-conexao-mobile/>. [Acedido: 25-Ago-2014].
82. “Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options - developer.force.com.” [Online]. Disponível: http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options. [Acedido: 25-Ago-2014].
83. “Non-functional requirement” [Online]. Disponível: http://en.wikipedia.org/wiki/Non-functional_requirement. [Acedido: 25-Ago-2014].

Anexos

Anexo A – Descrição dos Casos de Uso

Neste anexo estão descritos detalhadamente os casos de uso do sistema com uma relação às *User Stories* referidas na secção 3.1. *User Stories*

Caso de Uso	User Story	Descrição
Aceder ao Perfil	6	Este caso de uso permite ao utilizador ver os dados detalhados do seu perfil assim como editar os seus dados pessoais.
Aceder à caixa de mensagens	7	O caso de uso serve para o utilizador aceder às mensagens privadas do sistema assim como dar resposta às mesmas. Este caso de uso permite aceitar ou não pedidos de associação a viagens.
Inserir Viagens	3	Este caso de uso tem como função a inserção de uma viagem para partilha com outros utilizadores do sistema.
Atribuir Avaliações	5	O caso de uso serve para um utilizador que tenha viagens partilhadas com outro possa atribuir uma classificação relativamente à forma como decorreu a viagem.
Efectuar Registo	1	Neste caso de uso o utilizador que pretenda utilizar o sistema MoCas terá de efectuar o seu registo no sistema, caso contrário não terá acesso a um conjunto de funcionalidades.
Efectuar Autenticação	2	Este caso de uso tem como função a autenticação no sistema de um utilizador registado, ou de um utilizador que possua conta no Facebook.
Procurar Viagens	11	Este caso de uso serve para um utilizador autenticado ou não, possa pesquisar as viagens que deseje.
Marcar Viagem	4	O caso de uso permite a um utilizador efectuar uma marcação de uma viagem.
Inserir Veículos	10	Nesta caso de uso é dada a possibilidade do utilizador associar veículos à sua conta para posteriormente poder partilhar viagens como condutor.
Criar Grupos / Eventos	8	O caso de uso permite ao utilizador criar grupos ou eventos e associar viagens e utilizadores a estes.
Ver informação de outros Utilizadores	12	Este caso de uso permite a cada utilizador aceder à informação de perfil de outros utilizadores para que possam partilhar viagens com as mesmas preferências.
Recuperar Credenciais de Acesso	9	Neste caso de uso é concedida a possibilidade de recuperar a <i>password</i> de acesso ao sistema caso o utilizador se tenha esquecido da mesma,

Tabela 10 - Descrição dos casos de uso do MoCaS.

Anexo B – Protótipos das Aplicações Web e Móvel do Sistema MoCaS

Neste anexo estão representados os protótipos iniciais da aplicação web e móvel do sistema MoCaS respectivamente.

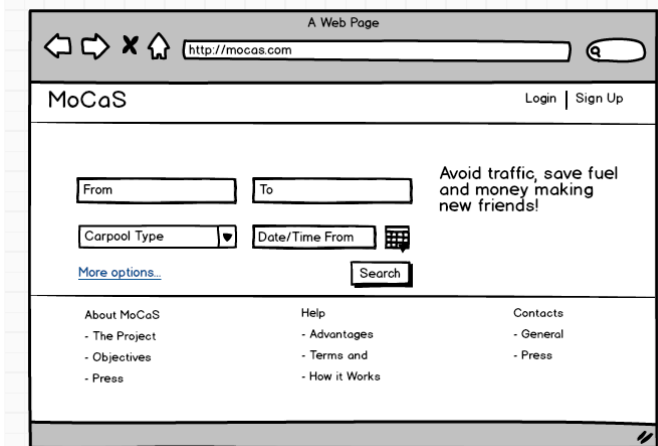


Figura 98 - Protótipo do ecrã inicial do website MoCaS.

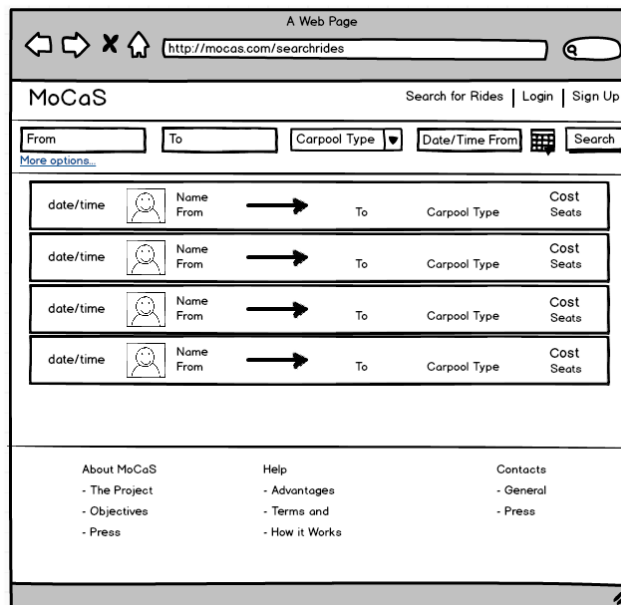


Figura 99 - Protótipo da listagem de viagens do website MoCaS.

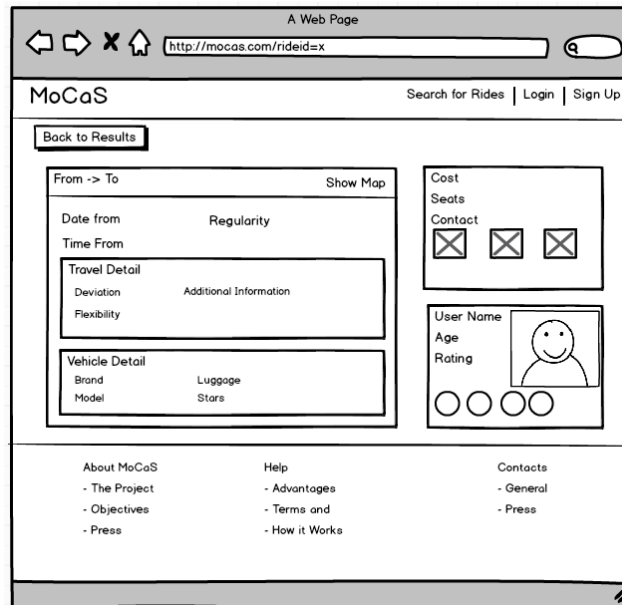


Figura 100 - Protótipo do ecrã de detalhe de viagem do *website* MoCaS.



Figura 101 - Protótipos do ecrã de entrada (à esquerda) e *login* (à direita) da aplicação móvel MoCaS.

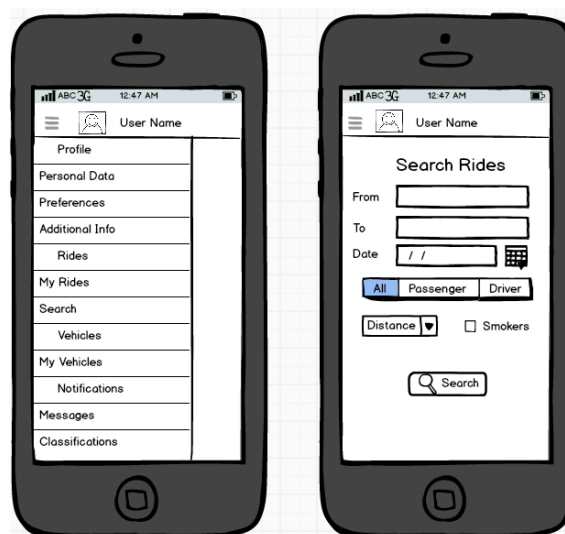


Figura 102 - Protótipos do ecrã de gestão (à esquerda) e pesquisa de viagens (à direita) da aplicação móvel MoCaS.

Anexo C – Simbologia do Sistema MoCaS

Neste anexo estão representadas algumas figuras presentes no sistema MoCaS com os respectivos significados.



Figura 103 - Logotipo do sistema MoCaS



Figura 104 - Ícone da definição de condutor na aplicação móvel MoCaS.



Figura 105 - Ícone da definição de passageiro na aplicação móvel MoCaS.



Figura 106 - Marcador de posição do veículo no mapa de viagens em tempo-real na aplicação móvel MoCaS.



Figura 107 - Marcador de posição de pontos de paragem no mapa de viagens em tempo-real na aplicação móvel MoCaS.



Figura 108 - Botão de pesquisa de viagens na aplicação móvel MoCaS.



Figura 109 - Botão de actualização de viagens na aplicação móvel MoCaS.

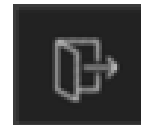


Figura 110 - Botão de *Logout* na aplicação móvel MoCaS.

Anexo D – Listagem dos Serviços Disponíveis no *Web Service*

Neste anexo é apresentada detalhadamente toda a informação relativa aos serviços disponíveis no *web service*.

Nome do Serviço	Parâmetros de Entrada	Descrição
AddRatingsUserRow	string user, string user_classified, int user_classification, int punctuality, int flexibility, int driving, string comment	Actualiza a avaliação entre dois utilizadores.
AddRatingsVehicleRow	string user, int vehicle, int vehicle_classification, int confort, int consumption, int luggage, string comment	Actualiza a avaliação de um veículo de um determinado utilizador.
BookTravel	int id_travel, int seats, string id_user	Insere uma marcação de viagem para um determinado utilizador.
GetAllTravelVehicleDetail	string status	Devolve todos os veículos para viagens com um determinado estado.
GetBookTravel	int id_travel, string id_user	Devolve a informação relativa a uma viagem marcada de um determinado utilizador.
GetCarMakes	string knownCategoryValues, string category	Devolve a lista de marcas de veículos.
GetCostVariables	string country	Devolve toda a informação para cálculo automático do custo de viagem para um determinado país.
GetCountUserTravelsByDate	string id,DateTime date	Devolve o número de viagens de um determinado utilizador numa determinada data.
GetFlexibilityWithStatus	string status	Devolve todas as flexibilidades de viagem para viagem com um determinado estado.
GetLiveTravels		Devolve a lista de viagens que estão a decorrer no momento.
GetModelsForMake	string knownCategoryValues, string category	Devolve a lista de modelos de veículos para uma determinada marca.
GetRatingsCount	string user1,string user2	Devolve o número de avaliações inseridas para dois utilizadores.
GetRatingsCountVehicle	string user, int vehicle	Devolve o número de avaliações inseridas para um veículo de um determinado utilizador.
GetRegularityWithStatus	string status	Devolve todas as regularidades de viagem para viagem com um determinado estado.
GetSeats	string knownCategoryValues, string category, string contextKey	Devolve a lista de lugares para um determinado veículo.

GetStops	string status	Devolve a lista de paragens de uma viagem com um determinado estado.
GetTravelOrders	int id_travel, string status	Devolve todos os pedidos de associação a viagem de um determinado utilizador e com um determinado estado.
GetTravelPeriod	string id_period	Devolve o período em que decorre uma viagem.
GetUpcommingTravels	string date	Devolve todas as viagens a decorrer depois de uma determinada data e hora.
GetUserBookedTravels	string id_user	Devolve todas as viagens que não estão terminadas em que um determinado utilizador está associado.
GetUserCars	string knownCategoryValues, string category, string contextKey	Devolve a lista de veículos de um determinado utilizador.
GetUserInfo	string id	Devolve toda a informação de um determinado utilizador.
GetUserMessageDetail	int id_message	Devolve toda a informação relativa a uma determinada mensagem privada.
GetUserOrderDetail	int id_order	Devolve a informação de um determinado pedido de adesão de viagem.
GetUserTravel	int id_travel	Devolve toda a informação de uma determinada viagem.
GetUserTravelFlexibility	int id_travel	Devolve toda a informação de flexibilidade de uma determinada viagem.
GetUserTravelRegularity	int id_travel	Devolve toda a informação de regularidade de uma determinada viagem.
GetUserTravelStops	int id_travel	Devolve toda a informação de todas as paragens de uma determinada viagem.
GetUserTravels	string id	Devolve a lista de todas as viagens criadas por um determinado utilizador.
GetUserVehicleDetail	int id_vehicle	Devolve toda a informação detalhada de um determinado veículo.
InsertCurrentPosition	int id_travel, string status, string latitude, string longitude	Insere a posição actual em que o utilizador se encontra no decorrer da viagem.
InsertFlexibility	DataSet ds	Insere uma ou mais flexibilidades temporais numa determinada viagem.
InsertLeaver	string email, string recomend, string cause, string comment	Regista as razões que levam um determinado utilizador a anular o registo no sistema MoCaS.
InsertLiveTravelOrder	string user_order, int id_travel, int seats, string status, int?	Insere um pedido de associação a uma viagem a decorrer no momento.

	id_section, string live_local, string latitude, string longitude	
InsertRatingsRow	string user1, string user2	Inserir um novo registo de avaliação entre dois utilizadores.
InsertRatingsVehicleRow	string user, int vehicle	Inserir um novo registo de avaliação de um veículo de um determinado utilizador.
InsertRegularity	DataSet ds	Inserir a regularidade numa determinada viagem.
InsertRide	string local_from, string local_to, DateTime? date_time_start, DateTime? date_time_end, bool driver, string additional_comment, string travel_creator, float cost, bool book, int? seats_available, int? max_deviation, string longitude_from, int? vehicle, string longitude_to, string latitude_to, string latitude_from, bool coordinates_automatic, bool highway, int? luggage	Inserir uma viagem no sistema e devolver o seu número de identificação.
InsertStops	DataSet ds	Inserir uma ou mais paragens de uma determinada viagem.
InsertTravelOrder	string user_order, int id_travel, int seats, string status, int? id_section	Inserir um pedido de associação a uma viagem por parte de um determinado utilizador.
InsertVehicle	string vehicle_photo, string user, int id_vehicle_detail, int seats_number, int class_num, int color_id, int year, int fuel_type	Inserir um novo veículo associado a um determinado utilizador.
RemoveBookedTravel	int id_travel, string id_user	Remover a marcação de uma viagem para um determinado utilizador.
RemoveFlexibility	int id_travel	Remover toda a flexibilidade temporal de uma determinada viagem.
RemoveRegularity	int id_travel	Remover toda a regularidade de uma determinada viagem.
RemoveStops	int id_travel	Remover todas as paragens de uma determinada viagem.
SendPrivateMessage	string user_from, string user_to, string subject, string message, DateTime date_hour, int? id_msg_root, bool? seen, bool automatic, int? order_id	Enviar uma mensagem privada para um determinado utilizador.
SendPrivateMobileMessage	string user_from, string user_to, string subject, string message, string date_hour, int? id_msg_root, bool? seen, bool automatic, int? order_id	Enviar uma mensagem privada para um determinado utilizador.

UpdateBookTravel	int id_travel, int seats, string id_user	Actualiza o número de lugares de uma marcação de uma viagem de um determinado utilizador.
UpdateFlexibility	DataSet ds	Actualiza a informação relativa às flexibilidades temporais de uma determinada viagem.
UpdateMessageAccepted	int id_message, bool accepted	Actualiza o estado de uma mensagem privada associada a um pedido de associação de uma viagem.
UpdateMessageState	int id_message, bool seen	Actualiza o estado de visualização de uma de terminada mensagem privada.
UpdateOrderState	int id_order, string status	Actualiza o estado do pedido de adesão de viagem de um determinado utilizador.
UpdateRegularity	bool monday, bool tuesday, bool wednesday, bool thursday, bool friday, bool saturday, bool sunday, int num_occurr, int period, DateTime? time_travel, DateTime? time_return, int id_regularity	Actualiza a informação relativa à regularidade de uma determinada viagem.
UpdateRide	int id_travel, string local_from, string local_to, DateTime? date_time_start, DateTime? date_time_end, bool driver, string additional_comment, float cost, bool book, int? seats_available, int? max_deviation, string longitude_from, int? vehicle, string longitude_to, string latitude_to, string latitude_from, bool coordinates_automatic, bool highway, int? luggage	Actualiza os dados de uma viagem que ainda não decorreu.
UpdateStops	DataSet ds	Actualiza a informação relativa às paragens de uma determinada viagem.
UpdateTravelSeats	int id_travel, int seats	Actualiza o número de lugares disponíveis numa determinada viagem.
UpdateVehicle	string vehicle_photo, int id_vehicle_detail, int seats_number, int class_num, int color_id, int year, int fuel_type, int id_vehicle	Actualiza toda a informação relativa a um determinado veículo.
ValidateLogin	string email, string password	Valida se as credências de acesso são válidas e devolve a informação do utilizador caso o sejam.

Tabela 11 - Listagem e descrição dos serviços disponíveis no *web service* do sistema MoCaS.