

UNIVERSIDADE DE LISBOA  
FACULDADE DE CIÊNCIAS  
DEPARTAMENTO DE INFORMÁTICA



## **Evolução de Sistema de Análise de Trajetos com Rosas dos Ventos**

Luís Paulo Garcia Ferreira

**Mestrado em Engenharia Informática**  
Especialização em Sistemas de Informação

Dissertação orientada por:  
António Manuel Silva Ferreira  
Ana Paula Pereira Afonso



# Agradecimentos

Esta dissertação de mestrado é o resultado de muitas horas de trabalho, mas não teria sido possível sem a colaboração, estímulo e empenho de várias pessoas que me ajudaram em mais uma etapa da minha vida.

Os meus agradecimentos vão em primeiro lugar para os meus pais, irmã e restante família, que foram sempre um pilar bastante importante durante todo o meu percurso. Estiveram presentes comigo nos bons e maus momentos, tendo-me ajudado a ultrapassar os tempos mais difíceis.

Em segundo lugar, gostaria de agradecer também o apoio e orientação dada pelo Professor António Ferreira e pela Professora Ana Paula Afonso. A concretização desta tese resulta da sua dedicação, disponibilidade, visão crítica e oportuna nos momentos cruciais e que contribuíram para enriquecimento deste trabalho, sendo sempre incentivado.

Gostaria ainda de agradecer aos meus amigos e colegas de curso pelo convívio, amizade e apoio demonstrado nos momentos que passámos juntos. Foram eles que me ajudaram a ultrapassar as barreiras e momentos mais difíceis, mas também foi com eles em que vivi grandes momentos.

Também quero agradecer aos participantes nos testes de utilizadores. A sua participação foi muito importante na obtenção de dados para quantificação de métricas importantes na avaliação do sistema.

Por fim, a todos aqueles que contribuíram direta ou indiretamente, para a realização desta dissertação o meu sincero agradecimento.

Este projeto contou com o financiamento da Fundação para a Ciência e a Tecnologia (FCT) com a referência UID/CEC/00408/2013 (LaSIGE)



*À minha família e amigos.*



# Resumo

Este trabalho utiliza o conceito de rosa dos ventos, que consiste num diagrama radial usado na área da meteorologia, onde são registados os dados dos ventos, como a velocidade e direção. Porém, no âmbito desta dissertação a ideia da rosa foi utilizada para a análise de dados de trajetos urbanos, permitindo representar e agregar este tipo de informação.

A rosa dos trajetos é uma forma de representação e visualização que tem por objetivo analisar percursos urbanos e ser uma ferramenta de apoio na tomada de decisão por parte dos analistas. Por exemplo, pode ser importante perceber o tráfego urbano de uma cidade ao longo de uma semana, em diferentes períodos do dia e locais de uma cidade. Para tal, um analista pode colocar um marcador num ponto do mapa com trajetos urbanos, sendo criada uma região de agregação onde são interseçados trajetos. Desta interseção resulta uma rosa com pétalas (os trajetos) que contém informação agregada, como a quantidade de trajetos, a velocidade, e a direção. Podem ser criadas várias rosas para possibilitar a visualização simultânea e análise comparativa entre estas, incluindo o caso particular de duas ou mais rosas estarem associadas ao mesmo ponto no mapa, contemplando, por exemplo, diferentes períodos do dia.

Este trabalho teve dois objetivos principais: evoluir um sistema de análise de trajetos do ponto de vista de um analista e fazer a avaliação deste sistema, através de testes com utilizadores.

Relativamente ao primeiro objetivo, a biblioteca de funcionalidades da rosa de trajetos foi revista e aumentada, através da inclusão de uma *framework* para fazer um processamento de trajetos mais rápido e da possibilidade de criar rosas duplicadas. Também foram realizadas melhorias na interação com o utilizador, sendo permitida uma maior liberdade de escolha dos intervalos temporais em análise, movimentar os marcadores das rosas com atualização automática dos seus dados agregados, e a possibilidade de deteção e análise de movimentos pendulares. Estas modificações devem facilitar a inspeção e análise de trajetos mais pormenorizada e exploratória que é o foco do trabalho dos analistas.

O segundo objetivo correspondeu à avaliação do novo sistema de análise de trajetos a partir de uma aplicação de testes onde foram feitas sessões de avaliação com utilizadores. O propósito destes testes foi avaliar a facilidade de inspeção e análise de trajetos a partir de um conjunto de tarefas pedidas. Outro ponto importante em estudo, foi perceber qual o nível de usabilidade do sistema, a partir das respostas dadas a um questionário SUS.

Os resultados dos testes registaram um aumento no número de erros e no tempo de execução das tarefas face ao sistema anterior, mas isto deveu-se ao aumento da complexidade das tarefas de análise, pois houve uma maior preocupação de aproximar as questões, tarefas e cenários ao trabalho com que os analistas se deparam. Todos os utilizadores conseguiram concluir com sucesso as tarefas pedidas e consideraram que o sistema apresenta um alto nível de usabilidade. Os resultados sugerem que o sistema concretizado facilita a análise e visualização de trajetos agregados, por exemplo na gestão do tráfego urbano de uma cidade.

**Palavras-chave:** Rosa dos Trajetos; Visualização Analítica; Agregação de Trajetos; Análise de Trajetos; Movimentos Pendulares.



# Abstract

This work uses the concept of wind rose, which is a radial diagram used in the area of meteorology, where wind data such as speed and direction are recorded. However, in the scope of this dissertation the idea of the rose was used for the analysis of urban route data, allowing to represent and aggregate this type of information.

The tracks rose is a form of representation and visualization that aims to analyze urban tracks and to be a support tool in decision making by analysts. For example, it may be important to understand urban traffic in a city over the course of a week, at different times of the day, and in locations within a city. To do this, an analyst can place a marker on a map point with urban tracks, creating an aggregation region where tracks are intersected. This intersection results in a rose with petals (the tracks) that contains aggregate information such as the number of paths, the speed, and the direction. Multiple roses can be created to enable simultaneous visualization and comparative analysis between them, including the particular case of two or more roses being associated with the same point on the map, contemplating, for example, different times of the day.

This work had two main objectives: to evolve a path analysis system and to evaluate the new system through tests with users.

Regarding the first objective, the tracks rose feature library has been revised and expanded by including a framework for faster track processing and the possibility of creating duplicate roses. Improvements have also been made in the interaction with the user, allowing greater freedom of choice of the time intervals under analysis, moving rose markers with automatic updating of their aggregate data, and the possibility of detecting and analyzing pendulum movements. These modifications should facilitate the more detailed and exploratory track inspection and analysis that is the focus of analysts' work.

The second objective corresponded to the evaluation of the new track analysis system from a testing application where user evaluation sessions were held. The purpose of these tests was to evaluate the ease of inspection and analysis of tracks from a set of requested tasks. Another important point under study was to understand the level of usability of the system, based on the answers given to a SUS questionnaire.

Test results showed an increase in the number of errors and task execution time compared to the previous system, but this was due to the increased complexity of the analysis tasks, as there was a greater concern to approximate issues, tasks and scenarios to the work analysts face. All users were able to successfully complete the requested tasks and considered the system to have a high level of usability. The results suggest that the implemented system facilitates the analysis and visualization of aggregate tracks, for example in the urban traffic management of a city.

**Keywords:** Track Rose; Analytical Visualization; Aggregation of tracks; Track Analysis; Pendular Movements.



# Conteúdo

Capítulo 1 Introdução .....	1
1.1 Motivação .....	1
1.2 Objetivos .....	2
1.3 Planeamento e execução.....	2
1.4 Contribuições.....	4
1.5 Notação.....	5
1.6 Organização do documento .....	5
Capítulo 2 Conceitos e trabalho relacionado.....	7
2.1 Conceitos.....	7
2.1.1 Rosa dos ventos.....	7
2.1.2 Trajetos, dados GPX e mapas .....	9
2.1.3 Agregação de dados.....	11
2.1.4 Visualização analítica de dados.....	11
2.1.5 Trabalho dos analistas.....	12
2.1.6 Serviços web.....	13
2.2 Aplicações existentes.....	13
2.2.1 Aplicações de visualização analítica .....	13
2.2.2 Protótipo anterior do sistema de análise de trajetos .....	14
2.3 Sumário .....	17
Capítulo 3 Desenvolvimento e demonstração do sistema.....	19
3.1 Ferramentas e processo de desenvolvimento.....	19
3.2 Requisitos do sistema.....	20
3.3 Sistema de análise de trajetos .....	20
3.3.1 Arquitetura do sistema.....	21
3.3.2 Base de dados geográfica.....	22
3.3.3 Serviços <i>web</i> .....	23
3.3.4 Biblioteca da rosa dos ventos.....	27
3.3.5 Aplicação <i>web</i> de análise de trajetos ( <i>WindRoseApp</i> ) .....	28

3.3.6 Bibliotecas externas.....	36
3.4 Análise de conjunto de trajetos .....	36
3.4.1 Dados de trajetos de táxis .....	36
3.4.2 Carregamento de trajetos da base de dados e seleção de trajetos.....	36
3.4.3 Criação de uma rosa .....	37
3.4.4 Criação de uma nova rosa para um mesmo marcador.....	38
3.4.5 Movimentação de uma rosa .....	40
3.4.6 Destruição de uma rosa.....	40
3.4.7 Simulação de dados trajetos em tempo real.....	41
3.5 Sumário .....	42
Capítulo 4 Avaliação do sistema.....	43
4.1 Propósito.....	43
4.2 Estudo com utilizadores .....	43
4.2.1 Participantes.....	43
4.2.2 Aparato .....	44
4.2.3 Tarefas .....	45
4.2.4 Procedimento .....	49
4.3 Resultados.....	51
4.3.1 Tempos de resposta .....	51
4.3.2 Tempos de recuperação.....	53
4.3.3 Erros .....	53
4.3.4 Usabilidade do sistema.....	54
4.4 Discussão.....	56
4.4.1 Tempos de resposta .....	56
4.4.2 Erros .....	57
4.4.3 Usabilidade do sistema.....	57
4.4.4 Influência de experiência prévia .....	58
4.4.5 Fluxos de trabalho .....	58
4.4.6 Comparação com os resultados do protótipo anterior .....	59
4.5 Sumário .....	60
Capítulo 5 Conclusões .....	61
5.1 Principais contribuições.....	61
5.2 Competências adquiridas.....	62
5.3 Dificuldades encontradas.....	62

5.4 Trabalho futuro .....	63
Bibliografia.....	65
Apêndice A Póster no 3 <sup>rd</sup> LASIGE Workshop.....	69
Apêndice B Participação no Ciência Viva 2018 .....	71
Apêndice C Proposta de artigo científico.....	73
Apêndice D Documentação da biblioteca <i>windroseLib</i> .....	85
Apêndice E Guião de avaliação de testes de utilizadores da aplicação <i>web windroseApp</i> .....	107
Apêndice F Questionário inicial dos testes .....	111
Apêndice G Folheto informativo .....	113
Apêndice H Formulário de consentimento informado .....	115
Apêndice I Questionário de usabilidade (SUS).....	117



## Lista de figuras

Figura 2.1 – Rosa dos ventos do aeroporto da ilha do Pico, nos Açores agregando ventos registados ao longo de um ano (Meteoblue, 2016).....	8
Figura 2.2 – Visualização de um percurso registado através da aplicação <i>Geo Tracker</i> (Ilya Bogdanovich, 2016). .....	10
Figura 2.3 – Representação visual do estado do trânsito em Pequim, numa hora de pico, num dia de semana, entre as 7 e 8 da manhã (J. Zhang et al., 2017). .....	10
Figura 2.4 – Arquitetura do Sistema de Análise de Trajetos com Rosas dos Ventos (Vaz, 2017).....	14
Figura 2.5 – Cenário de criação de um ficheiro GPX com trajetos para a aplicação de análise de trajetos do sistema (Vaz, 2017). .....	15
Figura 2.6 – Exemplo de funcionamento da aplicação do sistema de análise de trajetos, com uma rosa dos ventos (Vaz, 2017). .....	16
Figura 2.7 – Cenário de criação de marcador/rosa com a distribuição das responsabilidades entre a biblioteca da rosa e da aplicação de análise de trajetos do sistema (Vaz, 2017).....	17
Figura 3.1 – Arquitetura do sistema de análise de trajetos com rosas dos ventos.....	21
Figura 3.2 – Modelo EA de dados de um trajeto. ....	22
Figura 3.3 – Tempos de pesquisa sem e com ferramenta <i>Slim Framework</i> da aplicação.....	27
Figura 3.4 – Ecrã da aplicação de análise <i>trackRoseApp</i> quando carrega a página pela primeira vez. .	29
Figura 3.5 – Exemplo de criação de marcador e rosa dos trajetos na aplicação de análise <i>trackRoseApp</i> . .....	30
Figura 3.6 – Painel de filtros modificado.....	32
Figura 3.7 – Rosas dos trajetos em dois locais diferentes do mapa, sendo que para o local 1 existem duas rosas, uma para o período da manhã e a outra para a tarde, para o estudo de movimentos pendulares.	35
Figura 3.8 – Cenário de conversão e carregamento de um ficheiro GPX, com posterior consulta de trajetos (Vaz, 2017).....	37
Figura 3.9 – Página <i>web</i> do serviço <i>selectTracks</i> , onde se selecionam os trajetos da aplicação <i>windRoseApp</i> (Vaz, 2017). .....	37
Figura 3.10 – Cenário de criação de um marcador na aplicação de análise <i>windRoseApp</i> . .....	38
Figura 3.11 – Cenário de nova rosa para um mesmo marcador. ....	39
Figura 3.12 – Cenário de criação de duplicado da rosa na aplicação de análise <i>windroseApp</i> .....	39
Figura 3.13 – Cenário de movimentação de um marcador na aplicação de análise <i>windroseApp</i> e atualização automática do conteúdo da rosa.....	40
Figura 3.14 – Cenário de destruição da rosa e marcador na aplicação de análise <i>windroseApp</i> e atualização automática do conteúdo da rosa.....	41
Figura 4.1 – Questão do tipo SEMAP que apareceu durante os testes de utilizador.....	46
Figura 4.2 – Questão do tipo FISE que apareceu durante os testes de utilizador. ....	47

Figura 4.3 – Questão do tipo SERO que apareceu durante os testes de utilizador. ....	48
Figura 4.4 – Questão do tipo INSE que apareceu durante os testes de utilizador.....	48
Figura 4.5 – Panfleto com esquema e legenda da aplicação, computador usado nos testes, folheto informativo e consentimento informado. ....	50
Figura 4.6 – Tempos de resposta dos participantes, por questão. ....	51
Figura 4.7 – Tempos de resposta dos participantes, por tipo de questão.....	52
Figura 4.8 – Tempos de resposta dos participantes, por género e tipo de questão.....	52
Figura 4.9 – Tempos de resposta dos participantes, por ordem de questão. ....	52
Figura 4.10 – Tempos de resposta, por participante.....	53
Figura 4.11 - Erros por questão. ....	53
Figura 4.12 - Erros por tipo de questão.....	54
Figura 4.13 - Erros por género.....	54
Figura 4.14 – Classificações de usabilidade do sistema.....	55
Figura 4.15 – Classificações de usabilidade do sistema, por género do participante.....	55
Figura 4.16 – Pontuações atribuídas em cada pergunta do questionário <i>SUS</i> . ....	56

## Lista de tabelas

Tabela 1 – Planeamento e execução das tarefas do projeto .....	3
Tabela 2 – Serviços <i>Web</i> disponíveis .....	23
Tabela 3 – URIs de acesso a recursos e métodos HTTP aceites .....	24
Tabela 4 – Exemplo de pedido AJAX.....	25
Tabela 5 – Questões do teste e respetivas descrições.....	49



# Capítulo 1

## Introdução

No contexto de Projeto de Engenharia Informática (PEI) do 2.º ano do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa, foi proposto um projeto para construção de um novo protótipo de um sistema de análise de trajetos agregados com rosas dos ventos. Pretendeu-se evoluir as funcionalidades existentes, passando a ser possível efetuar análises sobre maiores quantidades de trajetos, com possibilidade de criar várias rosas em simultâneo, permitir a movimentação das rosas com atualização automática dos dados e concretização de filtros de tempo mais sofisticados.

Neste capítulo é apresentada a motivação do trabalho, os objetivos estabelecidos para o projeto, o planeamento e execução das tarefas, as principais contribuições do trabalho, e a notação e organização do documento.

### 1.1 Motivação

Com a proliferação de novas tecnologias, tornaram-se acessíveis grandes quantidades de informação. Por exemplo, a partir de dispositivos de posicionamento global (GPS) ubíquos, podemos registar trajetos percorridos por entidades/objetos (pessoas, veículos, barcos, aviões, etc.). Por essa razão, começaram a surgir estudos de mobilidade para a análise de trajetos, com base nestes registos de movimentos de dados não tratados. Por exemplo, para identificar movimentos pendulares, temos o estudo sobre as deslocações casa-trabalho-casa no sistema de transportes da cidade de Brasília, no Brasil (Izawa, Oliveira, Cajueiro, & Mello, 2017). Também é possível identificar locais críticos, onde há congestionamento do trânsito, através de mapas animados, como o estudo feito em Pequim (J. Zhang et al., 2017).

São estes dados de trajetos que foram trabalhados ao longo deste trabalho. Porém, é necessário processar, analisar e tratar estes dados para que possam ser utilizados para diferentes propósitos.

Apesar de os mapas animados já permitem obter alguma informação útil sobre dados de trajetos, podendo aplicar filtros temporais e espaciais, o decisor tem de analisar o mapa no seu todo para encontrar padrões de movimentos, não possuindo ferramentas ou sistemas de apoio que apresentem de forma resumida os principais resultados (Andrienko & Andrienko, 2010).

Deste modo, para permitir uma melhor análise de trajetos, pretendeu-se evoluir um sistema que possibilite ao analista incorporar num mapa, gráficos inspirados nas rosas dos ventos, que apresentam dados agregados de trajetos que intersejam uma dada região. Uma rosa dos ventos é um gráfico radial muito usado na área da meteorologia para identificação de ventos dominantes numa determinada região, como por exemplo o aeroporto da ilha do Pico (Meteoblue, 2016). Contudo, em vez de o sistema incorporar dados de ventos, são usados dados de trajetos. Assim, a pétala maior passa a ser a da direção que agrega mais trajetos e a maior secção colorida dentro de cada pétala representa o intervalo de velocidades mais comum. Para além destes elementos gráficos, devem existir outros que forneçam a um analista mais informação e possibilidades de interação, como, por exemplo, filtros temporais e espaciais, e a indicação da direção da origem e destino dos trajetos que atravessam a região de agregação da rosa.

O trabalho que serviu de base a este projeto foi desenvolvido no âmbito da tese de mestrado do Ricardo Vaz (Vaz, 2017). O sistema por ele desenvolvido é composto por uma aplicação *web*, uma biblioteca, e serviços *web* suportados por uma base de dados geográfica. No entanto, esse sistema tem um conjunto de limitações, como por exemplo, ao nível de carregamento dos trajetos, pois é sempre necessário inserir o ficheiro de trajetos para correr a aplicação, mesmo estes já se encontrando na base de dados. Não permite movimentar as rosas e atualizar de forma automática o conteúdo dos seus dados e também não permite analisar diferentes rosas associadas a um mesmo ponto no mapa com diferentes faixas horárias. Estas restrições são ultrapassadas com o atual sistema desenvolvido que é descrito neste documento.

## 1.2 Objetivos

Para evoluir o protótipo de sistema de análise de trajetos com rosas dos ventos, no sentido de ter uma ferramenta que facilite o trabalho dos analistas, foram definidos os seguintes objetivos:

### Objetivo 1: Evolução do sistema de análise de trajetos

O primeiro objetivo correspondeu a evoluir o sistema anterior com as seguintes funcionalidades:

- 1) Permitir que o local no mapa associado à rosa dos ventos possa ser movido livremente, com atualização automática do conteúdo da rosa (de forma fluida, com o mínimo de pausas);
- 2) Suportar a atualização automática dos trajetos que aparecem no mapa, assumindo que chegam periodicamente dados novos (por exemplo, a cada 90 segundos).
- 3) Permitir a criação de várias rosas dos ventos num mesmo local, com possibilidade de aplicação de filtros com diferentes períodos do dia, para análise de movimentos pendulares. No sistema anterior era possível ter apenas uma rosa dos ventos associada a um local.
- 4) Concretização de filtros de tempo e espaço mais sofisticados. Por exemplo, para contemplar o ajuste do que se consideram ser as horas de ponta, e pelo reforço da componente interativa dos elementos gráficos da rosa dos ventos. Atualmente conseguem-se saber todas as direções de origens e destinos dos trajetos que cruzam um dado local, devendo passar a ser permitida a visualização do subconjunto das direções dos destinos a partir de uma direção de origem selecionada, entre outras possibilidades.

### Objetivo 2: Avaliação do novo sistema de análise de trajetos

O segundo objetivo correspondeu a avaliar o novo sistema e as suas novas funcionalidades através de testes com utilizadores, onde foram recolhidos dados do desempenho destes, nomeadamente tempos de resposta e número erros cometidos. Para isso foi criada uma aplicação de testes onde o utilizador realiza um conjunto de tarefas no âmbito da análise de trajetos com rosas dos ventos. Desta forma pode-se avaliar o grau de usabilidade do sistema, a partir das respostas dadas a um questionário SUS e tentar compreender se este tipo de abordagem é útil no apoio à tomada de decisão, no trabalho dos analistas.

## 1.3 Planeamento e execução

A Tabela 1 mostra a duração prevista e real das tarefas que foram realizadas ao longo deste projeto. Para além dos objetivos propostos foi incluída a escrita de um artigo para uma conferência. Contudo, a execução não correu conforme o planeado devido ao facto de não ter tido aproveitamento a três disciplinas da componente curricular, tendo uma disciplina do primeiro semestre e duas do segundo semestre. Outra limitação foram as dificuldades na implementação de um dos objetivos do trabalho a atualização de novos dados (simulação em tempo real), pois é uma funcionalidade muito recente e apresenta um número reduzido de ferramentas *open source*, sendo na sua maioria pagas. As que existem apresentam

documentação limitada ou muitas lacunas para quem inicia ou não domina esta área. Isto levou a um atraso do trabalho e foi necessário reajustar os objetivos. Além disso, também comecei a trabalhar numa empresa nos Açores, o que contribuiu para o adiamento da entrega do relatório final.

Tabela 1 – Planeamento e execução das tarefas do projeto

Tarefas	Duração prevista (meses)	Duração real (meses)	Intervalo de execução
1. Revisão de ferramentas e linguagens: <ul style="list-style-type: none"> <li>PHP, JavaScript, JQuery</li> </ul>	2	2	outubro a novembro 2017
2. Pesquisa bibliográfica: <ul style="list-style-type: none"> <li>Literatura sobre visualização analítica e rosas dos ventos;</li> <li>Literatura sobre movimentos urbanos e preocupações dos analistas;</li> <li>Sistemas já desenvolvidos ou relacionados com tema de agregação de trajetos para visualização analítica (ex: tipos de representação; tipos de mapas)</li> </ul>	2	2	outubro a novembro 2017
3. Análise de protótipo anterior: <p>Compreensão da arquitetura do sistema desenvolvido:</p> <ul style="list-style-type: none"> <li>Base de dados geográfica desenvolvida com o <i>PostGIS</i>;</li> <li>Biblioteca com funções para rosas dos ventos;</li> <li>Aplicação de arranque do sistema (carregamento dos trajetos e seu formato);</li> <li>Serviços disponibilizados pelo Servidor <i>Web</i>;</li> <li>Exploração da interface e da aplicação de análise de trajetos (manipulação de rosas, trajetos e filtros para inspeção e aferição dos resultados apresentados).</li> </ul> <p>Resultados da avaliação do sistema com utilizadores</p> <ul style="list-style-type: none"> <li>Avaliar os resultados de usabilidade para os tempos de resposta e erros associados a cada cenário (<i>SUS</i>).</li> </ul> <p>Opinião de perito sobre o protótipo</p> <ul style="list-style-type: none"> <li>Perceber os pontos fortes e menos bem conseguidos do protótipo. Outros tipos de abordagens tipo de funcionalidades, representações de trajetos urbanos.</li> </ul>	2	2	outubro a novembro 2017
4. Escrita de um artigo (extra): <ul style="list-style-type: none"> <li>Sobre análise de trajetos com rosa dos ventos, com base nos resultados da avaliação com utilizadores obtidos do protótipo anterior. Tradução para versão inglesa.</li> </ul>	-	2	novembro 2017 e março 2019
5. Escrita do relatório preliminar: <ul style="list-style-type: none"> <li>Sobre os três objetivos a desenvolver ao longo do todo o projeto, para a análise de trajetos com rosa dos ventos;</li> <li>Análise de outras funcionalidades de interação com o utilizador;</li> <li>Especificação de um cenário de análise de dados e suas tarefas de decisão;</li> </ul>	1	1	janeiro 2018
6. Desenho e Codificação do Sistema: <ul style="list-style-type: none"> <li>Desenho e codificação do sistema de um protótipo funcional: evolução da aplicação dos serviços web, manipulação direta no mapa (outras formas geométricas) e novos filtros de espaço e tempo.</li> </ul>	6	9	fevereiro a novembro 2018
6. Desenho e Codificação de uma aplicação de teste do protótipo: <ul style="list-style-type: none"> <li>Desenho e codificação de testes com utilizadores com o novo protótipo funcional com o intuito de tirar conclusões quanto às novas funcionalidades no apoio à análise de trajetos no contexto criado.</li> </ul>	2	3	dezembro 2018 a fevereiro 2018
7. Avaliações da usabilidade do protótipo e respetivas correções: <p>Avaliação do impacto da rosa dos ventos num cenário de tomada de decisão</p>	2	2	março e abril 2019
9. Escrita do relatório final	1	4	maio a setembro 2019

## 1.4 Contribuições

A concretização do trabalho deu origem a quatro contribuições que se descrevem de seguida.

A primeira contribuição foi a evolução da infraestrutura da aplicação *web* com utilização da ferramenta *Slim Framework*, que permitiu redefinir e adicionar um conjunto de serviços *web* para a interação nos acessos à base de dados geográfica e redução dos tempos de resposta.

A segunda contribuição correspondeu à reestruturação da biblioteca da rosa dos trajetos e inclusão de novas funcionalidades, como a movimentação e atualização automática do conteúdo das rosas e a possibilidade de criação e análise de várias rosas, associadas a um mesmo ponto no mapa. Foram melhorados alguns elementos gráficos das rosas com a inclusão de pontos cardeais e a modificação da escala de cores. Foi melhorada a interface do painel de filtros com adição de novos filtros de tempo dinâmicos, e a sinalização de incoerências de valores de rosas selecionadas no painel de filtros. Estas alterações melhoraram a capacidade de análise de trajetos de acordo com o trabalho realizado de um ponto de vista de um analista.

A terceira contribuição correspondeu à avaliação da nova versão do sistema de análise de trajetos com rosas dos ventos. Esta avaliação foi feita a partir de testes de utilizadores tendo em conta o número de erros ocorridos e o tempo de resposta de cada tarefa. Nestes testes foram aferidos o desempenho dos participantes e a usabilidade do sistema. Os resultados revelaram que os utilizadores levaram mais tempo na resposta e cometeram mais alguns erros, mas era esperado dado o aumento da complexidade das questões, pois eram realizadas no contexto do trabalho dos analistas. Contudo os utilizadores conseguiram concluir as tarefas pedidas e o sistema revelou ter um bom grau de usabilidade.

Estes resultados sugerem que o sistema de análise de trajetos com rosas dos trajetos é uma boa ferramenta para a visualização de dados de trajetos agregados, com potencial para simplificar o trabalho dos analistas e ajudar na tomada de decisões.

Adicionalmente, participei em diferentes eventos, como: Encontro de Ciência e Tecnologia 2018<sup>1</sup> que decorreu a 4 de julho de 2018, no Centro de Congressos de Lisboa (ver Apêndice A), e a *Jobshop Ciências 2019*<sup>2</sup>, a 11 de abril de 2019, na Faculdade de Ciências, onde fiz demonstrações do sistema de análise de trajetos. Também colaborei no *3rd LASIGE Workshop*<sup>3</sup>, a 7 de junho de 2018, para o qual preparei um poster alusivo ao sistema desenvolvido (ver Apêndice B). Participei num seminário “Lisboa a caminho de ser uma cidade inteligente”, onde foram abordados temas relacionados com o trabalho desenvolvido para o projeto *Smart Cities*, em que a principal objetivo é criar sistemas de apoio à decisão eficientes para a rentabilização de recursos e para melhoria da qualidade de vida das pessoas, por exemplo, na gestão do tráfego urbano, onde tive oportunidade de mostrar o meu sistema.

Por último, foi realizada a escrita de um artigo em inglês, ainda não submetido sobre o anterior e atual protótipo do sistema (ver Apêndice C).

---

<sup>1</sup> <http://www.encontrociencia.pt/2018/home/>

<sup>2</sup> <http://jobshop2019.campus.ciencias.ulisboa.pt/>

<sup>3</sup> <https://workshop2018.lasige.di.fc.ul.pt/index.php/program/>

## 1.5 Notação

A língua escolhida para a escrita deste relatório foi o português. As palavras noutra idioma são escritas em itálico. Todo o código de computador é apresentado na fonte `Courier New`.

## 1.6 Organização do documento

O documento está organizado em quatro capítulos, que se apresentam abaixo:

No Capítulo 1 está definido o plano geral do trabalho: motivação, objetivos, planeamento e execução do projeto, a notação adotada e a organização do documento.

O Capítulo 2 abrange o estado de arte do trabalho com os conceitos de rosa dos ventos, movimentos de trajeteto (trajetórias) com dados espaço-temporais, tipos de soluções atuais para agregação e formas de visualização analítica de agrupamento deste tipo de dados. Também como trabalho relacionado, é analisado o protótipo anterior do sistema, e apresentadas algumas das suas limitações.

No Capítulo 3 é relatado todo o processo de evolução do sistema de análise de trajeteto com rosas dos ventos, incluindo o fluxo de trabalho, tecnologias e instrumentos utilizados e respetiva explicação dos vários componentes que constituem o novo protótipo. Depois, apresentam-se cenários de utilização com as novas funcionalidades adicionadas ao sistema.

O Capítulo 4 corresponde à avaliação do protótipo, onde foram realizados testes com utilizadores, a partir de uma aplicação criada para o efeito, sendo apresentados os resultados e respetivas conclusões.

Por fim, o Capítulo 5 descreve as principais contribuições do trabalho, as competências adquiridas, dificuldades encontradas e ideias para trabalho futuro.



## Capítulo 2

### Conceitos e trabalho relacionado

Neste capítulo são apresentados os conceitos teóricos relacionados com o sistema desenvolvido incluindo a rosa dos ventos, o tipo de dados envolvido neste projeto e a sua visualização em mapas. Adicionalmente, são analisados alguns estudos de mobilidade urbana e o tipo de problemas que os analistas pretendem resolver, no âmbito do seu trabalho de análise de trajetos. Também são apresentados os tipos de técnicas de agregação de dados, e mencionadas as aplicações anteriormente desenvolvidas de análise de trajetos com rosas dos ventos.

#### 2.1 Conceitos

Nesta secção são apresentados os conceitos base a serem utilizados no desenvolvimento do trabalho, incluindo os trajetos, a rosa dos ventos, a agregação de dados, a visualização analítica de dados, os serviços *web* e o tipo de análise que os peritos nos estudos de mobilidade urbana pretendem realizar.

##### 2.1.1 Rosa dos ventos

A primeira utilização da rosa dos ventos remete para o século XIV, por altura dos descobrimentos portugueses, e surgiu da necessidade de indicar exatamente uma direção, para ilustrar mapas cartográficos, tendo por base a direção dos principais ventos sentidos no Mar Mediterrâneo (Brasseur, 2005; Wallis, 1987). As rosas dos ventos são figuras circulares radiais que agregam dados dos ventos, muito utilizadas na área da meteorologia para visualização da velocidade, frequência e direção de ventos, de um local (Crutcher, 1957).

A Figura 2.1 mostra uma rosa dos ventos com os ventos dominantes no aeroporto da ilha do Pico, sendo apresentados os dados, ao longo de um ano (Meteoblue, 2016). A partir da rosa, podemos observar, que existe um conjunto de círculos concêntricos, que partilham o mesmo ponto central. A cada circunferência está associada uma escala de ocorrência, em horas anuais, que começa nas 0 horas, o ponto central e termina nas 1000 horas, a circunferência mais exterior.

As direções dos ventos são obtidas a partir de divisões dos vários círculos, em ângulos de 90°, no sentido horário. Cada secção corresponde aos 4 pontos cardeais: 0° corresponde à direção Norte (N), 90° a Este (E), 180° a Sul e, finalmente, os 270° a oeste (W).

Cada pétala da rosa representa a direção do vento, que está associada à sua respetiva posição no círculo, que pode ser um ponto cardeal, colateral ou subcolateral. Podemos identificar na Figura 2.1, o ponto cardeal Sul (S), o seu colateral, o Sudoeste (SW) ou o seu subcolateral, o Su-Sudoeste (SSW). O tamanho da pétala indica o número de dados que foram agregados. Ao analisarmos, a Figura 2.1 verificamos que a maior pétala é na direção do Sudoeste, sendo esta a que agrega o maior conjunto de ventos, com ocorrências superiores a 886 horas anuais. Dentro desta pétala existem seis divisões com cores diferen-

tes, para diferenciar as velocidades dos vários ventos agregados. Verifica-se, que a maioria das velocidades ocorreu, entre os 19 e os 28 Km/h. A escala de cores das pétalas está indicada no rodapé da figura e começa em 0 Km/h até valores superiores a 61 Km/h.

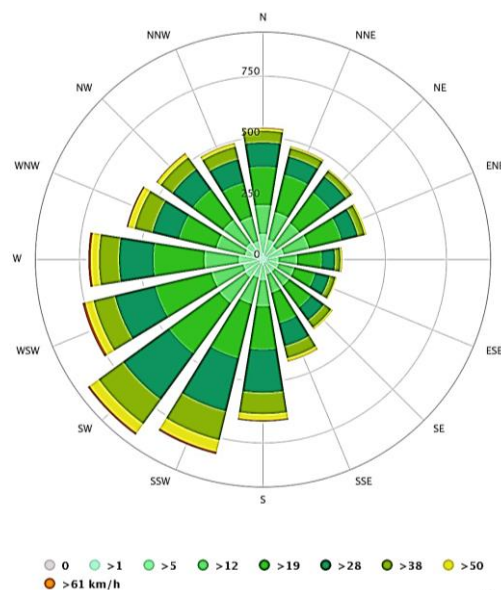


Figura 2.1 – Rosa dos ventos do aeroporto da ilha do Pico, nos Açores agregando ventos registados ao longo de um ano (Metcobluc, 2016).

Na área do planeamento de infraestruturas, os aeroportos necessitam de um estudo cuidado na conceção de uma pista, devendo por isso, ser definida a melhor orientação da pista aterragem/descolagem de um local, com os ventos dominantes. O sistema ARO usa a analogia da rosa dos ventos para processar grandes conjuntos de dados de ventos e desta forma facilitar a execução deste tipo de tarefa (Jia, Chung, Huang, Petrilli, & The, 2004).

Com objetivo idêntico ao trabalho anterior, foi realizado um estudo do escoamento do ar no aeroporto da Ilha do Pico, uma vez que este está sujeito a fenómenos de turbulência na aproximação e aterragem das aeronaves. Com os dados representados nas rosas dos ventos foi possível identificar fenómenos característicos, que contribuiu para identificar padrões e mapear possíveis zonas mais problemáticas. A partir deste estudo foi possível criar novos procedimentos de aproximação para partidas e chegadas, melhorando significativamente a operacionalidade do aeroporto, uma vez que a pista é, penalizada por ventos do quadrante sudoeste, que é dominante (Medeiros, 2009).

Outra aplicação das rosas, corresponde ao suporte visual para a interpretação dos dados, como o diagrama em rosa de Nightingale, ou também conhecido por diagrama de área polar. Florence Nightingale criou este diagrama em rosa, para representar graficamente a mortalidade, durante a guerra da Crimeia (Brasseur, 2005; Magnello, 2012). Este diagrama é composto por 12 setores radiais a partir de um ponto central, como as pétalas de uma flor, que correspondem aos meses do ano. Cada setor é subdividido em segmentos associados às causas principais de mortalidade. Esta representação ajudou Nightingale, a persuadir o governo britânico a instituir reformas para melhoria das condições sanitárias do exército.

No contexto, dos sistemas de transporte para resolução de problemas de congestionamento de tráfego urbano foi utilizada a representação em rosa para dar um visão geral espaço-temporal da distribuição dos locais de paragem dos autocarros, na cidade de Helsínquia, na Finlândia (Bak, Packer, Ship, & Dotan, 2012).

A informação apresentada na rosa permite deste modo sustentar e aumentar o conhecimento e é um meio de apoio na tomada de decisões. É também possível aplicar esta a outros tipos de dados, sem ser o vento.

### 2.1.2 Trajetos, dados GPX e mapas

De seguida é descrito o tipo de dados utilizado no sistema desenvolvido, bem como o tipo de sistemas e mapas para visualizar este tipo de dados. É ainda descrito um formato que permite armazenar dados de trajetos e explicada a sua representação através da rosa dos ventos.

#### Trajetos

Na literatura, um trajeto (trajetória) consiste numa sequência temporal de posições espaço-temporais, isto é, um pares (instante, ponto) guardados durante o movimento de um objeto.

Os trajetos constituem o tipo de dados analisado pelo sistema desenvolvido. Cada trajeto é composto por um ou mais segmentos que, por sua vez, agregam pontos que tem associados a latitude, longitude, elevação, velocidade instantânea, instante de registo e sequência de ocorrência. Foram definidos dois tipos de abordagem para as trajetórias: vista trajetória orientada (*trajectory-oriented view*) para conjunto de trajetórias de todas as entidades e vista de tráfego orientada (*traffic-oriented view*) para sequências temporais de situações de tráfego (Andrienko & Andrienko, 2008).

#### Dados GPX

O *GPX (GPS Exchange Format)* é um formato *XML* usado para registar trajetos através de aplicações que usam o sinal GPS. Tipicamente, no GPX existe uma hierarquia de níveis que permite descrever os percursos que nele estão contidos. A separação do conteúdo é feita através de marcas. A primeira é a marca de meta dados que contém a descrição geral de todo o ficheiro (nome, tempo de criação, programa). Segue-se a marca *track (trk)*, que corresponde ao trajeto registado, com a sua descrição (*desc*) e o nome da *track (name)*. Cada *track* inclui segmentos (*trkseg*), que representam troços da *track*. Por fim, cada segmento é constituído por pontos (*trkpt*), com atributos como a latitude, longitude, elevação, velocidade instantânea e instante de registo. A existência de cada marca *trk*, *trkseg* e *trkpt*, estabelece-se uma hierarquia dentro de cada trajeto, permitindo acesso aos elementos do trajeto.

#### Mapas

Uma visualização comum de um trajeto consiste em unir com uma linha pontos geográficos registados sucessivamente por um dispositivo GPS (*Global Positioning System*), sendo estas linhas representadas num mapa. Esta é a abordagem dos serviços de mapas da *Google* ou da *Microsoft*. Também pode ser encontrada em aplicações móveis, como o *Geo Tracker* para *Android* (Ilya Bogdanovich, 2016), que permite a qualquer utilizador registar os seus trajetos a partir do GPS do seu telemóvel e visualizá-los mais tarde com os mapas da *Google*. A Figura 2.2 mostra um exemplo de um percurso e a sua representação através desta aplicação móvel.

Apesar de ser fácil visualizar trajetos num mapa, quando estes são em elevado número ou se sobrepõem torna-se complexo realizar análise e é comum solucionar este problema através da realização de agregação de trajetos. Um exemplo de solução usada corresponde a agrupar trajetos com percursos similares num único traço destacado (Andrienko & Andrienko, 2008).

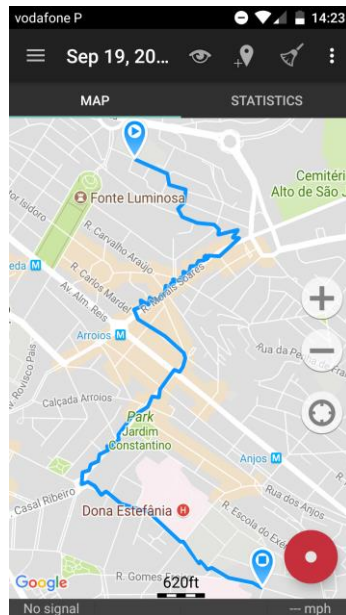


Figura 2.2 – Visualização de um percurso registrado através da aplicação *Geo Tracker* (Ilya Bogdanovich, 2016).

Outro tipo de técnica de visualização de trajetos é através de mapas animados que descrevem trajetórias através de pontos e segmentos (Seraj, Meratnia, & Havinga, 2017).

Os estudos até agora realizados pretendem perceber a forma como as pessoas se movimentam dentro de uma cidade, através de mapas que permitam visualizar dados agregados que acompanham o estado do trânsito ao longo de um dia. Na Figura 2.3 estão representadas as zonas de tráfego, na cidade de Pequim (J. Zhang et al., 2017), que mostram as zonas da região com diferentes fluxos de passageiros. É visível que o maior congestionamento de tráfego ocorre nas principais entradas da cidade, sendo que no centro da cidade o fluxo de passageiros é menor. Apesar de ser possível obter uma visão geral do trânsito, esta visualização não permite a obtenção de informação adicional dos trajetos descritos, nomeadamente a origem e o destino dos veículos que circulam numa determinada zona. Ainda assim, com esta representação é possível obter alguma noção dos padrões de movimento dos veículos, através de sucessivas visualizações da animação. No entanto, este procedimento está sujeito a erros de interpretação e acaba por ser moroso.

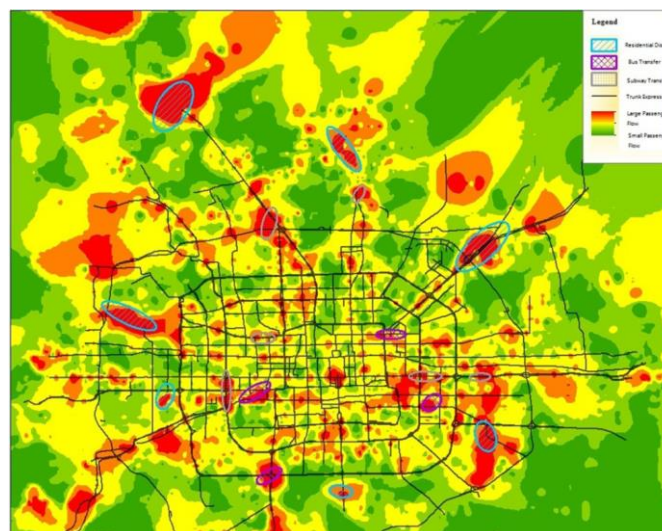


Figura 2.3 – Representação visual do estado do trânsito em Pequim, numa hora de pico, num dia de semana, entre as 7 e 8 da manhã (J. Zhang et al., 2017).

## **Adaptação da rosa dos ventos a trajetos**

A análise dos trajetos pode não ser feita diretamente no mapa, mas sim numa estrutura própria, como a rosa dos ventos, desenhada sobre o mapa e trajetos. Esta possibilidade, até agora pouco explorada, permite que sejam analisadas as mesmas variáveis associadas à representação de ventos, como a direção e a velocidade, mas adaptadas aos trajetos que passam num determinado ponto do mapa. Existe ainda a possibilidade de acrescentar informação com gráficos aos elementos que compõem a rosa, por exemplo com a criação de setas em torno da rosa que indicam as direções de origem e destino e a velocidade média dos trajetos.

### **2.1.3 Agregação de dados**

O processamento dos dados relativo a trajetos permite criar relatórios de dados agregados que podem ser úteis a decisores, nomeadamente quando pretendem tomar a melhor decisão para o negócio de uma organização. Deste modo, é muito importante que a informação agregada presente nestes relatórios seja de fácil perceção, análise e conclusão.

Existe um conjunto de técnicas de prospeção de dados de trajetos que pode ser dividido em categorias, tais como: pré-processamento, indexação, descoberta de padrões e tendências, entre outras. Estas técnicas são abordadas num artigo que pretende retratar uma visão geral do paradigma da análise de trajetos (Andrienko & Andrienko, 2010). As técnicas de pré-processamento de trajetos podem ser por aplicação de filtros de ruído (para exclusão de pontos erróneos), a segmentação (para diferenciação de trajetos distintos) e a deteção de paragens (para anular oscilações indesejadas nos trajetos). Estas técnicas foram utilizadas no sistema desenvolvido no âmbito desta dissertação e permitiram melhorar a qualidade do conjunto de trajetos. É ainda importante referir que, embora o artigo em questão apresente um vasto conjunto de técnicas de prospeção de trajetos, não existe qualquer referência alusiva a rosas dos ventos, sendo esta a principal técnica de agregação do sistema desenvolvido.

O conceito de agregação é importante na análise de trajetos, uma vez que facilita a compreensão de grandes conjuntos de dados deste tipo, permitindo descobrir aspetos que não são evidentes com a simples visualização num mapa, como é o caso da movimentação pendular de veículos numa determinada zona.

### **2.1.4 Visualização analítica de dados**

Um dos desafios associados à visualização analítica é a representação de grandes quantidades de dados que ajudem os humanos a perceber e interpretar a informação apresentada (Eppler & Mengis, 2004). Com o desenvolvimento de sistemas inteligentes de transporte (*Intelligent Transportation Systems*) os investigadores deparam-se com este problema e um dos maiores desafios é encontrar a melhor forma para representar, analisar e visualizar estes mesmos dados (J. Zhang et al., 2017).

Deste modo, surge a necessidade de aplicar métodos de agregação, filtragem e sumarização dos dados que permitam descobrir comportamentos, tendências e padrões (Andrienko & Andrienko, 2010). A representação visual gráfica dos dados agregados é essencial para um analista humano compreender os dados (perceção) e extrair a informação relevante para o seu conhecimento (cognição) (McCormick, DeFanti, & Brown, 1987). A interação com a representação gráfica também deve ser tida em consideração na idealização e suporte na tomada de decisões importantes (Thomas & Cook, 2006). Contudo, para compreender como utilizar os dados de um movimento e antes de fazer uma análise computacional, um analista necessita primeiro de olhar para os dados e explorá-los, a partir de uma representação visual apropriada dos mesmos.

No trabalho desenvolvido neste projeto a componente interativa está contemplada com a possibilidade de criar rosas dos ventos em quaisquer pontos do mapa, e a possibilidade de utilizar filtros de espaço e tempo que irão facilitar o trabalho dos analistas.

### **2.1.5 Trabalho dos analistas**

A análise de trajetos é extremamente útil para determinar congestionamento de trânsito e movimentos pendulares de uma grande cidade. Como exemplo, temos o estudo realizado, do sistema de transportes da cidade de Brasília, que descreve os movimentos de casa para o trabalho de manhã e de trabalho para casa, à noite. (Izawa et al., 2017). Este permite identificar, no mapa, quais os horas críticas de operação de tráfego e prevê qual a melhor altura para viajar para uma região particular da cidade. Em outro estudo é analisado o fluxo de passageiros de autocarros, em Pequim, ao longo de um dia, usando um método espaço-temporal com mapas animados que identificam locais críticos (congestionamento do trânsito). A partir deste, o decisor pode analisar e reorganizar o tráfego de passageiros da cidade, adequando o número de autocarros e o número de paragens de um circuito (J. Zhang et al., 2017). Outro motivo por trás dos movimentos das pessoas relaciona-se com determinados propósitos/atividades, em diferentes locais (por exemplo, escola, escritório, centro comercial, etc.). À medida que as atividades de comportamento humano são detetadas nestes dados espaço-temporais podemos estudar o comportamento humano e, em seguida, organizar os recursos/redes de transporte da cidade (Zeng et al., 2017).

Existem, portanto, dois tipos abordagens no estudo da mobilidade nos sistemas de transporte:

- Planeamento urbano: foca-se em viagens de estimativa de padrão de mobilidade usando dados de aplicações móveis. Estes dados ajudam os investigadores e os designers a desenhar redes de transportes públicos, estimar tempos de viagem de uma rota ou modelar rotas.
- Operações de Transporte: foca-se nos sistemas de apoio à decisão para operações de tráfego. Por exemplo, prever tempo de viagem, deteção de anomalias e incidentes de tráfego, cargas de congestionamento dinâmico (Amini, Gerostathopoulos, & Prehofer, 2017).

Os gestores de tráfego de cidades pretendem ver analisadas as seguintes tarefas (Andrienko & Andrienko, 2008):

1. Estimar a média de fluxos (número de passageiros) entre regiões de interesse e a sua variação em diferentes períodos de tempo e na presença de eventos extraordinários (greves, jogos de futebol);
2. Estimar o tempo de viagem entre regiões e suas variações;
3. Estimar o impedimento de uma rua (obstrução ao movimento) e a sua variação;
4. Estimar a proporção de carros saindo de uma estrada principal com diferentes saídas;
5. Compreender os percursos atuais utilizados pelas pessoas a partir de um ponto ou região de interesse para outro.

As tarefas 1, 2 e 5 são mais adequadas para vistas orientadas ao tráfego, porque denotam posições do movimento de todas as entidades com atributos como a velocidade, direção e aceleração. As tarefas 3 e 4 são mais orientadas para vistas de trajetórias orientadas quando o interesse é apenas a origem e o destino das viagens.

Atualmente os analistas não utilizam dados dos movimentos de pessoas e veículos, uma vez que não existem ferramentas adequadas que permitam essa análise e apenas utilizam estatísticas descritivas sem uma análise profunda sobre a influência de fatores como o tempo, velocidade, distância, etc. Por exemplo, a maioria dos estudos concentra-se apenas em horários de pico ou mesmo dados diários agregados

sem divisão horária, o que pode levar a perda repentina das horas de pico e não reflete a volatilidade horária do sistema de trânsito ou negligência a diferença temporal. Também, por vezes, a análise adicional apenas tem em conta certos indicadores diretos, como a velocidade e o tempo de viagem, enquanto que os indicadores indiretos como o congestionamento do tráfego, tendem a ser desprezados devido à menor atenção e dificuldade em obter quantidades fixas (K. Zhang, Sun, Shen, & Zhu, 2017).

### 2.1.6 Serviços *web*

A evolução da aplicação *web* referida nos objetivos deste trabalho, pode envolver a alteração dos serviços *web*. Segundo o W3C (*World Wide Web Consortium*), um serviço *web* é um software projetado para desempenhar uma dada tarefa, estando acessível a um vasto conjunto de clientes, sendo que estes conseguem interagir com o serviço através de um formato definido por um *Web Service Description Language* (WSDL) ou uma *Application Programming Interface* (API) (“Web services architecture”, 2004).

Existem dois protocolos principais de serviços *web*:

- SOAP (*Simple Object Access Protocol*) – baseia-se numa invocação remota de um determinado método (funcionalidade oferecida pelo serviço). É necessário especificar o endereço do serviço, o nome do método e os argumentos que devem ser passados ao mesmo. Todos estes dados são formatados em XML, e passados para o serviço por HTTP (*Hypertext Transfer Protocol*), que é um protocolo de comunicação aplicacional. A maior parte das linguagens de programação consegue gerar as mensagens SOAP de forma simplificada, o que facilita a utilização do serviço. A sua principal desvantagem é a sobrecarga associada ao processamento das mensagens (serialização/desserialização), o que torna este tipo de serviço mais lento que os restantes.
- REST (*Representational State Transfer*) – é um protocolo de troca de mensagens baseado em HTTP. Porém, o REST permite alterar a formatação de mensagens, mas tem de ser especificado o comportamento do serviço em causa. O formato das mensagens é definido pelo programador. Os serviços *web* com REST são mais rápidos na execução, mas a sua principal desvantagem é a liberdade de especificação dos formatos de mensagem, que pode conduzir a dificuldades de comunicação entre vários sistemas, uma vez que não é seguido um formato padrão.

Os serviços *web* desenvolvidos são importantes no tratamento e processamento dos dados dos trajetos analisados pelo sistema, pois permite aliviar a carga de processamento doutros componentes do sistema.

## 2.2 Aplicações existentes

Nesta secção são apresentados exemplos de aplicações de visualização analítica que utilizam trajetos, sendo feita uma comparação com as funcionalidades atuais e uma pequena descrição do protótipo do sistema de análise de trajetos com rosas dos ventos, desenvolvido num trabalho anterior e apresentadas algumas das suas limitações atuais.

### 2.2.1 Aplicações de visualização analítica

As técnicas de representação de trajetos mais comuns passam pela utilização de uma camada visual, sobreposta num mapa, em que são traçados trajetos ou pontos. Para os mapas animados são utilizadas estas técnicas para representação de dados sobre um globo terrestre 3D. Contudo a maioria das abordagens na construção dos mapas são estáticas, sendo que a maior parte se limita à representação de dados

estatísticos, por exemplo: o *Birdeye* (“*Birdeye information visualization and visual analytics library*”) e o *Google Vis* (“*Visualization API Reference*”). Nenhuma das técnicas/abordagens anteriores permite a representação de agregações de trajetos com rosas dos ventos.

Existem aplicações que oferecerem suporte a métodos de análise automática de dados (algoritmos de agrupamento e classificação), mas nenhum permite análise com trajetos e requerem o pagamento de licenças, o que é uma grande limitação. Contudo, existe a aplicação WRPLOT que é gratuita e permite construir rosas a partir dados de ventos (direção, velocidade, data de ocorrência e localização geográfica). Nesta é possível efetuar agregações, aplicar filtros e personalizar de aspetos gráficos da rosa, como a escala de cores para a velocidade dos ventos e o número de setores da grelha da rosa (Lakes Environmental, 2016). O formato das rosas é o *KML* (*Keyhole Markup Language*) e podem ser visualizadas em diferentes aplicações, como por exemplo no *Google Maps* (“*Google Maps*”). A desvantagem é a impossibilidade de utilizar a rosa, com outros tipos de dados, que no caso do trabalho desenvolvido neste projeto são os trajetos.

## 2.2.2 Protótipo anterior do sistema de análise de trajetos

No âmbito de um projeto de mestrado (Vaz, 2017) foi desenvolvido um sistema constituído por uma biblioteca e uma aplicação *web* que disponibilizam um conjunto de funcionalidades que permitem a visualização de agregações de trajetos, tendo como base a representação da rosa dos ventos. Este trabalho é o ponto de partida para o desenvolvimento deste novo projeto.

Os vários componentes incluem a base de dados geográfica, os serviços *web*, a biblioteca da rosa dos ventos e as aplicações de análise e gestão de trajetos, juntamente com bibliotecas externas, que estão representados na Figura 2.4.

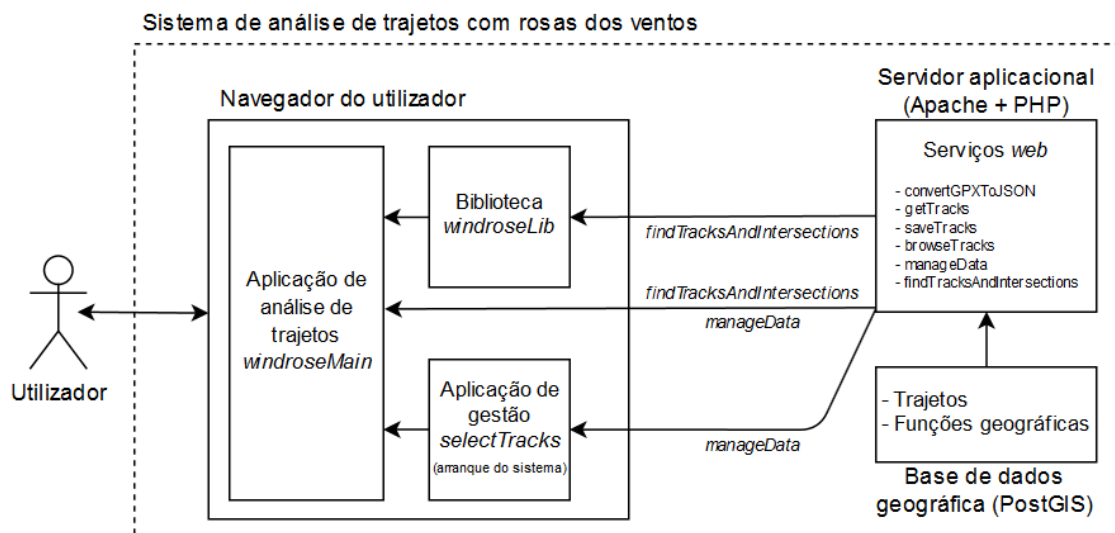


Figura 2.4 – Arquitetura do Sistema de Análise de Trajetos com Rosas dos Ventos (Vaz, 2017).

No lado esquerdo do diagrama está representado o navegador *web* que permite ao utilizador o acesso a duas aplicações distintas: à aplicação de gestão de trajetos *selectTracks*, que serve de ferramenta de arranque do sistema, e à aplicação de análise de trajetos *windroseMain*. A biblioteca da rosa dos ventos *windroseLib* é também executada a partir do navegador através da aplicação de análise de trajetos.

No canto superior direito encontra-se o servidor aplicativo Apache, onde são executados todos os serviços *web*. Estes serviços, para além de auxiliarem a manipulação de trajetos e permitirem a comunicação entre as aplicações executadas no navegador *web*, servem de ponto de ligação entre o navegador *web* e a base de dados geográfica, permitindo inserir e consultar trajetos guardados na mesma.

No canto inferior direito do diagrama está visível a base de dados geográfica, responsável por persistir toda a informação dos trajetos. Para além do armazenamento dos dados, é disponibilizado um conjunto de funções geográficas que facilitam a manipulação da informação.

### Base de dados geográfica

O *PostGIS* é um sistema de gestão de base de dados geográfica que possui um conjunto de ferramentas que facilitam a manipulação deste tipo de dados, como funções geográficas predefinidas que melhoraram a descoberta dos trajetos que cruzam a área de agregação da rosa.

### Serviços *web*

Os serviços *web* servem de apoio à aplicação *web*, oferecendo funcionalidades, tais como, a leitura de trajetos de ficheiros *GPX*, carregamento de dados de trajetos para a base de dados geográfica, e consulta dos mesmos. Permite conversão de *GPX* para *JSON*. Uma das limitações deste protótipo refere-se a uma maior complexidade na construção de sistemas dos serviços *web* e que torna o processamento de dados mais lento.

### Intermediário de trajetos entre as aplicações de gestão e análise de trajetos

Para que seja possível enviar o conteúdo de trajetos entre a aplicação de gestão e a aplicação de análise de trajetos, existe um serviço intermediário *manageData*, que guarda temporariamente trajetos provenientes da aplicação *selectTracks*, de forma a que estes possam ser lidos pela aplicação *windroseMain*. Uma das limitações deste protótipo corresponde a este passo intermédio, uma vez que é sempre necessário carregar o ficheiro com os dados dos trajetos provenientes do *selectTracks* para carregar a aplicação principal de análise de trajetos, como pode ser observado na Figura 2.5 a tracejado. Idealmente, este processo deveria ser automático e incluído na aplicação de gestão *selectTracks*.

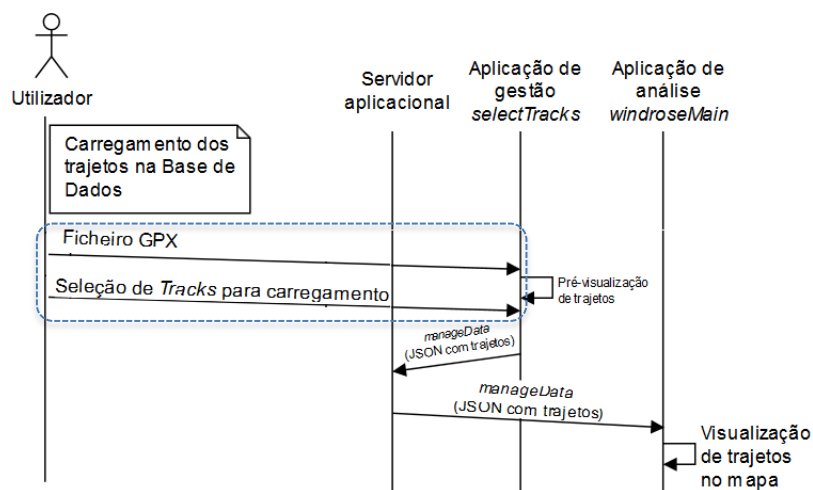


Figura 2.5 – Cenário de criação de um ficheiro GPX com trajetos para a aplicação de análise de trajetos do sistema (Vaz, 2017).

### Biblioteca da rosa dos ventos

A biblioteca de funções do sistema de análise de trajetos com rosas dos ventos, *windroseLib*, cria os elementos gráficos das rosas dos ventos e permite, de uma forma simples, a construção de rosas com

dados agregados de trajetos. A rosa criada pela biblioteca agrega dados de trajetos que estão contidos num círculo com um determinado raio (em metros) que pode ser modificado, possibilitando uma maior agregação de trajetos. Cada pétala presente na rosa representa um conjunto de trajetos com uma determinada direção. O tamanho das pétalas está diretamente relacionado com o número de trajetos que estas abrangem relativamente ao total de trajetos. Existe ainda uma escala de cores que informa sobre a velocidade média dos trajetos agregados por cada pétala. É ainda possível visualizar a origem e o destino dos vários percursos agregados, através de setas colocadas em torno do rebordo da rosa.

### Aplicação *web* de análise de trajetos

A aplicação de análise de trajetos, *windroseMain*, é a principal componente do sistema do ponto de vista do utilizador, comportando-se como um cliente da biblioteca da rosa dos ventos e dos serviços *web*, tirando partido da base de dados geográfica.

A aplicação *web* permite num contexto espacial representar os dados na rosa gerada pela biblioteca. Recorre à *API* do *Google Maps* para gerar o mapa onde são apresentados todos os trajetos presentes num ficheiro *GPX*, bem como elementos gráficos desenhados pelas funções da biblioteca da rosa. Cada trajeto é um objeto clicável, que oferece a possibilidade de criação de uma rosa desde que sobre um ponto de qualquer um dos percursos visíveis.

Um dos contributos e aspetos interessantes deste trabalho é a possibilidade de efetuar análises aos trajetos, dentro da área de agregação da rosa, como pode ser observado na Figura 2.6 onde ainda é possível a aplicação de filtros a várias rosas em simultâneo. Esta análise deverá conter dados sobre a secção do trajeto dentro da área de agregação, como por exemplo, o tempo decorrido desde que entrou até ter saído.

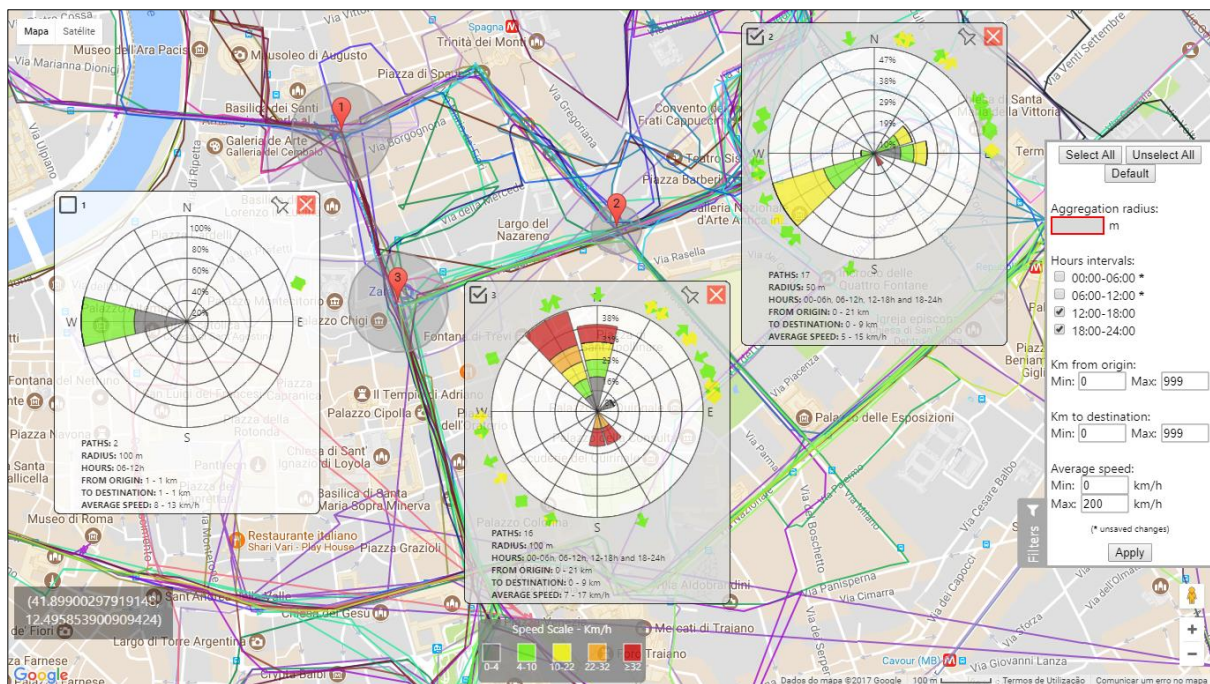


Figura 2.6 – Exemplo de funcionamento da aplicação do sistema de análise de trajetos, com uma rosa dos ventos (Vaz, 2017).

Para diferenciar a distribuição de responsabilidades entre a biblioteca e a aplicação *web*, apresenta-se o seguinte diagrama de sequência da Figura 2.7 que mostra o fluxo de execução da aplicação *web*, desde que é criado um marcador e as invocações das funções da biblioteca da rosa dos ventos até ao desenho da rosa junto ao marcador.

De notar que com este protótipo, atualmente não é possível atualizar o objeto *WindRose* (rosa criada) se o utilizador pretender por exemplo, mover o marcador para uma nova posição. Isto torna a utilização pouco prática do ponto de vista de um analista que pretenda fazer uma análise exploratória dos trajetos, uma vez que tem de estar sempre a remover e criar novas rosas. Outra situação, corresponde à análise de movimentos pendulares ao longo de um dia para estudar o comportamento dos trajetos no sentido casa-trabalho e trabalho-casa, que neste momento não é possível. De facto não existe a possibilidade de criar várias rosas associadas ao mesmo ponto do marcador associadas a diferentes períodos do dia.

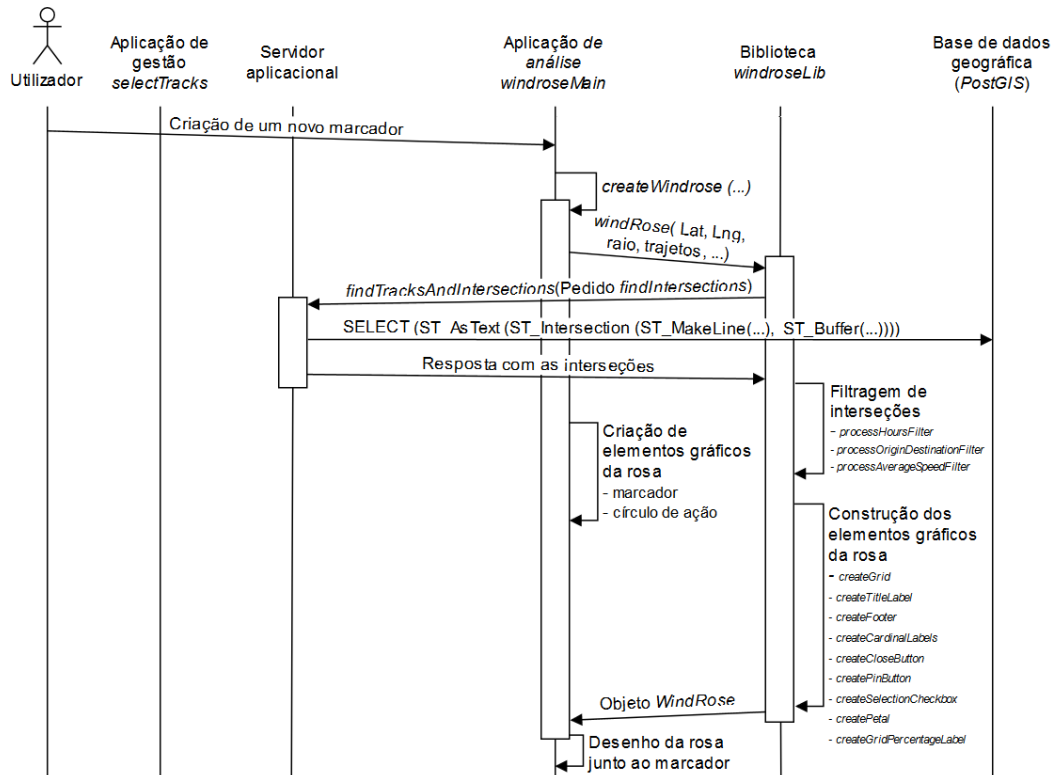


Figura 2.7 – Cenário de criação de marcador/rosa com a distribuição das responsabilidades entre a biblioteca da rosa e da aplicação de análise de trajetos do sistema (Vaz, 2017).

Outra limitação atual corresponde aos filtros aplicados aos trajetos com valores pré-definidos. Não existe a possibilidade de ajustar os valores desses mesmos parâmetros, como por exemplo, intervalos de horas personalizados.

Relativamente à interface gráfica podem ser melhorados alguns aspetos, nomeadamente a densidade e sobreposição de setas de origem e destino, quando existe uma quantidade elevada de trajetos que intersejam a área de agregação da rosa. Podem ser igualmente feitas algumas alterações na disposição do painel de filtros, de forma a que o ajuste de determinados parâmetros possa ser feito de uma forma mais sugestiva, como por exemplo, a inclusão de campos editáveis com um limite mínimo e máximo para os campos de velocidade média e distância à origem e destino.

## 2.3 Sumário

Neste capítulo foram explicados os conceitos e dado a conhecer trabalhos relacionados com o sistema desenvolvido. Foram também apresentados tipos de aplicações que usam rosas dos ventos, exemplos de sistemas com capacidade de análise de trajetos e explorado o tipo de trabalho que é realizado pelos analistas, nomeadamente que tipo de informação é importante no apoio à tomada de decisão. Foi também

realizada uma análise geral do protótipo anterior do sistema, que serviu de base para a evolução do sistema atual que está descrito neste relatório.

## Capítulo 3

# Desenvolvimento e demonstração do sistema

Neste capítulo apresenta-se o trabalho realizado, desde o tipo de tecnologias e ferramentas analisadas, assim como a discussão de possíveis soluções para cenários de determinadas tarefas e funcionalidades. É caracterizado o processo de desenvolvimento do protótipo, desde a definição de requisitos, concepção e evolução da arquitetura e desenho com respectiva codificação das novas funcionalidades. Por último, é feita uma descrição dos cenários de utilização das novas funcionalidades que permitem os utilizadores interagirem com o novo sistema de análise de trajetos com rosas dos ventos e sua aplicação do ponto de vista de um decisor.

### 3.1 Ferramentas e processo de desenvolvimento

O trabalho foi realizado num computador portátil com um processador *Intel Core i7 @ 2.0 GHz*, 12 GB de memória *RAM* e sistema operativo *Windows 10 Home x64*, nos laboratórios do LaSIGE.

O processo de desenvolvimento deste projeto consistiu em reuniões semanais, com a presença dos orientadores e do aluno. Em cada reunião foram discutidos aspetos sobre as novas funcionalidades de acordo com os objetivos do trabalho, tipo de ferramentas e tecnologias a utilizar (bibliotecas e *API*), sendo feito o ponto de situação em cada sessão. Depois de concretizadas as funcionalidades, estas foram testadas de forma a corrigir pequenos erros ou para permitir outros cenários de utilização que não tinham sido previamente definidos.

As ferramentas e tecnologias utilizadas são consequência dos conhecimentos adquiridos durante a licenciatura, a componente curricular do mestrado, o trabalho desenvolvido no protótipo anterior e estudo de novas tecnologias *open source*.

Nas próximas seções são apresentados os elementos principais que compõem o novo sistema e as decisões que foram tomadas durante a sua concretização: a base de dados geográfica; a definição de serviços *web* com *Slim Framework* (“*Slim a micro framework for PHP*”); a inclusão de novas funcionalidades na biblioteca da rosa; criação de uma aplicação para interagir com as diferentes funcionalidades da biblioteca na análise de trajetos urbanos num mapa de uma cidade.

Os dados dos percursos inseridos na base de dados correspondem a trajetos de táxis na cidade de Roma, em formato *GPX*, aquando do primeiro carregamento para a base de dados. Estes são previamente convertidos num formato compatível para esse efeito (*JSON*), a partir de serviços *web*.

Para a camada de dados, utilizou-se o *PostGIS* (“*PostGIS: Spatial and geographic objects for PostgreSQL*” 2016), que é uma extensão do *PostgreSQL*, pois permite realizar pesquisas com dados geográficos de maneira rápida e eficiente.

Usou-se o *XAMPP* (“*XAMPP installers and downloads for Apache Friends*,” 2017) que é um software gratuito que disponibiliza o Apache (Foundation) para execução da aplicação e os serviços *web* numa máquina local, sem ser necessárias muitas configurações. Para facilitar o trabalho de processamento e

construção de serviços *web* foi utilizado o *Slim Framework* que simplifica a escrita de interrogações (chamadas) à base de dados.

Para o auxílio de escrita do código da página cliente da aplicação em *PHP* (Team, 2016), *HTML* (W3C, 2017), *JavaScript* e *JQuery* (“The JS Foundation, «jQuery») utilizou-se a *IDE JetBrains PhpStorm 2018* (JetBrains, 2018) .

## 3.2 Requisitos do sistema

Iniciou-se o projeto com análise da arquitetura do sistema anterior e identificação das suas limitações. Primeiro, decidiu-se modificar a forma como era feito o carregamento dos dados dos trajetos para a aplicação de análise. Agora ao iniciar a aplicação, esta obtém os dados a partir da base de dados e não é preciso estar a inserir o ficheiro base com dados de trajetos.

De seguida foram estipulados quais os requisitos e funcionalidades a implementar na evolução do sistema. Estes tiveram em conta, o contexto do trabalho dos analistas de tráfego urbano, como a deteção de movimentos pendulares. O sistema deverá ser simples e claro na manipulação da sua interface para contribuir na tomada das melhores decisões, por parte dos decisores.

O primeiro requisito do sistema (R1) correspondeu a evoluir a infraestrutura de aplicação dos serviços *web*, de forma a possibilitar a atualização contínua dos dados que chegam à aplicação dos utilizadores. Este tem dois cenários. O primeiro corresponde a mover livremente uma rosa no mapa, com a atualização automática do seu conteúdo. O segundo corresponde a suportar a atualização de novos dados de trajetos que chegam periodicamente ao mapa. Este requisito surge da necessidade de agilizar o processo de análise numa outra localização.

O segundo requisito do sistema (R2) diz respeito à manipulação direta de várias rosas (duplicadas) numa mesma região de agregação de um local do mapa. Assim, tendo duas ou mais rosas associadas a um mesmo ponto (marcador) será possível selecionar diferentes períodos do dia em cada rosa e analisar. Esta situação é relevante do ponto de vista de monitorização para uma análise histórica, muito usada no trabalho dos analistas.

Por fim, o último requisito do sistema (R3) corresponde à concretização de filtros de espaço e tempo mais sofisticados. Por exemplo, ser possível ajustar e definir quais os intervalos de tempo a analisar, no caso das horas de ponta. Estas modificações podem ser feitas individualmente ou no conjunto de várias as rosas, para permitir a análise comparativa de vários pontos no mapa. Também deve ser reforçada a componente interativa dos elementos gráficos da rosa dos ventos, nomeadamente saber a direção do destino de um dos trajetos tendo selecionado a direção de origem.

## 3.3 Sistema de análise de trajetos

Nesta secção descrevem-se os elementos do sistema de análise de trajetos com rosas dos ventos, sendo apresentada a vista global da arquitetura do sistema. O sistema é composto por uma base de dados geográfica, serviços *web* em *framework Slim*, a biblioteca da rosa dos ventos e a aplicação de análise de trajetos. O sistema utiliza ainda algumas bibliotecas externas para o painel e inspetor de filtros, desenho dos trajetos e mapa.

### 3.3.1 Arquitetura do sistema

A arquitetura do sistema é apresentada na Figura 3.1 na qual são visíveis os principais elementos que compõem o protótipo e como se interligam.

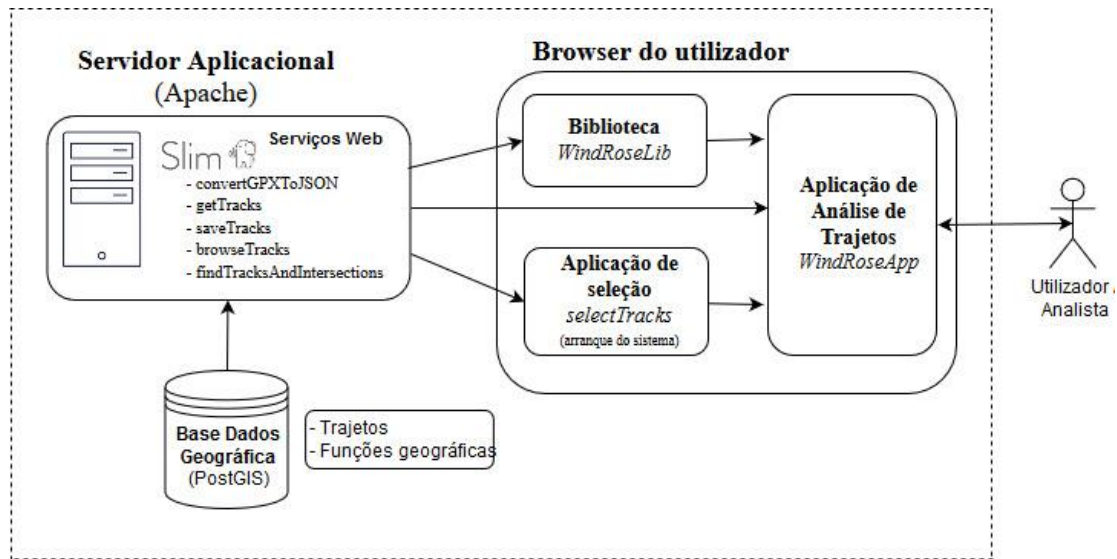


Figura 3.1 – Arquitetura do sistema de análise de trajetos com rosas dos ventos.

No lado direito do esquema está a *browser* do utilizador. Quando este inicia podem ocorrer dois cenários. Se a base de dados geográfica contiver dados, esta carrega a página de análise de trajetos *windRoseApp*. Caso a base de dados não tenha trajetos disponíveis, carrega a aplicação de seleção de trajetos *selectTracks*. A biblioteca da rosa dos ventos *windroseLib* é ativada a partir do *browser* com a aplicação de análise *windRoseApp*.

No lado esquerdo encontra-se o servidor da aplicação *Apache + Slim Framework*, onde são executados todos os serviços *web*. Estes serviços permitem realizar um conjunto de operações com trajetos como obter trajetos da base de dados.

Por último, na base do esquema temos a base dados geográfica que armazena dados de trajetos e disponibiliza um conjunto de funções geográficas para manipular este tipo de dados.

Manteve a base de dados geográfica, mas foi modificado o conceito de trajeto para que este fosse mais genérico. Os serviços *web* foram reformulados com a ferramenta *Slim Framework* que ajuda a codificar aplicações *web* e *APIs* num único local. Antes, os serviços *web* tinham de ser todos escritos em *PHP*, o que tornava o processo moroso e mais difícil.

Outra importante alteração correspondeu ao carregamento inicial da aplicação, pois era necessário estar sempre a selecionar o ficheiro de trajetos na página da aplicação de seleção para começar a mexer na aplicação. Agora se a base de dados já conter trajetos, esta carrega automaticamente a aplicação de análise de trajetos.

De seguida, descrevem-se os restantes elementos do sistema de análise (Figura 3.1). Primeiro as alterações que foram realizadas na base de dados geográfica, depois as novas operações dos serviços *web* com a ferramenta *Slim Framework*, de seguida a reestruturação da biblioteca da rosa dos ventos e, termina na aplicação de análise de trajetos.

### 3.3.2 Base de dados geográfica

Com o objetivo de resolver os possíveis problemas relacionados com o processamento de grandes quantidades de dados geográficos foi explorada uma alternativa, as bases de dados não relacionais – *NoSQL*.

A base dados explorada, foi o *Neo4J-Spatial*, baseado em grafos, que permite algumas operações espaciais de dados geográficos armazenados e otimização de consultas espaciais (Baas, 2012). Contudo, o *Neo4J-Spatial* ainda não possui um conjunto de funcionalidades espaciais tão completo como o *PostGIS* e as funcionalidades só poderem ser acedidas externamente à interface *Neo4j Browser* através da linguagem *Java* ou *REST* enquanto que no *PostGIS* permite o acesso às suas funcionalidades, através de uma mesma interface. Outra limitação refere-se às poucas referências do *Neo4J-Spatial*, o que dificulta o trabalho de configuração de uma base de dados geográfica orientada a grafos, tornando o processo moroso. Além disso, o suporte técnico nem sempre se encontra completo ou atualizado e a comunidade de utilizadores ainda é reduzida (Roberto, Holanda, & Victorino, 2017).

Deste modo, dadas as limitações anteriores optou-se por manter o *PostGIS* (“PostGIS: Spatial and geographic objects for PostgreSQL,” 2016).

A etapa seguinte correspondeu à alteração das tabelas da base de dados geográfica *PostGIS*. Foi reformulado o modelo de dados anterior para generalizar os elementos presentes no conceito de um trajeto. Abaixo, apresenta-se na Figura 3.2 os seguintes conceitos: *Entity*, *Track* e *Point* que correspondem às tabelas da base de dados.

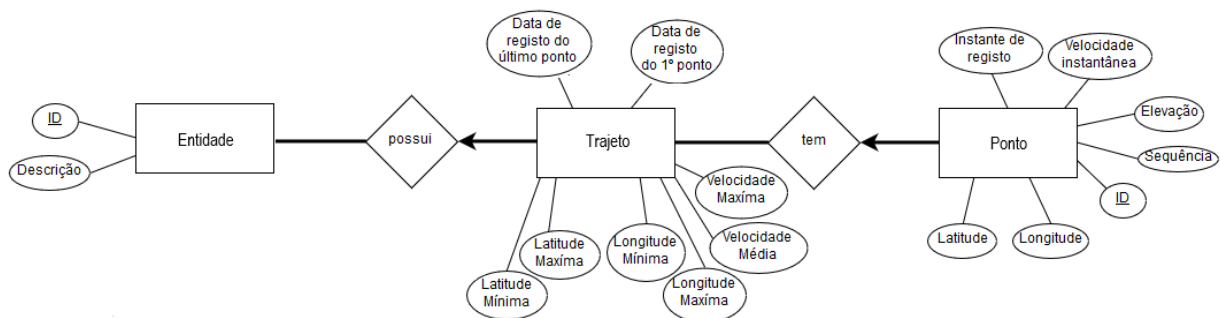


Figura 3.2 – Modelo EA de dados de um trajeto.

Uma *Entity* é composta por dois atributos: um identificador único que é chave primária, um nome e uma descrição.

```
CREATE TABLE Entitys (
  id          INT PRIMARY KEY,
  name       VARCHAR(20) ,
  sequence   INT NOT NULL,
  description VARCHAR(100)
);
```

Um *Trajeto* é constituído por diferentes atributos: a latitude e longitude máxima e mínima, a velocidade média e máxima e informações dos pontos iniciais e finais do *Trajeto*. Possui um identificador único e sequência temporal sendo que cada *Trajeto* pertence a uma única *Entidade*. As latitudes e longitudes máximas e mínimas, permitem criar uma *bounding box* que delimita o local onde estão presentes trajetos.

```
CREATE TABLE Tracks (
  id          SERIAL PRIMARY KEY,
  sequence   INT NOT NULL,
  entity_id  INT REFERENCES Entitys(id) NOT NULL,
```

```

firstPointDate      TIMESTAMP NOT NULL,
lastPointDate       TIMESTAMP NOT NULL,
minLatitude         DECIMAL (8,6) NOT NULL,
maxLatitude         DECIMAL (8,6) NOT NULL,
minLongitude        DECIMAL (8,6) NOT NULL,
maxLongitude        DECIMAL (8,6) NOT NULL,
maxSpeed            DECIMAL (9,6) NOT NULL,
averageSpeed        DECIMAL (8,6) NOT NULL
);

```

Um *Ponto* tem no seu conjunto: um identificador único, uma latitude, longitude, elevação, velocidade instantânea, instante de registo e sequência de ocorrência. Cada *Ponto* pertence a um único *Trajetos*.

```

CREATE TABLE Points (
  id          SERIAL PRIMARY KEY,
  sequence    INT NOT NULL,
  track_id    INT REFERENCES Tracks(id) NOT NULL,
  latitude    DECIMAL (8,6) NOT NULL,
  longitude   DECIMAL (8,6) NOT NULL,
  elevation   DECIMAL (4),
  speed       DECIMAL (5,2) NOT NULL,
  time        TIMESTAMP NOT NULL,
  wkb_geometry geometry (Point,4326) NOT NULL
);

```

### 3.3.3 Serviços web

Os serviços *web* foram construídos com a ferramenta *Slim Framework* que é uma microserviço *PHP* que ajuda os programadores a desenvolver aplicações e *APIs* de maneira rápida e fácil. O *Slim* é um *dispatcher* que recebe uma solicitação *HTTP* ou *URLs* específicos, invoca uma rotina de retorno de chamada apropriada e retorna uma resposta *HTTP*. Esta é uma ferramenta ideal para criar *APIs* que consomem, adaptam ou publicam dados e permitem prototipagem rápida, pois tem pouco código.

Para a sua utilização é necessário um servidor *web*, neste caso foi usado o *Apache (XAMPP)*. Foram definidos um conjunto de serviços *web* a serem usados pela aplicação de análise de trajetos *win-dRoseApp*, que se encontram descritos na Tabela 2.

Tabela 2 – Serviços *Web* disponíveis

Serviço web	Descrição
Pesquisa de trajetos (Entities)	Permite ao utilizador pesquisar todos os trajetos da base de dados.
Pesquisa de segmentos (Tracks)	Permite ao utilizador pesquisar todos os segmentos da base de dados.
Pesquisa de pontos (Points)	Permite ao utilizador pesquisar todos os pontos da base de dados.
Pesquisa de as coordenadas de latitudes e longitudes para desenhar trajetos (coordinates)	Permite ao utilizador pesquisar todas as latitudes e longitudes para desenhar os trajetos.
Pesquisa das interseções de um raio de agregação de um trajeto (intersections)	Permite ao utilizador pesquisar todas as latitudes e longitudes para desenhar os trajetos.
Pesquisa de trajetos e interseções	Permite ao utilizador pesquisar todos os trajetos e interseções de uma região

## Pedido e resposta

O *Slim* trabalha diretamente com objetos do tipo pedido/resposta. Esses objetos representam o pedido *HTTP* real recebido pelo servidor da *web* e a respectiva resposta *HTTP* que é devolvida ao cliente (utilizador). Cada rota recebe os objetos *Request* e *Response* atuais como argumentos para sua rotina de retorno de chamada e suporta correspondência de padrões.

De seguida apresenta-se um exemplo de escrita de código em *Slim*:

```
$app = new \Slim\App; // Create App Slim
$app->get('/tracks', function($request, $response, $args) {
    $data = array();
    $query = "select * from tracks order by id";
    $result = pg_exec($query);
    while ($row = pg_fetch_array($result)){
        array_push($data, $row);
    }
    $response = $response->withJson($data);
    return $response;
});
```

Os serviços *web* servem de apoio à aplicação, para obter dados dos trajetos da base de dados. Abaixo na Tabela 3 são descritos os serviços que foram construídos e que são utilizados pelo sistema.

Tabela 3 – URIs de acesso a recursos e métodos HTTP aceites

URI	Método	Formato/ Parâmetro	Efeito
/tracks	GET	-	Devolve todos os trajetos (tracks) da base de dados.
/tracks/{id}	GET	id	Devolve o trajeto (track) com certo id.
/segments	GET		Devolve todos os segmentos (segments) da base de dados.
/segments/{id}	GET	id	Devolve segmento (segment) com certo id.
/points	GET	id	Devolve todos os pontos (points) da base de dados.
/points/{id}	GET	id	Devolve ponto (point) com certo id.
/coordinates	GET		Devolve todas as coordenadas latitudes e longitudes dos pontos na base de dados.
/intersections	GET		devolve todos os trajetos intersetedos numa região de agregação
/intersections/{region_agg_id}	GET	re- gion_agg_id	devolve os trajetos intersetedos da região de agregação id.
/intersections/{region_agg_id}/{radius}	GET	re- gion_agg_id , radius	devolve o raio do círculo central da região de agregação id.

## Pesquisa de trajetos na base de dados

A base de dados já tem disponível um conjunto de funções geográficas que se podem aplicar aos trajetos guardados. Uma dessas funções permite selecionar todos os pontos que estejam contidos num determinado círculo, com centro num par de coordenadas (latitude e longitude) e raio fornecidos pelo utilizador.

Estes serviços *web* com diferentes *URLs* de acesso a métodos *HTTP* com e sem parâmetros fazem interações à base de dados, através do serviço em questão.

Um exemplo de um pedido ao serviço feito através da linha de endereço do browser *web*, pode ser dado como:

`http://localhost/wind-rose/public/intersections/41.9046/12.4787/100` que corresponde a `'/intersections/{latitude}/{longitude}/{radius}'`

No exemplo acima, estão definidos três argumentos: a latitude e longitude do ponto da região de agregação (círculo) e o raio. Os valores são respetivamente 41.9046, 12.4787 e 100 (metros). Este serviço *web* devolve todos os pontos contidos dentro da região demarcada pela circunferência. De seguida, são verificados os trajetos onde pertencem esses pontos. Finalmente, apresenta todos os resultados com os trajetos que intersejam a circunferência com o raio.

São usadas duas funções geográficas, do *PostGIS*: o *ST\_MakePoint* e o *ST\_Distance\_Sphere*, que permitem identificar os trajetos intersejados pela região de agregação. A função *ST\_MakePoint* cria o ponto central da circunferência e a função *ST\_Distance\_Sphere* calcula a distância de um dado ponto, até ao ponto central, cujo valor raio foi indicado pelo utilizador. No final, apresentam-se os trajetos que contêm pelo menos um ponto com uma distância até ao centro da circunferência igual ou inferior ao valor do raio (que estão dentro da circunferência).

Este serviço funciona como um filtro de trajetos, e pode ser útil em situações de elevada densidade dos mesmos, ou até mesmo em cenários que se pretenda avaliar os movimentos pendulares, presente numa determinada região. Este serviço *web* (*findTracksAndIntersections*) filtra os trajetos, utilizado pela biblioteca *windroseLib* para calcular todas as interseções de trajetos com os raios de agregação das rosas dos ventos.

Na Tabela 4 seguinte, é mostrado um exemplo de pedido *AJAX* (*Asynchronous JavaScript and XML*) (“Ajax - Asynchronous JavaScript and XML”) feito pela aplicação de análise de trajetos, para a consulta dos dados da base de dados.

Tabela 4 – Exemplo de pedido AJAX

Pedido em AJAX
<pre>\$.ajax({   url: webserviceURL,   type: "GET",   dataType: "json",   async: false,   success: function (response) {     //Store the response from PHP file     result = response;   },   error: function (response) {     console.log("PHP error: " + response);   } });</pre>

## Consulta de trajetos na base de dados

A partir do serviço *findTracksAndIntersections* pode ser feito o pedido com os seguintes argumentos:

- *findIntersections* – Efetua várias chamadas a funções geográficas da base de dados para descobrir os pontos onde existem interseções de trajetos de acordo com o valor do raio da circunferência.

A mensagem de pedido ao serviço deve possuir quatro campos:

- *tracks* – quais as *Tracks* onde existem interseções com a região de agregação (circunferência);
- *centerLat* – latitude do centro do círculo;
- *centerLon* – longitude do centro do círculo;
- *radius* – raio do círculo (metros).

A resposta é uma *array* de interseções dos trajetos com a circunferência, da seguinte forma [[IDentidade, IDtrack, PrimeiroPontoDeInterseção, ÚltimoPontoDeInterseção], ...]. O formato devolvido é em JSON.

Para a determinação das interseções foi implementado o seguinte algoritmo do trabalho anterior (Vaz, 2017):

1. Para cada *track* de uma *Entity*, enviadas no campo *tracks*, é criado um objeto na base de dados, através do comando *ST\_MakePoint* com os pontos do track;
2. Cria-se uma circunferência a duas dimensões na base de dados, com *ST\_Buffer*. A circunferência é centrada no par de coordenadas e o raio definidos pelos argumentos no pedido;
3. Para cada track definido na base de dados é verificada a interseção com a circunferência, através do *ST\_Intersection*;
4. Se interseção não for vazia:
  - 4.1. A interseção devolvida abrange todos os pontos que intersejam a circunferência em formato objeto, o resultado é convertido para texto a partir do *ST\_AsText*, pois facilita a manipulação do objeto;
  - 4.2. Retira-se o primeiro e último ponto da interseção com *ST\_StartPoint* e *ST\_EndPoint*;
  - 4.3. O resultado é colocado na mensagem de resposta, com o formato [[IDentidade, IDtrack, PrimeiroPontoDeInterseção, ÚltimoPontoDeInterseção], ...]
5. Caso a interseção seja vazia, é removida e é passada à interseção seguinte até todas serem analisadas.

O pedido *findIntersections* é essencial na construção dos elementos gráficos da rosa dos ventos, pois devolve os resultados com os dados para a orientação, tamanho e cor das pétalas da rosa.

Em termos de resultados o tempo de carregamento dos dados ficou um pouco mais rápido com o *Slim*. Por exemplo, para o mesmo pedido *findIntersections*, em termos médios demora menos 1,50 segundos como se pode ver na Figura 3.3.

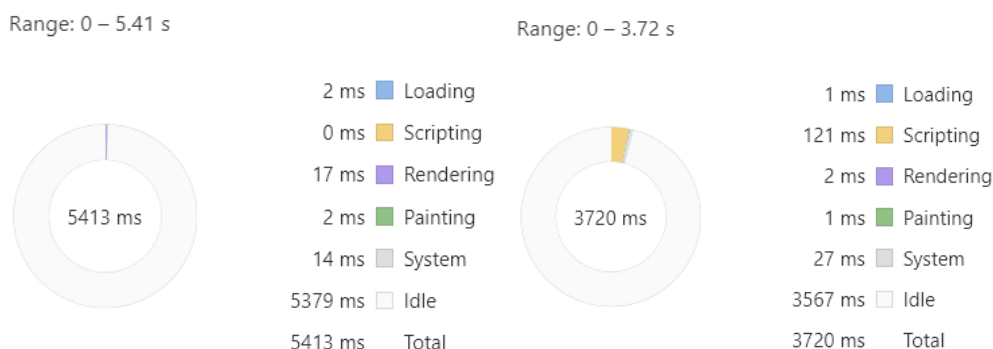


Figura 3.3 – Tempos de pesquisa sem e com ferramenta *Slim Framework* da aplicação.

### 3.3.4 Biblioteca da rosa dos ventos

Nesta etapa é caracterizada a biblioteca da rosa dos ventos (*WindRoseLib*) onde são descritas as alterações que foram realizadas em relação ao protótipo anterior e as novas funcionalidades que foram criadas e que correspondem ao objetivo 1 do trabalho.

Esta biblioteca agrega um conjunto de funções que permitem construir rosas com dados agregados de trajetos que foram intersetados pela região de agregação com apresentação dos seus respectivos elementos gráficos que a compõe.

Deste modo, foram definidas as seguintes funcionalidades:

- Permitir a movimentação do marcador, do local associado a uma rosa dos trajetos e atualização automática do seu conteúdo no novo ponto do mapa;
- Permitir a criação de uma nova rosa dos trajetos (duplicada), a partir de um mesmo local de um marcador, onde já exista uma rosa;
- Definir eventos suportados pela biblioteca para a manipulação das rosas das funcionalidades anteriores;
- Oferecer a capacidade de análise de movimentos pendulares da população;
- Modificar os elementos gráficos: coordenadas geográficas, escala de cores, informação apresentada, setas de origem/destino e botão de duplicar;

A construção da rosa dos trajetos a biblioteca foi codificada em *JavaScript*. O seu construtor principal aceita um conjunto de parâmetros que definem os elementos gráficos, os dados dos trajetos, os filtros a serem aplicados (região de agregação, faixas horárias, velocidades e distâncias) e o local da posição da rosa (latitude, longitude). A partir desta instanciação é criada a rosa de acordo com a sua localização e os trajetos intersetados na região de agregação. Para isso a biblioteca utiliza o serviço *web findTracksAndIntersections*, descrito atrás. Os dados das interseções devolvidos por esse serviço são utilizados na conceção dos componentes gráficos da rosa.

#### Elementos gráficos da rosa

Os vários elementos gráficos da rosa são construídos com o *SVG (Scalable Vector Graphics)* (“*W3C SVG Working Group, «SVG: Scalable vector graphics»*”), que é um formato de imagem baseado em *XML* para definir elementos gráficos a duas dimensões e que facilita o trabalho de animação e interação.

Para este sistema foi escolhida uma escala de cores divergente para caracterização dos valores de velocidade dos trajetos. Assim, para velocidades baixas (associados a trânsito parado) foram utilizadas cores próximas do vermelho, laranja, seguindo-se o amarelo e para velocidades mais altas o verde. Esta escala é utilizada para a cor das pétalas e a direção do par de setas origem/destino dos trajetos.

### **Eventos de atualização e duplicação de uma rosa**

Foi acrescentada uma nova função para a atualização da informação do conteúdo da rosa, aquando da sua movimentação de um marcador para um novo local. O novo método é o *updateRose* que remove os dados da rosa da posição anterior e atualiza a rosa com os novos dados de acordo com a nova localização. Por exemplo, um local que tenha mais de que um trajeto vai dar origem a uma rosa com várias pétalas, enquanto que numa região que tem apenas um trajeto é criada apenas uma pétala.

Com o objetivo de analisar o comportamento de diferentes rosas ao longo de um dia para um mesmo local no mapa foi adicionado um novo método, o *duplicateRose* que fica ativo quando é acionado o botão de duplicar na rosa. Este botão permite criar uma cópia da rosa original, sendo depois possível aplicar outros filtros. Por exemplo, a rosa original fica com dados dos trajetos referentes ao período entre as 00-12 horas e a cópia contém o período da tarde entre as 12-24h. Esta nova funcionalidade permite comparar as duas rosas, em diferentes períodos do dia associadas ao mesmo local.

Cada uma das funções anteriores tem associado um conjunto de eventos sempre que existe alguma interação com as rosas, suportando funcionalidades como duplicar ou atualizar a rosa. Os eventos têm atributos associados que indicam em pormenor sempre que é desencadeado um evento. Estes eventos são descritos na subsecção 3.3.5, referente à aplicação de análise de trajetos, e são ativos quando selecionados elementos gráficos da rosa.

### **Objeto *WindRose***

Na criação de um objeto *WindRose* é necessário definir um conjunto de dados relativos a uma rosa, tais como: trajetos, elementos gráficos de objetos *SVG* e métodos que permitem aceder à informação guardada na rosa, como as interseções dos trajetos com a região de agregação.

A biblioteca continua a ser flexível, pois podem ser adicionadas mais funcionalidades no futuro. Utilizou-se o *JSDoc3* (“«JSDoc 3 - API documentation generator for JavaScript»,” 2018), uma ferramenta que permite gerar documentação *JavaScript* de forma automática, para documentar todos os métodos e ajuda na rápida aprendizagem. A documentação da biblioteca pode ser consultada no Apêndice D.

Na secção seguinte é descrita a aplicação de análise de trajetos que faz uso de todas as funcionalidades da biblioteca descritas ao longo desta subsecção, bem como dos serviços *web* descritos na secção 3.3.3.

## **3.3.5 Aplicação *web* de análise de trajetos (*WindRoseApp*)**

A aplicação de análise, *windRoseApp* corresponde à parte mais importante do sistema onde vai decorrer toda a interação no trabalho de análise de trajetos com rosas dos trajetos por parte do utilizador/analista. Esta faz uso da biblioteca das rosas e dos serviços *web* disponibilizados no acesso à base de dados geográfica. Inicialmente é realizada uma explicação global da aplicação *windRoseApp* e de seguida é detalhado o seu funcionamento e a sua utilização: criar rosas, perceber como manipular o painel de filtros/inspetor e como são despoletados os eventos quando se interage com as rosas.

## Arranque do sistema

O sistema desenvolvido tem agora duas formas de arranque de sistema. Caso a base de dados já tenha presente conjuntos de dados de trajetos carrega diretamente para a página de aplicação de análise *windRoseApp*. Por outro lado, se a base de dados ainda não tiver trajetos abre uma página inicial para seleção de trajetos de um ficheiro GPX, para análise.

A Figura 3.4 apresenta a página da aplicação de análise *windRoseApp* onde apresenta um mapa com vários trajetos de táxis na cidade de Roma tendo à sua direita um painel de filtros e no canto inferior esquerdo um local de coordenadas (latitude e longitude) que muda consoante o cursor do rato se movimenta no mapa.

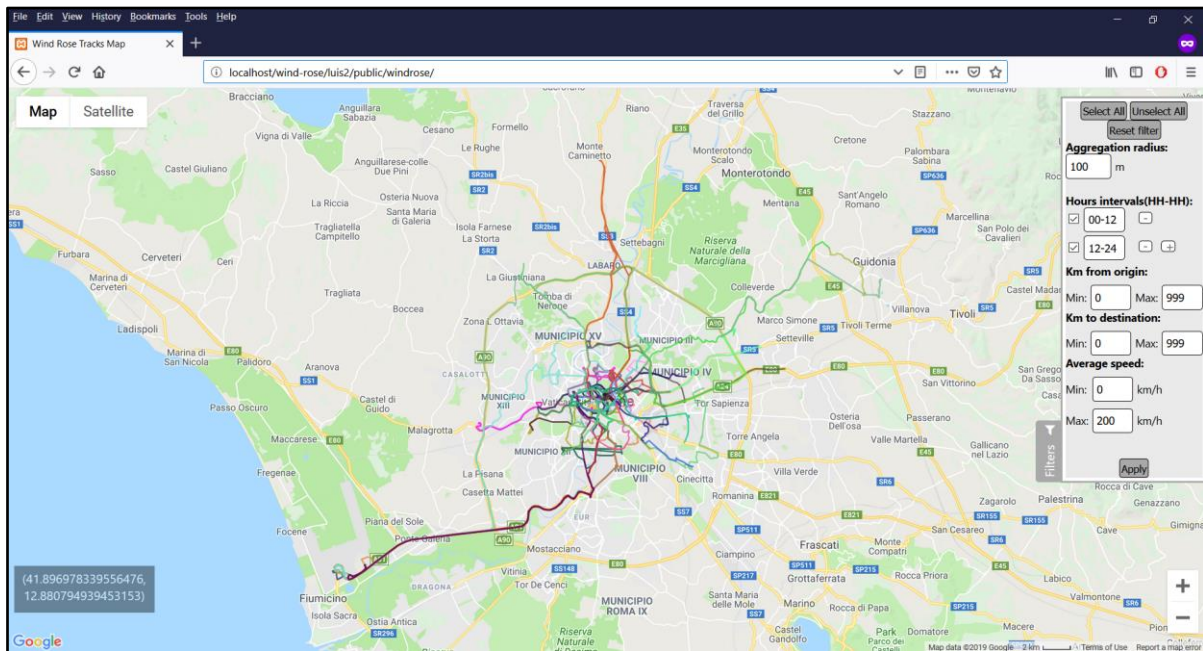


Figura 3.4 – Ecrã da aplicação de análise *windRoseApp* quando carrega a página pela primeira vez.

## Marcador, região de agregação e rosa de agregação de trajetos

A Figura 3.5 mostra um cenário de análise de trajetos de um local do mapa. Cada trajeto é assinalado por uma cor diferente e aleatória. Também é apresentado um marcador, uma região de agregação (círculo cinza que envolve o marcador) e uma rosa associada ao local em análise. Na rosa é possível visualizar os trajetos agregados que foram interseccionados e estão contidos na região de agregação do marcador. Os trajetos fora desta região não são considerados.

Para criar um marcador e gerar uma rosa com dados tem de se clicar com o botão direito do rato numa zona do mapa com trajetos. Cada marcador está identificado com um número único e a rosa tem associado o mesmo número do marcador no canto superior esquerdo. No exemplo, em concreto da Figura 3.5 temos o marcador 2, a sua região de agregação com 100 metros de raio e à direita temos a rosa dos trajetos associada.

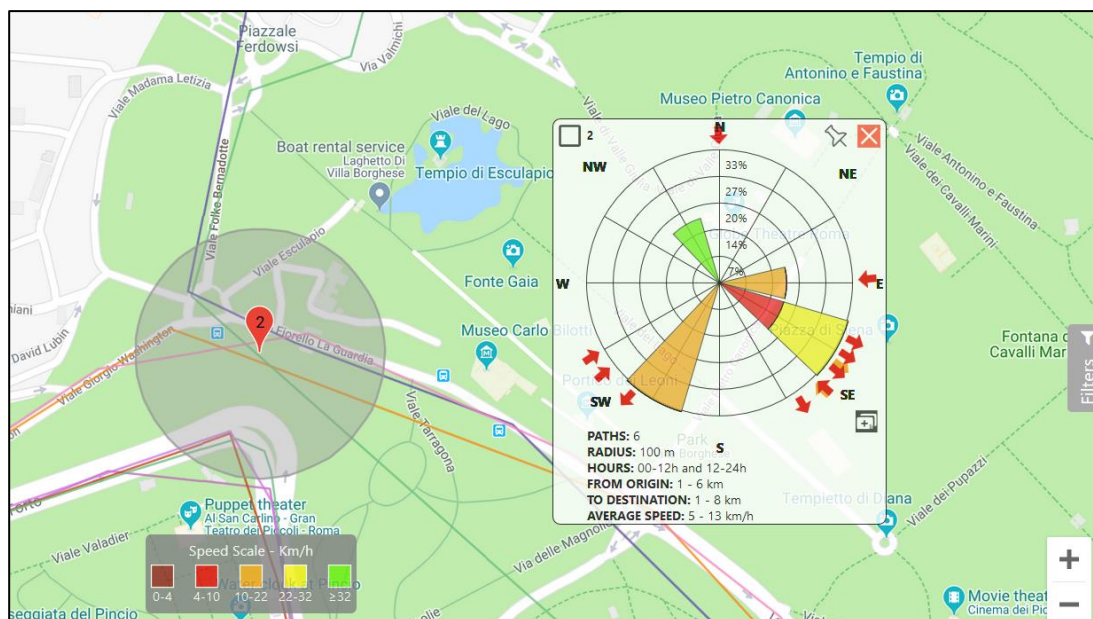


Figura 3.5 – Exemplo de criação de marcador e rosa dos trajetos na aplicação de análise *windRoseApp*.

Na rosa dos trajetos existem cinco botões diferentes. No topo começando da esquerda temos o botão de seleção da rosa e à sua direita o botão de pin e fecho. Na parte inferior direita do desenho da rosa temos o botão de duplicar. O botão de seleção permite selecionar ou desselecionar uma rosa. Para modificar a informação da rosa é preciso que esta esteja selecionada. O botão pin quando clicado fixa a posição da rosa no mapa e esta já não pode ser movida pelo rato. O botão de fecho quando ativo elimina a rosa, o marcador e área de agregação que lhe estavam associadas. Por fim, o último botão é o de duplicar que permite criar uma cópia da rosa original do mesmo local.

A rosa possui uma grelha com vários círculos com uma escala relativa, que se ajusta consoante o máximo de trajetos que uma pétala pode agregar.

No caso da rosa da Figura 3.5, temos um total de 6 trajetos que foram agregados em 4 pétalas principais:

- Duas pétalas mais pequenas a verde e amarelo que agregaram 1 trajeto cada uma;
- Duas pétalas de maiores dimensões que agregaram 2 trajetos cada uma.

Relativamente às pétalas de maiores dimensões temos dois casos: a pétala que vem da direção SE apresenta duas sub-pétalas, porque agrega dois trajetos com velocidades diferentes. A pétala que vem da direção SW agrega dois trajetos com a mesma velocidade que estão representados numa única pétala.

O máximo de trajetos agregados por direção foi de 33%, uma vez que existem duas pétalas principais que agregam no máximo dois trajetos que contêm todos percursos que interseam a região de agregação. Cada um dos valores da escala está associado a um dos círculos da grelha da rosa dos ventos, sendo que a escala cresce do centro até ao último círculo da grelha.

Cada pétala da rosa possui as seguintes propriedades:

- A direção corresponde a direção de entrada com que um trajeto interseou, pela primeira vez, a região de agregação do marcador da rosa.
- O comprimento da pétala depende da quantidade total de trajetos que a pétala consegue agregar relativamente ao total de trajetos que estão no *PATHS* do rodapé da rosa. No caso da pétala que vem da direção SE e apresenta 2 sub-pétalas, esta agregou 2 trajetos de um total de 6 trajetos

daquela região de agregação, o que corresponde a uma percentagem de 33% que é o máximo que é representada na grelha da escala.

- A cor das pétalas define as várias velocidades com que os trajetos podem intersestar a região de agregação. No caso da rosa da Figura 3.5 esta apresenta vários trajetos com 4 velocidades de interseção distintas. Para velocidades baixas as cores correspondem ao vermelho/laranja, para velocidades médias o amarelo e para velocidades altas o verde. A escala de cores segue a lógica do trânsito de uma cidade: vermelho para trânsito parado e verde tráfego a circular. No exemplo, a pétala verde e amarela contém um trajeto cada. A pétala que apresenta duas sub-pétalas tem uma vermelha e outra laranja. A última e maior pétala contém dois trajetos com cor laranja. A soma dos trajetos de cada uma das pétalas/sub-pétalas corresponde ao total de trajetos agregados pela pétala principal. Para saber o número de trajetos agregados em cada pétala/sub-pétala, o utilizador/analista deve mover o cursor sobre cada uma, onde surge uma caixa com informação do número de trajetos que foram agregados em cada pétala.

No exterior da grelha das pétalas da rosa temos setas que representam a origem e destino de cada trajeto. A orientação de cada par de setas origem/destino está de acordo com o local onde começou e terminou o trajeto, em relação ao ponto atual do marcador da rosa no mapa. Também no exterior da grelha estão definidos os vários pontos cardeais e colaterais (N, NE, W, SW, S, SE, NW) para ajudar na análise.

A cor do par de setas origem/destino representa a velocidade média de todo o trajeto completo, enquanto que a cor das pétalas corresponde à velocidade quando existe a primeira interseção com a região de agregação da rosa. A relação entre as cores e as velocidades utiliza uma escala divergente que se apresenta no rodapé da aplicação: *Speed Scale* que varia entre o vermelho escuro e o verde.

Por baixo da rosa existem um conjunto de informações que resumem e apresentam dados dos trajetos da rosa. O campo *PATHS* dá o total de trajetos que intersestaram a região de agregação, o *RADIUS* o raio da área de agregação, o *HOURS* corresponde às faixas horárias selecionadas para essa visualização de trajetos. Os últimos três campos (*FROM ORIGIN*, *TO DESTINATION*, *AVERAGE SPEED*), representam mínimos e máximos relativos à distância, em Km, da origem/destino (distância entre o marcador e o primeiro/último ponto do trajeto) e velocidade média, em Km/h, respetivamente. Estes intervalos são referentes às propriedades dos trajetos representados na rosa.

O campo *HOURS* e os últimos três campos referidos atrás podem ser modificados com a ajuda de uma ferramenta, o painel de filtros.

### **Painel e Inspetor de Filtros**

O painel de filtros está localizado à direita na aplicação e permite alterar os diferentes parâmetros da rosa ou conjunto de rosas dos trajetos existentes no mapa. Neste existem igualmente *checkboxes* com intervalos de faixas horárias que podem ser ativadas quando selecionadas.

A Figura 3.6 mostra o painel de filtros com os diferentes parâmetros que podem ser modificados: Aggregation radius, hours intervals, km from origin/destination e average speed mínimo e máximo. Podem ser adicionadas tantas faixas até ao limite do tamanho do ecrã da aplicação e os campos das faixas horárias podem ser modificados.

Existem quatro botões no painel de filtro: o *Select All*, o *Unselect All*, o *Reset filter* que estão localizados no topo do painel de filtros e o botão de *Apply* que está localizado no fim do painel de filtros.

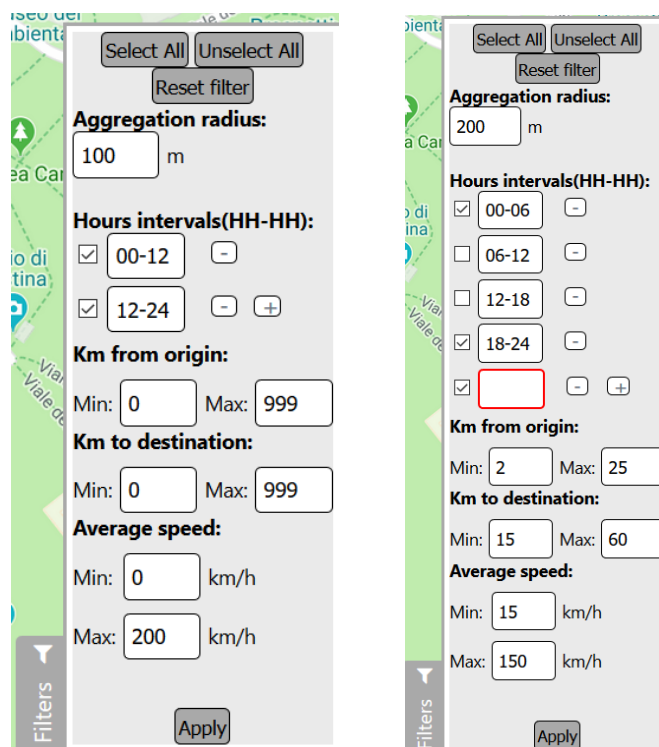


Figura 3.6 – Painel de filtros modificado.

A interação com painel de filtros é feita aquando da criação de rosas ou na modificação das rosas existentes. Para melhor perceber esta interação temos o exemplo da Figura 3.7, com dois marcadores presentes no mapa, sendo uma rosa associada ao marcador 2 e as outras duas rosas associadas ao marcador 1. A rosa do marcador 2 e uma das rosas do marcador 1 encontram-se seleccionadas, porque têm um *check* no botão de seleção ativo e estão sombreadas a cinza.

Quando o utilizador/analista cria um marcador no mapa é gerada uma região de agregação e a rosa respetiva que assume os valores que estão presentes nos campos do painel de filtros. Se for inserido algum valor inválido é dado um alerta de mensagem de erro.

Para modificar uma rosa é necessário que esta esteja sempre com o botão de seleção ativo e mudar os parâmetros do painel de filtros. Para proceder à confirmação dos campos a alterar é necessário carregar no botão de *Apply*. Sempre que estejam rosas seleccionadas, e existam alterações nalgum campo do painel de filtros é mostrado um asterisco (\*) para assinalar que o campo assumiu um novo valor, mas que ainda não foi aplicado. Quando o botão *Apply* é ativado, o asterisco (\*) desaparece e o novo valor é aplicado na rosa.

Se o utilizador estiver a analisar diferentes rosas torna-se necessário ter uma maneira expedita de seleccionar ou desseleccionar todas as rosas, por exemplo se quiser aplicar os mesmos filtros em diferentes rosas. Por isso, existem dois botões, o *Select All* e o *Unselect All* que permitem fazer rapidamente uma marcação/desmarcação de todas as rosas presentes no mapa. Caso o utilizador pretenda voltar aos valores e configuração inicial do painel de filtros basta clicar no botão *Reset filter* volta aos valores base.

Na Figura 3.7 também está presente um Inspetor no painel de filtros que verifica se existe alguma incoerência quando compara duas ou mais rosas em simultâneo. Quando se selecciona uma rosa, o painel de filtros atualiza o valor dos seus parâmetros com os valores dessa rosa. Se uma outra rosa é seleccionada e tiver algum campo com valor diferente da primeira rosa, o inspetor alerta para essa incoerência, colocando o campo a cinza sem valor e com um rebordo vermelho à sua volta. A incoerência acaba quando

uma ou mais rosas com valores de parâmetros divergentes são desseleccionadas ou modificamos e aplicamos o mesmo valor no campo que possui a incoerência. Enquanto existir alguma incoerência, não será possível criar rosas e aparece uma mensagem de alerta de erro.

Por exemplo na Figura 3.7, observa-se que o valor do raio de agregação, o primeiro campo da faixa horária, o máximo de destino e o mínimo de velocidade média das rosas 1 e 2 seleccionadas apresentam os campos bloqueados a cinza e sombreados a vermelho. Isto significa que estamos perante uma incoerência pois estão a ser analisadas rosas com valores diferentes e é dado um alerta ao utilizador para esse facto. Para sinalizar a incoerência das *checkboxes* das faixas horárias que estão ou não ativas é feita um sombreamento da faixa horária respetiva a amarelo.

### **Movimentos pendulares**

A análise de movimentos pendulares é um aspeto importante para a compreensão do movimento da população ao longo de um dia numa zona urbana, por exemplo quando as pessoas se deslocam de casa para o trabalho, no centro da cidade e depois quando regressam a casa. Este movimento pode ser detetado nas rosas dos trajetos do sistema. Para isso, é necessário criar rosas com diferentes períodos do dia associadas a um mesmo ponto no mapa.

No caso da Figura 3.7, temos o marcador 1 ligado a duas rosas, sendo que a primeira contém a rosa com dados filtrados entre o período entre as 00-12 horas (madrugada/manhã) e segunda rosa com dados compreendidos entre as 12-24h (tarde/noite).

De seguida, podemos comparar as duas rosas. Na primeira rosa (00-12h) podemos verificar uma pétala com duas sub-pétalas com direção de SW-S, o que significa que temos dois trajetos que vem dessa direção com diferentes velocidades. Também existe uma pétala que vem da direção N-NE que corresponde a um trajeto. Ambos os trajetos correspondem ao percurso casa-trabalho. Na segunda rosa (12-24h), verificamos o comportamento inverso com uma pétala com três sub-pétalas que vem da direção N-NE e outra pétala que corresponde a um trajeto que vem da direção SW-S que corresponde ao percurso trabalho-casa.

Esta análise e visualização é muito útil para conhecer o fluxo de pessoas dentro de cidade para um ponto do mapa. Podemos concluir, neste exemplo em concreto que não existem grandes problemas de trânsito no período da manhã, contudo verificamos que no período da tarde o trânsito fica mais lento.

### **Eventos para manipulação de rosas**

No sistema anterior já tinham sido definidos alguns eventos que são despoletados pela aplicação de análise *windRoseApp* ou pelas instâncias da biblioteca *windroseRose*. Cada evento tem associado um *listener* que o capta e realiza a sua função respetiva. Já existiam os seguintes eventos *roseSelected*, *roseUnselected*, *roseCreated* e *roseDeleted* para permitir as alterações feita na rosa como a funcionalidade *applyFilters*.

A seguir, são apresentadas as alterações que foram feitas nos eventos da versão anterior do sistema e os novos eventos que foram definidos e que são utilizados entre a aplicação *windRoseApp* e a biblioteca *windRoseLib*. É dada uma explicação de como cada evento é desencadeado, qual seu objetivo, onde foi implementado o seu *listener* e como se comporta.

- *deleteRose* – é assinalado por uma instância *windRose*, sempre que o utilizador carrega no botão de fecho da rosa. O evento tem os seguintes parâmetros: o identificador único da rosa que deve ser

destruída. O *listener* deste evento está implementado na aplicação web e sempre que deteta a ocorrência do evento, trata de iniciar o processo de destruição da instância *windRose* em causa (apaga o marcador e círculo da rosa e chama o método de destruição da instância da rosa em causa).

- *updateRose* – é lançado pela aplicação *windroseApp* sempre que o utilizador movimenta o marcador no mapa. O evento tem os seguintes parâmetros: o identificador único da rosa e a nova posição do marcador. O *listener* deste evento está implementado em todas as instâncias *windRose*, sendo que sempre que exista a movimentação do marcador, trata de chamar a função *updateRose* que atualiza o conteúdo de todas as rosas associadas de acordo com a nova posição do marcador.
- *duplicateRose* – é despoletado por uma instância *windRose*, sempre que o utilizador carrega no botão de duplicar na rosa. O evento é composto por um parâmetro de identificador único da rosa que deve ser duplicada. O *listener* deste evento está implementado na aplicação web e sempre que deteta a situação do evento, trata de chamar a função *duplicateRoseHandler* que trata de duplicar a rosa original.
- *applyFilters* – é desencadeado pela aplicação *windroseApp*, sempre que o utilizador clica o botão Apply, no painel de filtros. Este evento tem os parâmetros do painel de filtros. O *listener* está presente em todas as instâncias *WindRose*. Quando ocorre o evento, consulta os dados dos filtros a aplicar e trata de chamar a função de aplicação de filtros da rosa;
- *selectedRose/unselectedRose* – são ativados pela aplicação *windroseApp*, sempre que o botão *Select All/UnselectAll*, no painel de filtros, é carregado. Não tem associados parâmetros. O *listener* está codificado nas instâncias *WindRose*. Estes eventos alteraram o estado da rosa selecionada ou desselecionada, respetivamente;
- *roseCreated* – é assinalado por uma instância *WindRose*, sempre que a mesma acaba de criar uma rosa. O evento tem como parâmetro o identificador único da rosa criada. O *listener* deste evento está implementado na aplicação *windRoseApp* e sempre que deteta a ocorrência do evento, imprime, na consola, uma mensagem a indicar que a rosa foi criada.
- *roseDeleted* - é assinalado por uma instância *WindRose*, quando uma rosa é removida. O evento tem como parâmetro o identificador único da rosa que foi removida. O *listener* deste evento está codificado na aplicação *WindRoseApp* e sempre que deteta a ocorrência do evento, imprime, na consola, uma mensagem a indicar que a rosa foi removida;
- *roseSelected/roseUnselected* – são ativados por uma instância *WindRose*, sempre que a rosa muda para o estado selecionada ou desselecionada, respetivamente. O evento contém como parâmetro o identificador único da rosa. O *listener* deste evento está concretizado na aplicação *windRoseApp*. Quando deteta a ocorrência de um evento, e exista mais do que uma rosa selecionada, trata de verificar a ocorrência de incoerências entre as várias propriedades das rosas selecionadas. Caso existam incoerências, são assinaladas pelo inspetor do painel de filtros;

A utilização de eventos entre a aplicação *windroseApp* e a biblioteca *windroseLib* servem de canal de comunicação e permite manusear várias rosas ao mesmo tempo.

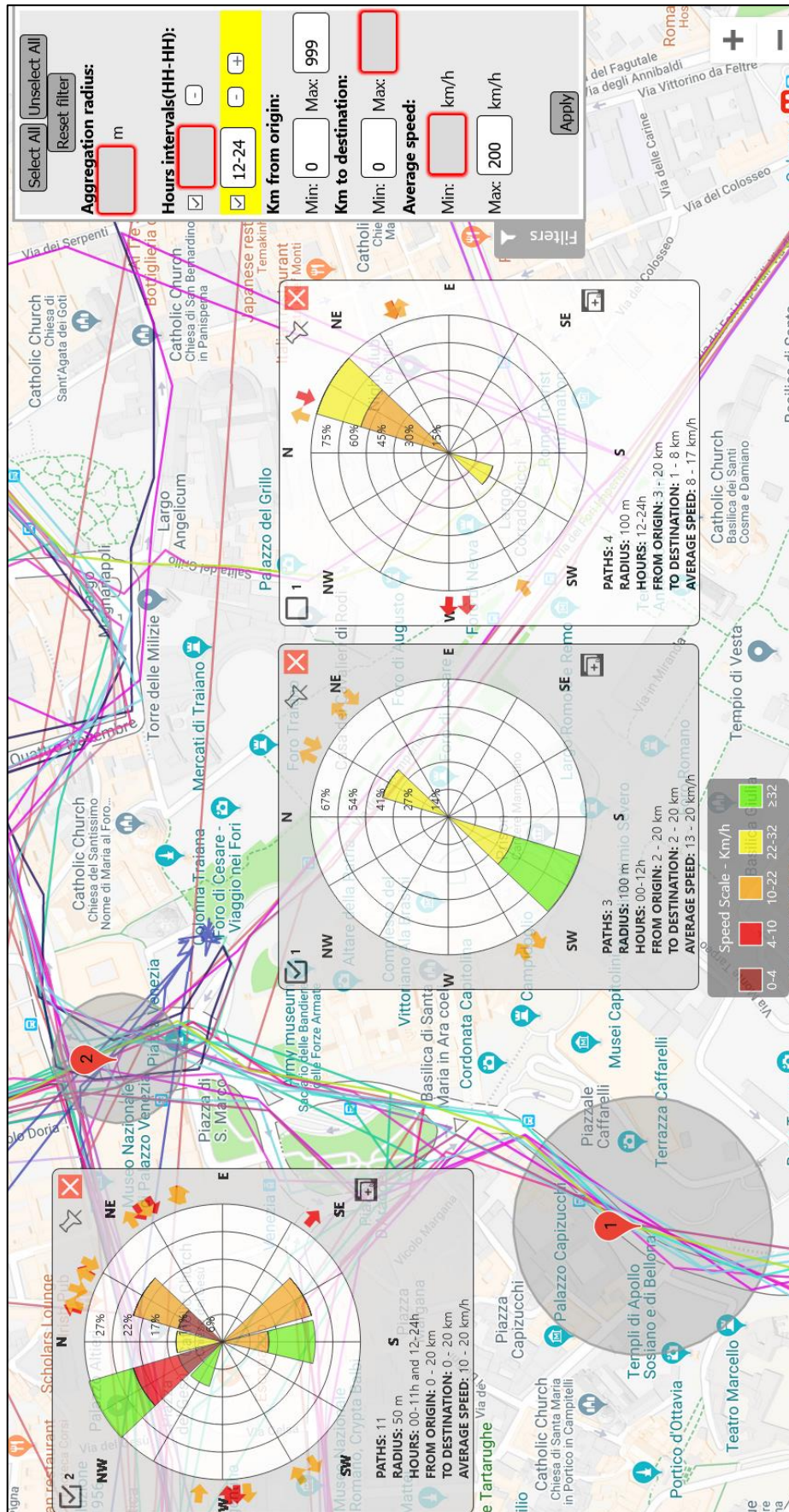


Figura 3.7 - Rosas dos trajetos em dois locais diferentes do mapa, sendo que para o local 1 existem duas rosas, uma para o período da manhã e a outra para a tarde, para o estudo de movimentos pendulares.

### 3.3.6 Bibliotecas externas

As bibliotecas externas utilizadas no desenvolvimento do sistema foram as seguintes:

- *Google Maps JavaScript API* (“Google, «Google Maps JavaScript API: Simple polylines»,” 2018) – foi utilizado em todas as funcionalidades que usam o mapa ou componentes associados ao mesmo, como o desenho de trajetos, criação e movimentação de marcadores e regiões de agregação;
- *jQuery* (“The JS Foundation, «jQuery»”) – serviu de auxílio na escrita do código *JavaScript*, porque simplificava muito determinadas ações como manipular e obter elementos *HTML* e para pedidos *AJAX* de serviços *web*.
- *jQuery UI* (“The JS Foundation and the jQuery UI Team, «jQuery UI - jQuery user interface extension»,” 2012) – foi usado para tornar o arraste das rosas funcional pois basta associar a cada objeto *windRose* a propriedade *draggable* desta biblioteca;
- *Date-ranges-selector* (Albert Gonzalez, 2018) – foi aplicado na implementação das faixas horárias dinâmicas no painel de filtros para inserir e remover faixas horárias;
- *Font Awesome* (D. Gandy, 2018) – foi empregue na inserção dos símbolos na rosa, com licença de utilização gratuita, como o ícone de duplicar;

Estas revelaram-se extremamente úteis no auxílio de conceção das funcionalidades pois resolveram vários problemas que de outra forma a sua implementação teria sido mais difícil e demorada.

## 3.4 Análise de conjunto de trajetos

Nesta parte explica-se a forma como os vários componentes do sistema interagem perante um determinado conjunto de trajetos, através de diagramas de sequência. São mostrados vários cenários que vão desde o carregamento dos trajetos, até à sua análise com a criação de rosas na aplicação do sistema.

Foram utilizados os mesmos dados de trajetos de táxi, na cidade de Roma nesta nova versão do protótipo, devido à grande dificuldade em achar dados de trajetos deste tipo. Houve a hipótese de uma empresa de transporte fornecer dados, mas acabou por não se concretizar. Para simular a situação de movimentos pendulares foram gerados alguns trajetos específicos com base nos dados originais com o objetivo de se analisar movimentos da população em diferentes períodos do dia.

### 3.4.1 Dados de trajetos de táxis

Para testar o sistema foram utilizados dados da plataforma de *CRAWDAD*, que disponibiliza conjuntos de dados obtidos a partir de equipamentos com acesso a comunicações sem fios. Os dados correspondem a movimentação de 320 táxis na cidade de Roma durante o mês de Fevereiro de 2014 (Bracciale et al., 2014) e que foram tratados no trabalho anterior (Vaz, 2017), com um pré-processamento para gerar um ficheiro de formato *GPX* final com os trajetos de oito táxis.

### 3.4.2 Carregamento de trajetos da base de dados e seleção de trajetos

O carregamento dos dados de trajetos na base de dados é necessário caso não existam dados na base de dados. Existem dois passos intermédios desenvolvidos no trabalho anterior (Vaz, 2017) que consistem em converter o ficheiro do formato *GPX* em formato *JSON* compatível com a base de dados geográfica

(serviço *web convertGPXtoJSON*) e depois fazer o carregamento do formato gerado em JSON, com informação dos trajetos, na base de dados.

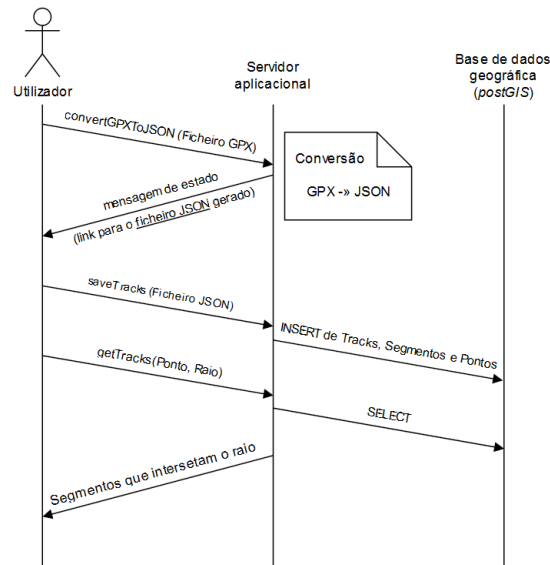


Figura 3.8 – Cenário de conversão e carregamento de um ficheiro GPX, com posterior consulta de trajetos (Vaz, 2017).

No protótipo anterior esta aplicação de gestão de trajetos *selectTracks* (selecionar quais os trajetos a analisar) tinha de ser realizada sempre em primeiro lugar, antes de se prosseguir para a aplicação análise de trajetos *windRoseApp*. Neste projeto retificou-se este fluxo e desde que já existam trajetos na base de dados de dados geográfica, o utilizador já pode utilizar a aplicação de análise do sistema *windRoseApp*.

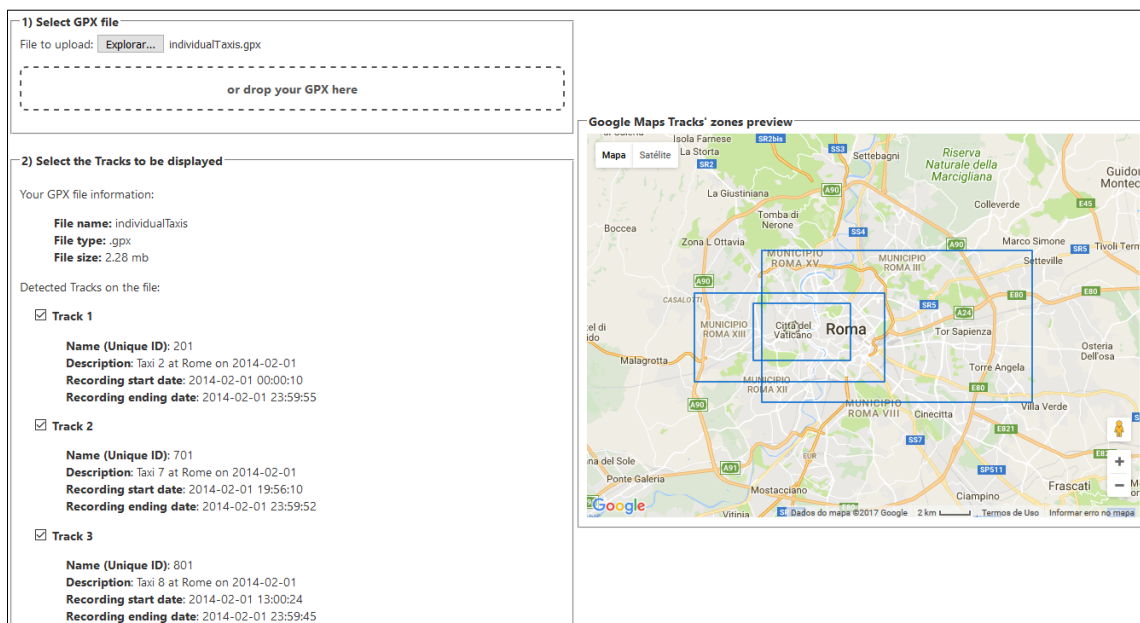


Figura 3.9 – Página web do serviço *selectTracks*, onde se selecionam os trajetos da aplicação *windRoseApp* (Vaz, 2017).

### 3.4.3 Criação de uma rosa

Depois de carregada a aplicação de análise *windRoseApp*, os trajetos estão prontos a ser analisados e já é possível criar rosas a partir de instâncias da biblioteca *windRoseLib*.

Quando a página *windRoseApp* é carregada aparecem conjuntos de trajetos no mapa e o utilizador/analista pode iniciar a criação de um novo marcador no mapa. Para criar um marcador o utilizador escolhe um ponto do mapa com trajetos e faz um clique com o botão direito do rato. Esta operação cria um marcador, uma região de agregação (circunferência), e coloca esse marcador numa *array* de marcadores e executa a função *createWindRose* que corresponde ao construtor da biblioteca *windRoseLib*. Nesta chamada da função são passados vários argumentos que constituem as várias características da rosa, por exemplo, o identificador do marcador da rosa, altura e comprimento da caixa da rosa e informações dos trajetos do mapa, entre outras. Esse objeto *windRose* é colocado num *array* associado à posição do respetivo marcador e também é inicializado um *array* de duplicados dessa rosa (ver Figura 3.10).

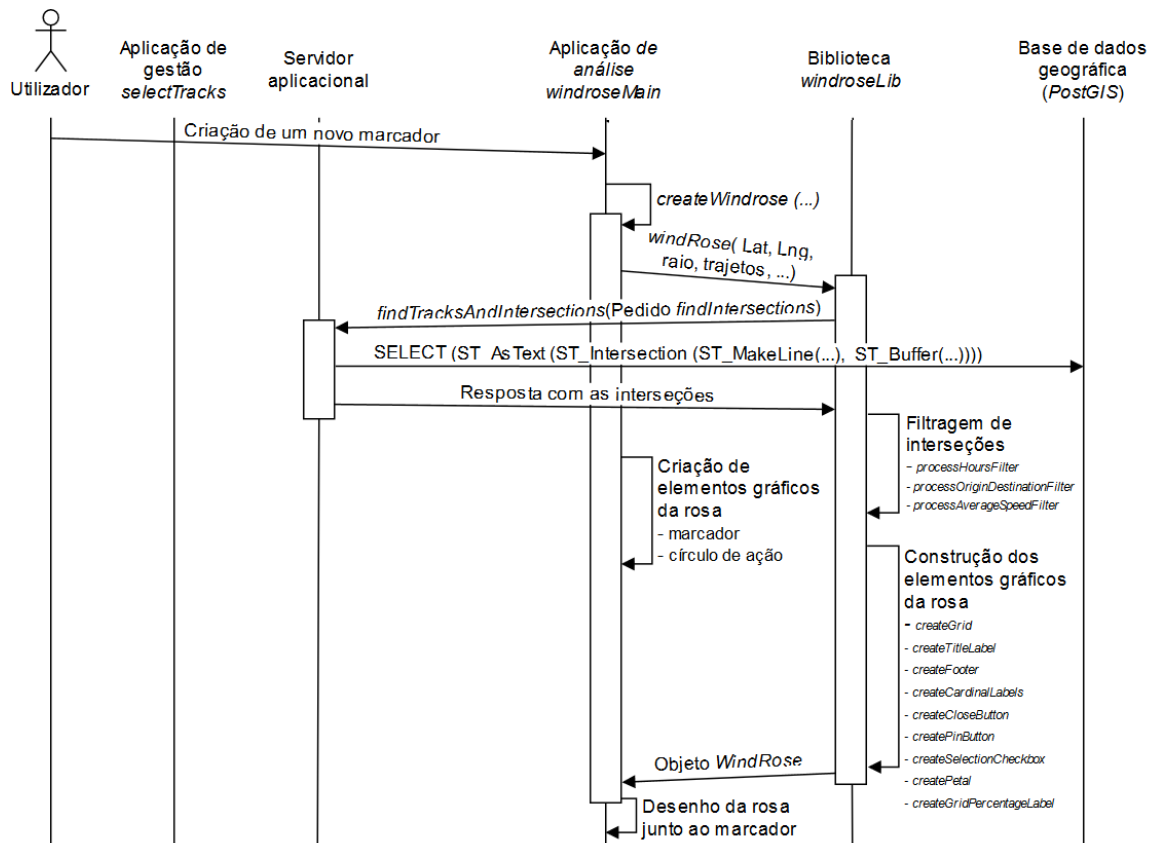


Figura 3.10 – Cenário de criação de um marcador na aplicação de análise *windRoseApp*.

Agora com a instância de um objeto do tipo *windRose*, a primeira instrução corresponde a obter as interseções dos trajetos do mapa com a região de agregação definida. Esta ação é feita a partir do serviço *findTracksAndIntersections*. Após obter as interseções são aplicados os filtros de acordo com as propriedades que vieram em argumento no construtor da rosa. A informação e elementos gráficos *SVG* a apresentar na rosa são o resultado da aplicação desses filtros. O objeto *windRose* gerado aparece junto do marcador. O aspeto do conteúdo da rosa pode ser modificado através da interação com o painel de filtros por exemplo, mudar a região de agregação para 50 metros de raio e considerar apenas trajetos realizados entre as 18-24h.

### 3.4.4 Criação de uma nova rosa para um mesmo marcador

Quando na aplicação já se encontra no mapa um marcador e uma rosa associada com dados, o utilizador pode querer analisar várias rosas ao longo de vários períodos do dia. Como exemplo, na Figura 3.11 foi criada uma nova rosa (duplicada), a partir da rosa original que está associada ao marcador 1. Cada rosa está ligada a um período do dia (madrugada/manhã e tarde/noite).

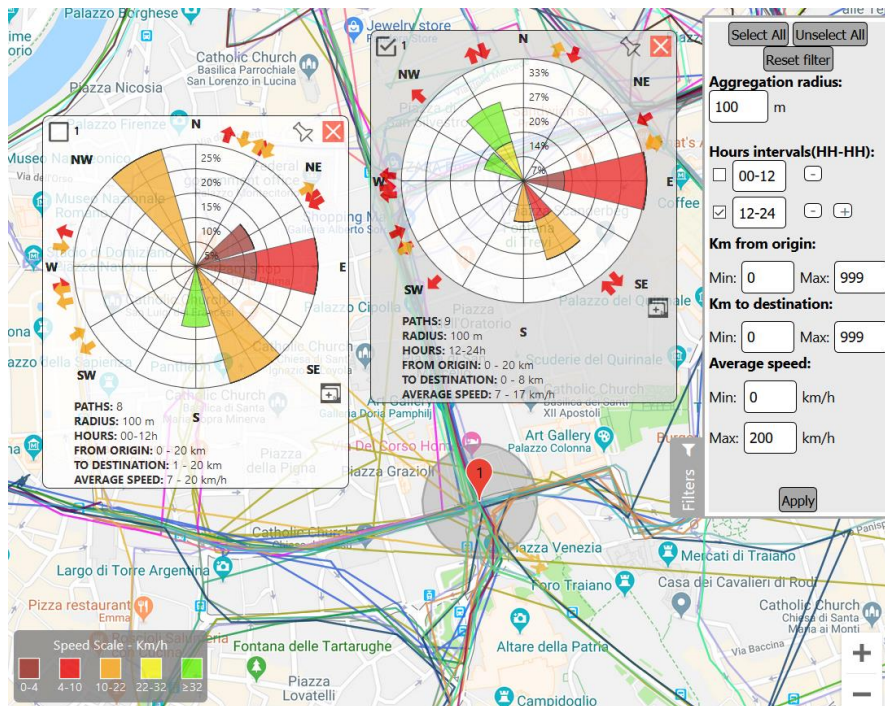


Figura 3.11 – Cenário de nova rosa para um mesmo marcador.

A Figura 3.12 mostra o cenário da ação de duplicar que é acionada quando o botão de duplicar é pressionado na rosa original. É na aplicação que este evento vai ser tratado pelo *listener* que cria uma rosa igual à original em termos de dados. A diferença surge no seu identificador único que é diferente internamente para o sistema, contudo para o utilizador é como se fosse o mesmo. Este é efetuado pelo *duplicateRoseHandler* que incrementa e adiciona o duplicado do objeto do tipo *windRose* no *array roseObjects* de duplicados do marcador associado.

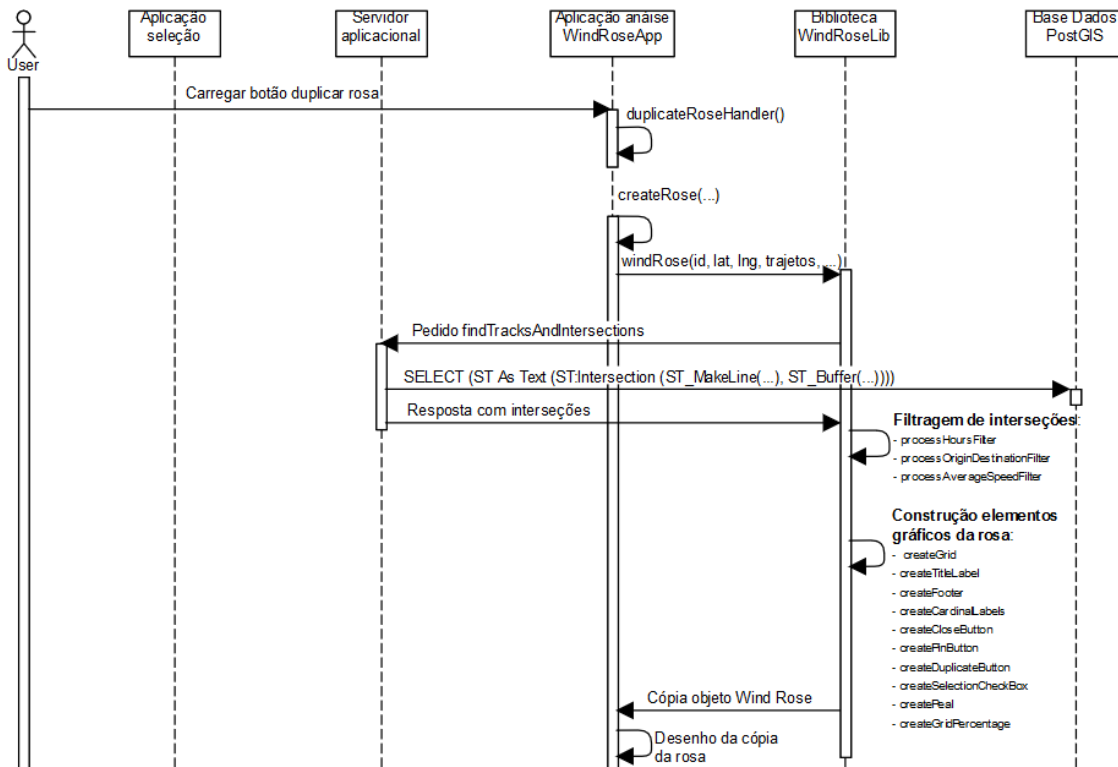


Figura 3.12 – Cenário de criação de duplicado da rosa na aplicação de análise *windroseApp*.

### 3.4.5 Movimentação de uma rosa

Se já existirem algumas rosas com os respectivos marcadores em vários locais do mapa, é possível movimentar um determinado marcador para um novo ponto no mapa. Para isso, seleciona-se e arrasta-se o marcador com o cursor do rato. Como consequência, a nova posição do marcador leva a uma atualização do conteúdo da rosa.

Esta funcionalidade é muito útil, porque um utilizador não precisa de estar sempre a destruir ou a criar rosas sempre que quiser analisar um novo local no mapa. Também o utilizador pode realizar pequenos ajustes, quando estiver em modo exploratório de certos cenários que pretenda identificar.

A Figura 3.13 mostra o cenário da ação de movimentar o marcador. É na aplicação que este evento vai ser tratado pelo *listener* que apaga os dados da rosa e coloca os novos dados de acordo com a nova posição do marcador. Este é efetuado pelo *updateRoseHandler* que atualiza o conteúdo da rosa do objeto do tipo *windRose* e do *array roseObjects* de duplicados do marcador associado.

### 3.4.6 Destruição de uma rosa

Num cenário em que temos várias rosas com duplicados no mapa, a análise pode ser dificultada pela existência de muitos gráficos, no mapa. De modo a resolver, esta situação disponibiliza-se a opção de destruição de uma ou mais rosas. Deste modo, quando é carregado o botão de fecho é lançado um evento com o identificador único que vai ser tratado pelo *listener* que elimina essa rosa. Este é efetuado pelo *removeRoseHandler* que remove o duplicado ou a rosa original do objeto do tipo *windRose* no *array* de duplicados do marcador associado. O marcador e região de agregação continuam a aparecer enquanto tiver pelo menos uma rosa no *array* de duplicados do marcador. Assim que é eliminada a última rosa associada ao marcador, este último é removido e a região de agregação (circunferência).

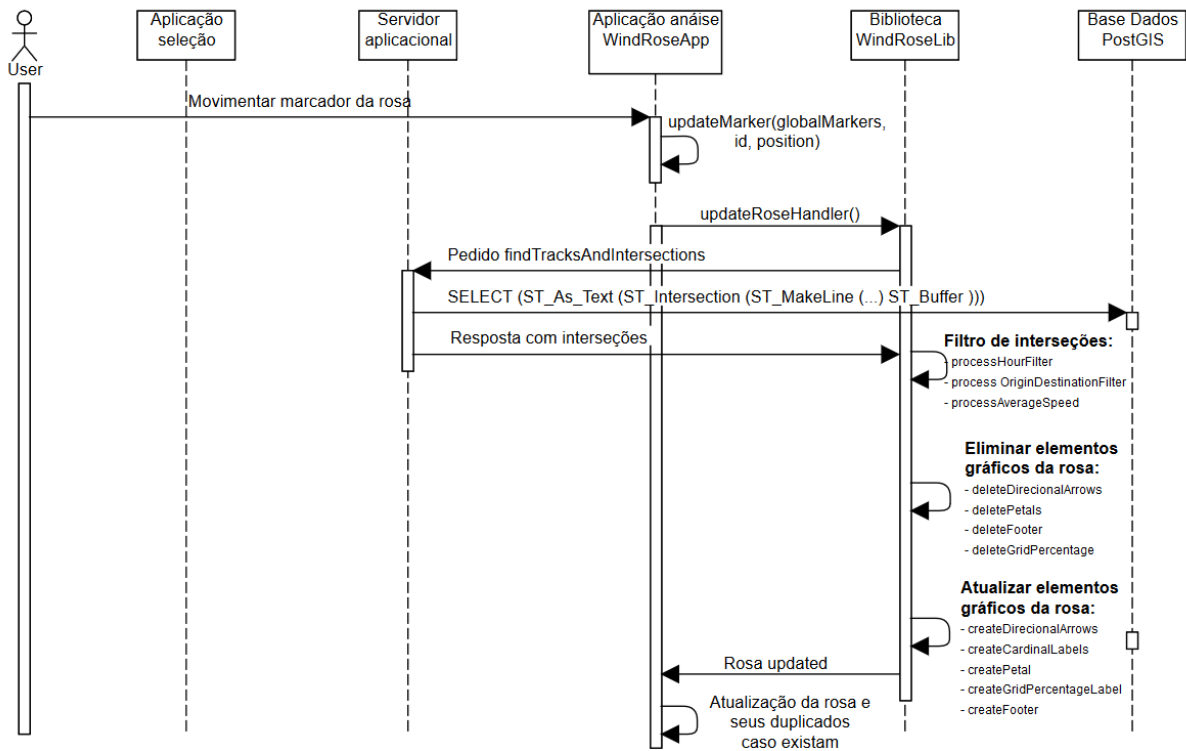


Figura 3.13 – Cenário de movimentação de um marcador na aplicação de análise *windroseApp* e atualização automática do conteúdo da rosa.

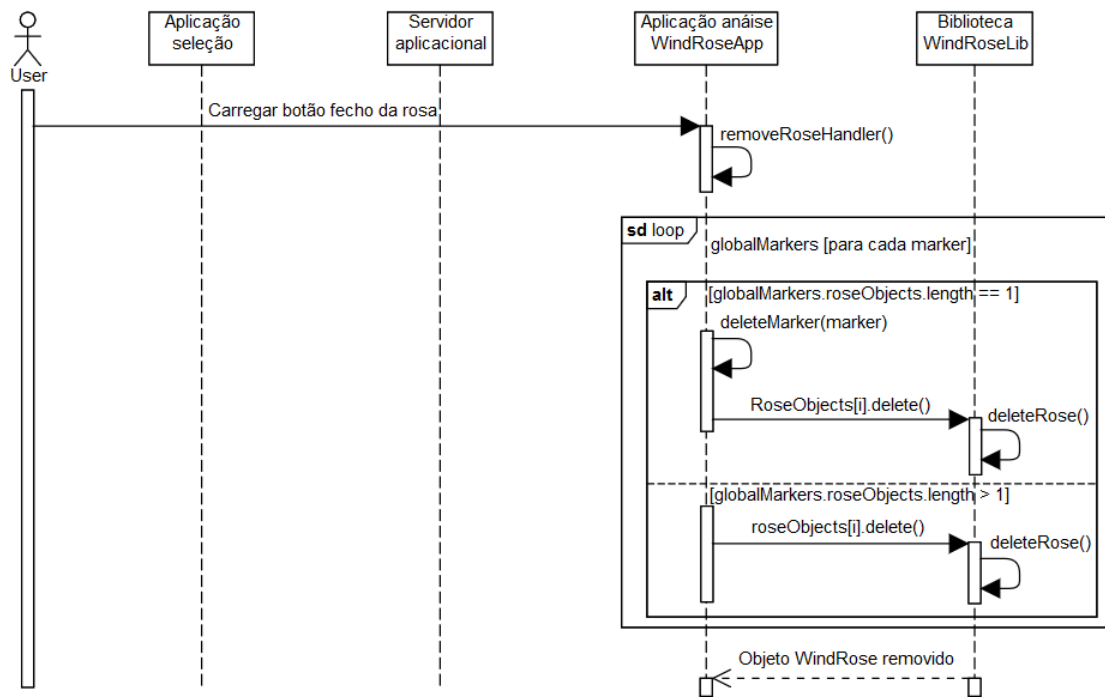


Figura 3.14 – Cenário de destruição da rosa e marcador na aplicação de análise *windroseApp* e atualização automática do conteúdo da rosa.

### 3.4.7 Simulação de dados trajetos em tempo real

Esta funcionalidade não se concretizou neste sistema devido à sua complexidade e não ter conhecimentos aprofundados de processamento distribuído. Ainda foram analisados alguns exemplos, mas não foi possível implementar o que provocou um atraso nas etapas seguintes. Um tipo de ferramenta explorada e que poderá servir de guião de trabalho futuro é o *GeoMesa* (Commonwealth Computer Research, n.d.) que é um software livre que faz análise geo-espacial em larga escala de sistemas de computação distribuída e processamento de fluxo de dados do tipo espaço-temporais.

O *GeoMesa* pode usar um *datastore* de *streaming* baseado no *Apache Kafka*, que seria indicado para suportar simulação de dados na visualização de mapas. Por exemplo, o sistema vai lendo dados de posição de veículos que depois seriam renderizados e animados numa camada do mapa (Hughes, Zimmerman, Eichelberger, & Fox, 2016). Além disso, o *GeoMesa* pode aproveitar o *Apache Spark* (“Apache Spark - Unified Analytics Engine for Big Data,” 2018), uma estrutura de computação que está substituindo o uso direto do *MapReduce* para processamento muito mais rápido e abre algumas possibilidades de trabalho com dados espaço-temporais.

Uma limitação desta ferramenta é estar programada em *Scala*, o que é uma limitação para quem pretenda analisar o código fonte e usar métodos diretamente se não estiver familiarizado com a linguagem.

Das outras alternativas existentes, estas são ainda em número muito reduzido e grande parte são pagas, por exemplo o *Microsoft Azure* (“Azure Stream Analytics Serviço de análise em tempo real”) e o *AWS* (“Real-time Analytics on AWS,” 2018), para trabalhar com dados de *streaming* e análise analítica.

## 3.5 Sumário

Neste capítulo foi apresentada a arquitetura geral do sistema que possibilita a análise de trajetos com rosas. Explicou-se o seu modo de funcionamento e como foram concretizados os objetivos do trabalho, como a reestruturação da biblioteca com aumento das funcionalidades para permitir a atualização automática e criar rosas duplicadas num mesmo ponto. Igualmente foi descrita a construção de uma ferramenta que permitiu evoluir e implementar mais facilmente os serviços web, o *Slim Framework*. Foram ainda descritos alguns cenários de interação das rosas com o painel de filtros.

No capítulo seguinte são apresentados os testes com utilizadores para avaliar a usabilidade do sistema desenvolvido e o desempenho dos utilizadores.

## Capítulo 4

### Avaliação do sistema

Neste capítulo é abordada uma análise detalhada dos testes com utilizadores para avaliar a usabilidade do sistema desenvolvido e o desempenho dos utilizadores. Primeiro é feita uma descrição do propósito das sessões de avaliação, com caracterização do local dos testes e dos participantes. De seguida, são apresentados os objetivos de cada tipo de tarefa e como se concretizaram. Depois, mostram-se os resultados e, por fim, é feita uma discussão dos mesmos, incluindo a comparação com os resultados do protótipo do trabalho anterior.

#### 4.1 Propósito

Os objetivos principais dos testes com utilizadores foram os seguintes:

- Determinar o grau de usabilidade do sistema a partir do procedimento SUS (Bangor, Kortum, & Miller, 2005);
- Entender como os utilizadores se comportam quando usam rotas dos trajetos como apoio visual e analítico em cenários de excesso de informação (trajetos) num mapa.

De seguida, é explanado como é que as sessões de avaliação com utilizadores foram executadas.

#### 4.2 Estudo com utilizadores

Esta seção descreve o estudo realizado com utilizadores para avaliação do sistema. É feita uma caracterização dos participantes no estudo, uma descrição do aparato, da aplicação de testes que foi utilizada nas sessões de avaliação e dos tipos de tarefas com o objetivo de aferir o desempenho das pessoas e respetiva usabilidade do sistema.

##### 4.2.1 Participantes

Foram selecionadas 25 pessoas, que se voluntariaram para realizar os testes com utilizadores. Destas, 15 eram homens e 10 eram mulheres, com idades compreendidas entre os 22 e os 30 anos, sendo nesta estreita faixa etária que existiram pessoas disponíveis para fazer testes deste tipo.

Foi feito um questionário inicial para saber se os utilizadores tinham experiência em ler e analisar mapas e concluiu-se que cerca de 90% dos participantes já usou aplicações como o *Google Maps* (“Google Maps,” n.d.), *Moovit* (“Moovit,” n.d.), *Waze* (“Waze - GPS, mapas e trânsito,” n.d.). O uso típico dessas aplicações foi para saber como chegar a um destino, a partir das várias alternativas de percursos apresentadas no mapa. Também foi reportada pelos participantes a utilização da visualização em tempo real do trânsito do *Google Maps*. Apenas dois participantes usam aplicações móveis no contexto de desporto, nomeadamente corrida, onde registam trajetos que depois podem ser vistos no mapa juntamente com um resumo com a velocidade, distância, e tempo total do percurso. A maioria dos utilizadores nunca fez

análise de trajetos sobre mapas. Aliás, numa das questões do questionário inicial (ver Apêndice F) foi pedida a análise de um mapa quanto à facilidade ou dificuldade de compreender um conjunto de trajetos, tendo a maioria respondido que era: “Demasiada informação e de difícil análise”.

## 4.2.2 Aparato

A avaliação com utilizadores decorreu em locais com baixo nível de ruído, boa luminosidade e sem quaisquer interferências de terceiros (biblioteca do Departamento de Informática da FCUL). A única interação com terceiros prevista seria no caso de o participante não querer responder a uma questão ou se tivesse alguma dúvida de português nas questões do teste. Cada sessão foi realizada por um participante, num computador portátil com um ecrã de 15.6” e resolução de 1920x1080 pixéis, com a luminosidade no máximo e foram desativadas todas as notificações do computador (atualizações de sistema, antivírus e outras) para evitar interrupções a meio do teste. A interação do utilizador com a aplicação de testes do sistema foi feita exclusivamente com o rato.

## Aplicação de testes

Foi implementada uma aplicação de testes sobre o novo protótipo do sistema de análise de trajetos, na qual cada participante realiza um conjunto de tarefas consoante o cenário proposto, sendo depois recolhidos dados sobre o seu desempenho, como o tempo de resposta e erros. Cada cenário é acompanhado por uma questão, cuja resposta deve ser dada através de cliques em componentes gráficos da aplicação como o marcador no mapa ou a rosa.

No decorrer de cada sessão de teste são guardados os seguintes atributos e respetivos valores:

- *UserID* – identificador único do participante, inserido antes de iniciar a sessão de avaliação;
- *QuestionID* – identificador da questão da tarefa;s
- *QuestionOrder* – ordem da questão da tarefa que surge no teste;
- *Errors* – número de erros cometidos pelo participante nas suas respostas. É contabilizado um erro sempre que um participante clique num componente gráfico do sistema que não é o correto tendo em conta a tarefa pedida na questão;
- *ExecutionTime* – tempo de resposta, em segundos, desde o momento em aparece uma questão da tarefa, até ao instante em que é dada uma primeira resposta (pode ser logo a mensagem com resposta correta);
- *ExecutionTimeWithErrors* – tempo de resposta, em segundos, desde o momento em que apareceu a questão da tarefa, até ao instante em que é dada a resposta certa (pode ter errado em primeiras tentativas);
- *ErrorsTime* – tempo, em segundos, desde o momento em que participante errou na pergunta da tarefa e carregou no *ok* da mensagem de alerta para tentar novamente outra resposta;
- *Skipped* – se o participante desistiu de responder, pois tentou por diversas vezes e não conseguiu saber a resposta, sendo forçado o carregamento da questão seguinte. Isto é feito com um *div HTML* oculto no cabeçalho da questão da pergunta, no canto superior direito, que só pode ser usado por quem conduz o teste.
- *Date* – data no formato “DD/MM/AAAA”, quando aparece a mensagem de resposta correta ou foi uma resposta *skipped*;

- *Hour* – tempo com formato “HH:MM:SS” quando aparece a mensagem de resposta correta ou foi uma resposta *skipped*.

Existem dois modos de obter os dados do desempenho dos participantes. A partir de um *log* num elemento *div* de *HTML* oculto no canto inferior direito da questão da tarefa. Quando ativa, a *div* é tornada visível, e surge uma *scroll bar* na margem direita do ecrã que permite revelar todo o conteúdo do *log*. O outro modo corresponde a gerar automaticamente um ficheiro de *log* com o nome *userID.json* no fim do teste, que contém os dados das respostas de todas as questões. Este último é mais prático e seguro, porque não é necessário copiar os dados do *log* na página para depois serem guardados num ficheiro, e evita-se a perda dos dados no elemento *div* se a aplicação de testes terminar abruptamente.

Cada resposta dada pelo participante às questões pedidas fica registada numa linha do ficheiro de *log*. Abaixo está um exemplo de *log* gerado a partir de um teste do utilizador com *UserID* 12 (*12.json*). Cada atributo que se encontra na primeira linha é delimitado por vírgulas (“,”). A segunda linha regista o começo do teste e a última linha quando terminou.

```
UserID, QuestionID, QuestionOrder, Errors, ExecutionTime, ExecutionTimeWithErrors, ErrorsTime, Skipped, Date, Hour
----- Início de teste 28/02/2019, 16:08:06 -----
12, 8, 1, 0, 74.4, 74.4, 0.0, normal, 28/2/2019, 16:09:24
12, 9, 2, 0, 334.4, 334.4, 0.0, normal, 28/2/2019, 16:15:00
12, 7, 3, 0, 295.7, 295.7, 0.0, normal, 28/2/2019, 16:20:01
12, 4, 4, 0, 195.7, 195.7, 0.0, normal, 28/2/2019, 16:23:20
12, 5, 5, 1, 51.7, 54.3, 2.5, normal, 28/2/2019, 16:24:16
12, 10, 6, 0, 86.3, 86.3, 0.0, normal, 28/2/2019, 16:25:50
12, 6, 7, 0, 161.0, 161.0, 0.0, normal, 28/2/2019, 16:28:32
12, 2, 8, 0, 23.6, 23.6, 0.0, normal, 28/2/2019, 16:29:02
12, 3, 9, 0, 165.6, 165.6, 0.0, normal, 28/2/2019, 16:31:50
12, 1, 10, 0, 148.2, 148.2, 0.0, normal, 28/2/2019, 16:34:20
----- Fim de teste 28/02/2019, 16:34:23 -----
```

Do ficheiro *de log* conseguimos saber que o participante número 12 demorou 26 minutos e 16 segundos (início às 16:08:06 e fim às 16:34:23) no dia 28 de fevereiro de 2019 a fazer o teste completo. Na linha que se apresenta a sublinhado verificamos que a quinta questão do teste tem o identificador 5 e que o participante errou uma vez, com tempo total de resposta 54.3 segundos, sendo que 2.5 segundos correspondem ao tempo que demorou até aceitar a mensagem de erro e tentar de novo responder à questão. Não foi necessário forçar o carregamento para a questão seguinte (valor “normal” na coluna “Skipped”).

### 4.2.3 Tarefas

No contexto da análise de trajetos por parte de um analista foi criado um conjunto de tarefas para exercitar e testar as novas funcionalidades do sistema, tendo sido definidos 4 tipos de questões:

- SEMAP (*SElect Marker Pendular*) – selecionar o marcador no mapa que corresponde ao par de rosas onde podemos identificar movimentos pendulares;
- FISE (*FIlter SElect*) – criar cópias a partir de uma rosa inicial e aplicar filtros em cada uma das rosas e, de acordo com a tarefa pedida, selecionar a rosa que respeite a condição da questão;
- SERO (*SElect ROse*) – selecionar, de entre um conjunto de rosas, aquela que corresponde ao marcador que aparece no ecrã;
- INSE (*INspect and SElect*) – inspecionar um local com trajetos no mapa, dentro de um limite a tracejado, e identificar o ponto por onde passam os trajetos que respeitem a condição da questão.

Foram formuladas 10 questões associadas a um conjunto de tarefas para testar o sistema. Cada utilizador respondeu às mesmas perguntas, porém estas surgiram por ordem aleatória para garantir que cada participante responde a uma sequência de perguntas diferente da dos testes anteriores.

As questões do tipo SEMAP têm por objetivo a identificação de movimentos pendulares dado um conjunto de três pares de rosas desenhadas no mapa. O utilizador deve validar a resposta com um duplo clique no marcador que considerar ser aquele onde está localizado o local onde o respetivo par de rosas mais se aproxime de um movimento pendular. Foram definidas duas questões deste tipo.

A Figura 4.1 mostra um exemplo deste tipo de questão, em que a resposta correta é o marcador 11 porque é aquela que apresenta rosas inversas (período manhã/tarde). As rosas do marcador 12 ficam automaticamente excluídas porque não se encontra nenhum padrão, e nas do 13 podem existir algumas dúvidas, mas devido à dispersão das direções dos trajetos, também se podem excluir.

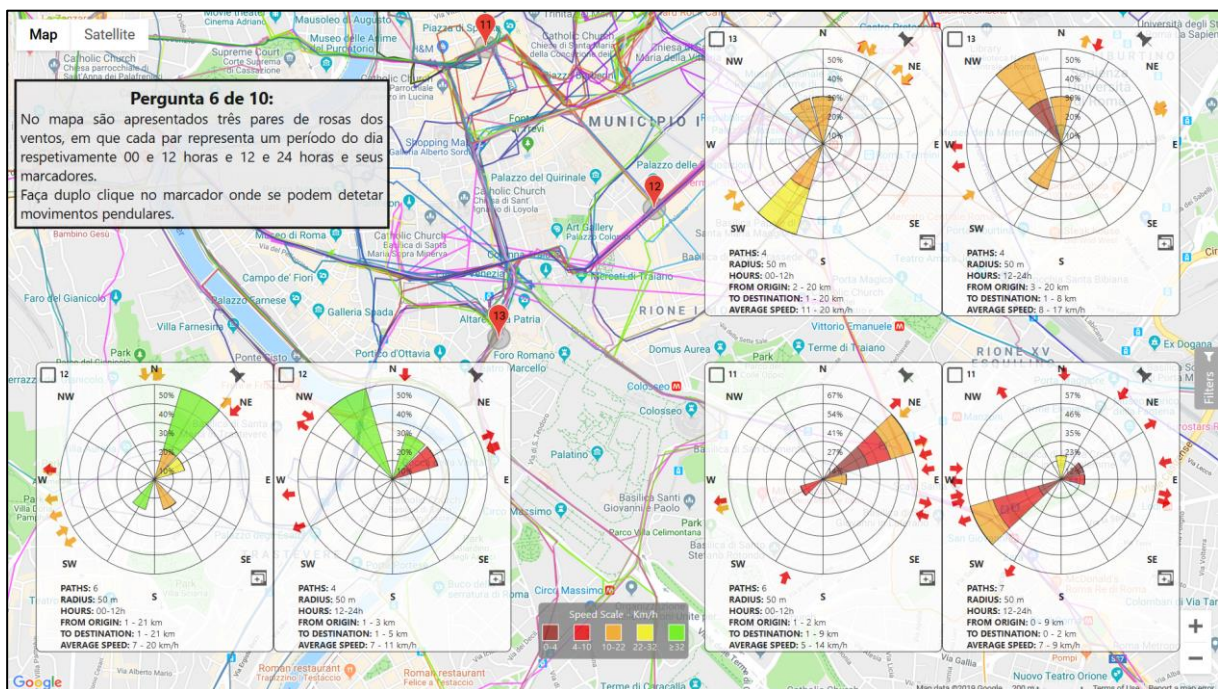


Figura 4.1 – Questão do tipo SEMAP que apareceu durante os testes de utilizador.

Em relação às questões do tipo FISE, pretendem-se criar mais duas ou três rosas duplicadas a partir do mesmo marcador no mapa e de uma rosa inicial, e aplicar filtros em cada uma de acordo com condições dadas na tarefa. A validação da resposta é feita através de um duplo clique na rosa que o utilizador considerar correta. Foram criadas três questões deste tipo.

A Figura 4.2 exemplifica uma questão deste tipo em que é pedido para ser escolhida a rosa que tem trajetos com maior velocidade média. A resposta correta é o marcador que está selecionado e sombreado a cinza, porque apresenta uma velocidade média entre os 11-17 km/h; nas outras rosas o valor é inferior.

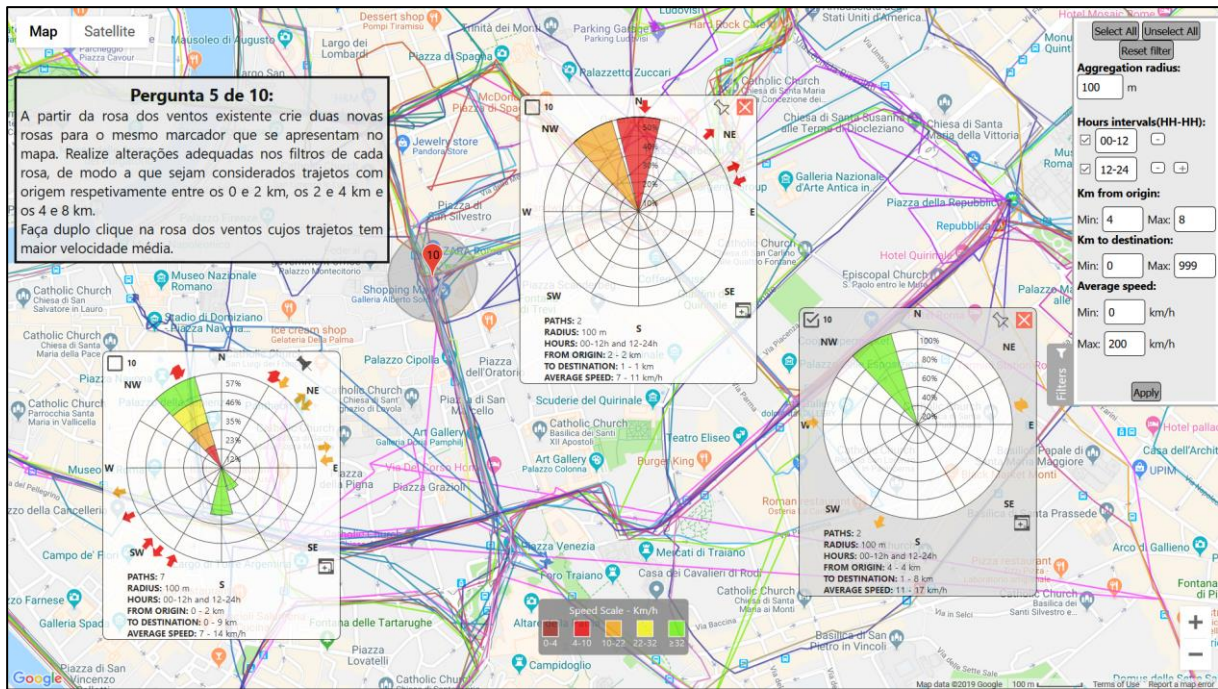


Figura 4.2 – Questão do tipo FISE que apareceu durante os testes de utilizador.

Para as questões do tipo SERO existe um marcador no mapa e três rosas dos ventos. O utilizador deve, a partir dos trajetos que intersectam a região de agregação em torno do marcador, escolher a rosa dos ventos correta, fazendo duplo clique. Foram definidas duas questões deste tipo.

A Figura 4.3 demonstra este tipo de questão, cujo marcador correto é o 13 visto que é o único que tem as pétalas localizadas na mesma direção da interseção de trajetos na região de agregação. A rosa 15 é logo excluída porque as pétalas estão localizadas em direções em que não existem quaisquer trajetos intersectados. A que pode gerar confusão é a 14, mas as suas pétalas na direção SW não têm correspondência com os trajetos que intersectam a região de agregação.

Por fim, nas questões do tipo INSE é pedido ao utilizador que entre em modo exploratório dentro de uma área delimitada a tracejado no mapa. Pretende-se testar a sua capacidade de análise através da colocação de rosas dos ventos em vários locais por onde passam trajetos. Após a visualização das rosas, o utilizador pode movimentá-las até achar o local no mapa que respeite as condições do enunciado da tarefa, fazendo duplo clique num marcador de rosa. Foram criadas três questões deste tipo.

No exemplo da Figura 4.4 é preciso identificar a zona por onde passam veículos com a velocidade média mais baixa entre as 14 e as 21 horas. Caso o marcador se encontre numa zona considerada incorreta, a aplicação regista o erro e mostra uma caixa de aviso (*alert box*) com essa informação, podendo o utilizador tentar de novo. A questão seguinte é carregada após o clique no botão 'Ok' de uma *alert box*, sempre que é dada uma resposta certa.

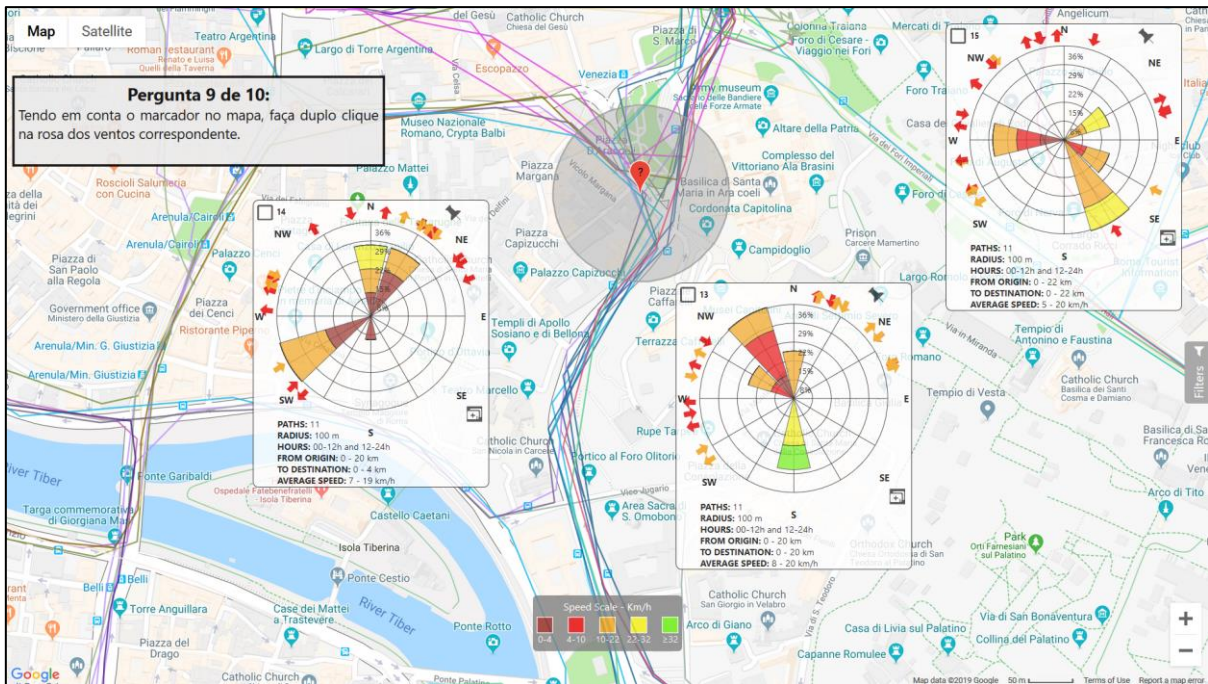


Figura 4.3 – Questão do tipo SERO que apareceu durante os testes de utilizador.

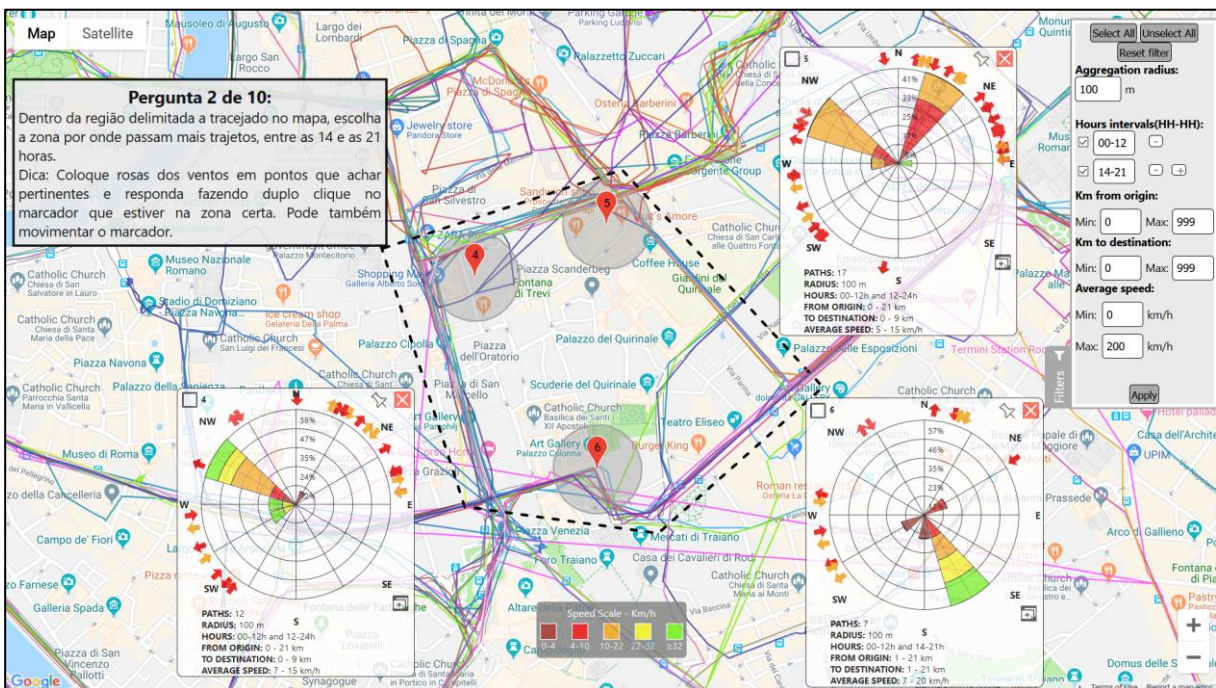


Figura 4.4 – Questão do tipo INSE que apareceu durante os testes de utilizador.

A Tabela 5 mostra a correspondência entre os vários identificadores das questões dos testes e a descrição correspondente.

Tabela 5 – Questões do teste e respectivas descrições

Tipo de questão	Descrição das questões
SEMAP	1. No mapa são apresentados três pares de rosas dos ventos, em que cada par representa um período do dia respetivamente 00 e 12 horas e 12 e 24 horas e seus marcadores. 1.1. Faça duplo clique no marcador onde se podem detetar movimentos pendulares. 1.2. Faça duplo clique no marcador onde <u>não</u> se podem detetar movimentos pendulares.
FISE	2. A partir da rosa dos ventos existente crie duas novas rosas para o mesmo marcador que se apresentam no mapa. Realize alterações adequadas nos filtros de cada rosa, de modo a que sejam considerados trajetos com ... 2.1. origem respetivamente entre os 0 e 2 km, os 2 e 4 km e os 4 e 8 km. Faça duplo clique na rosa dos ventos cujos trajetos tem maior velocidade média. 2.2. faixas horárias respetivamente entre as 00 e 08 horas, as 08 e 16 horas e as 16 e 24 horas. Faça duplo clique na rosa dos ventos cuja faixa horária passam pelo menos quatro trajetos e com maior velocidade média. 2.3. com faixas horárias respetivamente entre as 00 e 06 horas, as 06 e 12 horas, as 12 e 18 horas e as 18 e 24 horas (Nota. São criadas 3 rosas).
SERO	3. Tendo em conta o marcador no mapa, faça duplo clique na rosa dos ventos correspondente. 3.1. Cenário 1: com poucos trajetos. 3.2. Cenário 2: com muitos trajetos.
INSE	4. Dentro da região delimitada a tracejado no mapa, escolha a zona por onde ... 4.1. passam mais trajetos entre as 14h e as 21h. 4.2. passam mais trajetos e com origem mais distante, entre as 12h e 24 horas. 4.3. passam mais trajetos com velocidade média mais alta, entre as 00 e as 12 horas.

#### 4.2.4 Procedimento

Cada sessão de teste com um participante seguiu um guião, com os seguintes pontos:

1. Apresentação do sistema, enquadramento no contexto de análise de trajetos;
2. Preenchimento de questionário inicial e de consentimento informado, e explicação de como se processa a avaliação e como é feita a recolha e tratamento de dados;
3. Treino com a aplicação;
4. Avaliação da aplicação (teste);
5. Questionário final de usabilidade.

O primeiro ponto do guião consistiu na apresentação do trabalho e do que foi desenvolvido e explicada a necessidade de visualização de dados agregados devido à grande quantidade de dados que dificulta o processo de análise. Foi mencionado o tipo de trabalho dos analistas, como o estudo dos movimentos numa cidade e a resolução de problemas relacionados com trânsito em áreas urbanas.

No ponto dois explicou-se como é feita a recolha de dados a partir de um folheto informativo (ver Apêndice G) e pediu-se o consentimento informado, de acordo com as regras da comissão de ética para a proteção de dados (ver Apêndice H), para a recolha de informações como o género e a idade. Posto isto, o utilizador respondeu a um inquérito inicial *online*, elaborado no *Google Forms* (ver Apêndice F), para

obter dados pessoais básicos, informar sobre a experiência com trajetos, e recolher a opinião relativamente às características que o utilizador acha importantes na análise de trajetos (por exemplo, velocidade, distância percorrida, noção de origem/destino) e também sobre se considera existir excesso de informação num dado mapa com trajetos.

No terceiro ponto realizou-se um treino com a aplicação *windroseApp* do sistema. Foi mostrada uma imagem, plastificada numa folha A4 como a apresentada na Figura 4.5, com a interface da aplicação de análise e duas rosas dos trajetos criadas num ponto em que passavam trajetos num mapa. Cada elemento principal da aplicação: marcador, região de agregação, rosas e seu conteúdo, painel de filtros e escala de cores foram legendados, permitindo que o participante entendesse os conceitos e as funcionalidades da interface do sistema. De seguida, o participante utilizou a aplicação livremente, com a finalidade de facilitar a aprendizagem e foram pedidas algumas tarefas de treino. Se surgissem dúvidas, estas eram esclarecidas, e se fossem dadas sugestões, estas eram anotadas.

O penúltimo ponto correspondeu à avaliação da aplicação que consistiu na resposta a 10 questões usando a aplicação de testes. Durante a realização dos testes, foi dada a possibilidade de o participante falar enquanto estava a pensar e, se necessário, foram dadas dicas simples, mas sempre sem dar qualquer solução. Este procedimento não afetou os resultados, uma vez que era feito antes da resposta poder ser dada. Após o teste, foi gerado um ficheiro em formato *JSON* com as respostas.

Para as questões *INSE*, nos casos em que os marcadores foram colocados num ponto muito próximo da zona correta, mas fora da área delimitada a tracejado, o erro detetado pela aplicação foi ignorado. Foi dito ao utilizador que colocasse o marcador numa zona em que a validação ocorresse corretamente. Nestes casos, foi descontado o erro do utilizador e subtraído o tempo decorrido desde o instante em que o falso erro ocorreu, até que foi dada a resposta correta.

Por fim, o último ponto da sessão de testes com utilizadores foi responder a um questionário de usabilidade (ver Apêndice I), que também foi elaborado no *Google Forms*, para avaliar o nível de usabilidade do sistema, segundo o método *SUS* (Bangor et al., 2005). No final, foi dada a chance aos participantes de sugerirem melhorias à interface ou novas funcionalidades. Além disso, questionou-se quais foram as dificuldades que sentiram durante o uso da aplicação e no final foi feito um agradecimento pela sua participação no teste.

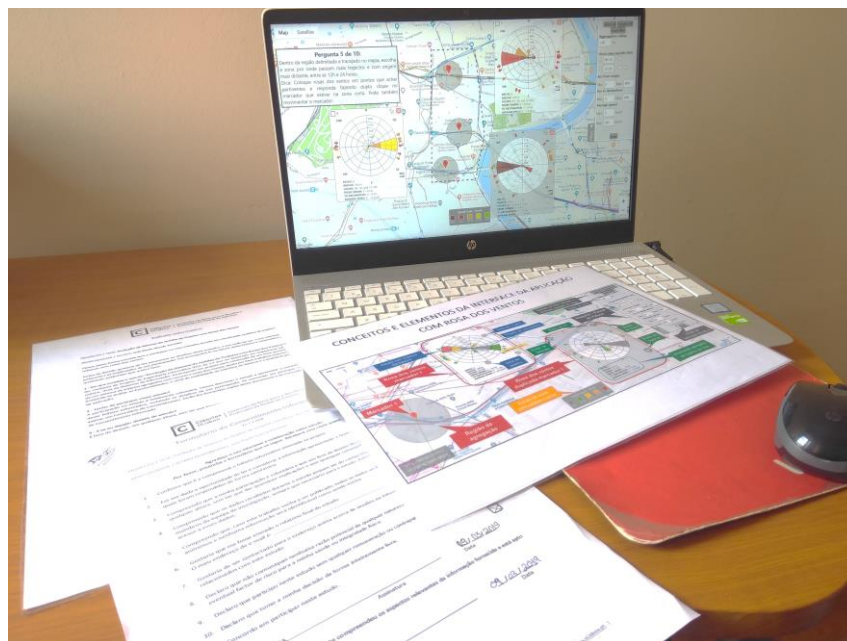


Figura 4.5 – Questão do tipo INSE que apareceu durante os testes de utilizador.

## 4.3 Resultados

Esta secção descreve os principais resultados dos testes da aplicação com utilizadores. O processamento dos dados dos ficheiros de *log* de cada utilizador e dos questionários inicial e de usabilidade foram feitos através do *RStudio* (RStudio, 2017), onde também foram construídos os gráficos respetivos.

### 4.3.1 Tempos de resposta

O tempo de resposta a todas as 10 questões do teste variou entre os 15 minutos e 2 segundos (participante número 9) e os 32 minutos e 7 segundos (participante número 4). O tempo médio dos testes foi de 23 minutos e 36 segundos.

Os gráficos seguintes mostram *box plots* dos tempos de resposta dos participantes às questões dos testes, em segundos.

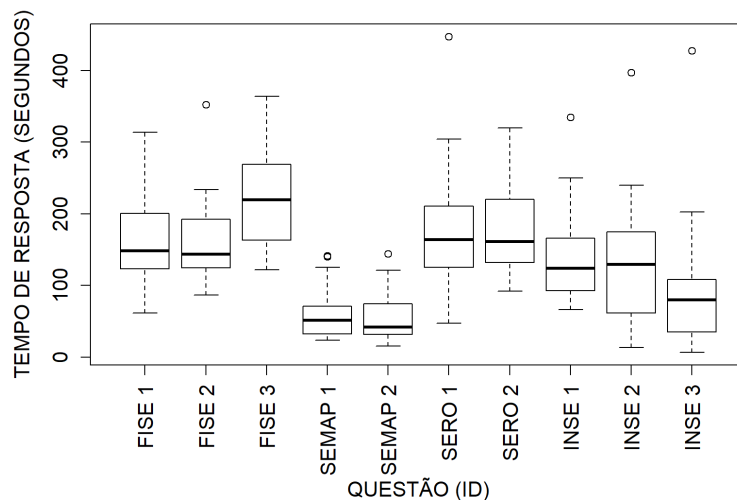


Figura 4.6 – Tempos de resposta dos participantes, por questão.

Na Figura 4.6 são apresentados os identificadores de cada questão, sendo que, por exemplo, *SERO 1* e *SERO 2*, do tipo *SERO*, estão associados os cenários 3.1 e 3.2, da Tabela 5. A questão que os utilizadores demoraram mais tempo a responder foi o *FISE 3* (cenário 3 do tipo *FISE*). Note-se que apesar dos identificadores serem fixos, a ordem das perguntas foi sempre aleatória para cada participante.

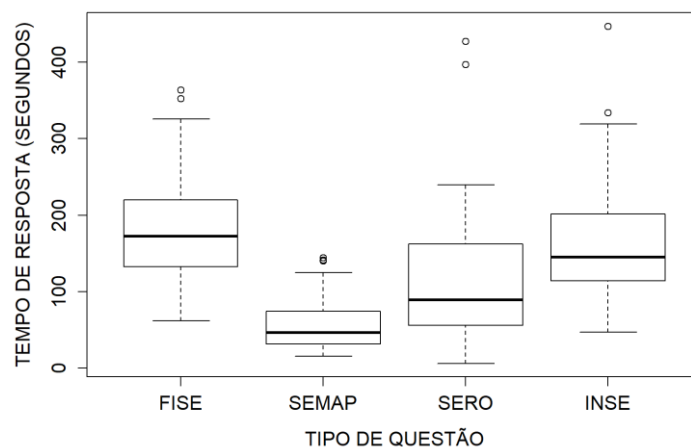


Figura 4.7 – Tempos de resposta dos participantes, por tipo de questão.

Na Figura 4.7 verifica-se que o maior tempo de resposta, independentemente dos vários cenários, correspondeu aos tipos *FISE* e *INSE*, com tempos máximos entre os 300-325 segundos. Seguiu-se o tipo *SERO* com tempos de resposta intermédios máximos de 220 segundos, e por fim o tipo *SEMAP*, com respostas mais rápidas de tempo máximo de 100 segundos.

Na Figura 4.8, por tipo de questão, ao separarmos os participantes por género, verificamos que o tempo de resposta foi relativamente semelhante nos tipos *FISE* e *SERO* em ambos. Contudo o tempo de resposta nas perguntas no tipo *INSE* e *SEMAP* foi menor nos participantes do sexo masculino que nos participantes do sexo feminino.

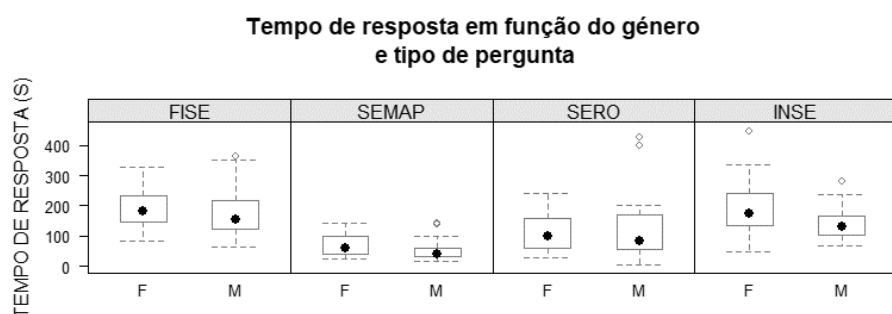


Figura 4.8 – Tempos de resposta dos participantes, por género e tipo de questão.

Na Figura 4.9 pode observar-se que as medianas dos tempos de resposta segundo a ordem de apresentação das questões aos participantes estão relativamente próximas.

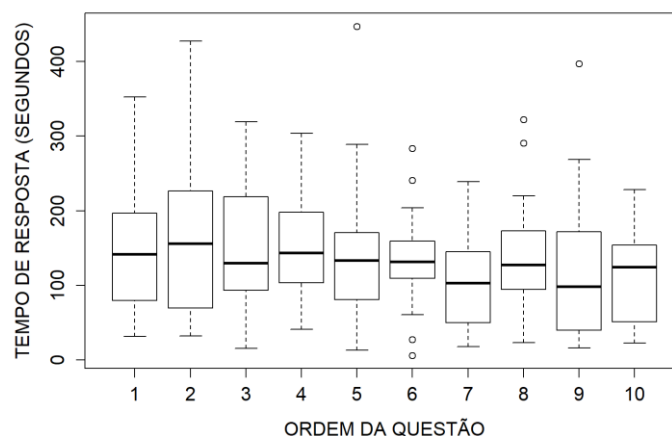


Figura 4.9 – Tempos de resposta dos participantes, por ordem de questão.

A Figura 4.10, apresenta o tempo de resposta por participante, por ordem crescente de mediana, podendo ser constatado que o mais lento (participante 4, com 1920 segundos) demorou 2 vezes mais tempo que o mais rápido (participante 9, com 920 segundos).

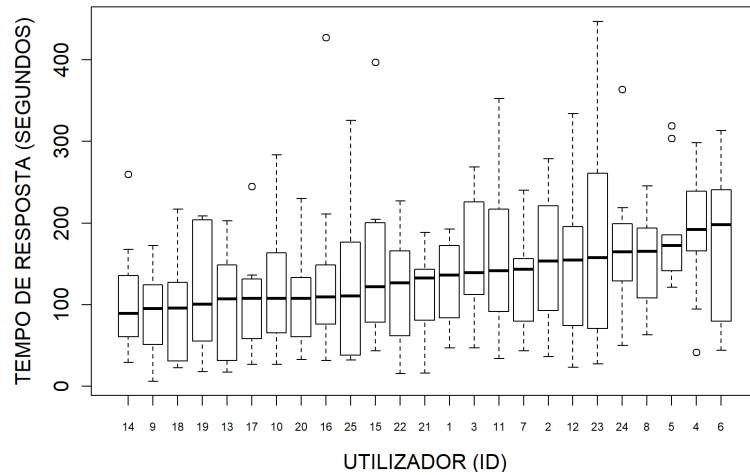


Figura 4.10 – Tempos de resposta, por participante.

### 4.3.2 Tempos de recuperação

Cada erro tem associado um tempo, desde o momento em que ocorreu o primeiro erro até ao instante em foi dada a resposta certa. De uma forma geral, os tempos de recuperação dos erros foram reduzidos, tendo variado entre 1,4 e 23,1 segundos (com mediana de 16 segundos), o que mostra que os utilizadores chegaram rapidamente à resposta correta da pergunta.

### 4.3.3 Erros

Seguem-se os resultados relativos ao número de vezes que um participante clicou num componente gráfico do sistema que não era a resposta correta.

No total ocorreram 36 erros de um total de 250 respostas, em 25 testes realizados. Isto dá uma média de 1,4 erros cometidos por participante em cada teste. 4 participantes conseguiram responder corretamente a todas as questões colocadas no seu teste. O número máximo de erros registados numa pergunta foram 2, sendo cometidos por 3 participantes e eram perguntas do tipo INSE.

A Figura 4.11 mostra que o número de erros durante as sessões de avaliação foram reduzidos, tendo o máximo de erros por questão sido 2, sendo que os participantes falharam mais na questões de tipo *FISE* 2 e *SERO* 1.

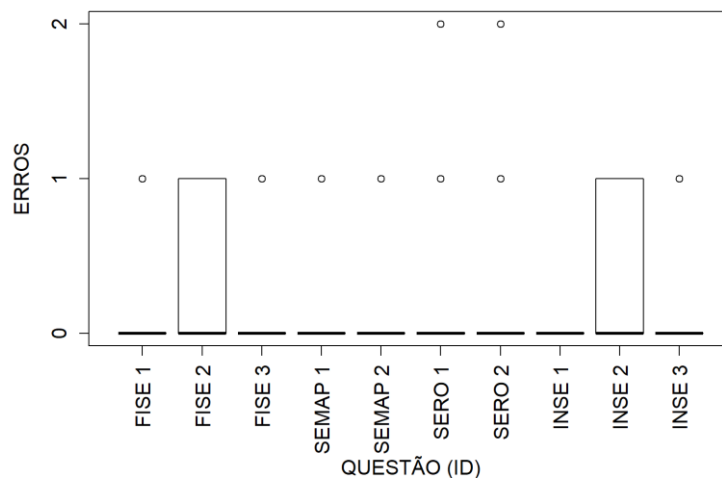


Figura 4.11 - Erros por questão.

A Figura 4.12 mostra que foram cometidos mais erros no tipo de questão SERO. Por género, os participantes do sexo masculino deram mais erros do que os do sexo feminino, como se pode ver na Figura 4.13.

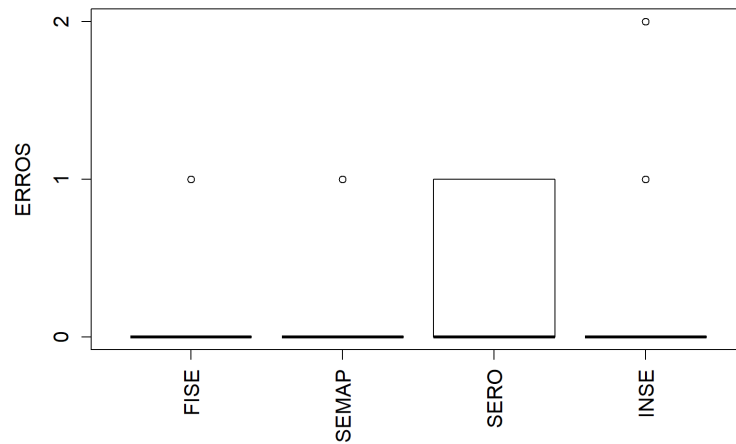


Figura 4.12 - Erros por tipo de questão.

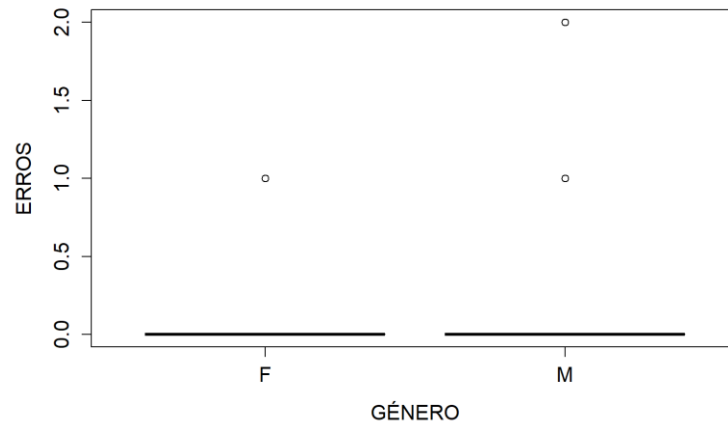


Figura 4.13 - Erros por género.

#### 4.3.4 Usabilidade do sistema

Nesta subsecção são mostrados os gráficos das classificações de usabilidade do sistema, pontuadas com base no questionário *SUS* (ver Apêndice I).

A Figura 4.14 apresenta as classificações globais de usabilidade, que seguiram a regra que faz uma ligação entre intervalos de pontuação e as classificações (Bangor et al., 2005). As pontuações globais de usabilidade foram calculadas de acordo com fórmulas do *SUS*, onde cada questão possui um peso definido para o cálculo da pontuação de usabilidade final.

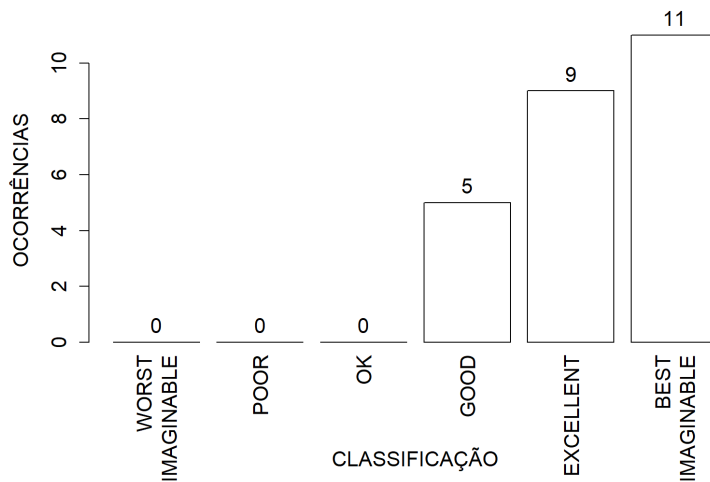


Figura 4.14 – Classificações de usabilidade do sistema.

Na Figura 4.15 é visível a distribuição da pontuação de usabilidade atribuída por género dos participantes. As medianas de ambos os géneros são semelhantes, existindo uma maior variação nos utilizadores do sexo masculino.

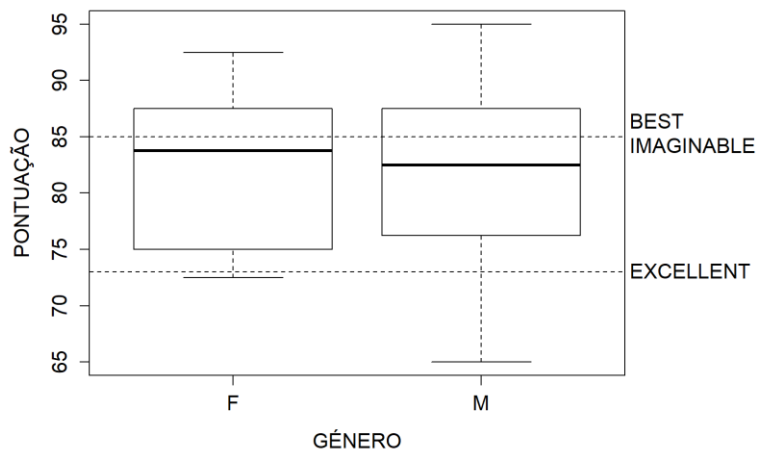


Figura 4.15 - Classificações de usabilidade do sistema, por género do participante.

A Figura 4.16 mostra as pontuações dadas pelos participantes às 10 questões do método *SUS*. Cada pergunta tem uma afirmação sobre a utilização do sistema e foi avaliada numa escala entre 1 a 5, relativamente ao nível de concordância do participante com a mesma. O valor 1 corresponde a discordância total e o valor 5 a concordância total. No Apêndice I, temos as 10 afirmações do questionário *SUS*.

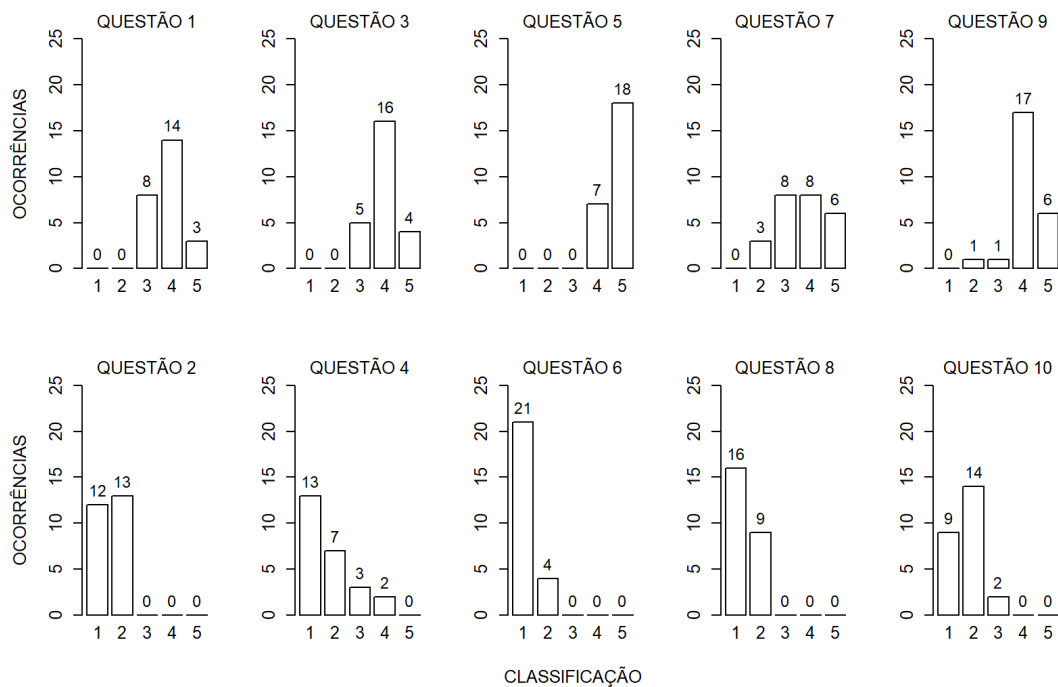


Figura 4.16 – Pontuações atribuídas em cada pergunta do questionário *SUS*.

As 10 afirmações têm como objetivo apurar o grau de concordância dos participantes relativamente ao sistema que acabaram de utilizar. A Figura 4.16 mostra as pontuações atribuídas a cada afirmação. Na primeira linha da figura as afirmações ímpares estão expressas de forma positiva, por outro lado as afirmações pares que estão na segunda linha correspondem a asserções com sentido negativo.

Verificou-se que nível de concordância dos utilizadores, relativamente às afirmações positivas (primeira linha) foi no geral de nível 4 com concordância, sendo que a questão 7 gerou menos consenso. Por outro lado, nas afirmações negativas (linha dois), o nível mais avaliado ficou-se entre o 1 e 2 de discordância.

## 4.4 Discussão

Nesta secção os resultados obtidos são discutidos e são tiradas algumas conclusões sobre a avaliação com os utilizadores. Também é feita uma comparação com os resultados do protótipo anterior do sistema, para verificar qual foi a evolução do sistema atual em relação ao anterior.

### 4.4.1 Tempos de resposta

Em relação aos tempos de resposta por cada questão, verifica-se que as do tipo *SEMAP* tiveram os tempos de resposta mais curtos com tempo máximo de 100 segundos. Logo, estas foram as mais fáceis e rápidas de responder. Neste tipo de questão pretendia-se analisar três pares de rosas e de seguida identificar qual deles se aproximava mais de um movimento pendular.

Seguem-se os tempos de resposta do tipo *SERO* com um máximo de 220 segundos. Para este tipo de questão foram criadas rosas que suscitassem dúvidas quanto à orientação das pétalas e número de trajetos agregados, o que levava algum tempo na resposta.

Por outro lado, registaram-se tempos de resposta elevados na ordem dos 300-325 segundos, nas questões do tipo *FISE* e *INSE*. As do tipo *FISE* implicavam criar rosas duplicadas, aplicar filtros e interpretar os

resultados de acordo com o enunciado da questão. As do tipo *INSE* envolviam uma análise exploratória do local que respeitasse as condições da questão. Estes tipos de questões implicavam um maior esforço e trabalho mais elaborado, e, como consequência, os participantes levaram mais tempo a responder.

Como demonstram os resultados da Figura 4.10, não foram detetados efeitos de aprendizagem, porque as medianas são muito próximas, desde a primeira até à última pergunta. Caso ocorresse efeito de aprendizagem, os valores das medianas teriam diminuído nas últimas perguntas.

O protótipo atual tinha questões de maior complexidade, em relação ao protótipo anterior, pois envolviam um maior fluxo de trabalho, uma vez que era necessário ter atenção em respeitar várias condições do enunciado. Isto levou a um maior esforço por parte dos participantes, o que se traduziu num maior tempo de resposta a este tipo de questões.

#### 4.4.2 Erros

Tal como referido nos tempos de resposta, o aumento do nível de complexidade das questões contribuiu para o incremento da ocorrência de erros, porque envolviam um maior fluxo de trabalho.

Analisando por tipo de questão, verificou-se que um dos erros nas respostas às perguntas do tipo *SEMAP* foi a incompreensão da representação dos movimentos pendulares nas rosas. Para verificarmos esse movimento pendular as duas rosas deveriam apresentar pétalas inversas.

Relativamente às respostas das questões do tipo *SERO*, o principal erro foi não associar a direção das pétalas à direção do local de interseção da região de agregação nos trajetos, pois alguns utilizadores estavam a ter em conta o local de onde vinham os trajetos e não o ponto de interseção. Também o facto de os trajetos terem cores parecidas, levou a alguns utilizadores a pensar que se tratava do mesmo trajeto. Estas duas situações levaram aos participantes a gastar mais tempo na decisão da escolha da rosa correta.

As questões do tipo *FISE* apresentavam o enunciado mais extenso, o que envolvia um maior fluxo de trabalho. A maioria dos erros cometidos ocorreu na decisão final entre 2 rosas, que apresentavam dois intervalos de velocidade. Deveria ser escolhida a rosa com a média mais alta desses intervalos e alguns utilizadores responderam apenas tendo em conta o valor máximo do intervalo. Por exemplo, a primeira rosa variava entre 5-14km/h e o segunda entre os 7-13km/h. Apesar de a primeira apresentar um valor máximo de 14km/h, em termos médios a rosa com valores entre os 7-13km/h tinha trajetos com maior velocidade.

Por fim, nas questões do tipo *INSE*, surgiram erros quando os utilizadores não respeitaram os limites a tracejado onde deveria ser colocado o marcador e a região de agregação, ou quando faltou explorar um local do mapa. Também quando dentro desse limite, um determinado local podia apresentar mais trajetos, contudo quando se aplicava um filtro a limitar uma faixa horária, esse local poderia passar a ser o que tem menos trajetos, o que causou dúvidas em alguns participantes.

Em relação ao protótipo anterior, houve um aumento em 8% no número de erros, mas esta situação já era previsível dado o aumento da dificuldade das questões.

#### 4.4.3 Usabilidade do sistema

Os resultados de usabilidade, no global, revelaram que os participantes consideraram o sistema simples de utilizar, tal como mostra a Figura 4.14. De realçar que maioria das classificações (75%) foi *Excellent* e *Best Imaginable*, e os restantes 25% foram *Good*.

Em cada afirmação do teste *SUS*, da Figura 4.16, verificou-se que das asserções positivas (números ímpares) a que gerou menos acordo foi a número 7 (“Acredito que a maioria das pessoas facilmente

aprenderá a usar a aplicação.”). Por outro lado, das afirmações negativas (números pares) a que gerou menos consenso foi a número 4 (“Precisaria da ajuda de uma pessoa com conhecimentos técnicos para utilizar a aplicação”). Estas afirmações refletem, em certa medida, situações inversas uma da outra, e têm interpretação subjetiva, daí ter sido questionado por alguns participantes em que situações estas deveriam ser consideradas.

Por um lado, no caso da afirmação 7, alguns utilizadores questionaram se se deveria considerar qualquer pessoa ou alguém que costuma a usar aplicações semelhantes (por exemplo, a faixa etária idosa não usa muito tecnologias). Foi indicado para considerar um utilizador de tecnologias ou aplicações. Uma hipótese, para alguns utilizadores não concordarem com a afirmação 7 poderá ter sido por considerarem que este sistema é mais adequado para analistas de tráfego urbano aquando do trabalho análise de trajetos e não por qualquer pessoa.

Por outro lado, na afirmação 4 os participantes perguntaram se a questão deveria ser respondida no contexto da sessão de teste onde foi dada uma breve explicação do sistema e suas funcionalidades (ou existência de um manual de instruções) ou se nada é dito e autonomamente o utilizador começa a utilizar o sistema. Foi indicado para considerarem que depois de dada uma explicação do sistema e suas funcionalidades ou ter um guia com instruções se conseguiriam mexer no sistema sozinhos sem necessitar de apoio de um técnico. Podemos supor que os participantes que concordaram com a asserção 4 assumiram que ter um guia ou uma breve explicação seria suficiente para mexer no sistema, enquanto que os que discordaram, consideraram que ainda teriam dúvidas.

#### **4.4.4 Influência de experiência prévia**

A maioria dos utilizadores já utilizou aplicações que usam mapas e percursos, como o *Google Maps*, para saber o trânsito na cidade e o *Moovit* (“Moovit”) e *Waze* (“Waze - GPS, mapas e trânsito”) para saber as alternativas de como chegar a um destino, mas não de um ponto de vista de análise de grandes quantidades de dados de trajetos.

Os dados mostram que aqueles que já utilizavam aplicações de mapas e trajetos não tiveram melhores resultados nos testes, e cometeram um número semelhante de erros em relação aos participantes que nunca utilizam frequentemente aplicações deste tipo. De facto, foi registado um fenómeno interessante, o grupo de participantes sem contacto prévio com aplicações de trajetos e mapas teve um tempo médio de resposta 7.5 segundos mais rápido do que o grupo que já esteve em contato com esse tipo de sistemas. Se considerarmos a mediana do tempo de resposta, foi de 121.7 segundos para os participantes que não tiveram contacto com essas aplicações, enquanto que os que disseram que tinham experiência, o valor subiu para 131.4 segundos.

#### **4.4.5 Fluxos de trabalho**

O modo como os participantes interagiram com as funcionalidades do sistema variou um pouco, e dependeu sobretudo do tipo de tarefa. Um aspeto curioso da tarefa *SERO*, foi um dos participantes ter deslocado as rosas sobre o local dos trajetos para verificar qual delas se ajustava melhor ao local no mapa e a partir daí escolher a rosa correta.

Relativamente às questões *FISE*, a tarefa envolvia 3 etapas: criar rosas duplicadas, aplicar filtros a cada uma, e depois decidir com base no cenário pedido. A maioria criou os duplicados sempre a partir da rosa original e depois selecionou cada rosa obtida e ajustava o filtro de acordo com o pedido. Existiram 3 perguntas associadas a este tipo. Como podemos observar na Figura 4.6, as *FISE* 1 e 2 apresentaram tempos de resposta com valores de mediana aproximados, mas o cenário do *FISE* 3 apresentou um desvio de mediana positivo. Esta diferença justifica-se devido ao fato de no cenário 1 e 2 serem pedidos

para criar dois duplicados, enquanto que no cenário 3 pediu-se para criar três duplicados, o que contribuiu para aumentar o tempo de resposta. Notou-se que quando era pedido para aplicar os filtros que manipulavam intervalos de tempo, a maioria dos utilizadores não adicionaram mais faixas horárias (botão +), apenas modificaram os valores que já existiam nas faixas horárias e aplicavam o filtro.

Por fim, nas *INSE* os participantes tinham de explorar os trajetos à volta de uma região com linha delimitada a tracejado e decidir com duplo clique no marcador, o local que respeitasse a condição da tarefa. Depois de criada a primeira rosa, na globalidade os utilizadores usavam a funcionalidade de movimentação do marcador para atualizar o conteúdo das rosas, e não recorreram ao apagar e criar nova rosa para analisar um novo local. Foi nesta tarefa que se verificou muito o uso dos botões de marcar/desmarcar do painel de filtros, para aplicar o mesmo filtro a todas as rosas. Também na Figura 4.6 nota-se uma diferença nos tempos do *INSE* 1 e 2 para o *INSE* 3, devido à diferença no número de trajetos que passavam dentro do limite. O valor de mediana e de tempo total de resposta do cenário 3 é mais baixo que os cenários 1 e 2, porque nesse limite estavam menos trajetos.

Tanto as *INSE* como as *FISE* não eram de resposta imediata, o que aumentou a complexidade das questões visto que tinham de respeitar certas condições como: “ escolha a zona por onde passam mais trajetos e com origem mais distante, entre as 12h e 24 horas”. Por exemplo havia locais onde havia rosas com mais trajetos, mas os seus pontos de origem não eram os mais distantes.

Nas primeiras respostas dadas, os utilizadores tinham a tendência para olhar e usar o inspetor, em vez de olharem para a informação no rodapé de cada rosa. Porém, verificavam de seguida que, para a tomada de decisão final, tinham de recorrer à informação do rodapé e dos elementos gráficos da rosa. Isto poderá ter sido devido à informação apresentada no rodapé não estar muito destacada, podendo eventualmente passar despercebida ao início.

#### **4.4.6 Comparação com os resultados do protótipo anterior**

O número de participantes neste estudo foi de 25 utilizadores, enquanto que no estudo anterior tinha sido de 21 utilizadores. Fazendo uma comparação com o protótipo anterior, constata-se que, com o novo sistema, os participantes demoraram mais tempo a responder e cometeram mais erros, mas há que ter em conta que, propositadamente, foram incluídas mais funcionalidades e as tarefas passaram a ser mais exigentes e complexas.

As tarefas semelhantes nos testes dos dois protótipos do sistema de análise de trajetos foram as do tipo *SERO* e *INSE*. Neste novo protótipo verificou-se que os utilizadores demoraram mais 18,7 segundos em média a responder a tarefas do tipo *SERO* do que no protótipo anterior, possivelmente explicado por nos testes do protótipo anterior terem sido considerados menos trajetos e menos pétalas nas rosas.

O mesmo aconteceu nas tarefas do tipo *INSE*, em que houve um aumento médio de 23,5 segundos nos tempos de resposta. Uma explicação para este aumento deve-se ao facto de as tarefas pedidas terem de respeitar duas condições, enquanto que nos testes do protótipo anterior era pedida uma condição. Também se verificou os participantes recorrerem muito à movimentação do marcador para a atualização do conteúdo da rosa, sendo que alguns exploraram de forma muito minuciosa todos os locais onde existiam trajetos. Por contrapartida, no outro protótipo era necessário eliminar e criar uma nova rosa sempre que pretendiam analisar outro local, o que não incentivava exploração cuidadosa no mapa.

Também foi registado um aumento de erros, que passaram de 9 para 36, o que demonstra o aumento das dificuldades sentidas pelos participantes. Pode supor-se que as perguntas realizadas no protótipo anterior eram demasiado fáceis ou então que esta nova versão do protótipo complicou a usabilidade do sistema.

Por isso, confrontaram-se as respostas do questionário SUS dos dois protótipos do sistema. Quanto à afirmação “5. As várias funcionalidades da aplicação estão muito bem integradas” no novo protótipo a globalidade das respostas situou-se no nível 5 e depois 4, enquanto que no protótipo anterior variou entre o nível 3 e 5.

Em relação à afirmação “6. A aplicação apresenta um elevado grau de incoerência”, os resultados em ambos os protótipos foram muito semelhantes, sendo avaliados no nível 1. Isto sugere que a usabilidade não ficou reduzida com a inclusão das novas funcionalidades e interfaces gráficas.

Do ponto de vista de facilidade de utilização, nota-se que os utilizadores sentiram mais dificuldade. Na afirmação “3. Achei a aplicação fácil de se usar”, os utilizadores pontuaram o novo sistema entre o nível 3 e 5, estando a maioria situada no grau 4. Por outro lado, na versão anterior a maioria das classificações estava entre o nível 4 e 5.

Verificou-se alguma discordância na questão “7. Acredito que a maioria das pessoas facilmente aprenderá a usar a aplicação” com distribuição equitativa entre os níveis 2 e 5 no novo protótipo, enquanto no anterior a maioria tinha classificado entre os níveis 3 e 5. Apesar disso, os utilizadores sentiram-se mais confiantes a usar esta nova versão, dado que para a asserção “9. Senti-me muito confiante ao usar a aplicação” deram melhor pontuação (maioria em 4) à nova versão em relação à anterior (variou entre 2 e 4).

## **4.5 Sumário**

Neste capítulo foi descrito o processo de avaliação com utilizadores do sistema de análise de trajetos, incluindo a caracterização dos participantes, a construção da aplicação de testes, o tipo de tarefas a avaliar, e o procedimento utilizado nas sessões de avaliação. De seguida foram apresentados e discutidos os resultados relativamente aos tempos de respostas, erros, e usabilidade do sistema. Por fim, foi feita uma comparação entre os resultados do protótipo atual e anterior.

## Capítulo 5

### Conclusões

Este capítulo faz uma apreciação global do trabalho desenvolvido, incluindo as principais contribuições, competências adquiridas, dificuldades encontradas e ideias para trabalho futuro.

#### 5.1 Principais contribuições

Começando pelo primeiro objetivo do sistema (evolução da infraestrutura da aplicação *web*), foi reformulada a estrutura dos serviços *web* com uso da ferramenta *Slim Framework*. Isto consistiu em redefinir e adicionar um conjunto de serviços *web* para acesso à base de dados geográfica. Além disso, foi alterado o modo de carregamento da aplicação de análise de trajetos, para que não fosse necessário estar sempre a inserir um ficheiro (*GPX*) a partir da página de seleção de trajetos (se existirem trajetos na base de dados, a página de análise de trajetos abre de imediato).

O primeiro objetivo incluiu também a criação de novas funcionalidades. Deste modo, para garantir a flexibilidade do sistema, acrescentaram-se novas funções e reformulou-se a estrutura da biblioteca *WindRose*. O Apêndice D tem a descrição de cada método da biblioteca.

A primeira funcionalidade (movimentação do marcador com atualização automática do conteúdo da sua rosa) foi cumprida. O código da biblioteca *WindRose* foi modificado e implementada a função de movimentação do marcador (*updateRose*), a partir da criação de eventos do *JavaScript*. Sempre que se movimenta o marcador da rosa é detetado um evento que procede à atualização dos dados da rosa referentes à nova posição.

Saltando para a terceira funcionalidade (criação de várias rosas dos ventos num mesmo local), esta também foi cumprida, passando a existir a possibilidade de aplicar filtros diferentes (por exemplo, períodos da manhã e tarde) a cada uma das rosas. Isto é relevante para o trabalho dos analistas pois permite a análise de movimentos pendulares. Esta funcionalidade foi concretizada a partir da utilização de eventos do *JavaScript*, da criação de um novo método, e da inclusão de um botão para duplicar uma rosa. Do ponto de vista de análise da velocidade dos trajetos, a escala de cores passou a ser divergente de acordo com a lógica do trânsito (cores vermelhas, velocidade lenta, cores verdes, trânsito fluído). Ainda ao nível da interface gráfica, foi resolvido o problema de sobreposição de setas de origem e destino quando existe uma quantidade elevada de trajetos que intersectam a área de agregação da rosa. Agora, ao passar o cursor do rato numa seta de origem ou destino é realçada a seta de destino ou origem respetiva.

A quarta e última funcionalidade (novos filtros de espaço e tempo) foi cumprida através da inclusão de um componente dinâmico em que é possível modificar e adicionar várias faixas horárias. Sempre que for adicionada uma nova faixa e aplicado um filtro, todas as faixas são ordenadas automaticamente. Foi também alterada a interface do painel de filtros e inspetor, para melhorar e destacar botões, e sinalizar eventuais incoerências de valores nos filtros quando várias rosas estão selecionadas.

Para o objetivo dois (avaliação do sistema com teste de utilizadores), foi implementada uma aplicação de testes para recolher dados sobre o desempenho dos participantes na análise de trajetos com rosas. Os

resultados dos testes sugerem que a aplicação de análise de trajetos apresenta um grau de usabilidade elevado e que as rosas permitem cumprir várias tarefas de análise de trajetos que são propostas no contexto do trabalho dos analistas para a tomada de decisões. Além disto, os resultados apresentaram um reduzido número de erros.

Adicionalmente, contribuí para a divulgação deste trabalho na comunidade científica através da participação em diferentes eventos, tais como: Encontro de Ciência e Tecnologia 2018 que decorreu a 4 de julho de 2018, no Centro de Congressos de Lisboa (ver Apêndice A) e o *JobShop* Ciências 2019, a 11 de abril de 2019, na Faculdade de Ciências, onde fiz demonstrações do sistema de análise de trajetos. Também colaborei no *3rd LASIGE Workshop*, que se realizou a 7 de junho de 2018, onde realizei um póster alusivo ao sistema desenvolvido (ver Apêndice B). Tive ainda a oportunidade de mostrar o sistema no seminário “Lisboa a caminho de ser uma cidade inteligente”, onde foram abordados temas relacionados com a criação de sistemas de apoio à decisão eficientes para a rentabilização de recursos e para a melhoria da qualidade de vida das pessoas, por exemplo, na gestão do tráfego urbano.

Por último, foi realizada a escrita de um artigo em inglês, ainda não submetido sobre o anterior e atual protótipo do sistema (ver Apêndice C).

## 5.2 Competências adquiridas

Ao longo do trabalho foram ganhas várias competências, experiências e capacidades.

Em termos de programação, foi aprofundado o conhecimento em *JavaScript*, *JQuery*, *PHP* e *HTML*, pois foi necessária uma familiarização com o código da aplicação *web* e da biblioteca *WindRose*. Pela primeira vez usei o *SVG* para a criação de elementos gráficos e utilizei eventos em *JavaScript* e manipulei ficheiros *GPX* em pormenor.

A utilização do *RStudio* permitiu melhorar os conhecimentos da linguagem *R*, como a criação de gráficos com informação estatística e interpretação dos dados.

Ao nível da base de dados, foi utilizado o *PostGIS*, que é uma base de dados geográfica onde pude utilizar funções geográficas para manipular dados de trajetos.

A reformulação dos serviços *web* com a *Slim Framework* permitiu formular de forma mais eficiente pedidos e respostas entre o cliente e os serviços *web*, o que contribuiu para um menor tempo de resposta. Aprendi a configurar um servidor aplicacional, instalar e configurar uma base de dados geográfica. Também foi adaptado o sistema para correr em sistema *Linux*, para além de *Windows*.

A capacidade de adaptação foi exercitada durante o projeto, uma vez que perante imprevistos e dificuldades, foi exigida alguma capacidade de ajuste para que o trabalho prosseguisse.

Por fim, a experiência de realizar testes com utilizadores foi nova e muito interessante, pois envolveu organização e gestão do espaço e tempo, com a criação de um guião de avaliação, questionários e documentos de autorização de recolha de dados. Estas sessões de avaliação resultam do trabalho que foi desenvolvido e é uma forma de ver o sistema em ação: as suas funcionalidades e a sua utilização prática no contexto de trabalho de análise de trajetos urbanos, e o modo de interação dos utilizadores e respetivos resultados.

## 5.3 Dificuldades encontradas

A maior dificuldade e limitação encontrada durante o projeto correspondeu à segunda funcionalidade do objetivo 1 (atualização de dados novos de trajetos que chegam periodicamente ao mapa), a qual não

foi concretizada. Este tipo de funcionalidade de suporte para atualização automática de novos dados de trajetos (simulação em tempo real) é relativamente recente e apresenta um número reduzido de ferramentas *open source*, com documentação limitada ou com lacunas para quem não domina esta área. Em particular, foi explorada a utilização da base de dados chave-valor *Kafka* (Apache Software Foundation, 2017), a partir da ferramenta *GeoMesa* (Commonwealth Computer Research, 2017), mas não foi possível obter resultados utilizáveis na aplicação de análise de trajetos.

Outra dificuldade encontrada ao longo do desenvolvimento do trabalho prendeu-se com a construção dos serviços *web*, mais precisamente com os mecanismos da formulação de pedidos e envio de respostas. Foi necessário investir algum tempo a aprender a sintaxe e forma de uso das mesmas com a ferramenta *Slim Framework*.

Outra complexidade foi a aprendizagem de manipulação de eventos e uso de *JavaScript* onde por vezes, foi necessário utilizar o *JQuery* para simplificar a manipulação de objetos e alterar as suas características. No início do projeto, foi difícil perceber a lógica dos eventos na implementação da função *update-Rose*, pois a atualização estava a ser feita numa rosa de cada vez, o que em termos de visualização não era adequado. Em vez disso, todas as rosas deviam ser atualizadas em simultâneo, situação que depois foi resolvida.

Tentou-se obter novos conjuntos de dados de trajetos para utilizar na aplicação, mas os *datasets* de trajetos ou percursos urbanos são limitados. Houve a possibilidade de uma empresa de transporte fornecer dados, mas acabou por não acontecer, daí terem sido usados os mesmos dados do protótipo anterior. Deste modo, foi necessário simular dados para gerar rosas com movimentos pendulares.

## 5.4 Trabalho futuro

Como trabalho futuro foram sugeridas melhorias ao sistema de análise de trajetos pelos participantes nos testes com utilizadores e foram também surgindo ideias durante o desenvolvimento do projeto.

Ao nível da interface gráfica, podem ser melhorados os casos de deteção de incoerências. Por exemplo, além de mostrar as discrepâncias nos vários parâmetros no painel de filtros, também deveriam ser realçadas no rodapé das rosas onde existem essas incoerências. Outro aspeto refere-se ao carregar no botão de *Apply* do painel de filtros. O utilizador não tem um *feedback* imediato da rosa ou conjunto de rosas e parâmetros do rodapé da rosa que foram modificados, o que pode confundir quando existirem várias rosas. Por isso, foi sugerido realçar o limite da rosa ou das diferentes rosas e respetivos campos do rodapé da rosa que sofreram modificações. Outra situação relatada foi quando existem muitas rosas e o espaço do ecrã fica reduzido. Pode ser incluída uma função de esconder (desvanecer) as rosas que não interessam, de forma a que apenas apareçam as que sejam relevantes na análise.

Relativamente a novas funcionalidades, a que foi mais destacada pelos participantes foi a atualização dos trajetos a partir da simulação de dados em tempo real nas rosas. No contexto de trabalho dos analistas faz sentido a análise de dados históricos, como no planeamento do tráfego urbano, porém para um utilizador comum, o sistema seria útil em cenários do momento presente. Por exemplo, estando o utilizador num determinado local, este quer saber qual melhor direção (menor fluxo de trânsito) para chegar ao seu destino (alertas). Com esta funcionalidade, as rosas seriam atualizadas, de uma forma periódica, com a chegada de novos dados de trajetos à base de dado.

Outra sugestão interessante foi a possibilidade de selecionar pétalas da rosa e esta atualizar a informação contida no rodapé sobre essa seção da rosa. Isto permitiria um estudo mais pormenorizado de um trajeto, ou conjunto de trajetos específicos, mas sempre com a opção de voltar à informação geral da rosa. No caso de existirem muitos trajetos aquando da seleção da pétala, destacar (*highlight*) os trajetos no mapa

associados a essa pétala. Também no caso de seleção dos pares de setas origem/destino deveria ser destacado o trajeto respectivo no mapa.

Por fim, outra funcionalidade sugerida foi a aplicação deste sistema em outros meios de transporte, como barcos e aviões. No caso da análise de trajetos de aeronaves, o sistema teria de ter uma dimensão *3D*, passando a incluir a altitude. Por exemplo, este sistema podia ser útil no controlo e gestão do tráfego aéreo das aeronaves, como saber as direções de origem e destino dos aviões e o fluxo de aeronaves, num determinado período, e respetivas velocidades.

## Bibliografia

- «JSDoc 3 - API documentation generator for JavaScript». (2018). Acedido a 27 de janeiro de 2019. Disponível em: <http://usejsdoc.org/about-getting-started.html>
- Ajax - Asynchronous JavaScript and XML. Acedido a 7 de março de 2018. Disponível em: <https://developer.mozilla.org/en-US/docs/AJAX>.
- Albert Gonzalez. (2018). Css date-ranges-selector. Acedido a 25 de setembro de 2018. Disponível em: <http://www.albertgonzalez.coffee>
- Amini, S., Gerostathopoulos, I., & Prehofer, C. (2017). *Big Data Analytics Architecture for Real-Time Traffic Control*. (Tum Llc), 710–715.
- Andrienko, G., & Andrienko, N. (2008). Spatio-temporal aggregation for visual analysis of movements. *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST'08)*, 51–58. <https://doi.org/10.1109/VAST.2008.4677356>
- Andrienko, G., & Andrienko, N. (2010). A General Framework for Using Aggregation in Visual Exploration of Movement Data. *The Cartographic Journal*, 47(1), 22–40. <https://doi.org/10.1179/000870409X12525737905042>
- Apache Software Foundation. (2017). Apache Kafka a distributed streaming platform. Acedido a 11 de dezembro de 2018. Disponível em: <http://kafka.apache.org/>
- Apache Spark - Unified Analytics Engine for Big Data. (2018). Acedido a 21 de novembro de 2018, Disponível em: <https://spark.apache.org/>
- Azure Stream Analytics Serviço de análise em tempo real. (2016). Acedido a 11 de novembro de 2018, Disponível em: <https://azure.microsoft.com/pt-pt/services/stream-analytics/>
- Baas, B. (2012). *NoSQL spatial – Neo4j versus PostGIS*. 121. Acedido a 14 de dezembro de 2018, Disponível em <http://igitur-archive.library.uu.nl/student-theses/2012-0822-200532/UUindex.html>
- Bak, P., Packer, E., Ship, H. J., & Dotan, D. (2012). Algorithmic and visual analysis of spatiotemporal stops in movement data. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, 462. <https://doi.org/10.1145/2424321.2424390>
- Bangor, A., Kortum, P., & Miller, J. (2005). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3), 114–123.
- Birdeye information visualization and visual analytics library. Acedido a 21 de fevereiro de 2018, Disponível em: <https://code.google.com/archive/p/birdeye/>
- Bracciale, L., Bonola, M., Loreti, P., Bianchi, G., Amici, R., & Rabuffi, A. (2014). CRAWDAD dataset Roma/taxi (v.2014-07-17). <https://doi.org/10.15783/C7QC7M>
- Brasseur, L. (2005). Florence Nightingale's visual rhetoric in the Rose Diagrams. *Technical Communication Quarterly*, 14(2), 161–182. <https://doi.org/10.1207/s15427625tcq1402>
- Commonwealth Computer Research. (2013). GeoMesa. Acedido a 21 de novembro de 2018, Disponível em: <https://www.geomesa.org/>
- Crutcher, H. L. (1957). On the standard vector deviation wind rose. *Journal of Meteorology*, Vol. 14, pp. 28–33.
- D. Gandy, «Font Awesome: the iconic font and CSS toolkit». Acedido a 12 de maio de 2019, Disponível em: <http://fontawesome.io/>
- Eppler, M. J., & Mengis, J. (2004). The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *Information Society*, 20(5), 325–344. <https://doi.org/10.1080/01972240490507974>

- Google, «Google Maps JavaScript API: Simple polylines». (2018). Acedido a 1 de julho de 2019 Disponível em: <https://developers.google.com/maps/documentation/javascript/>
- Google, Google Maps Acedido a 21 de julho de 2019 Disponível em: <https://www.google.com/maps>
- Hughes, J. N., Zimmerman, M. D., Eichelberger, C. N., & Fox, A. D. (2016). *A survey of techniques and open-source tools for processing streams of spatio-temporal events*. 1–4. <https://doi.org/10.1145/3003421.3003432>
- Ilya Bogdanovich. (2016). Geo Tracker - GPS tracker. Acedido a 15 de julho de 2018. Disponível em: [https://play.google.com/store/apps/details?id=com.ilyabogdanovich.geotracker&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.ilyabogdanovich.geotracker&hl=pt_PT).
- Izawa, M. M., Oliveira, F. A., Cajueiro, D. O., & Mello, B. A. (2017). Pendular behavior of public transport networks. *Physical Review E*, 96(1), 1–9. <https://doi.org/10.1103/PhysRevE.96.012309>
- JetBrains. (2018). PhpStorm. Acedido a 17 de maio de 2019. Disponível em: <https://www.jetbrains.com/phpstorm/>
- Jia, X., Chung, D., Huang, J., Petrilli, M., & The, L. (2004). ARO: Geographic information systems-based system for optimizing airport runway orientation. *Journal of Transportation Engineering*, 130(5), 555–559. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2004\)130:5\(555\)](https://doi.org/10.1061/(ASCE)0733-947X(2004)130:5(555))
- Lakes Environmental. (2016). WRPLOT View. Acedido a 1 de julho de 2019. Disponível em: <http://www.weblakes.com/products/wrplot/index.html>
- Magnello, M. E. (2012). Victorian statistical graphics and the iconography of Florence Nightingale's polar area graph. *BSHM Bulletin*, 27(1), 13–37. <https://doi.org/10.1080/17498430.2012.618102>
- McCormick, B., DeFanti, T., & Brown, M. (1987). Definition of visualization. *SIGGRAPH Comput. Graph*, 21(6).
- Medeiros, R. F. da C. (2009). *Escoamento do ar em torno da Ilha do Pico e a operacionalidade do seu aeroporto*.
- Meteoblue. (2016). Climate Pico Airport. Acedido a 26 de outubro de 2017, Disponível em: [https://www.meteoblue.com/en/weather/forecast/modelclimate/pico-airport\\_portugal\\_6618992](https://www.meteoblue.com/en/weather/forecast/modelclimate/pico-airport_portugal_6618992)
- Moovit. Acedido a 9 de março de 2019. Disponível em: <https://moovitapp.com/>
- PostGIS: Spatial and geographic objects for PostgreSQL. (2016).
- Real-time Analytics on AWS. (n.d.). Retrieved from <https://aws.amazon.com/big-data/real-time-analytics-featured-partners/>
- Roberto, E., Holanda, M. T. De, & Victorino, M. (n.d.). *Dados Geográficos em Banco de Dados Orientado a Grafo Geographic Data in a Graph Oriented Database*. (9), 2–7.
- RStudio – Open source and enterprise-ready professional software for R. (2017). Acedido a 2 de novembro de 2017. Disponível em: <https://www.rstudio.com>
- Seraj, F., Meratnia, N., & Havinga, P. J. M. (2017). An aggregation and visualization technique for crowd-sourced continuous monitoring of transport infrastructures. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, 219–224. <https://doi.org/10.1109/PERCOMW.2017.7917561>
- Slim a micro framework for PHP. (2018). Acedido a 2 de fevereiro de 2018. Disponível em: <http://www.slimframework.com/>
- Team, P. development. (2016). PHP: Hypertext Preprocessor. Acedido a 26 de novembro de 2017. Disponível em: <http://php.net/manual/en/index.php>
- The Apache Software Foundation. (2017). Apache HTTP Server Support - The Apache HTTP Server Project. Acedido a 26 de novembro de 2017. Disponível em: <https://httpd.apache.org/support.html>
- The JS Foundation, «jQuery». (2016). Acedido a 6 de maio de 2018. Disponível em: <https://jquery.com>

- The JS Foundation and the jQuery UI Team, «jQuery UI - jQuery user interface extension». (2012). Acedido a 6 de maio de 2018. Disponível em: [from http://jqueryui.com](http://jqueryui.com)
- Thomas, J. J., & Cook, K. a. (2006). Visualization Viewpoints: a visual analytics agenda. *IEEE Computer Graphics and Applications*, 26(February), 10–13. <https://doi.org/10.1109/MCG.2006.5>
- Vaz, R. (2017). *Visualização Analítica de Agregações de Trajetos com Rosas dos Ventos*.
- Visualization API Reference. Acedido a 9 de março de 2019. Disponível em: <https://developers.google.com/chart/interactive/docs/reference>
- W3C. (2017). W3C - HTML. Acedido a 9 de março de 2019. Disponível em: <https://www.w3.org/html/>
- W3C SVG Working Group, «SVG: Scalable vector graphics». Acedido a 12 de maio de 2019. Disponível em: <https://www.w3.org/Graphics/SVG>
- Wallis, H. (1987). *Cartographical innovations: An international handbook of mapping terms to 1900*. Published by Map Collector Publications in association with the International Cartographic Association.
- Waze - GPS, mapas e trânsito. Acedido a 9 de março de 2019. Disponível em: <https://www.waze.com/pt-PT>
- World Wide Web Consortium , Web services architecture. (2004). Acedido a 12 maio de 2019. Disponível em: <https://www.w3.org/TR/ws-arch/>
- XAMPP installers and downloads for Apache Friends. (2017). Acedido a 21 de novembro de 2018. Disponível em: <https://www.apachefriends.org/index.html>
- Zeng, W., Fu, C. W., Muller Arisona, S., Schubiger, S., Burkhard, R., & Ma, K. L. (2017). Visualizing the relationship between human mobility and points of interest. *IEEE Transactions on Intelligent Transportation Systems*, 18(8), 2271–2284. <https://doi.org/10.1109/TITS.2016.2639320>
- Zhang, J., Chen, Z., Liu, Y., du, M., Yang, W., & Guo, L. (2017). Space–time visualization analysis of bus passenger big data in Beijing. *Cluster Computing*, 1–13. <https://doi.org/10.1007/s10586-017-0890-8>
- Zhang, K., Sun, D. (Jian), Shen, S., & Zhu, Y. (2017). Analyzing spatiotemporal congestion pattern on urban roads based on taxi GPS data. *Journal of Transport and Land Use*, 10(1), 675–694. <https://doi.org/10.5198/jtlu.2017.954>



# Apêndice A

## Póster no 3<sup>rd</sup> LASIGE Workshop

### Visualization of Spatio-Temporal Information in a Track Analysis System with Wind Roses

Luís Ferreira, Ricardo Vaz,  
António Ferreira, Ana Paula Afonso

#### 1. Motivation

It is easy to show many tracks on a map (using Google Maps or similar systems), and this is widely used in organizations, for example, for urban planning. But there are limitations, such as **overlapping tracks** (which blocks the view of a part of the tracks), which compromise their correct analysis.



Visualization of **aggregated tracks** is necessary for the **study/analysis of patterns and trends**, and it is fundamental to have a **decision support** system that allows interaction with these visualizations and with the data, for example, by applying spatio-temporal filters.

#### 2. Objective

Development of a system that **allows the analysis of tracks** in maps with graphics inspired by wind roses.

Wind rose charts can be adapted in two scopes: **local** and **global**.

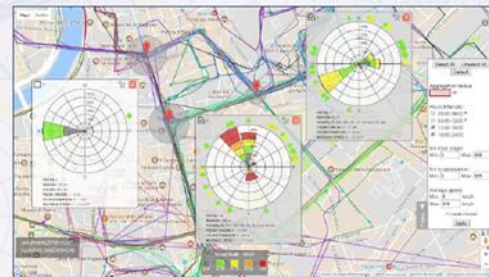


In the first case, it is the **analysis of the tracks within a region of aggregation**, and in the second it is intended to **study the origin and destination of the same routes**. To the best of our knowledge, this is an **innovative idea**.

#### 3. Work

**Key features** of the web application for analysing tracks using wind roses:

- **placing several roses** on map points
- use of an **inspector** to apply **spatio-temporal filters** to one or more roses at the same time



Possibilities for **decision support**:

- **choose the typical hours** of urban pendular movement
- **select tracks** from/to the periphery of a city, and consider tracks in congested traffic

On going work:

1. Evolution of the supporting web services
2. Direct manipulation on the map
3. New space and time filters



# Apêndice B

## Participação no Ciência Viva 2018



Luís Paulo Garcia  
Ferreira

LASIGE, Faculdade de Ciências  
Universidade de Lisboa

**ENCONTRO  
COM A CIÊNCIA  
E TECNOLOGIA  
EM PORTUGAL**  
**2-4** Centro de  
Congressos  
de Lisboa  
**JULHO**





# Apêndice C

## Proposta de artigo científico

### *Visualization of Spatio-Temporal Information for Track Analysis System with Wind Roses metaphor*

Luis Ferreira<sup>1</sup>, Ricardo Vaz<sup>1</sup>, António Ferreira<sup>2</sup>, Ana Paula Afonso<sup>2</sup>

LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal  
<sup>1</sup>{fc45509,fc41939}@alunos.fc.ul.pt, <sup>2</sup>{amferreira,apafonso}@fc.ul.pt

#### Abstract

In this paper we present an evolution of previous track analysis system using wind roses, through graphic visualization. Each track rose is formed by petals that add paths according to the direction around a marked point in an aggregation region on a map. The user can visualize the directions and speeds of the routes that pass around a place, as well as manipulate several roses with a time and space filters on the same map. The focus now was study manners of analyze pendular movements and patterns of tracks, that includes the possibility of studying behavior of tracks in different periods of the day with several roses at same place and allows exploratory analysis of the paths with automatic update of track roses when move to a new place of the map. An evaluation with users exposed a near absence of errors in the answer to questions of interpretation of the aggregations of the routes, and response times that did not differ significantly with the gender group, but which suggest improvements in the graphical interface. This version of the application offers new features, contributing to increase the knowledge in the area of analytical data visualization and is an important tool for decision support for urban traffic analysts. The application has the potential to be used in several domains, for example, in the analysis and optimization of traffic in a city.

**Keywords:** - visual analytics; wind rose; tracks aggregation; pendular movements

#### I. INTRODUCTION

With the proliferation of new technologies, access to large amounts of information has become very easy. For example, from ubiquitous global positioning devices (GPS), we can track paths traveled by entities / objects (people, vehicles, boats, airplanes, etc.) and it is easy to show them on a map (using Google Maps or

similar systems). This is widely used in organizations, for example, for urban planning, but there are limitations, such as overlapping tracks (which blocks the view of a part of the tracks), which compromise their correct analysis (Amini, Gerostathopoulos, & Prehofer, 2017).

Mobility studies for track analysis began to emerge, based on these records of untreated data movements. For example, to identify pendular movements, we have the study carried out in the transport system of the city of Brasília in Brazil (home-work-home travel) (Izawa, Oliveira, Cajueiro, & Mello, 2017). It is also possible to identify critical places, where there is traffic congestion, through animated maps, such as the study done in Beijing (Zhang et al., 2017). Animated maps already allow us to obtain some useful information, and we can apply temporal and spatial filters, but the decision maker has to worry about analyzing the map as a whole to find patterns of movement and do not have any tool / support system that presents in a compact and summarized way the main results.

In this article we effort to go beyond the information provided by animated maps. We describe below a system capable of applying time and space filters for analyzing specific traffic in period of the day, showing, which tracks that come and go in aggregated rose. Consequently, instead of the decision makers having to interpret multiple images in search of patterns for paths or problem with overlapping tracks, they will have explanations on the traffic.

The system integrates a web application, a track rose library and services supported by a geographic database to give the decision maker the possibility to

analyze paths on a map by placing and interacting with track roses.

The paper is structured as follows: in Section II we present the concepts and related work; in Section III we describe the components and functionalities of the application; in Section IV we show the results of a preliminary assessment with users, and we discuss in Section V the existing limitations; finally in Section VI we present the conclusions and ideas for future work.

## II. CONCEPTS AND RELATED WORK

### A. Wind Roses

Wind roses are radial circular figures that aggregate wind data, widely used in the area of meteorology to visualize the speed, frequency, and direction of prevail winds, from a certain region, for example, data wind of Pico Island Airport (Meteoblue, 2016). The size of petals depends on accumulated frequency of wind in the angle covered by each petal. For example, in a rose with four petals it would be added winds covering  $45^\circ$  on one side and another on the north, west, south and east cardinal points. Within each petal has a scale of speeds, where the stronger winds are on exterior limits of petals and weaker winds are in the closest to the center of the petals rose (Crutcher, 1957).

### B. Tracks

In the literature, the movement of a path consists of a temporal sequence of spatio-temporal positions, that is, a pair (instant, point) stored during the movement of an object. Each path is composed of one or more segments which, in turn, aggregate points that have associated latitude, longitude, elevation, instantaneous velocity, recording instant and occurrence sequence. Two types of approach were defined for trajectories: trajectory-oriented view for trajectory-oriented set of all entities and traffic-oriented view for temporal sequences of traffic situations (Andrienko & Andrienko, 2008).

### C. Applications with wind roses

The first use of the wind roses dates to the fourteenth century, by the time of the discoveries by

Portuguese, and arose from the need to indicate exactly the direction of the main winds, to illustrate cartographic maps (Wallis, 1987).

In a study, a rose was used to visual support for data interpretation, the Nightingale rose diagram, or also known as polar area diagram. Florence Nightingale created this diagram to graphically represent mortality during the Crimean War, that is divided into 12 sectors that represents each month. Each sector is subdivided into segments of the major causes of mortality. This representation helped Nightingale persuade the British government to institute reforms to improve the sanitary conditions of the army (Brasseur, 2005).

Other study, in the context of transport systems to solve urban traffic and congestion problems was the rose representation to give a general spatio-temporal overview of the distribution of bus stops in the city of Helsinki, Finland. That was an example of well-known visualization paradigm of detecting and analyzing stops as type of geographic movement patterns (Bak, Packer, Ship, & Dotan, 2012).

In terms of infrastructure systems, the ARO system seeks to facilitate the task of guiding the runways of an airport using wind roses to allow analysis of the prevailing winds if they are favorable to take-off and landing of airplanes (Jia, Chung, Huang, Petrilli, & The, 2004).

The WRPLOT application lets you draw roses with wind data (direction, speed, date, time and geographic information). The rose can be customized: changing the color scale of the speeds, number of petals, and the data can be filtered to, for example, display only the winds by month (Lakes Environmental, 2016). The principal limitation of this application is cannot be adapted to other types of data, such as tracks.

As we can see, wind roses have many useful applications as visual support for data interpretation or for decisions makers.

### D. Visualization tracks

It is easy to show many tracks on a map (using Google Maps or similar systems), and this is widely

used in organizations, for example, for urban planning. But there are limitations, such as overlapping tracks (which blocks the view of a part of the tracks), which compromise their correct analysis. So, visualization of aggregated tracks is necessary for the study/analysis of patterns and trends, and it is fundamental to have a decision support system that allows interaction with these visualizations and with the data, for example, by applying spatio-temporal filters. One example is using animated maps like studies of traffic in Brasilia (Izawa et al., 2017) and Beijing (Zhang et al., 2017) or can be done with a graphical visual representation like the study of bus stops in Helsinki.

Currently, analysts do not use data on the movements of people and vehicles, since there are no adequate tools to allow this analysis and only use descriptive statistics. There is no interaction with the user. For example, to apply time or space filters, or to consider specific tracks elements, such as their origin and destination, the distance traveled, total time elapsed, and others. It is precisely some of these elements, we created a system that allows the analysis of tracks in maps with graphics inspired by wind roses.

Wind rose charts can be adapted in two scopes: local and global. In the first case, it is the analysis of the tracks within a region of aggregation, and in the second it is intended to study the origin and destination of the same routes. To the best of our knowledge, this is an innovative idea and we described in the next section.

### III. SYSTEM DESCRIPTION

We organized the description of the route analysis system with wind roses in three parts: first we present the main components of the architecture; then we show a wind roses for routes generated on a point on the map, containing filters accessible to the user; and finally, we've covered some performance optimizations.

#### A. Architecture of the System

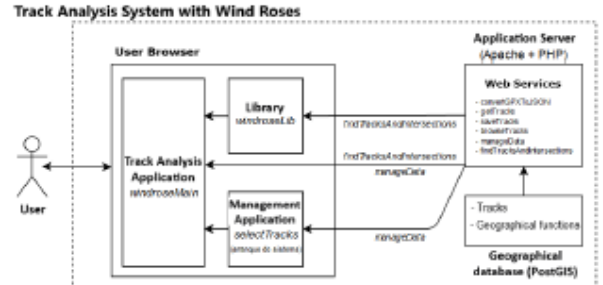


Figure 1. Architecture of Track Analysis System with Wind Roses

The system has four main components: web services programmed in PHP (1); a library with data structure and wind roses functions for paths, in JavaScript (2); a geographic database (3) in PostGIS ; a web application written in JavaScript that allows the placement of roses from the winds on a map with paths (4);

#### 1. Web Services

The web services support the web application, offering features such as:

- i. reading paths in GPX format;
- ii. loading tracks to the geographic database;
- iii. transforms track data in GPX format into JavaScript Object Notation (JSON);
- iv. sent tracks in JSON format to the geographic database;
- v. creating a session shared by a preliminary route selection page contained in a GPX file and the route analysis web application.
- vi. Get tracks data from database, as intersections.

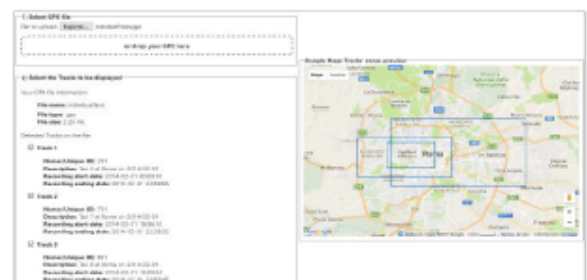


Figure 2. Tracks list and bounding boxes, respectively.

## 2. Wind Rose Library

The wind rose library has methods to build wind roses with aggregated data from map points and draw the respective graphic of a rose, such as the petals and speed scale. The JavaScript object representing a rose has access to the paths that were uploaded to the web application, in order to facilitate the generation of the graphic elements that constitute it.

## 3. Geographic database

PostGIS is a geographic database management system that has a set of tools that facilitate the manipulation of this type of data, such as predefined geographical functions that improve the discovery of the paths that cross the area of rose aggregation.

## 4. Track Analysis Web Application

The track analysis web application is the main component of the system from the user's point of view of the user, behaving like a client.

The creation of a rose is done by clicking the right mouse button on a point on the map, drawing a marker and a circle with the area of aggregation. Initially, the radius of aggregation is 100 meters which can be changed later. When a marker is placed on the map, the web application begins discovering the tracks that intersect the circle of the rose aggregation area, taking advantage of PostGIS functions. Once the intersections with the circle are obtained, the web application uses the wind rose library to draw the graphic elements of a rose, such as the petals and speed scale, which is placed next to its marker.

The web application allows the user to change the various aspects of each rose individually, such as the radius of route aggregation or the application of time filters, for example from 100 to 200 meters radius and considering only routes performed in the afternoon. However, it is also planned to control all roses simultaneously, especially on a map that contains several strategic locations to analyze, which facilitates comparison of the data shown on roses.

## B. Example of the system wind roses for tracks



Figure 3. It is a scenario where markers and respective wind roses are created, associated with the paths.

The Figure 3 shows a scenario of when each marker was created on the map, loaded with taxi routes in Rome (Vaz, 2017). Each path is marked by a different color, assigned randomly in the loading phase of the data. The colors of the paths assume no meaning apart from the differentiation function.

Each marker is created by clicking the right mouse button on any area of the map and has a unique identifier number. Around the element, the gray path aggregation area is delimited. This area delimits the space where the paths are analyzed, and those that do not intersect the gray area are ignored. Beside each marker there is a box containing the corresponding wind rose. The number displayed in the upper left corner of the wind rose box represents the corresponding marker number.

In the box header there are three distinct buttons, from left to right: select, pin button and close. The select button is used to indicate whether or not the rose can be modified, since only selected roses can be changed. The pin button allows you to fix the position of the rose on the screen, which prevents the rose box from being dragged with the mouse. The close button serves to erase the rose box, along with the corresponding marker and banding area. The wind rose has a grid with a relative occurrence scale, which adjusts according to the maximum path a petal can contain. Each of the scale values is associated with one of the grid circles of the wind rose, with the scale growing from the circle near the center to the outer circle.

The petal that is drawn in the rose has three distinct characteristics:

1. Direction (paths intersected, for the first time, in radius of aggregation of the marker of the rose);
2. Length of the petal (total of paths that the petal aggregates relative to the total of paths);
3. Color of petal (different speeds of the paths intersect in the radius of aggregation).

The size of each section is related to the number of paths each of them adds. Although it is not visible in the image, if the user leaves the mouse over a petal section, a tooltip with the total of paths corresponding to the section pointed by the mouse is shown.

At the outer border of the area of rose petals are placed arrows of origin and destination of each route. The orientation of the source arrow is defined by the direction that points from the beginning of a path to the rose on the map. The orientation of the destination arrow is given by the direction of the marker to the last point of the route.

The colors and speeds are represented on footer of the web application, with the title Speed Scale. The color of the pair of arrows represents the average velocity of the corresponding path, while the color in the sections of the petals corresponds to the velocity at the intersection with the area of aggregation of the rose.

Finally, the footer of the winds rose contains information about the parameters the paths that it adds. The first three fields (PATHS, RADIUS and HOURS) show the number of paths that intersect the rose aggregation area and the allowed time intervals for the route visualization, respectively. The last three fields (FROM ORIGIN, TO DESTINATION, AVERAGE SPEED) represent the minimum and maximum relative to the distance in km to the origin / destination (distance between the marker and the first / last point of the route) and the average speed in Km / h, respectively.

### C. Filters Panel and Inspector

The application dashboard is displayed in the right margin of the web application screen, as shown in Figure 3, and allows you to set the parameters of the roses in the map. When the user creates a new marker on the map, a rose is created next to the marker with the values that are defined in each of the fields in the filter panel. Creating new roses with invalid parameters causes an error message.

The application of the filters is subjected to the use of the button Apply, which motivate the notification of unsaved changes with \* symbol is displayed in front of the field. When Apply button is used, the notification disappears. For the convenience of the user, two buttons (Select All and Unselect All) have been created allowing to quickly mark/unmark all the roses presented on the map. The rose parameters can only be changed if they are selected.

Whenever a rose is selected, the filter panel adjusts the various fields to the values that are set on the selected roses. If any inconsistency is found, a red border is placed and the value of the field in question is removed from the filter panel. The incoherence disappears with the unmarking of one or more roses that are generating the conflict or simply with the application of a common value to the field that has the inconsistency. Furthermore, when there are inconsistencies, an error message is displayed, and creation of roses is disabled.

Considering Figure 3, it can be observed that the radius of aggregation of roses 2 and 3 is different, which motivated the appearance of the notification of incoherence of values.

### D. Performance optimization

The track analysis system with wind roses has a functional prototype, that is version two, used in the example of Figure 3. A third prototype is under development that contemplates some improvements and optimizations that we describe next.

New features of the third system for analyzing tracks using wind roses: placing several roses with different periods of day at same place, for identifying

pendular movements and automatic update of roses when the marker is moved. In addition, a loading track analysis application page has been changed so that we do not have to always insert the file (GPX) from the selection page and immediately open the page if there are tracks on database.

In terms of optimization the web services application was reformulated using Slim Framework tool that allowed to redefine and add a set of web services in the accesses to the geographic database and improved performance time. Another change from the second prototype was new time filters including a dynamic component in which it is possible to modify and add multiple slots, which are sorted automatically. The filter panel and inspector interface has been changed to enhance and highlight buttons and signalling in-coherence of selected rose values in the filter panel.

#### IV. USER EVALUATION

We have done a user evaluation to assess the usability of the web application: to understand how users are able to perceive scenarios of information overload in a map, using wind roses and quantify the level of usability of the web application of the system through the method SUS.

##### A. Participants

For this, the prototype was evaluated by 21 people, 11 men and 10 women, aged between 17 and 27 years. Most of the participants had no mapping experience. Only 3 people had previously used applications to perform route analysis on mobile devices.

##### B. Apparatus

Each evaluation session was done with a user, taking place in a closed room, without external interferences and with low levels of noise. A laptop computer was used along with a mouse to facilitate interaction with the system interface. The purpose of the test and the concepts under study were explained. Time was given for the participants to explore the application, before answering the questions.

##### i. Application Test

We developed a test application of the system analysis application, in order to collect human performance data in various wind rose analysis scenarios. There are 10 questions in the task bank of the application. Each user responds to the same questions, which are randomly reordered, at the beginning of each test. This means there is an extremely high probability of each user responding to a test with a sequence of questions different from the rest. This application captures user activity data, in a way that interferes little on the windroseMain application interface.

##### C. Tasks

Each scenario is accompanied by a task, whose response must be given through clicks on graphical elements that are shown in the application. During the test, values are stored in a set of variables: user id, question id, question order, errors, runtime (in seconds), whether the question was answered or skipped, date and time of the response.

In order to explore and test the different elements of the system interface, 4 types of question were created, namely:

1. SEMA - Select the marker that corresponds to the wind rose shown on the screen;
2. SEPE - Select the petal with the highest number of paths and speed or the petal with the highest speed of the rose that is being shown on their screen;
3. SERO - Select from among several the wind rose corresponding to the marker shown on the screen;
4. INSE - Inspect a certain area with tracks on the map, delimited by dashed lines, and mark the place that respect the characteristics presented in the question.

For each of the question types, there were 2 associated scenarios, except for the INSE questions, where there are 4. The creation of different scenarios

was designed in order to test the learning effect among the various issues.

#### D. Procedure

The procedure adopted for each test performed by each participant was composed of a set of objectives, listed in order:

1. initial questionnaire;
2. training with the application of system analysis;
3. tests with the application of test;
4. a final usability questionnaire.

#### E. Results

We applied the Shapiro-Wilk statistical test only to user response times to the questions, since the number of errors was greatly reduced. As most of the data did not follow a normal distribution ( $p \leq 2,495e-06$ ) the Wilcoxon non-parametric test for two samples was used in the following comparisons for response times.

##### i. Response Times to tasks

The median response times varied from 25.20 seconds in SEPE to 109.50 seconds in INSE, SEMA had a median of 68.45 seconds and the SERO 85.00 seconds. There were significant differences in response times between the four types of questions, being higher between SEMA and SEPE type ( $V = 887$ ,  $p = 5.354e-08$ ) and type SEMA and INSE ( $V = 97$ ,  $p = 1.644e-06$ ).

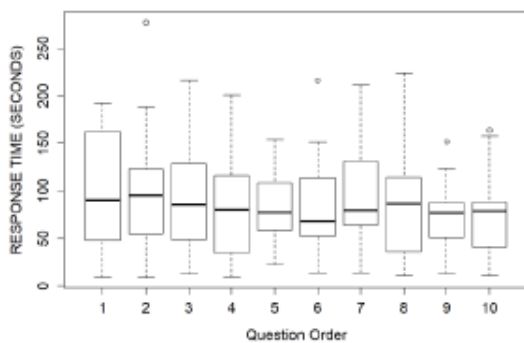


Figure 4 – User response times, in order of tasks.

Regarding possible learning effects between questions of the same type, no significant differences were found. The medians of the response time according to the order of the test questions are quite balanced, as shown in figure 4.

In the results of figure 5 it is verified that the greater response time, by the users, is associated with INSE-type questions. On the other hand, SEPE-type questions have the lowest response times. The SEMA and SERO types have intermediate response times.

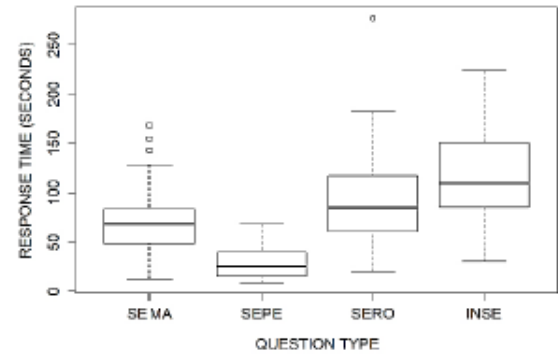


Figure 5 – User response times by question type

We also analyzed possible gender effects on response time with the results shown in Figure 6. No significant difference was found between men and women ( $p \geq 0.288$ ).

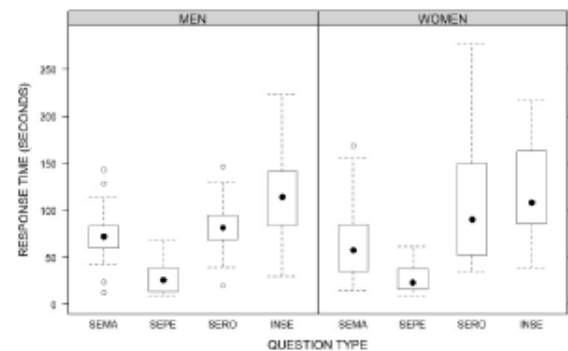


Figure 6 – Response times by gender of users, by type of question

### ii. Errors in the tasks (questions)

Errors represent the number of times users clicked on system interface elements that did not match the correct answer to the question they were responding to. The results are in Figure 7. It was verified that the presence of errors during the tests was practically non-existent, regardless of the type of question to which the users were submitted.

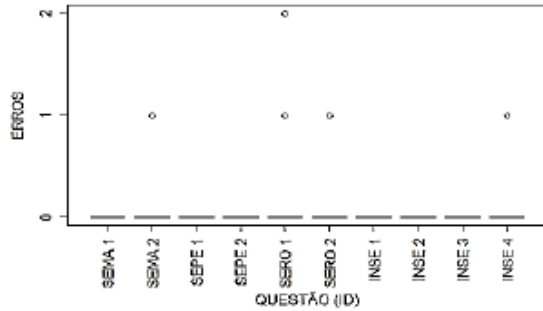


Figure 7 – Number of errors, by question

### iii. Usability of the system, based on the tasks (questions)

An evaluation of the usability of the system with a score attributed by the users was made, based on the SUS questionnaire (Bangor, Kortum, & Miller, 2005). These results are presented in Figure 8 where it is possible to verify the total usability classifications assigned by the users.

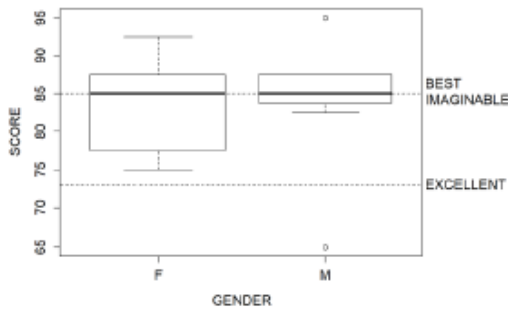


Figure 8 - User usability classifications assigned by users

The total scores were calculated according to SUS formulas, in which each question has a certain weight for the calculation of the final usability score. Most

users felt that the system was easy to use, so their scores corresponded to the maximum scale rating.

We also verified the distribution of the usability score attributed by the users' gender. It was found that the medians of both genders are practically the same, and there is a slightly higher variation in female users, which can be seen in figure 9.

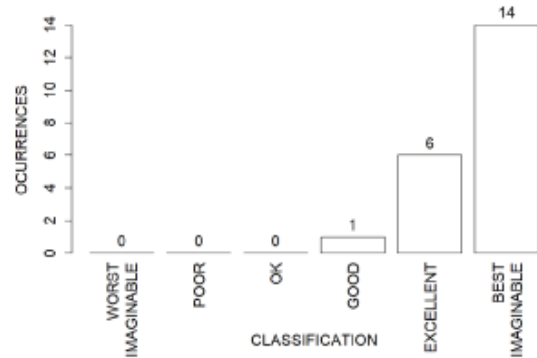


Figure 9 – System usability ratings, attributed by user gender.

The classifications are scored by the users, to the 10 questions of the SUS questionnaire. Each question was rated on a scale of 1 to 5, regarding the degree of agreement of the user with the statement. The value 1 represents total disagreement, with 5 representing total agreement.

## V. DISCUSS AND LIMITATIONS

### i. Response Time

The most easy-to-answer questions were those of the SEPE-type, where the answers were almost immediate after reading the statement of the question, followed by SEMA-type. On the other hand, there have been high response times for SERO-type tasks. In the scenarios of this type of question, wind roses were purposely created to raised doubts as to the orientation of the petals and the number of routes. Finally, as expected, INSE-type questions contain the highest response times, since they require a more elaborate workflow compared to the other types of question.

One of the problems noted throughout the answers to SEMA questions was the use of random colors to represent different paths on the map. Often there were very similar shades of color, which led the user to think that they were the same path to intersect the radius of rose aggregation. In most cases, this has meant that more time is spent on color differentiation than on the analysis of the elements of the roses in the winds.

The tests for the existence of possible learning effects revealed that this effect was non-existent. The median is very similar across the 10 questions of each 21-user test. Once the order of questions was changed for each test, it can be assumed that in the various scenarios of the 4 question types, users responded by reasoning about the information displayed on the screen. If the responses had taken place in a mechanized way, a median drop in the number of different orders should be checked.

#### **ii. Errors**

The absence of errors of this second prototype was noted along the course of the tests.

With the evaluation of this second prototype, questions were developed with a higher degree of complexity. The degree of complexity of the INSE-type questions led to elaborate workflow that could be susceptible to error (something that was expected in the results). In a rather surprising and quite positive way, users continued to make no mistake. when put to the test with analysis tasks with the system.

This result leads us to believe that the web application of the system is easy to perceive and interact, successfully fulfilling the purpose of the system: to analyze aggregate paths, on maps, resorting to roses from the winds.

#### **iii. SUS**

The usability questionnaire showed that users felt that the system is easy to use, where only one user assigned a final classification to the good system, while the rest assigned the following level (Excellent) or maximum (Best Imaginable).

#### **iv. Influence of previous experience with track analysis**

The number of participants with previous experience of analysis are only 3 people. This knowledge has not proved to be advantageous in terms of test performance.

#### **v. Workflow**

Most users preferred to make a rose creation with the predefined parameters by applying filters later. A minority of users adjusted the filters, according to what was asked in the statement of the question. This occurrence is believed to be due to the fact that users had little system uptime prior to the test. Two other features that fell into disuse were the select / unselect buttons for all the roses on the screen.

Most participants chose to mark the roses of the winds one by one. After the test, the participants were asked about the workflow for adjusting the properties of the roses of the winds. With the answers obtained, the main cause was reached: the buttons were not sufficiently elucidative in the function they performed, which caused some uncertainty as to its use.

Finally, some users have tended to make filter adjustments, rather than looking directly at the information in the footer of each rose. This scenario was recorded, for example, in the INSE question that asked to be marked the area where vehicles were traveling with the most distant destination. In the end, the participants ended up using the information shown in the footer of each rose and answered the question successfully. These occurrences may be related to the fact that the information shown in the footer does not have a title that shows that they are information of the routes that are aggregated by the rose of the winds in question.

#### **vi. Expert Opinion**

In order to better understand the applicability of the developed system, we interviewed an expert from the Transport Area of the Rodoviária de Lisboa to evaluate the developed system.

The expert considered the system developed to planning and management of vehicle routes important. For analysts, the system should be able to produce information reports of the various roses created on the screen, with a summary of the data displayed by them.

Thus, would leave interpretation of the system is not dependent on familiarization with the various graphical components of the interface.

Regarding the area of rose aggregation, it has been mentioned that it is very important to provide an analysis of what goes on within the radius of each marker, rather than outside, as it is in the current prototype. An example of this would be to realize where each path enters the radius of action and where it leaves, rather than a general aggregation of these data. This individual analysis could be accompanied by properties within the radius of aggregation, such as the average speed and distance traveled.

It was also noted that it would be interesting to note the average speed from the origin of the routes to the center of the marker, and from there to the destination. The need for all these details was justified by the possible analysis of alternative routes for buses, based on the flow of traffic within the areas of aggregation of roses created at strategic points.

The flexibility of setting parameters was one of the most appreciated aspects, since the system that the company uses to monitor their vehicles, does not allow the reconfiguration of certain filters. However, it was pointed out that it would be useful to reconfigure the time intervals in which routes are to be observed taking into account, for example, peak times.

Finally, the possibility of using the system with real-time data was pointed out as an innovative idea that would improve the way the company's vehicles are monitored.

The inclusion of movement delays alerts on the various roses created on the map would allow you to see which vehicles would be affected within the range of a given marker.

#### vii. System Limitations

This 2nd prototype does not have the possible to synchronously control various wind roses to make it easier to compare the aggregate paths at different locations on a map.

There is a problem in overlapping source and destination arrows when there are a large number of paths that intersect the radius of rose aggregation.

Additionally, it does not allow automatic updating of the various roses when the user changes the marker's position.

## VI. CONCLUSION AND FUTURE WORK

In this article we presented the evolution of a system that allows the analysis of routes in maps using graphics inspired by roses from the winds.

In this second prototype, the evolution of the wind rose system of the 1st prototype was carried out where new functionalities were developed to allow a greater ability to analyze a large set of track data. A new panel of filters and inspector has been created and a set of web services to support the operation of the system.

Finally, the system was submitted to user evaluation sessions. The results showed that the implemented system successfully fulfills its main function: to allow the analysis of aggregate paths using wind roses.

The aspects related to the future work of this project were compiled from the feedback obtained through user and expert evaluation. The implementation of the possibility of applying other types of filters to the routes, with greater freedom of adjustment of parameters, such as the choice of custom time intervals in which routes may occur. Some aspects like density and overlap of origin and destination arrows. The possibility of using new geometric forms for the area of path aggregation of a rose is something that can be explored. Other important feature was roses can be updated periodically, with the arrival of new route data to the database. This update would also be valid if the user wanted to move the rose marker to another on the map, which is currently not possible.

## VII. Biography references

Amini, S., Gerostathopoulos, I., & Prehofer, C. (2017). *Big Data Analytics Architecture for Real-Time Traffic Control*. (Tum Llcm), 710-715.

Andrienko, G., & Andrienko, N. (2008). Spatio-temporal

- aggregation for visual analysis of movements. *VAST'08 - IEEE Symposium on Visual Analytics Science and Technology, Proceedings*, 51–58. <https://doi.org/10.1109/VAST.2008.4677356>
- Bak, P., Packer, E., Ship, H. J., & Dotan, D. (2012). Algorithmic and visual analysis of spatiotemporal stops in movement data. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, 462. <https://doi.org/10.1145/2424321.2424390>
- Bangor, A., Kortum, P., & Miller, J. (2005). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3), 114–123.
- Brasseur, L. (2005). Florence Nightingale's visual rhetoric in the Rose Diagrams. *Technical Communication Quarterly*, 14(2), 161–182. <https://doi.org/10.1207/s15427625tcq1402>
- Crutcher, H. L. (1957). On the standard vector deviation wind rose. *Journal of Meteorology*, Vol. 14, pp. 28–33.
- Izawa, M. M., Oliveira, F. A., Cajueiro, D. O., & Mello, B. A. (2017). Pendular behavior of public transport networks. *Physical Review E*, 96(1), 1–9. <https://doi.org/10.1103/PhysRevE.96.012309>
- Jia, X., Chung, D., Huang, J., Petrilli, M., & The, L. (2004). ARO: Geographic information systems-based system for optimizing airport runway orientation. *Journal of Transportation Engineering*, 130(5), 555–559. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2004\)130:5\(555\)](https://doi.org/10.1061/(ASCE)0733-947X(2004)130:5(555))
- Lakes Environmental. (2016). WRPLOT View.
- Meteoblue. (2016). Climate Pico Airport. Retrieved October 26, 2017, from [https://www.meteoblue.com/en/weather/forecast/modelclimate/pico-airport\\_portugal\\_6618992](https://www.meteoblue.com/en/weather/forecast/modelclimate/pico-airport_portugal_6618992)
- Vaz, R. (2017). *Visualização Analítica de Agregações de Trajetos com Rosas dos Ventos*.
- Wallis, H. (1987). *Cartographical innovations: An international handbook of mapping terms to 1900*. Published by Map Collector Publications in association with the International Cartographic Association.
- Zhang, J., Chen, Z., Liu, Y., du, M., Yang, W., & Guo, L. (2017). Space-time visualization analysis of bus passenger big data in Beijing. *Cluster Computing*, 1–13. <https://doi.org/10.1007/s10586-017-0890-8>



# Apêndice D

## Documentação da biblioteca *windroseLib*

### Class: WindRose

#### Constructor

```
new WindRose(id, centerLat, centerLng, radius, hoursRanges, originMinDistance, originMaxDistance, destinationMinDistance, destinationMaxDistance, averageSpeedMinimum, averageSpeedMaximum, tracks, segments, points, googleMapTracksBounds, roseCircle, height, width, numberGridCircles, numberGridSectors, petalsAmp, petalsScaleUnit, petalsColorScale, mouseX, mouseY, pin, select)
```

Wind Rose Library constructor that creates a new WindRose instance

#### Parameters:

Name	Type	Description
id	number	Unique identifier
centerLat	number	Center latitude
centerLng	number	Center longitude
radius	number	Aggregation radius
hoursRanges	Array.<string>	Valid hours ranges
originMinDistance	number	Valid minimum distance to origin, in Km
originMaxDistance	number	Valid maximum distance to origin, in Km
destinationMinDistance	number	Valid minimum distance to destination, in Km
destinationMaxDistance	number	Valid maximum distance to destination, in Km
averageSpeedMinimum	number	Valid minimum average speed, in Km/h
averageSpeedMaximum	number	Valid maximum average speed, in Km/h
tracks	object	Structure containing tracks data
segments	object	Structure containing segments data
points	object	Structure containing points data
googleMapTracksBounds	Array.<object>	Array containing google.maps.LatLngBounds objects with tracks bounds

roseCircle	object	google.maps.Circle object representing the Wind Rose aggregation radius
height	number	Canvas height
width	number	Canvas width
numberGridCircles	number	Total grid circles
numberGridSectors	number	Total grid sectors
petalsAmp	number	Petals maximum amplitude
petalsScaleUnit	number	Petals scale unit
petalsColorScale	object	Structure containing the petals color scale and respective values
mouseX	number	Wind Rose box top left X coordinate
mouseY	number	Wind Rose box top left Y coordinate
pin	boolean	True, if pinned wind rose upon creation; False, otherwise
select	boolean	True, if selected wind rose upon creation; False, otherwise

## Members

**(inner) setHidden**

Hides the Wind Rose

## Methods

**delete ()**

Deletes the Wind Rose (includes all elements, listeners and handlers)

**getAggregationRadius () → {number}**

Returns the Wind Rose aggregation radius, in meters

**Returns:**

Aggregation radius, in meters

**getAverageSpeedMax () → {number}**

Returns the valid Maximum Average Speed filter from the Wind Rose

**Returns:**

Maximum Average Speed, in meters per second

**getAverageSpeedMin () → {number}**

Returns the valid Minimum Average Speed filter from the Wind Rose

**Returns:**

Minimum Average Speed, in meters per second

**getCanvas () → {Object}**

Returns the Wind Rose canvas Object

**Returns:**

The Wind Rose canvas object

**getColorScale () → {Array}**

Returns the Wind Rose color scale

**Returns:**

Data structure containing the speed color scale and respective speed intervals

**getDestinationMaxDistance () → {number}**

Returns the valid Maximum Destination Distance filter from the Wind Rose

**Returns:**

Maximum Destination Distance, in meters

**getDestinationMinDistance () → {number}**

Returns the valid Minimum Destination Distance filter from the Wind Rose

**Returns:**

Minimum Destination Distance, in meters

**getHours () → {Array}**

Returns the valid hours filter from the Wind Rose

**Returns:**

Valid hours on String Array

**getID () → {number}**

Returns the Wind Rose ID

**Returns:**

Wind Rose ID

**getIntersectionData () → {Array}**

Returns the intersection Data structure with intersection points over the aggregation radius

**Returns:**

The incoming petals data structure

**getOriginMaxDistance () → {number}**

Returns the valid Maximum Origin Distance filter from the Wind Rose

**Returns:**

Maximum Origin Distance, in meters

**getOriginMinDistance () → {number}**

Returns the valid Minimum Origin Distance filter from the Wind Rose

**Returns:**

Minimum Origin Distance, in meters

**getPetalsInData () → {Array}**

Returns the Incoming Petals Data structure

**Returns:**

The incoming petals data structure

**isSelected () → {boolean}**

Returns true or false, whether the Wind Rose is selected, or not

**Returns:**

True - If the Wind Rose is selected; False - Otherwise

**isVisible () → {boolean}**

Returns true or false, whether the Wind Rose is visible, or not

**Returns:**

True - If the Wind Rose is visible (visible canvas); False - Otherwise

**setTitle (newText)**

Changes the Wind Rose Title Label

**Parameters:**

Name	Type	Description
newText	string	String with the new title

**setVisible()**

Sets the Wind Rose visible

**(inner) arc(canvasX, canvasY, radius, initialAngle, finalAngle) → {Object}**

Returns the initial point and final points of an arc. An arc can be seen as an "unfinished circumference". Example: a circular shaped line from 30 degrees to 60 degrees is an arc

**Parameters:**

Name	Type	Description
canvasX		Canvas center x value
canvasY		Canvas center y value
radius		Reference circle radius
initialAngle		The initial angle, in degrees
finalAngle		The final angle, in degrees

**Returns:**

{ initCoordinates: {x, y}, finalCoordinates: {x, y} }

**(inner) averageSegmentSpeed(segmentID, pointsData) → {number}**

Calculates the average speed (in m/s) from a given segment

**Parameters:**

Name	Type	Description
segmentID		The segment ID
pointsData		Structure containing all points data

**Returns:**

Average segment speed in m/s

**(inner) calibrateGridPercentage(petalsData, gridSectors) → {number}**

Finds the petal with highest percentage on the rose and then, returns the calibrated grid percentage

**Parameters:**

Name	Type	Description
petalsData		Array containing wind rose intersection points data
gridSectors		Number of grid sectors

**Returns:**

Calibrated grid percentage

**(inner) circle(centerX, centerY, radius, strokeColor, strokeWidth, strokeFill, circleID, roseID) → {Object}**

Creates a circle element

**Parameters:**

Name	Type	Description
centerX		Center latitude
centerY		Center longitude
radius		Circle radius
strokeColor		Circle stroke color
strokeWidth		Circle width color
strokeFill		Circle filling color
circleID		The circle ID
roseID		The Wind Rose ID

**Returns:**

Object representing a <circle> element in the SVG reference

**(inner) createCanvas(roseID, canvasHeight, canvasWidth, backgroundColor, opacity, borderRadius, positionX, positionY) → {Object}**

Creates a new drawing area for the wind rose

**Parameters:**

Name	Type	Description
roseID		The Wind Rose ID
canvasHeight		Canvas Height
canvasWidth		Canvas Width
backgroundColor		Canvas background color
opacity		Canvas opacity
borderRadius		Canvas border Radius
positionX		Canvas Top-Left X (mouse position)
positionY		Canvas Rose Top-Left Y (mouse position)

**Returns:**

```

{ canvasX - Canvas center x position,
  canvasY - Canvas center y position,
  canvasObject - Variable to store the canvas output object,
  outerCircleRadius - Variable to store the circle radius output value
}

```

```

(inner) createCardinalLabels(cardinalPoints, textColor, canvasX, canvasY,
  canvas, outerCircleRadius, labelClassID, labelID, roseID)

```

Creates the Wind Rose Cardinal Points text labels

**Parameters:**

Name	Type	Description
cardinalPoints		Array containing cardinal points labels (ordered: 0° -> 90° -> 180° -> 270°)
textColor		Text color
canvasX		Canvas center X position
canvasY		Canvas center Y position
canvas		Canvas object
outerCircle-Radius		Grid's outer circle radius
labelClassID		The Cardinal Point Label class
labelID		Label ID
roseID		The Wind Rose ID

```

(inner) createCloseButton(coordinateX, coordinateY, canvas, buttonSize,
  xOffset, strokeColor, strokeWidth, strokeFill, classID, closeButtonID,
  roseID)

```

Creates the Wind Rose closing button

**Parameters:**

Name	Type	Description
coordinateX		Button top-left x position
coordinateY		Button top-left y position
canvas		The canvas object
buttonSize		Button size
xOffset		X element offset
strokeColor		Button stroke color

strokeWidth		Button width color
strokeFill		Button filling color
classID		Button class ID
closeButtonID		The Button ID
roseID		The Wind Rose ID

**(inner) createDirectionalArrows(petalsInData, outerCircleRadius, classID, centerLatitude, centerLongitude, pointsData, canvas, centerX, centerY, colorScale)**

Draws the "IN & OUT" arrows around the Wind Rose

**Parameters:**

Name	Type	Description
petalsInData		Structure containing the circle intersection points data (WARNING: same structure as Petals Data)
outerCircleRadius		Wind Rose outer circle radius
classID		Directional arrow class ID
centerLatitude		Wind Rose center latitude
centerLongitude		Wind Rose center longitude
pointsData		Segments Points Data structure
canvas		The Canvas object
centerX		Canvas center x coordinate
centerY		Canvas center y coordinate
colorScale		The Wind Rose color scale

**(inner) createFooter(canvasWidth, canvasHeight, canvas, petalsData, hours, footerClassID, footerID, roseID)**

Creates the Wind Rose footer

**Parameters:**

Name	Type	Description
canvasWidth		Canvas width
canvasHeight		Canvas height
canvas		Canvas object

petalsData		Wind Rose intersection points data structure
hours		Data structure containing the valid hours filter values
footerClassID		The footer class ID
footerID		The footer ID
roseID		The Wind Rose ID

**(inner) createGrid(strokeWidth, strokeColor, strokeFill, canvasX, canvasY, canvasObject, gridCircles, gridSectors, outerCircleRadius, roseID) → {Object}**

Creates the Wind Rose Grid (circles and lines)

**Parameters:**

Name	Type	Description
strokeWidth		Grid stroke width
strokeColor		Grid stroke color
strokeFill		Grid filling color
canvasX		Canvas center x position
canvasY		Canvas center y position
canvasObject		Canvas object
gridCircles		Number of grid circles
gridSectors		Number of grid sectors
outerCircleRadius		Outer circle radius
roseID		The Wind Rose ID

**Returns:**

{ gridGroup: Element, radiusDifference: number }

**(inner) createGridPercentageLabel(outerCircleRadius, radiusDifference, textColor, gridPercentage, gridCircles, canvas, canvasHeight, percentageLabelClass, percentageLabelID, roseID)**

Creates the Grid Percentage Labels

**Parameters:**

Name	Type	Description
outerCircleRadius		Grid's outer circle radius
radiusDifference		Grid's radius difference
textColor		Label text color

gridPercentage		Grid's percentage
gridCircles		Number of grid circles
canvas		The canvas object
canvasHeight		Canvas Height
percentageLabelClass		The percentage label class
percentageLabelID		The percentage label ID
roseID		The Wind Rose ID

**Requires:**

createPetal

**(inner) createPetal(direction, petalsData, roseScale, petalsAmplitude, strokeWidth, strokeColor, gridSectors, outerCircleRadius, canvas, canvasX, canvasY, petalID, roseID)**

Creates a Wind Rose Petal element

**Parameters:**

Name	Type	Description
direction		The petal direction
petalsData		Array containing all intersection points data
roseScale		Wind Rose scale
petalsAmplitude		Petals' amplitude
strokeWidth		Petal's stroke width
strokeColor		Petal's stroke color
gridSectors		Number of grid sectors
outerCircleRadius		Outer circle radius
canvas		Canvas object
canvasX		Canvas center X position
canvasY		Canvas center Y position
petalID		The Petal ID
roseID		The Wind Rose ID

**(inner) createPetalsInData(intersectionPointsArray, centerLatitude, centerLongitude) → {Array}**

Creates the petals Input Data Array

**Parameters:**

Name	Type	Description
intersectionPointsArray		Array containing the circle intersection point data
centerLatitude		Wind Rose center latitude
centerLongitude		Wind Rose center longitude

**Returns:**

[segmentID, direction, speed, hour, intersectionLat, intersectionLng, closestPointLat, closestPointLng, originDistance, destinationDistance, averageSpeed]

**(inner) createPinButton(coordinateX, coordinateY, height, width, unpinnedDir, pinnedDir, canvas, unpinnedID, pinnedID, roseID)**

Creates the wind rose pin button with two images (element "img" on svg)

**Parameters:**

Name	Type	Description
coordinateX		Button top-left x position
coordinateY		Button top-left y position
height		Image height
width		Image width
unpinnedDir		Unpinned image source
pinnedDir		Pinned image source
canvas		The canvas object
unpinnedID		Unpinned image ID
pinnedID		Pinned image ID
roseID		The Wind Rose ID

**(inner) createSelectionCheckbox(coordinateX, coordinateY, height, width, uncheckedDir, checkedDir, canvas, checkedID, uncheckedID, roseID)**

Creates the Selection Button element

**Parameters:**

Name	Type	Description
coordinateX		Button coordinate X
coordinateY		Button coordinate Y
height		Button height
width		Button width

uncheckedDir		Button unchecked image directory
checkedDir		Button checked image directory
canvas		Canvas object
checkedID		Button checked image ID
uncheckedID		Button unchecked image ID
roseID		The Wind Rose ID

**(inner) createTitleLabel(titleText, textColor, canvas, labelClassID, labelID, roseID)**

Creates the title text elements

**Parameters:**

Name	Type	Description
titleText		The title text
textColor		Title text color
canvas		The canvas object
labelClassID		The Label Class ID
labelID		The Label ID
roseID		The Wind Rose ID

**(inner) deleteDirectionalArrows(directionalArrowsClass)**

Removes all existing elements associated to the Wind Rose Directional Arrows

**Parameters:**

Name	Type	Description
directionalArrowsClass		Directional arrow elements class name

**(inner) deleteFooter(footerClass)**

Removes all existing elements associated to the Wind Rose footer

**Parameters:**

Name	Type	Description
footerClass		Footer elements class name

**(inner) deleteGridPercentage(gridPercentageClass)**

Deletes all Grid Percentage Labels elements

**Parameters:**

Name	Type	Description
gridPercentageClass		Grid Percentage Labels class name

**(inner) deletePetals(petalsClass)**

Removes all existing elements associated to the Wind Rose Petals

**Parameters:**

Name	Type	Description
petalsClass		Petals elements class name

**(inner) deleteRose()**

Deletes the current Wind Rose elements (SVG element along with all its children) and triggers a new 'roseDeleted' event

**(inner) distanceAndDirection(lat1, lng1, lat2, lng2, unit) → {Object}**

Method that calculates the distance between 2 points and the bearing, i.e., if a circle is drawn(imagined) around the first point at what degree would the second point be. Being North the degree 0

**Parameters:**

Name	Type	Description
lat1		first point's latitude (circle center latitude)
lng1		first point's longitude (circle center longitude)
lat2		second point's latitude
lng2		second point's longitude
unit		measure unit, meters or kilometers

**Requires:**

{string} unit - "m" (meters) or "Km" (kilometers)

**Returns:**

Object composed by distance, in meters or kilometers, and direction(bearing) in degrees

**(inner) findArrowColor(speed, roseScale) → {string}**

Returns the color value a speed represents, according to the color scale used on the Wind Rose

**Parameters:**

Name	Type	Description
speed		Speed value
roseScale		Data structure containing the color scale (speed values and corresponding colors)

**Returns:**

Hexadecimal color value

**(inner) findCircleIntersections(trackIDs, circleCenterLatitude, circleCenterLongitude, circleRadius) → {Array}**

Calculates intersection between the given tracks and circle

**Parameters:**

Name	Type	Description
trackIDs		Track IDs to be analysed for intersection
circleCenterLatitude		Circle center latitude
circleCenterLongitude		Circle center longitude
circleRadius		Circle radius

**Returns:**

Array with each track intersection points with the given circle

**(inner) findIntersectingTracksBounds(circleInstance, boundingBoxes, tracks) → {Array}**

Searches for track's ID's which bounding boxes intersect the given circle.

**Parameters:**

Name	Type	Description
circleInstance		The circle instance (element to be intersected)
boundingBoxes		The Tracks Bounding Boxes (stored in the same order as Tracks array)
tracks		The tracks objects array

**Returns:**

Object containing arrays of track IDs that intersect the circle

**(inner) firstLastSegmentPoints(segmentID, pointsData) → {Object}**

Retrieves the first and last point from a segment

**Parameters:**

Name	Type	Description
segmentID		The segment ID to search on
pointsData		All points data structure

**Returns:**

{firstPoint, lastPoint}

**(inner) generateFiltersData(segmentID, pointsData, referencePointLat, referencePointLon) → {Object}**

Generates the necessary data for petals creation and filters applying regarding a given intersection. This data contains: a point instance from the circle intersection closest point; the distance from the segment point to the Wind Rose center; the distance from the Wind Rose center to the segment last point; the segment average speed;

**Parameters:**

Name	Type	Description
segmentID		Target segment ID
pointsData		Data structure containing all point objects
referencePointLat		The reference point latitude
referencePointLon		The reference point longitude

**Returns:**

{ {Point} intersectionClosestPoint: Instance of the intersection closest point,  
 {Number} originDistance: Distance between the segment's first point and the Wind Rose center, {Number} destinationDistance: Distance between the segment's last point and the Wind Rose center, {Number} averageSpeed: Segment's average speed  
 }

**(inner) getInitAndFinalAngle(direction, amplitude, petalsData, gridSectors) → {Object}**

Filters data in order to get only data correspondent to the given direction's data. Returns a final and an initial angle in degrees, for a petal, according to the given amplitude percentage.

**Parameters:**

Name	Type	Description
direction		Petal direction in degrees
amplitude		Petal amplitude in percentage
petalsData		Array containing the intersection points Data
gridSectors		The number of grid sectors

**Returns:**

{{initialAngle: number, finalAngle: number, speedData: Array}}

**(inner) group(strokeFill, strokeColor, strokeWidth, classID, groupID, roseID) → {Object}**

Creates a group where added objects share the same attributes (defined on the group creation)

**Parameters:**

Name	Type	Description
strokeFill		Group stroke fill (filling color)
strokeColor		Group stroke color (line color)
strokeWidth		Group stroke width (line thickness)
classID		Group class name
groupID		Desired group ID
roseID		Wind rose ID

**Returns:**

Object that represents the element <g> in the SVG reference

**(inner) image(coordinateX, coordinateY, height, width, imageSource, rotation, imageID, roseID) → {Object}**

Creates an image element

**Parameters:**

Name	Type	Description
coordinateX		the x coordinate where the image will be displayed
coordinateY		the Y coordinate where the image will be displayed
height		image height
width		image width
imageSource		image source
rotation		image rotation
imageID		The image ID
roseID		The Wind Rose ID

**Returns:**

Object representing <img> element in the SVG reference

**(inner) line(x1, y1, x2, y2, lineID, roseID) → {Object}**

Creates a line element

**Parameters:**

Name	Type	Description
------	------	-------------

x1		Line's initial point x coordinate
y1		Line's initial point y coordinate
x2		Line's final point x coordinate
y2		Line's final point y coordinate
lineID		The line ID
roseID		The Wind Rose ID

**Returns:**

Object representing a <line> element in the SVG reference

**(inner) path(d, fill, pathID, roseID) → {Object}**

Creates a path element

**Parameters:**

Name	Type	Description
d		Set of commands that define the path, e.g., d="M 175 200 L 10 10" means that the drawing will start at point 175x,200y (the M means "move to") and will end at 10x,10y (the L means "line to").
fill		Filling color
pathID		The path ID
roseID		The Wind Rose ID

**Returns:**

Object representing a <path> in the SVG reference

**(inner) polarToCartesian(degreesAngle, radius, centerX, centerY) → {Object}**

Converts polar coordinates into cartesian coordinates

**Parameters:**

Name	Type	Description
degreesAngle		Angle in degrees
radius		Circle radius
centerX		Circle center latitude
centerY		Circle center longitude

**Returns:**

Cartesian coordinates values (x and y)

**(inner) polygon(points, strokeFill, strokeColor, rotation, classID, polygonID, roseID) → {Object}**

Creates a polygon element

**Parameters:**

Name	Type	Description
points		Polygon's points to draw
strokeFill		Polygon's filling color
strokeColor		Polygon's stroke color
rotation		Polygon's rotation value in degrees
classID		The polygon class ID
polygonID		The polygon ID
roseID		The Wind Rose ID

**Returns:**

Object representing a <polygon> element in the SVG reference

**(inner) processAverageSpeedFilter(averageSpeedMin, averageSpeedMax, petalsData) → {Array}**

Checks if a given intersection data stays inside the Average Speed range and returns a new structure with removed intersections that don't fulfill the given range.

**Parameters:**

Name	Type	Description
averageSpeedMin		Minimum average speed value
averageSpeedMax		Maximum average speed value
petalsData		Intersection data structure to filter

**Returns:**

Filtered intersection data

**(inner) processFilters(filterData)**

Applies all filters (sequentially) to the current Wind Rose. After applying all filters a new instance of filtered data intersections is stored on the proper global variable of the current Wind Rose

**Parameters:**

Name	Type	Description
filterData		Structure containing the filters values (scope, radius, hours)

**(inner) processHoursFilter(hoursValues, petalsData) → {Array}**

Returns a filtered data structure (from the original Petals In Data), containing only intersections that have 'hour' values in the given hoursValues

**Parameters:**

Name	Type	Description
hoursValues		Allowed hours values
petalsData		The original petals in data structure

**Returns:**

Filtered petals in data array, only containing valid intersections according to the filter

**(inner) processOriginDestinationFilter(originMin, originMax, destinationMin, destinationMax, petalsData) → {Array}**

Checks if a given intersection data stays inside the Origin/Destination distance range and returns a new structure with removed intersections that don't fulfill the given range.

**Parameters:**

Name	Type	Description
originMin		Minimum distance from segment origin to Wind Rose center
originMax		Maximum distance from segment origin to Wind Rose center
destinationMin		Minimum distance from Wind Rose center to segment destination
destinationMax		Maximum distance from Wind Rose center to segment destination
petalsData		Intersection data structure to filter

**Returns:**

Filtered intersection data

**(inner) rectangle(coordinateX, coordinateY, height, width, strokeFill, strokeColor, strokeWidth, classID, rectangleID, roseID) → {Object}**

Creates a rectangle element

**Parameters:**

Name	Type	Description
coordinateX		The X coordinate where the rectangle will be drawn
coordinateY		The Y coordinate where the rectangle will be drawn
height		The rectangle's height
width		The rectangle's width
strokeFill		The rectangle's color
strokeColor		The rectangle's border color
strokeWidth		The rectangle's border thickness
classID		The rectangle class ID
rectangleID		The rectangle ID

roseID		The Wind Rose ID
--------	--	------------------

**Returns:**

Object representing a <rect> element in the SVG reference

**(inner) removeEventListeners ()**

Removes all event listeners from the current Wind Rose instance

**(inner) selectRose ()**

Marks the current wind rose as selected and triggers a new 'roseSelected' event, if the current wind rose is not selected

**(inner) setCanvasStyle(roseID, backgroundColor, opacity, borderRadius, x, y)**

Creates the style elements for the rose canvas

**Parameters:**

Name	Type	Description
roseID		Canvas's rose ID
backgroundColor		Canvas background color
opacity		Canvas opacity
borderRadius		Canvas border radius
x		Canvas X position
y		Canvas Y position

**(inner) text(coordinateX, coordinateY, strokeFill, classID, textID, roseID) → {Object}**

Creates a text element

**Parameters:**

Name	Type	Description
coordinateX		The x coordinate where the text will be drawn
coordinateY		The y coordinate where the text will be drawn
strokeFill		Text color
classID		The text class ID
textID		The text ID
roseID		The Wind Rose ID

**Returns:**

Object representing a <text> element in the SVG reference

**(inner) toRadians (angle) → {number}**

Converts Degrees into Radians

**Parameters:**

Name	Type	Description
angle		The angle value in degrees

**Returns:**

The angle value in Radians

**(inner) unselectRose ()**

Marks the current wind rose as unselected and triggers a new 'roseSelected' event, if the current wind rose is selected



# Apêndice E

## Guião de avaliação de testes de utilizadores da aplicação *web windroseApp*

### 1. Introdução do trabalho

- Agradecimento pela participação no teste/projeto
- Introdução de quem está a fazer o teste (Nome, Área de mestrado, etc.)
- Enquadramento da aplicação (sem explicar propósito do teste):

- A visualização de dados agregados nem sempre é simples e clara. Grandes quantidades de dados podem esconder informações quando analisadas de forma separada. Existem muitos tipos de dados que podem ser agregados. No nosso caso, escolheram-se dados de trajetos, uma vez que estes podem ser úteis em ambiente urbano, quando aplicados a veículos/movimentos pendulares:

- Compreensão do fluxo de veículos em cidades (hora, dia, distância à origem/destino, etc.)

- Criou-se uma aplicação que permite visualizar dados agregados de trajetos, sobre mapas, recorrendo a rosas dos ventos
- Esta aplicação funciona através de um navegador web
- Nesta avaliação, foram carregados para a aplicação trajetos de táxis na cidade de Roma.

### 2. O que está a ser testado?

O objetivo deste estudo com utilizadores passa por avaliar o nível de usabilidade da interface do sistema e a qualidade geral da aplicação. A sua colaboração é muito importante para possamos quantificar estes aspetos da aplicação.

É importante salientar que não pretendemos avaliar o seu desempenho, mas sim o da aplicação.

### 3. Recolha e tratamento de dados

Antes de passarmos para a aplicação em si, vou fazer-lhe breves perguntas. Responderá a um simples questionário para recolher informações básicas sobre cada utilizador que efetua o teste (idade, género, etc). Os dados recolhidos serão usados apenas para este trabalho. Terá de assinar um documento em que consente fornecer esses dados. Poderá sair/desistir em qualquer momento.

No fim do teste responderá a um questionário SUS (A. I. Martins et al, 2017) relativo à usabilidade da aplicação.

#### ➤ Questionário Inicial

Este questionário serve meramente para retirar informações básicas sobre cada utilizador que efetua o teste (para efeitos estatísticos).

<https://goo.gl/forms/ZSWiEm9z3Su7CgnW2>

## ➤ Folheto Informativo

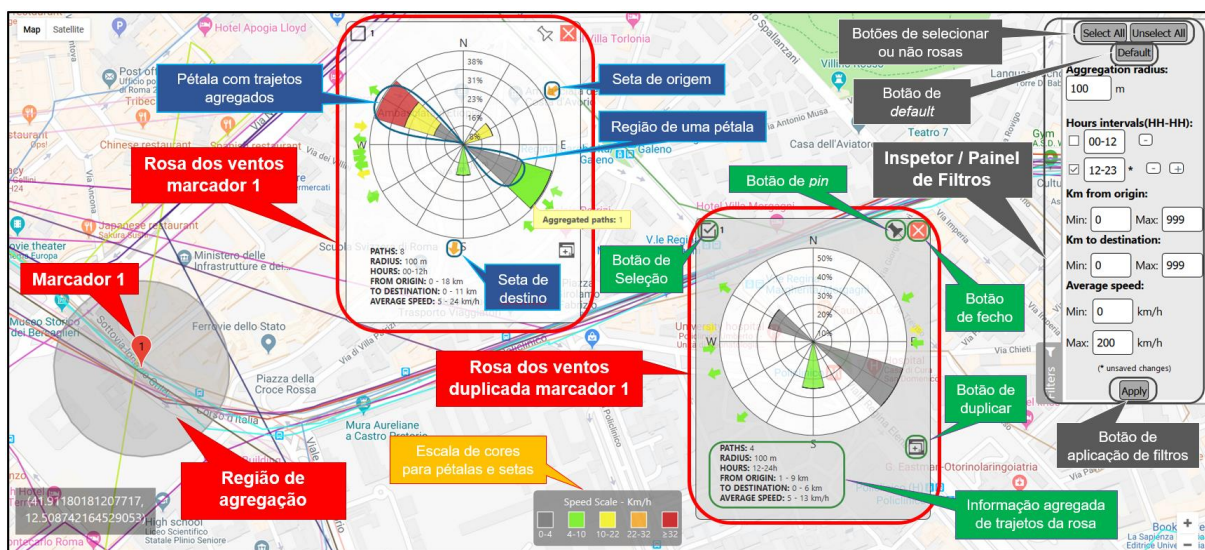
São apresentados os detalhes deste trabalho, a sua razão de ser, a sua utilidade potencial e as implicações da sua participação. Acompanhar o utilizador na leitura do folheto e responder a quaisquer perguntas que queira fazer (ver Apêndice G).

## ➤ Consentimento informado

São apresentadas várias alíneas em que o participante concorda em participar no estudo e assina em como autoriza a recolha de dados para o trabalho (ver Apêndice H).

## 4. Treino com a aplicação (Fase Treino)

- Explicação da interface e conceitos
- Mostrar a figura com a legenda da rosa dos ventos e explicar, por ordem (Marcador -> Rosa / Duplicado -> Elementos da Rosa -> Escala de cores -> Filtros)
- Botão direito em qualquer ponto para criar rosas dos ventos
- Explicar a escala de cores. Setas de origem/destino -> velocidade média do trajeto; Cores nas pétalas -> velocidade instantânea do trajeto quando interseta o raio de ação da rosa
- O funcionamento de filtros deve ser explicado recorrendo à aplicação “normal”
  - Mostrar o ajuste de filtros quando uma rosa é selecionada
  - Mostrar a criação de rosas com as propriedades do painel de filtros
  - Mostrar o aviso de alterações por guardar
  - Mostrar a notificação de incoerência entre rosas
  - Mover o marcador e verificar atualização da rosa
  - Mostrar a criação de cópias da rosa
  - Mostrar exemplo de movimentos pendulares
- Alertar para o facto de poderem existir cores muito parecidas e que na verdade são dois trajetos distintos! (Importante para os testes!)
- Deixar o utilizador explorar a aplicação com liberdade total



## **5. Avaliação com a aplicação (Fase de testes)**

O teste consiste em 10 perguntas. Cada pergunta apresenta um cenário com um desafio. As instruções de resposta de cada pergunta são dadas juntamente com o seu enunciado.

Caso acha que não consegue responder a uma determinada pergunta, não se preocupe. Passaremos à pergunta seguinte do teste.

Clique no botão 'Iniciar avaliação' assim que estiver pronto para iniciar o teste. Boa sorte!

## **6. Questionário final de usabilidade**

Para concluir esta sessão de avaliação, resta-lhe responder ao último questionário relativo à usabilidade da aplicação. Aqui, terá a oportunidade de apontar aspetos menos positivos da aplicação, bem como sugestões de melhoria.

<https://goo.gl/forms/IgJQspAXYs4MWA3I3>

## **7. Agradecimento Final**



# Apêndice F

## Questionário inicial dos testes

### Questionário inicial

Cada utilizador que testar e avaliar a aplicação de visualização analítica de trajetos a partir de rosas dos ventos responde a um breve questionário. Só serão recolhidas informações básicas do utilizador para fins estatísticos.

**\*Obrigatório**

1. Identificador do utilizador: \*

A sua resposta \_\_\_\_\_

2. Género: \*

Masculino

Feminino

3. Idade: \*

A sua resposta \_\_\_\_\_

4. Costuma visualizar percursos ou trajetos em mapas? \*

Sim

Não

5. Se respondeu Sim, que aplicações/ferramentas usa?

A sua resposta \_\_\_\_\_

6. Quando analisa trajetos, que características considera importantes? \*

Distância percorrida

Direção de origem/destino

Velocidade

Tempo decorrido

Período de ocorrência

Outra: \_\_\_\_\_

7. Avalie se teria facilidade ou dificuldade em analisar informação dos trajetos do mapa seguinte:



- Quantidade de informação suficiente e de fácil análise
- Quantidade de informação significativa e de razoável análise
- Demasiada informação e de difícil análise

**SUBMITER**

Nunca envie palavras-passe através dos Formulários do Google.

# Apêndice G

## Folheto informativo



**Ciências** | Comissão de Ética para a Recolha e  
**ULisboa** | Protecção de Dados de Ciências

### Folheto Informativo

**PROJECTO / TESE:** Evolução de Sistema de Análise de Trajetos com Rosas dos Ventos

**INVESTIGADOR / ALUNO:** Luís Paulo Garcia Ferreira

Vimos desta forma convidá-la/o a participar no meu trabalho focado em visualização analítica de trajetos urbanos com rosas dos ventos.

Antes de decidir, gostaria de lhe apresentar os detalhes deste trabalho, a sua razão de ser, a sua utilidade potencial e as implicações da sua participação. Irei acompanhá-lo na leitura deste folheto e responderá a quaisquer perguntas que queira fazer.

**1 - Em que consiste o estudo "Evolução de Sistema de Análise de Trajetos com Rosas dos Ventos"?**  
No contexto de Projeto de Engenharia Informática (PEI) do 2.º ano do Mestrado em Engenharia Informática, área de Sistemas de Informação, foi proposto um projeto para construção de um novo protótipo de um sistema de trajetos agregados com recurso a rosas dos ventos, tendo por base trabalhos anteriores. Pretende-se avaliar as funcionalidades, o nível de usabilidade da interface do sistema e a qualidade geral da aplicação.

**2 - Tenho de participar neste estudo?**

A participação no estudo é totalmente voluntária. Vamos descrever o estudo e apresentar o conteúdo deste folheto informativo, incluindo os detalhes da sua participação. Se concordar em participar, irá assinar um Formulário de Consentimento. Serão-lhe fornecidas cópias deste documento e do Formulário de Consentimento informado.

**3 - E se eu desejar desistir do estudo?**

É livre de desistir, em qualquer altura, sem ter que fornecer quaisquer razões ou explicações.

**4 - O que terei de fazer no âmbito do estudo?**

No âmbito do estudo, irá participar numa sessão que demorará um tempo máximo de 35 minutos mas não tem um tempo mínimo obrigatório. O teste consiste em 10 perguntas, em que é apresentado um cenário com um desafio. As instruções de resposta de cada pergunta são dadas juntamente com o seu enunciado. É importante salientar que não pretendemos avaliar o seu desempenho, mas sim o da aplicação. Todas as recomendações e melhorias sugeridas permitirão fazer evoluir o trabalho e são bem-vindas.

**5 - Quais as desvantagens e riscos de participar?**

Não estão previstos quaisquer riscos associados e a expectativa é de que a sessão em que participar seja uma experiência agradável.

**6 - Quais os possíveis benefícios de participar?**

De acordo com a nossa experiência, as pessoas gostam de participar em estudos que promovem a comunicação com cientistas. O seu envolvimento irá ajudar a perceber.

**7 - O que acontece quando o estudo terminar?**

A análise dos dados terminará em Março. Os resultados do estudo serão publicados em conferências e revistas académicas. Se desejar saber detalhes sobre os resultados e implicações do estudo, fazer-lhe-emos chegar uma cópia do relatório do estudo.



## Folheto Informativo

### 8 - A minha informação será mantida confidencial?

Sim. Seguiremos todas as práticas éticas e legais e toda a informação sobre si será tratada de forma absolutamente confidencial. Para garantir a anonimidade, os registos pessoais estarão apenas disponíveis na sua integralidade, para o investigador responsável, e os membros da equipa de investigação apenas terão acesso aos dados que necessitarem de conhecer. Se os seus dados forem usados para publicações ou apresentações, serão totalmente anonimizados, sem qualquer referência, directa ou indirecta, à sua identidade. Se forem tiradas fotografias, e for nossa intenção usá-las em alguma apresentação, será-lhe pedida autorização prévia. Pedir-lhe-emos primeiro que assine autorizações específicas com esse objectivo.

### 9 - Quem posso contactar relacionado com este estudo?

<p><b>Prof. António Ferreira</b> Faculdade de Ciências da Universidade de Lisboa <a href="mailto:amferreira@ciencias.ulisboa.pt">amferreira@ciencias.ulisboa.pt</a></p>	<p><b>Prof. Ana Paula Afonso</b> Faculdade de Ciências da Universidade de Lisboa <a href="mailto:apafonso@ciencias.ulisboa.pt">apafonso@ciencias.ulisboa.pt</a></p>	<p><b>Luís Ferreira</b> Faculdade de Ciências da Universidade de Lisboa <a href="mailto:fc45509@alunos.fc.ul.pt">fc45509@alunos.fc.ul.pt</a></p>
---	---	--

Sinta-se à vontade para os contactar em qualquer matéria relacionada com este estudo.

Muito Obrigado.

Obrigado por nos ter dedicado este seu tempo e por considerar participar neste estudo.

O Investigador/Aluno Responsável

Luís Ferreira

# Apêndice H

## Formulário de consentimento informado



Ciências | Comissão de Ética para a Recolha e  
ULisboa | Protecção de Dados de Ciências

### Formulário de Consentimento Informado

V2, 1-2-2018

PROJECTO / TESE: Evolução de Sistema de Análise de Trajetos com Rosas dos Ventos

INVESTIGADOR / ALUNO RESPONSÁVEL: Luís Paulo Garcia Ferreira

**Agradeço o seu interesse e colaboração neste estudo.  
Por favor, preencha o formulário que se segue. Receberá uma cópia quando sair.**

1. Confirmando que li e compreendi o folheto informativo associado ao projeto.
2. Foi-me dada a oportunidade de ler e considerar a informação apresentada, e fazer perguntas, as quais foram respondidas de forma satisfatória.
3. Compreendo que a minha participação é voluntária e que sou livre de desistir do estudo em qualquer altura, sem ter que dar quaisquer explicações e sem quaisquer consequências.
4. Compreendo que os dados recolhidos durante o estudo possam ser do conhecimento dos membros da equipa de investigação, sempre que necessário para o estudo. Autorizo que tenha acesso a esses dados.
5. Compreendo que, caso este trabalho venha a ser publicado, todos os dados serão mantidos anónimos e nenhuma informação será identificável como sendo minha.
6. Gostaria que me fosse enviado o relatório final do estudo.  
O meu endereço de e-mail é: \_\_\_\_\_
7. Gostaria de ser contactado para o endereço acima acerca de sessões ou estudos adicionais relacionados com este estudo.
8. Declaro que não comuniquei nenhuma razão potencial de qualquer natureza que constitua um eventual factor de risco para a minha saúde ou integridade física.
9. Declaro que participo neste estudo sem qualquer remuneração ou contrapartida.
10. Declaro que tomo a minha decisão de forma inteiramente livre.
11. Concordo em participar neste estudo.

\_\_\_\_\_  
Nome do Participante

\_\_\_\_\_  
Assinatura

\_\_\_\_/\_\_\_\_/\_\_\_\_  
Data

Sou da opinião que o participante compreendeu os aspectos relevantes da informação fornecida e está apto a tomar uma decisão informada.

\_\_\_\_\_  
Assinatura do Investigador / Aluno Responsável

\_\_\_\_/\_\_\_\_/\_\_\_\_  
Data



# Apêndice I

## Questionário de usabilidade (SUS)

### Questionário de Usabilidade (SUS) - Visualização e análise de trajetos com rosas dos ventos

Este é um questionário anônimo tem como objetivo inquirir o utilizador sobre a usabilidade da aplicação de visualização de trajetos com rosas dos ventos. Todos os dados pessoais recolhidos serão utilizados meramente para fins estatísticos.

Nas próximas 10 questões, classifique cada afirmação de acordo com o seu nível de concordância. Em caso de dúvida, opte pela opção 3.

\*Obrigatório

Identificação do utilizador: \*

A sua resposta

1. Gostaria de utilizar esta aplicação com frequência: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

2. Achei a aplicação desnecessariamente complexa: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

3. Achei a aplicação fácil de se usar: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

4. Precisaria da ajuda de uma pessoa com conhecimentos técnicos para utilizar a aplicação: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

5. As várias funcionalidades da aplicação estão muito bem integradas: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

6. A aplicação apresenta um elevado grau de incoerência: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

7. Acredito que a maioria das pessoas facilmente aprenderão a usar a aplicação: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

8. Achei a aplicação muito estranha de se usar: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

9. Senti-me muito confiante ao usar a aplicação: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

10. Precisei de aprender muitas coisas antes de conseguir usar a aplicação: \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

11. Sentiu alguma dificuldade ao usar a aplicação? \*

- Sim
- Não

12. Se respondeu afirmativamente à questão anterior, explique as dificuldades que encontrou:

A sua resposta

---

13. Sugestões para melhorar a aplicação:

A sua resposta

---

**SUBMITER**

Nunca envie palavras-passe através dos Formulários do Google.