

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



Visualização Analítica de Agregações de Trajetos com Rosas dos Ventos

Ricardo Neiva Lúcio Olaio Vaz

Mestrado em Engenharia Informática
Especialização em Sistemas de Informação

Dissertação orientada por:
Prof. Doutor António Manuel Silva Ferreira
Prof.^a Doutora Ana Paula Pereira Afonso

2017

Agradecimentos

A conclusão desta tese só foi possível com a ajuda de um conjunto de pessoas que sempre me apoiaram e ajudaram a alcançar os meus objetivos ao longo da minha vida.

Em primeiro lugar quero agradecer aos meus pais, irmão e avós, por todo o carinho e motivação que me deram ao longo de todos estes anos. Estiveram sempre comigo nos bons e maus momentos, tendo-me ajudado a ultrapassar situações menos fáceis.

Um grande obrigado ao Professor António Ferreira e à Professora Ana Paula Afonso, pela orientação, disponibilidade, dedicação e boa disposição que mostraram ao longo da elaboração da dissertação. A qualidade deste projeto também se deve a eles.

Aos meus amigos de longa data, pela camaradagem e apoio que deram. Passámos bons momentos juntos que serviram sempre de motivação para ultrapassar os desafios que me foram surgindo.

Aos meus colegas de curso que também participaram nesta caminhada comigo, desde os tempos da licenciatura.

A todos os participantes nas sessões de avaliação do sistema. A sua colaboração foi essencial para que fossem obtidos todos os dados necessários à quantificação de métricas importantes para o sistema.

Deixo também um agradecimento ao Professor João Telhada pelos contactos feitos que possibilitaram a ida à Rodoviária de Lisboa e ao Diretor do Centro de Atividades de Transporte de Bucelas da Rodoviária de Lisboa pela sua disponibilidade em avaliar o sistema desenvolvido nesta dissertação.

Um sincero obrigado a todos.

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.

Resumo

A rosa dos ventos é um gráfico radial usado em meteorologia para representar ventos dominantes num dado local, agregando registros de velocidade e direção recolhidos ao longo de um período de tempo, sendo usada, por exemplo, para apoiar a decisão da melhor direção para as pistas de um aeroporto. Pretendeu-se neste trabalho aplicar a rosa dos ventos a trajetos, em vez de ventos, sejam eles pedonais ou veiculares, e demonstrar a viabilidade desta ideia com o desenvolvimento de um sistema de análise de trajetos. Para isso, foram traçados três objetivos principais: a evolução de uma biblioteca de rosas dos ventos, o desenvolvimento de uma aplicação *web* para analisar trajetos em mapas, e a avaliação do sistema com utilizadores.

Relativamente ao primeiro objetivo, foram concretizadas novas funcionalidades da biblioteca, tais como a aplicação de novos filtros aos dados dos trajetos, bem como o suporte a uma base de dados geográfica e novas formas de manipular as rosas criadas através de um mecanismo de eventos. A biblioteca funciona de forma independente face a eventuais aplicações que tirem partido da mesma. Com estas novas funcionalidades, a biblioteca ficou mais flexível no que toca à sua utilização, mais rápida relativamente ao processamento de trajetos, e menos estrita ao nível de aplicação de filtros sobre os mesmos. A concretização do primeiro objetivo foi conseguida com a utilização das funcionalidades da biblioteca numa aplicação *web*, que está relacionada com o objetivo seguinte.

O segundo objetivo deste trabalho foi a construção de uma aplicação *web* que permite a análise de trajetos com rosas dos ventos, tirando partido das funcionalidades que a biblioteca oferece. A aplicação *web* tem uma interface gráfica que permite criar rosas dos ventos, sobre um mapa com trajetos e manipular as mesmas através de um conjunto de filtros. Os filtros podem ser definidos num painel lateral, que também serve de inspetor das configurações aplicadas a cada rosa.

O terceiro e último objetivo foi a avaliação do sistema através de testes com utilizadores, nos quais se pretendeu perceber a facilidade de análise de trajetos e o grau de usabilidade do sistema. Os resultados dos testes mostraram que o sistema possui um grau de usabilidade elevado e que todos os utilizadores conseguiram interpretar os dados

mostrados nas rosas, mesmo sem terem qualquer contacto prévio com este tipo de gráfico e análise de trajetos sobre mapas.

A aplicação tem potencial para ser utilizada em vários domínios, tendo sido feita uma demonstração numa empresa de transportes, onde foi sugerido por um perito que o sistema poderia ajudar na análise e otimização dos trajetos percorridos pela frota de autocarros.

Palavras-chave: Agregação de Trajetos; Visualização Analítica; Rosa dos Ventos; Base de Dados Geográfica.

Abstract

The wind rose is a radial graphic used in meteorology to represent prevailing winds at a given location, aggregating speed and direction records collected over a period of time. This data can be used, for example, to help determine the best direction for an airport highway. This project aims to apply the wind rose, not to winds, but to pedestrian or vehicular tracks, and demonstrate the viability of this idea, by building a track analysis system with wind roses. To achieve this, three main objectives were established: the evolution of an existing wind rose library, the development of a web application to analyze tracks over maps, and a system evaluation with users.

Regarding the first objective, new library functionalities were implemented, such as the application of new data filters, as well as the support for a spatial geographical data base, and new ways to manipulate wind roses through an events mechanism. The library works independently from any applications that might take advantage of it. With these new features, the library became more flexible, faster regarding tracks processing, and less strict regarding filters that can be applied to the tracks. The first objective was achieved by using the new library features in a web application, which is related to the next objective.

The second objective aimed to construct a web application for analyzing tracks using wind roses, taking advantage of the features provided by the library. The web application has a graphical user interface, which allows to create wind roses over a map with tracks and manipulate them through a set of filters. These filters can be applied using a side panel, which also serves to inspect the settings applied to each rose.

The third and last objective was the evaluation of the track analysis system with users. The purpose of this evaluation was to understand the ease of track analysis and degree of usability provided by the system. The results showed that the system has a high usability and all users successfully managed to interpret the data shown on the wind roses, even without having any previous experience with this type of graphic and track analysis over maps.

The application has potential applicability in multiple domains. A demonstration was made to an expert from a public bus company, where it was suggested that the system could help in the analysis and optimization of the tracks covered by the company's fleet.

Keywords: Track Aggregation; Visual Analytics; Wind Rose; Geographical Data Base.

Conteúdo

| | |
|---|----|
| Lista de figuras | ix |
| Lista de tabelas | xi |
| Capítulo 1 Introdução | 1 |
| 1.1 Motivação | 1 |
| 1.2 Objetivos | 2 |
| 1.3 Planeamento e execução..... | 4 |
| 1.4 Contribuições | 4 |
| 1.5 Notação..... | 5 |
| 1.6 Organização do documento | 6 |
| Capítulo 2 Enquadramento | 7 |
| 2.1 Conceitos | 7 |
| 2.1.1 Rosa dos ventos | 7 |
| 2.1.2 Trajetos, mapas e dados GPX | 9 |
| 2.1.3 Agregação de dados | 13 |
| 2.1.4 Visualização analítica de dados | 14 |
| 2.1.5 Serviços <i>web</i> | 15 |
| 2.2 Aplicações e sistemas existentes | 16 |
| 2.2.1 Aplicações de visualização analítica..... | 16 |
| 2.2.2 Protótipo anterior do sistema de análise de trajetos..... | 17 |
| 2.3 Sumário | 20 |
| Capítulo 3 Sistema de análise de trajetos com rosas dos ventos..... | 23 |
| 3.1 Ambiente de trabalho | 23 |
| 3.1.1 Ferramentas e tecnologias..... | 23 |
| 3.1.2 Processo de desenvolvimento | 24 |
| 3.2 Requisitos do sistema | 25 |
| 3.3 Desenvolvimento do sistema..... | 25 |
| 3.3.1 Arquitetura do sistema | 26 |
| 3.3.2 Base de dados geográfica..... | 27 |
| 3.3.3 Serviços <i>web</i> | 30 |
| 3.3.4 Biblioteca da rosa dos ventos..... | 37 |

| | | |
|------------|--|----|
| 3.3.5 | Aplicação <i>web</i> de análise de trajetos | 39 |
| 3.3.6 | Bibliotecas externas | 48 |
| 3.4 | Preparação e análise de um conjunto de trajetos | 49 |
| 3.4.1 | Dados de trajetos de táxis | 49 |
| 3.4.2 | Criação do ficheiro GPX com trajetos de táxis..... | 51 |
| 3.4.3 | Conversão e carregamento dos trajetos | 53 |
| 3.4.4 | Seleção de trajetos para a aplicação de análise..... | 54 |
| 3.4.5 | Criação de uma rosa dos ventos na aplicação de análise..... | 55 |
| 3.5 | Sumário | 57 |
| Capítulo 4 | Avaliação do sistema | 59 |
| 4.1 | Propósito..... | 59 |
| 4.2 | Estudo com utilizadores | 59 |
| 4.2.1 | Participantes..... | 59 |
| 4.2.2 | Aparato..... | 60 |
| 4.2.3 | Tarefas | 62 |
| 4.2.4 | Procedimento | 65 |
| 4.3 | Resultados | 67 |
| 4.3.1 | Tempos de resposta..... | 67 |
| 4.3.2 | Erros..... | 68 |
| 4.3.3 | Usabilidade do sistema | 69 |
| 4.4 | Discussão..... | 71 |
| 4.4.1 | Tempos de resposta..... | 71 |
| 4.4.2 | Erros..... | 72 |
| 4.4.3 | Usabilidade do sistema | 72 |
| 4.4.4 | Tempos de recuperação | 73 |
| 4.4.5 | Influência da experiência prévia com análise de trajetos | 73 |
| 4.4.6 | Fluxos de trabalho..... | 73 |
| 4.4.7 | Opinião de um perito | 74 |
| 4.5 | Sumário | 76 |
| Capítulo 5 | Conclusões | 77 |
| 5.1 | Principais contribuições | 77 |
| 5.2 | Competências adquiridas..... | 78 |

| | | |
|------------|--|-----|
| 5.3 | Desafios encontrados..... | 79 |
| 5.4 | Trabalho futuro..... | 79 |
| | Bibliografia..... | 81 |
| Apêndice A | Artigo apresentado na 12º CISTI..... | 85 |
| Apêndice B | Documentação da biblioteca <i>windroseLib</i> | 91 |
| Apêndice C | Questionário inicial dos testes | 107 |
| Apêndice D | Guia rápido da aplicação <i>web windroseMain</i> | 109 |
| Apêndice E | Questionário de usabilidade (SUS)..... | 111 |

Lista de figuras

| | |
|--|----|
| Figura 1 – Rosa dos ventos do aeroporto da Portela, em Lisboa, agregando ventos registados ao longo de 30 anos [5] | 8 |
| Figura 2 – Visualização de um percurso registado através da aplicação Geo Tracker ... | 10 |
| Figura 3 – Representação visual do estado do trânsito em Colónia, Alemanha, às 7:00h, num dia de semana, numa área de 400Km ² [3] | 11 |
| Figura 4 – Diagrama do modelo de visualização de Van Wijk [17] | 14 |
| Figura 5 – Aplicação <i>web</i> do primeiro protótipo do sistema [6]..... | 18 |
| Figura 6 –Funcionamento da aplicação <i>web</i> do primeiro protótipo do sistema | 20 |
| Figura 7 – Arquitetura do sistema de análise de trajetos com rosas dos ventos..... | 26 |
| Figura 8 – Modelo de dados em Entidade-Associação (EA)..... | 28 |
| Figura 9 – Exemplo de funcionamento da aplicação de gestão <i>selectTracks</i> | 41 |
| Figura 10 – Exemplo de funcionamento da aplicação de análise <i>windroseMain</i> , com uma rosa dos ventos | 42 |
| Figura 11 – Rosas dos ventos criadas em três pontos distintos do mapa para análise de trajetos | 45 |
| Figura 12 – Exemplo de um erro de leitura com sinal GPS | 52 |
| Figura 13 – Exemplo do erro de leitura quando um veículo se encontra parado num dado intervalo de tempo e volta a retomar a circulação | 52 |
| Figura 14 – Exemplo de uma situação em que ocorre uma incoerência temporal num dado trajeto, devido a perdas ou desligar de sinal..... | 52 |
| Figura 15 – Cenário de conversão e carregamento de um GPX para a base de dados, com posterior consulta de trajetos | 54 |
| Figura 16 – Cenário de criação de um ficheiro GPX com trajetos de táxis e carregamento para a aplicação de análise de trajetos do sistema | 55 |
| Figura 17 – Cenário de criação de um marcador na aplicação de análise <i>windroseMain</i> | 56 |
| Figura 18 – Uma questão do tipo INSE mostrada durante os testes do sistema..... | 64 |
| Figura 19 – Tempos de resposta dos utilizadores, por questão | 67 |
| Figura 20 – Tempos de resposta dos utilizadores, por tipo de questão | 67 |
| Figura 21 – Tempos de resposta dos utilizadores, por ordem de questão | 68 |

| | |
|---|----|
| Figura 22 – Tempos de resposta, por utilizador | 68 |
| Figura 23 – Erros ao longo dos testes, por tipo de questão | 68 |
| Figura 24 – Classificações de usabilidade do sistema, atribuídas pelos utilizadores | 69 |
| Figura 25 – Classificações de usabilidade do sistema, por género de utilizador..... | 69 |
| Figura 26 – Pontuações atribuídas em cada pergunta do questionário SUS | 70 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 – Planeamento e execução das tarefas do projeto..... | 4 |
| Tabela 2 – Exemplo de pedidos AJAX para o serviço <i>manageData</i> | 35 |
| Tabela 3 – Identificadores das questões do teste e respetivas descrições | 63 |

Capítulo 1

Introdução

No âmbito da disciplina de Projeto de Engenharia Informática (PEI) do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa, surgiu o projeto de construção de um novo protótipo de um sistema de análise de trajetos agregados com recurso a rosas dos ventos, já existente. O primeiro protótipo surgiu igualmente como projeto de mestrado, na mesma faculdade, sendo que neste projeto pretende-se evoluir as funcionalidades existentes, passando a ser possível efetuar análises sobre grandes quantidades de trajetos.

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

1.1 Motivação

É comum as organizações possuírem grandes quantidades de dados, que excedem a capacidade humana de processamento, havendo a necessidade de os agregar e filtrar para possibilitar a descoberta de padrões e tendências [1]. A representação gráfica dos dados agregados e a interação com os mesmos são aspetos importantes para o bom desempenho dos decisores, estando na essência da visualização analítica de dados [2].

Os dados tratados ao longo deste trabalho são trajetos, que podem ser percorridos por entidades, tais como pessoas, automóveis, aviões, e barcos, e que se presume terem sido registados por dispositivos de posicionamento global (GPS), ubíquos nos dias de hoje. A análise de trajetos é usada em tarefas de planeamento, por exemplo, urbano, em que a agregação dos trajetos permite visualizar movimentos pendulares da população e reorganizar o fluxo de tráfego, como foi feito na cidade de Colónia, na Alemanha [3]. Neste caso, os trajetos foram agregados com linhas de diferentes espessuras, e a passagem do tempo foi visualizada com mapas animados, técnicas também usadas na demonstração de um estudo com trajetos em Pequim, na China [4].

Neste trabalho procura-se ir além da informação disponibilizada pelos mapas animados: enquanto estes permitem aplicar sucessivamente filtros temporais que revelam,

por exemplo, onde há mais circulação automóvel em cada hora do dia, descreve-se neste documento um sistema capaz de mostrar, de forma agregada, de onde vieram e para onde foram esses condutores. Desta forma, em vez dos decisores terem de interpretar múltiplas imagens em busca de padrões para a acumulação de automóveis, passam a dispor de explicações sobre o trânsito.

O sistema integra uma aplicação *web*, uma biblioteca e serviços suportados por uma base de dados geográfica para oferecer ao decisor a possibilidade de analisar trajetos num mapa colocando e interagindo com gráficos inspirados nas rosas dos ventos. Uma rosa dos ventos é um gráfico radial convencionalmente usado na representação de ventos predominantes numa zona, tal como o aeroporto de Lisboa [5]. Aplicando a rosa dos ventos aos dados dos trajetos, a pétala maior é aquela cuja direção agrega mais trajetos e a maior secção colorida dentro de cada pétala representa o intervalo de velocidades mais comum. A estes elementos gráficos, acrescentaram-se filtros para, por exemplo, agregar trajetos realizados só de manhã ou de tarde, e setas de origem e destino dos trajetos no rebordo da rosa.

Este trabalho pretende dar continuidade a um primeiro protótipo desenvolvido no âmbito de uma tese de mestrado [6]. Esse protótipo conta com uma biblioteca de rosas dos ventos e uma aplicação *web* que tira partido das funcionalidades oferecidas pela biblioteca. No entanto, tem um conjunto de limitações, como por exemplo, a aplicação de filtros com valores muito estritos, a uma rosa de cada vez, e apenas ser aceite um único ficheiro com um conjunto de trajetos limitado. Estas limitações são ultrapassadas com o atual sistema descrito ao longo deste documento.

1.2 Objetivos

O grande propósito deste trabalho foi construir um sistema de análise de trajetos com rosas dos ventos que seja útil e fácil de usar, para o qual foram definidos os seguintes objetivos:

1. Evolução da biblioteca da rosa dos ventos
2. Construção de uma aplicação *web*
3. Avaliação do sistema de análise

De seguida é feita uma descrição de cada um dos três principais objetivos.

Objetivo 1: Evolução da biblioteca da rosa dos ventos

Este objetivo está relacionado com a evolução da biblioteca da rosa dos ventos já existente, construída no primeiro protótipo do sistema. A evolução da biblioteca é feita através da reestruturação do funcionamento da mesma, bem como da implementação de novas funcionalidades, tais como a inclusão de novos filtros para os dados dos trajetos. Pretende-se que todas estas funcionalidades estejam facilmente acessíveis a uma aplicação cliente, através da invocação dos métodos que a biblioteca oferece.

O cumprimento deste objetivo passou pela construção de uma aplicação *web* que faz uso das várias funcionalidades oferecidas pela biblioteca, a qual está relacionada com o segundo objetivo.

Objetivo 2: Construção de aplicação *web*

A aplicação *web* deve conseguir integrar as funcionalidades da biblioteca da rosa dos ventos com trajetos num mapa. Pretende-se que o utilizador seja capaz de criar e manipular rosas dos ventos em vários pontos de um mapa com trajetos, através da aplicação de filtros.

De forma a que o processamento dos trajetos seja mais flexível e eficiente, devem ser criados serviços *web* que permitam o armazenamento e consulta destes dados, o que motiva a criação de uma base de dados geográfica com os dados dos trajetos visualizados na aplicação *web*.

Este segundo objetivo é concretizado através de uma aplicação funcional, acedida por um *browser*, que tire partido das funcionalidades da biblioteca, dos serviços *web* e da base de dados, de forma transparente para o utilizador.

Objetivo 3: Avaliação do sistema

É importante perceber se as rosas dos ventos constituem um mecanismo eficaz de análise de trajetos agregados sobre um mapa. Para isso, deve-se recorrer a um conjunto de testes com utilizadores que ponham à prova as várias funcionalidades que o sistema oferece. No final, deve ser obtido um *feedback* quanto à usabilidade do sistema e eficácia na visualização de trajetos agregados com rosas dos ventos. Este objetivo é concretizado caso sejam reunidos dados de um número adequado de participantes nas sessões de avaliação do sistema e os resultados sejam positivos.

1.3 Planeamento e execução

O trabalho teve início no dia 30 de setembro de 2016. O prazo estipulado para a sua conclusão foi de 9 meses, o que previu que estivesse concluído a 30 de junho de 2017.

A Tabela 1 mostra que quase todas as tarefas foram cumpridas dentro do prazo estipulado. No entanto, a escrita e apresentação de um artigo numa conferência não estava prevista. Esta tarefa causou algum impacto no desenrolar das avaliações com utilizadores, o que fez com que o processo de desenvolvimento da aplicação de testes e início das avaliações se atrasasse e estendesse para dois meses. Este atraso motivou a que o relatório final tenha sido escrito no decorrer dos meses de julho e setembro, o que fez 11 meses para a conclusão de todo o projeto. Esta contagem não teve em conta o mês de agosto, uma vez que os departamentos e a maioria dos serviços académicos se encontravam encerrados.

Tabela 1 – Planeamento e execução das tarefas do projeto

| | Duração prevista (meses) | Duração real (meses) | Intervalo de execução |
|--|--------------------------|----------------------|-------------------------|
| Pesquisa da literatura sobre visualização analítica e rosas dos ventos | 2 | 2 | Outubro a novembro |
| Familiarização com biblioteca existente de funções para rosas dos ventos | 2 | 2 | |
| Análise das novas funcionalidades de interação com o utilizador | 2 | 2 | |
| Especificação de um cenário de análise de dados e suas tarefas de decisão | 2 | 2 | |
| Escrita do relatório preliminar | 2 | 2 | |
| Desenho e codificação da biblioteca aumentada e de um protótipo funcional | 6 | 6 | Dezembro a maio |
| Escrita do artigo e preparação da apresentação para a conferência CISTI | - | 1 | Abril e junho |
| Avaliações da correção e usabilidade do protótipo | 1 | 2 | Maio a junho |
| Avaliação do impacto da rosa dos ventos num cenário de tomada de decisão | 1 | 2 | |
| Escrita do relatório final | 1 | 2 | Julho e setembro |

1.4 Contribuições

Foram feitas 4 contribuições principais resultantes da execução do projeto.

A primeira contribuição está relacionada com a reestruturação e aumento de funcionalidades da biblioteca da rosa dos ventos anterior, o que levou a uma maior capacidade de análise de trajetórias, nomeadamente com a inclusão de novos filtros sobre

os dados que estão a ser mostrados em cada rosa. Foi criada uma base de dados geográfica e implementados serviços *web* como consequência do processo de desenvolvimento de aplicações na *web*, servindo principalmente para o processamento e armazenamento de dados de trajetos. Tome-se como exemplo, o caso em que o navegador *web* não consegue efetuar acessos diretos a uma base de dados, pelo que são necessários serviços adicionais para este efeito. A reestruturação da biblioteca alterou o funcionamento da aplicação de filtros, passando a basear-se em eventos, o que permite a manipulação de várias rosas em simultâneo.

A segunda contribuição foi a construção de uma aplicação *web*, onde foram implementadas novas funcionalidades que permitiram expandir a forma como o utilizador interage com as várias rosas dos ventos, criadas a partir da biblioteca. Estas funcionalidades incluem um novo painel de filtros, que permite manipular uma ou várias rosas em simultâneo e a integração de um inspetor no painel, que oferece a possibilidade de visualizar as definições que estão aplicadas a uma ou mais rosas, assinalando eventuais incoerências.

A terceira contribuição está associada à avaliação do sistema de análise de trajetos com rosas dos ventos, através de sessões com utilizadores. Para além da usabilidade, testou-se o grau de capacidade de análise de trajetos que as rosas dos ventos conferem aos utilizadores. Os testes mostraram dois aspetos importantes: o sistema possui um grau de usabilidade elevado e todos os utilizadores conseguiram cumprir com sucesso as tarefas que lhes foram propostas. Estes factos demonstram que as rosas dos ventos constituem uma boa ferramenta de análise para a visualização de dados de trajetos agregados.

A última contribuição está relacionada com a escrita e apresentação de um artigo sobre o sistema desenvolvido na 12^o Conferência Ibérica de Sistemas e Tecnologias de Informação, tendo decorrido em Lisboa durante o mês de junho [7] (consultar Apêndice A).

1.5 Notação

A língua escolhida para a escrita deste relatório foi o português. Todas as palavras noutra idioma são escritas em itálico. Todo o código é apresentado na fonte *Courier New*.

1.6 Organização do documento

Este documento está organizado em cinco capítulos, descritos a seguir.

No Capítulo 1 são dados a conhecer os seguintes aspetos: motivação, objetivos do trabalho, planeamento e execução de tarefas, contribuições, notação adotada no documento e organização do mesmo.

No Capítulo 2 é feito o enquadramento do trabalho. Inclui-se uma descrição do estado de arte das várias soluções existentes para a representação de trajetórias e visualização analítica de conjuntos de dados, bem como os tipos de dados agregados mais comuns e que tipo de operações se realizam com os mesmos. Como trabalho relacionado, é ainda abordada a existência de um protótipo anterior do sistema, bem como as limitações que este acarreta. São ainda apresentados conceitos teóricos relacionados com a temática do trabalho.

No Capítulo 3 é documentado todo o processo de desenvolvimento do sistema de análise de trajetos com rosas dos ventos, incluindo as ferramentas utilizadas, o fluxo de trabalho da equipa e a descrição e funcionamento dos vários componentes que compõem o sistema. São ainda apresentados vários exemplos de utilização do sistema com o intuito de mostrar a forma como os diferentes componentes interagem entre si.

No Capítulo 4 são apresentados os aspetos relacionados com a avaliação do sistema, nomeadamente, a forma como os testes foram realizados, incluindo a aplicação responsável pela execução dos testes, os resultados das avaliações e ainda a opinião de um perito.

O Capítulo 5 conclui o documento e apresenta as principais contribuições, competências adquiridas, dificuldades encontradas e ideias de trabalho futuro.

Capítulo 2

Enquadramento

Neste capítulo são introduzidos conceitos teóricos relacionados com o tema do projeto, como por exemplo, a rosa dos ventos e a agregação de dados, e mencionadas aplicações que apresentam soluções com semelhanças ao sistema desenvolvido. Estas aplicações incluem soluções com mapas estáticos e animados, visualização analítica e um primeiro protótipo do sistema de análise de trajetos com rosas dos ventos.

2.1 Conceitos

Nesta secção são apresentados os conceitos utilizados no desenvolvimento do trabalho, incluindo a rosa dos ventos, agregação de dados, visualização analítica de dados e serviços *web*.

2.1.1 Rosa dos ventos

A rosa dos ventos é um tipo de gráfico clássico que tem uma representação circular [8]. A sua utilização mais comum é na área da meteorologia, para visualizar a direção, frequência e velocidade dos ventos num determinado local.

A Figura 1 mostra uma rosa dos ventos com os ventos predominantes no aeroporto da Portela, agregando registos anuais recolhidos ao longo de várias décadas. Ao observar-se a rosa, verifica-se que existe um conjunto de 4 círculos em torno do ponto central. Cada círculo define os valores da escala de ocorrência em horas anuais, sendo que a escala começa a partir das 0 horas anuais no ponto central dos vários círculos e vai até às 2000 horas anuais no círculo da periferia.

As direções de ocorrência dos ventos são descritas a partir dos ângulos formados pelas divisórias dos vários círculos. Os 0° apontam para norte, 90° para este, 180° para sul e, por fim, 270° para oeste. A contabilização dos graus é feita no sentido horário.

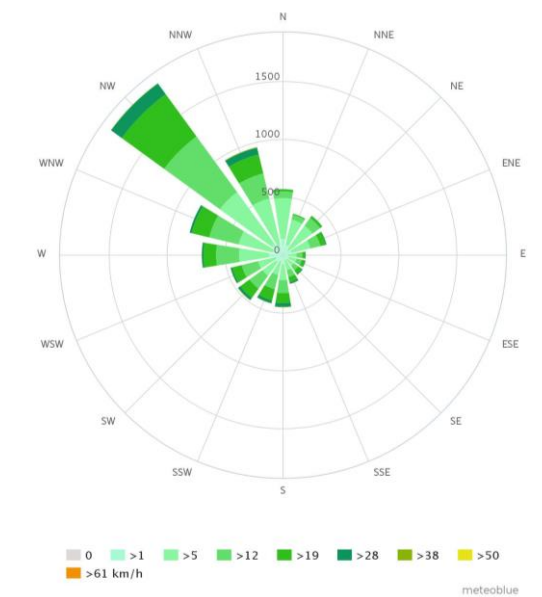


Figura 1 – Rosa dos ventos do aeroporto da Portela, em Lisboa, agregando ventos registados ao longo de 30 anos [5]

Cada pétala da rosa representa ventos provenientes de uma dada direção, sendo que a mesma é lida através do posicionamento no círculo e está associada a um ponto cardinal tal como oeste (O), colateral, como por exemplo nordeste (NE) ou subcolateral, como é o caso de nor-noroeste (NNW). O tamanho da pétala está diretamente relacionado com o número de dados que a mesma agrega. Por exemplo, na Figura 1 a maior pétala é a que se encontra voltada para noroeste, sendo esta a que agrega o maior conjunto de ventos, com ocorrências superiores a 1500 horas anuais. Dentro desta pétala existem quatro secções com cores distintas, que permitem distinguir as diferentes velocidades dos vários ventos agregados, sendo que a maioria ocorreu com velocidades entre os 5 e os 19 Km/h. A escala de cores das pétalas está definida no rodapé da figura e vai desde os 0 Km/h até acima dos 61 Km/h.

Apesar da sua função base estar relacionada com a análise de ventos, tanto a nível de direção como velocidade, é possível aplicar a rosa dos ventos a dados de domínios variados. Recentemente, foi elaborado um estudo com o intuito de perceber a influência dos ventos na movimentação das nuvens de fumo, nos dois dias que se seguiram após a ocorrência do incêndio de 25 de agosto de 1988, em Lisboa [9]. Para a concretização desta análise, foi utilizada uma rosa dos ventos com o centro definido no aeroporto da Portela, local onde estavam disponíveis registos de ventos referentes ao momento da ocorrência do incêndio. Através da rosa, foi possível descrever a evolução da direção

dos ventos ao longo de 24 horas, bem como a sua velocidade. Com estes dados e através de um algoritmo de simulação, foi possível obter as várias trajetórias das nuvens de fumo, resultantes da influência dos ventos presentes naqueles instantes.

Numa outra vertente, foi feito um estudo para avaliar se a poluição gerada ao longo da costa sudeste do lago Michigan nos Estados Unidos, maioritariamente proveniente das cidades, afetava os níveis de poluição do lago [10]. Neste caso, a rosa dos ventos serviu de mecanismo visual para a interpretação da distribuição de ventos ao longo de 10 anos e foram utilizados valores médios de velocidade e direção. Concluiu-se, com o auxílio dos dados fornecidos pela rosa, que a poluição atmosférica presente naquela zona, contribuiu significativamente para a presença de alguns elementos químicos poluentes na água do lago.

Na área do planeamento de infraestruturas, os aeroportos necessitam de uma análise cuidada relativamente à predominância de ventos na zona de construção desejada. Tal facto deve-se à necessidade de colocação estratégica da pista de aterragem/descolagem, que deve ser posicionada segundo os ventos predominantes. O sistema ARO procura facilitar a execução desta tarefa, recorrendo à analogia da rosa dos ventos para a análise de grandes conjuntos de dados de ventos em determinados locais [11].

Na mesma vertente que a aplicação anterior, existe um estudo das condições meteorológicas em cinco zonas do Dubai para a colocação de geradores eólicos [12]. Em cada uma dessas zonas foram recolhidos dados relativos à predominância de ventos, os quais foram analisados com o auxílio da rosa dos ventos. A rosa revelou-se útil para classificar os ventos predominantes em cada uma das cinco zonas, permitindo uma escolha adequada na colocação dos geradores eólicos.

2.1.2 Trajetos, mapas e dados GPX

De seguida é descrito o tipo de dados que o sistema desenvolvido analisa, bem como as visualizações comuns que lhe estão associadas. É ainda descrito um formato que permite armazenar dados de trajetos e como é que a rosa dos ventos pode ser adaptada a este domínio.

Trajetos e mapas

Os trajetos constituem o tipo de dados que o sistema desenvolvido analisa. 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 projetadas num mapa. Esta é a abordagem dos serviços de mapas da Microsoft [13] ou da Google [14]. Estas soluções também estão presentes em aplicações móveis, como é o caso do Geo Tracker para Android [15], onde existe a possibilidade do utilizador registar trajetos com o sinal GPS recebido no telemóvel e visualizá-los através do serviço de mapas da Google. A Figura 2 mostra uma captura de ecrã dessa aplicação, onde está a ser visualizado, num mapa da Google, um trajeto registado. Este tipo de aplicações enquadra-se na visualização com mapas estáticos.

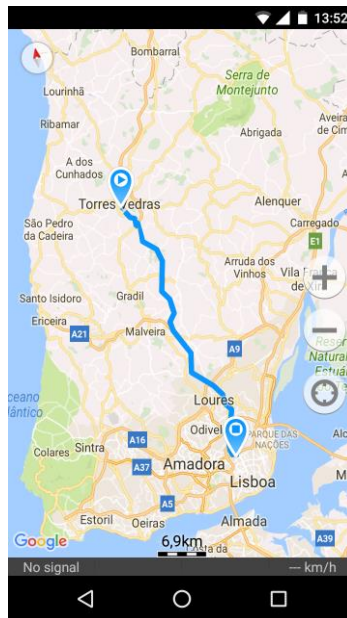


Figura 2 – Visualização de um percurso registado através da aplicação Geo Tracker

Apesar de ser intuitivo visualizar trajetos num mapa, quando estes são muitos ou se sobrepõem passa a ser adequado fazer agregações. Uma técnica habitual consiste em juntar vários trajetos semelhantes numa só linha com traço mais grosso. Este tipo de visualização pode ser animado com uma representação dos trajetos ao longo do tempo, resultando na produção de mapas animados. Estes mapas descrevem trajetórias através de pontos que se movem ao longo do tempo.

Existem estudos que procuram compreender a mobilidade de veículos de uma cidade, nomeadamente com a construção de um modelo de visualização de dados agregados que acompanha a evolução da circulação de veículos na rede de estradas do

centro de Colónia, na Alemanha [3]. Este modelo enquadra-se no conceito de mapas animados. Na Figura 3 estão representados pontos que correspondem a veículos em movimento na área representada. A cor de cada ponto está associada à velocidade a que o veículo se encontrava a circular no instante representado. É visível que o maior congestionamento de tráfego ocorre na zona central da cidade, sendo que na periferia existem veículos a circular de forma fluida com velocidades superiores a 90 Km/h.

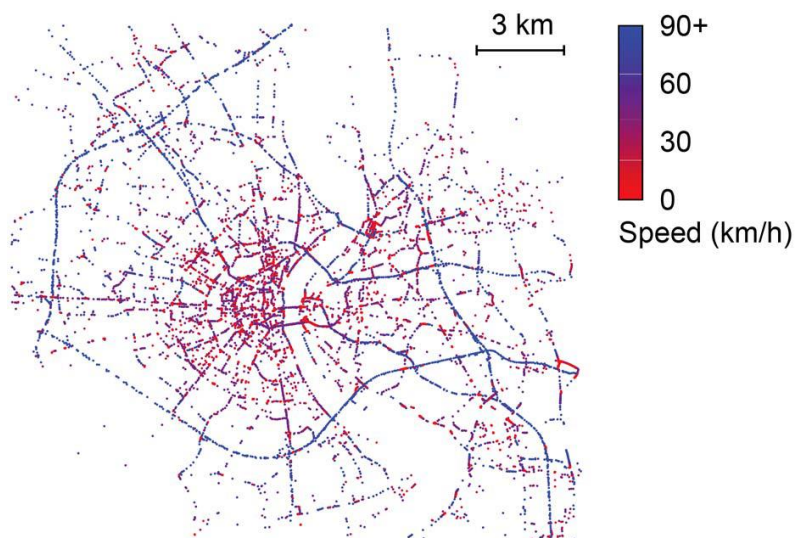


Figura 3 – Representação visual do estado do trânsito em Colónia, Alemanha, às 7:00h, num dia de semana, numa área de 400Km² [3]

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 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, uma vez que está dependente da duração da animação. Este tipo de visualização também foi utilizado num estudo de análise de tráfego que originou a construção do sistema T-Drive, cujo objetivo passou por obter o melhor conjunto de direções de navegação na cidade de Pequim, com base na circulação de aproximadamente 33000 táxis durante o período de três meses [4].

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 não 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 se estender os elementos que compõem a rosa, por exemplo com a criação de setas em torno da grelha que indiquem as direções de origem e destino e a velocidade média dos trajetos.

Dados GPX

O GPX (*GPS Exchange Format*) é um formato XML muito usado para registar trajetos por parte de aplicações que façam uso de 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. De seguida, é apresentado um excerto de um ficheiro GPX gerado pela aplicação Geo Tracker.

```
<trk>
  <name>Casa - FCUL</name>
  <desc>Viagem normal de manhã desde casa até à FCUL</desc>
  <src>Recorded in Geo Tracker for Android from Ilya
    Bogdanovich</src>
  <link href="https://play.google.com/store/apps/details?
id=com.ilyabogdanovich.geotracker"/>
  <extensions>
    <geotracker:meta>
      <length>45435.01</length>
      <duration>3667981</duration>
      <creationtime>2016-09-30T08:03:02.582Z</creationtime>
      <activity>0</activity>
    </geotracker:meta>
  </extensions>
  <trkseg>
    <trkpt lat="39.086399" lon="-9.259854">
      <ele>117</ele>
      <time>2016-09-30T08:03:06.687Z</time>
      <extensions>
        <geotracker:meta c="41.4" s="4.67" />
      </extensions>
    </trkpt> ...
</trk>
```

O ficheiro GPX possui uma primeira marca de metadados, ausente do excerto anterior, que contém uma descrição geral do conteúdo de todo o ficheiro, como por exemplo, o nome, o programa que o gerou e o instante de criação. De seguida, encontra-se a marca *track*, denotada por *trk*, que representa um trajeto registado. Os atributos utilizados dentro desta marca permitem descrever o trajeto, como são exemplo, a descrição (*desc*) e o nome da *track* (*name*). Dentro de cada *track* existem segmentos, denominados por *trkseg*, que representam troços da *track*. Por último, dentro de cada segmento encontram-se pontos, representados por *trkpt*. Estes últimos elementos possuem atributos como a latitude, longitude, elevação, velocidade instantânea e instante de registo. Através das marcas *trk*, *trkseg* e *trkpt*, estabelece-se uma hierarquia dentro de cada trajeto, permitindo uma consulta simplificada dos seus elementos.

2.1.3 Agregação de dados

O processamento desta informação leva à criação de relatórios de dados agregados que podem ser úteis a decisores que tenham de desempenhar um papel decisivo perante o negócio de uma organização. As decisões são tomadas com base na informação agregada mostrada nos relatórios.

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 [16]. Neste projeto, recorreu-se a técnicas de pré-processamento de trajetos, nomeadamente a 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 permitiram melhorar a qualidade do conjunto de trajetos utilizado no sistema desenvolvido, sendo detalhadas posteriormente nas devidas secções do relatório. 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 no que toca à análise de trajetos, uma vez que facilita a compreensão de grandes conjuntos de dados deste tipo, permitindo descobrir

aspectos 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 maiores problemas associados à visualização analítica de dados agregados, prende-se com a quantidade de informação mostrada ao utilizador [17]. As interfaces devem ser pensadas e trabalhadas no sentido de mostrar ao utilizador dados que o ajudem a executar as suas tarefas sem que a sua atenção seja desviada para pormenores de usabilidade.

O modelo de visualização de Van Wijk representado na Figura 4 é centrado no utilizador, ou seja, toda a atividade de interação e aquisição de conhecimento ocorre em torno do mesmo, podendo ser explicado da seguinte forma:

O utilizador possui conhecimento (K, de *knowledge*) que adquire através da sua perceção (P) do meio envolvente. A sua perceção é conseguida através da exploração interativa (E) da interface gráfica de uma aplicação. Por sua vez, esta exploração ocorre através de técnicas de manipulação (S, de *specifications*), disponibilizadas pela aplicação, sendo auxiliada pelo conhecimento (K) que o utilizador foi adquirindo ao longo do tempo. A interface da aplicação constitui a visualização (V) para o utilizador dos dados (D). A visualização é percecionada pelo utilizador e o ciclo no diagrama volta a ser iterado ao longo da utilização da aplicação.

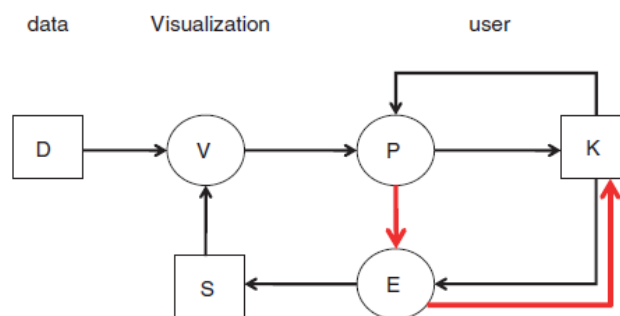


Figura 4 – Diagrama do modelo de visualização de Van Wijk [17]

O trabalho desenvolvido neste projeto está diretamente relacionado com as componentes de visualização (V) e exploração interativa (E), nomeadamente a visualização de agregações de dados que evitam o excesso de informação para o utilizador que seria causado pela análise de trajetos individuais no mapa. A vertente interativa está

presente na possibilidade de criação de rosas dos ventos em quaisquer pontos do mapa, juntamente com a aplicação de filtros sobre as rosas mostradas.

2.1.5 Serviços *web*

O desenvolvimento de uma aplicação *web*, referido nos objetivos do trabalho, pode motivar a construção de serviços *web* para auxiliar o funcionamento da aplicação em causa. 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) [18].

Existem dois protocolos principais de serviços *web*:

- SOAP (*Simple Object Access Protocol*) – baseia-se numa invocação remota de um determinado método, ou seja, uma determinada 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. No entanto, é apontada como principal desvantagem a sobrecarga associada ao processamento das mensagens (serialização e desserialização), o que torna este tipo de serviço mais lento que os restantes.
- REST (*Representational State Transfer*) – é igualmente um protocolo de troca de mensagens baseado em HTTP. No entanto, o REST distingue-se pela liberdade na formatação de mensagens, ao contrário do SOAP. O REST requer apenas que seja especificado o comportamento do serviço em causa. Isto leva a que o formato das mensagens possa ser especificado pelo programador. Os serviços *web* que utilizam o protocolo REST são, em geral, mais rápidos na execução. A principal desvantagem apontada ao REST está diretamente associada à sua vantagem, ou seja, a liberdade de especificação dos formatos

de mensagem pode levar a dificuldades de comunicação entre vários sistemas, uma vez que não é seguido um formato predefinido.

No âmbito do projeto, a opção REST é a mais adequada, uma vez que os serviços são utilizados a partir do navegador *web* e não requerem uma codificação muito elaborada. Os serviços *web* desenvolvidos desempenham um papel importante no tratamento e processamento dos dados dos trajetos analisados pelo sistema, permitindo aliviar a carga de processamento noutros componentes do sistema.

2.2 Aplicações e sistemas existentes

Nesta secção são apresentados exemplos de aplicações de visualização analítica relativamente a trajetos, sendo feito um paralelismo com as funcionalidades e objetivos do sistema desenvolvido neste projeto. Mencionam-se ainda aspetos gerais do primeiro protótipo do sistema de análise de trajetos com rosas dos ventos, juntamente com algumas limitações que esse sistema acarreta.

2.2.1 Aplicações de visualização analítica

Existem análises comparativas de aplicações que permitem a visualização analítica de conjuntos de dados agregados, elaboradas no sentido de se perceber quais as capacidades de cada *software* relativamente aos seus concorrentes no mercado.

Relativamente a soluções *open source*, uma análise comparativa de 2012 [19] enumera as várias funcionalidades das aplicações testadas. Tendo em conta que o foco deste projeto é a representação de trajetos agregados, são apenas indicadas as funcionalidades relacionadas com a representação deste tipo de dados. As técnicas de representação mais comuns passam pela utilização de uma camada visual, sobreposta num mapa, em que são traçados trajetos ou pontos. Soluções no âmbito de mapas animados são igualmente utilizadas, tais como cartogramas e representação de dados sobre um globo terrestre 3D. Quase todas estas abordagens são estáticas, sendo que a maior parte se limita à representação de dados estatísticos. O Birdeye [20], Google Vis [21] e Protovis [22], são alguns exemplos destas aplicações. Nenhuma das aplicações referidas na análise comparativa permite a representação de agregações de trajetos com rosas dos ventos.

Por outro lado, está disponível um comparativo de sistemas pagos [23], onde apesar da maior parte das soluções oferecerem suporte a métodos de análise automática de dados, como por exemplo, algoritmos de agrupamento e classificação, nenhuma está vocacionada para a análise de trajetos. Estas aplicações requerem licenças pagas, o que dificulta o seu acesso devido a limitações monetárias.

Em relação à visualização com rosas dos ventos, a aplicação gratuita WRPLOT [24] permite construir rosas a partir de um ficheiro com dados de ventos contendo a direção, velocidade e data de ocorrência de uma ou mais localizações geográficas. Após o carregamento dos dados, é possível efetuar agregações e aplicar filtros. Existe ainda a hipótese de personalização de aspetos gráficos da rosa gerada, como é o caso das cores da escala de velocidade dos ventos e o número de setores da grelha da rosa. As rosas geradas podem ser exportadas no formato KML (*Keyhole Markup Language*) para que possam ser visualizadas em vários contextos, como por exemplo no Google Maps. Uma limitação desta aplicação é aceitar exclusivamente dados de ventos, impossibilitando a utilização da rosa com outros tipos de dados, tais como trajetos.

2.2.2 Protótipo anterior do sistema de análise de trajetos

No âmbito de um projeto de mestrado em engenharia informática, foi desenvolvida, por Mauro Carreira, 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 [6]. Este trabalho serviu de base ao projeto descrito ao longo deste documento, sendo considerado o primeiro protótipo do sistema. A Figura 5 mostra a interface gráfica da aplicação *web* desse protótipo.

No que diz respeito à biblioteca, são disponibilizadas funções que permitem ao utilizador criar os vários elementos da rosa, como a grelha, as pétalas e as setas de direção de origem. Para além dos elementos gráficos, existe a possibilidade de se filtrarem os trajetos representados com base em intervalos fixos de horas e raio de agregação de trajetos em torno de cada rosa, também com valores predefinidos.

A aplicação *web* permite conferir um contexto geoespacial aos dados representados na rosa gerada pela biblioteca. Recorrendo à API do Google Maps, é gerado um mapa sobre o qual é mostrado um conjunto de trajetos de um 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.

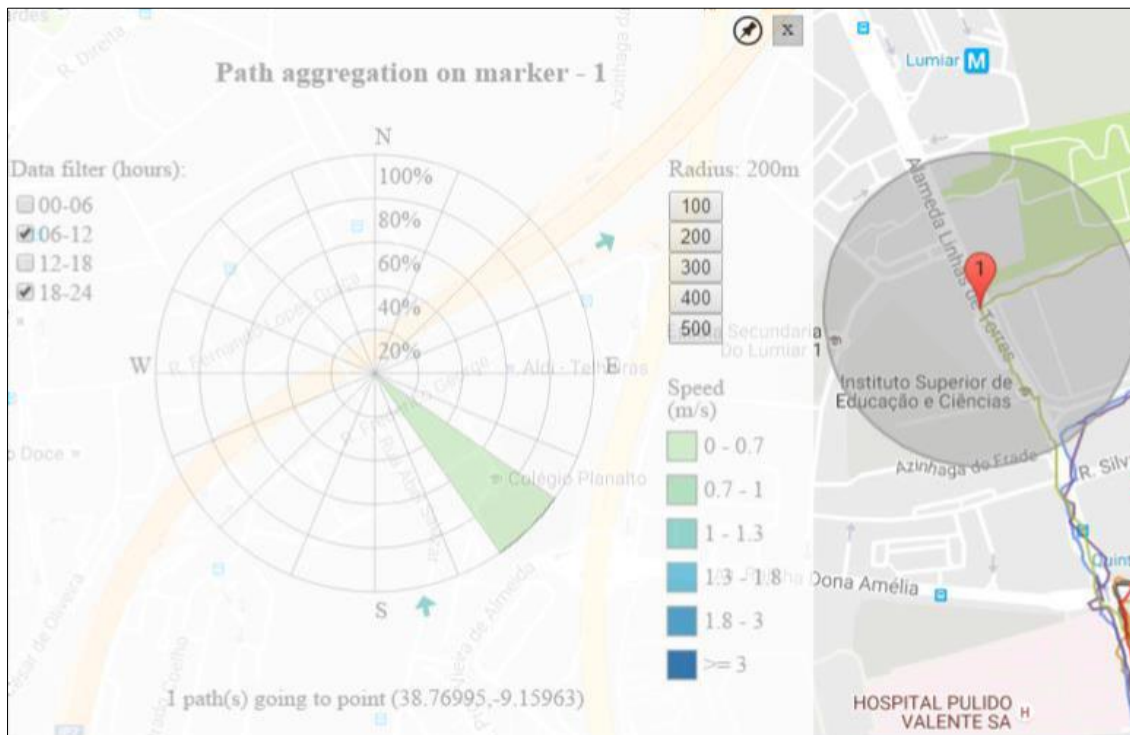


Figura 5 – Aplicação *web* do primeiro protótipo do sistema [6]

A rosa criada pela biblioteca agrega dados de trajetos que estão contidos num círculo com um determinado raio (em metros). Este raio de agregação pode ser alterado para valores fixos de 100, 200, 300, 400 e 500 metros, 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.

Todos os elementos desenhados pela biblioteca requerem dados com um formato concreto. Tomemos como exemplo a função de criação de uma pétala da rosa:

```
createPetal(direction, windroseData, windroseScale, amplitudePercentage,  
            petalStrokeWidth, petalStrokeColor)
```

Neste exemplo existem argumentos que requerem valores obtidos através da extração de informação do ficheiro GPX com os respetivos cálculos, como é o caso da direção da pétala (*direction*), em graus, e os dados correspondentes aos percursos representados pela pétala (*windroseData*) que contêm a direção e a velocidade média dos mesmos. Os dados previamente trabalhados e enviados para a biblioteca não são persistidos, o que leva a que, de cada vez que se pretenda criar uma nova rosa no mapa de trajetos, não sejam reaproveitadas as estruturas de dados já existentes.

O aspeto mencionado no parágrafo anterior estende-se a todos os métodos da biblioteca da rosa que requeiram informação agregada de trajetos. Com esta decisão de implementação surge um problema de desempenho e escalabilidade perante grandes conjuntos de dados: todo o processamento de trajetos é feito pela aplicação *web* para obtenção da estrutura de dados esperada pela biblioteca, sem qualquer auxílio de serviços externos, como é o caso dos serviços *web*. Acresce ainda a limitação de só poder ser utilizado um único ficheiro GPX, armazenado numa diretoria específica, o que confere pouca flexibilidade na seleção de trajetos.

Para dar ênfase à distribuição de responsabilidades entre a biblioteca e a aplicação *web*, construiu-se a Figura 6 que mostra o fluxo de execução da aplicação *web*, desde que é aberto um ficheiro GPX até à chamada das funções da biblioteca da rosa dos ventos. As setas a carregado correspondem ao fluxo de execução principal, enquanto que as setas a tracejado representam execuções de funções auxiliares, cujo objetivo é produzir estruturas de dados compatíveis com a biblioteca.

De notar que existe um tratamento de informação excessivo, denotado pela área a sublinhado na Figura 6, para que possam ser invocados os métodos da rosa. Isto torna a utilização da biblioteca difícil, uma vez que o utilizador deve conhecer com precisão as estruturas de dados que alimentam os métodos da mesma e por se perder a noção de trajeto à medida que o conteúdo extraído do ficheiro GPX é processado e moldado. A perda desse conceito faz com que deixe de ser possível trabalhar a informação para outras finalidades, como por exemplo, a aplicação de outros tipos de filtros nos trajetos, que não a velocidade média ou a hora do dia em que foi efetuado o registo.

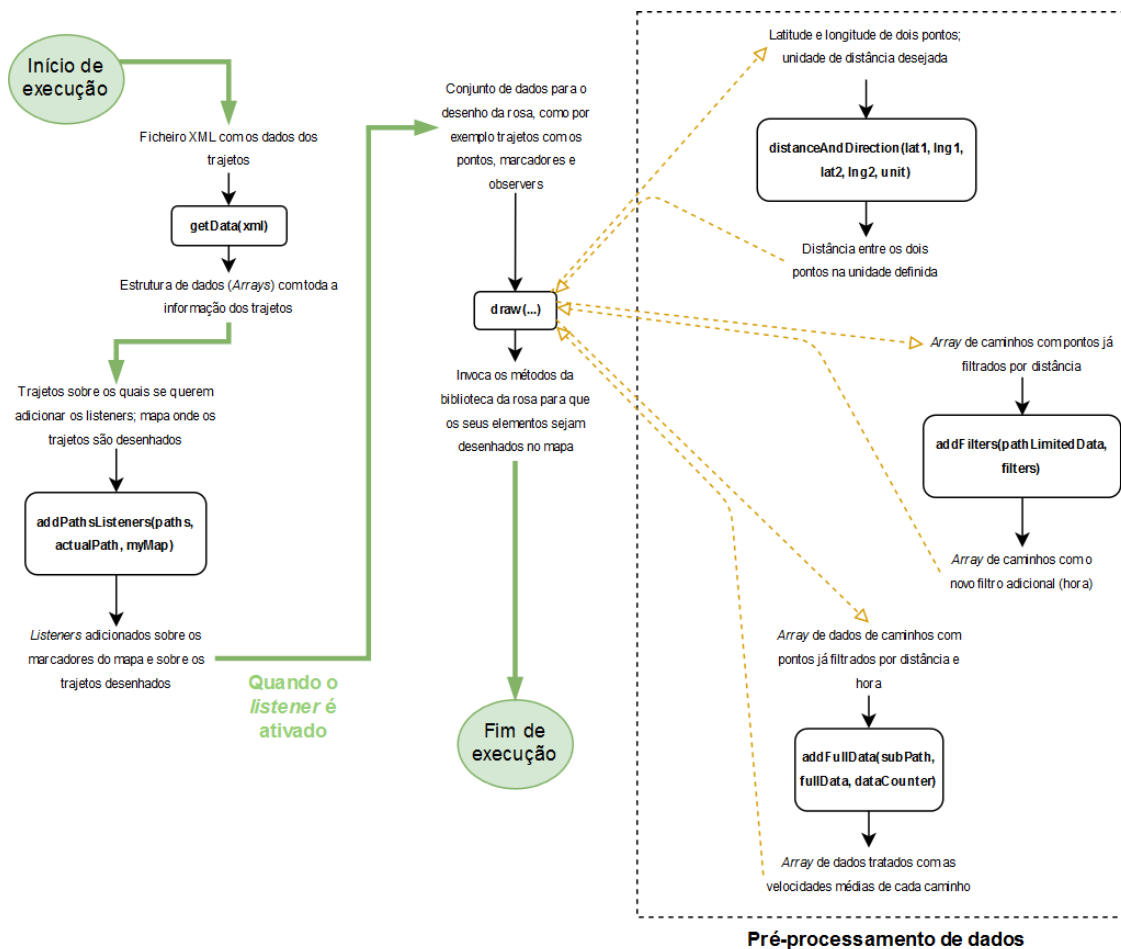


Figura 6 –Funcionamento da aplicação *web* do primeiro protótipo do sistema

Todas as limitações descritas nos parágrafos anteriores foram ultrapassadas no trabalho descrito neste relatório, com a implementação de serviços *web*, uma base de dados geográfica e alterações significativas na biblioteca e na aplicação *web*. No seguimento do relatório existem secções para cada um dos aspetos referidos, sendo também feita uma análise comparativa entre o sistema anterior e a solução desenvolvida neste projeto.

2.3 Sumário

Neste capítulo foram descritos os conceitos teóricos relacionados com o trabalho desenvolvido. Foram também apresentados um conjunto de aplicações de rosas dos ventos e sistemas que se inserem na temática do trabalho, nomeadamente na área de análise de trajetos. Foi também realizada uma análise geral do primeiro protótipo do sistema, o qual serviu de base à elaboração do sistema descrito ao neste relatório.

No capítulo seguinte é documentado o desenvolvimento do atual protótipo do sistema de análise de trajetos com rosas dos ventos, que visa cumprir os objetivos traçados inicialmente.

Capítulo 3

Sistema de análise de trajetos com rosas dos ventos

Neste capítulo estão descritos todos os aspetos relacionados com o desenvolvimento do sistema de análise de trajetos recorrendo a gráficos inspirados nas rosas dos ventos. Detalham-se o ambiente de desenvolvimento, os requisitos do sistema e a arquitetura e desenho deste. Por fim, apresentam-se cenários que permitem perceber a interação dos diferentes componentes do sistema quando são desempenhadas determinadas tarefas.

3.1 Ambiente de trabalho

Nesta secção inicial são apresentadas as ferramentas e linguagens utilizadas para o desenvolvimento do sistema, seguido de uma explicação do processo de desenvolvimento de *software*.

3.1.1 Ferramentas e tecnologias

Todo o trabalho foi realizado num computador portátil com um processador Intel Core i5 @ 2.4 GHz, 4 GB de memória RAM e sistema operativo Windows 10 Home x64.

Tendo em conta os objetivos do projeto, dentro do conjunto de tecnologias existentes para a realização dos mesmos, foram selecionadas aquelas em que já existe uma base de conhecimento minimamente sólida que permita agilizar a sua utilização. Nos parágrafos seguintes são descritas as opções tomadas ao longo do desenvolvimento das quatro componentes principais do sistema: uma base de dados geográfica, serviços *web*, uma biblioteca da rosa dos ventos e uma aplicação *web* que permite a colocação de rosas dos ventos sobre um mapa com trajetos.

Para o desenvolvimento dos serviços *web* REST, foi utilizada a linguagem PHP [25] juntamente com HTML [26]. O HTML permite a construção de uma página cliente simples para fazer o acesso ao serviço *web*. Para o auxílio de escrita do código PHP, HTML e JavaScript recorreu-se ao IDE JetBrains PhpStorm 2016 [27].

Para o tratamento de dados presentes em ficheiros de texto de grande dimensão, foi criado um programa escrito na linguagem Java. Esta decisão de desenvolvimento foi

tomada tendo em conta a facilidade que o Java oferece no tratamento de ficheiros e manipulação de linhas de texto.

Para a interpretação dos dados pré-processados, fez-se uso da ferramenta RStudio [28]. Esta ferramenta possui mecanismos de análise de conjuntos de dados, permitindo a construção de tabelas com filtros, criação de gráficos e sumários de estatística descritiva.

Para a camada de dados utilizou-se o PostGIS [29], uma extensão do PostgreSQL [30]. Com a utilização deste tipo de base de dados relacional, é possível fazer pesquisas baseadas em informação geográfica de uma forma simples. É importante referir que os dados dos percursos inseridos na base de dados foram obtidos a partir de duas fontes: trajetos a pé registados com telemóveis e trajetos de táxis que circularam em Roma. O registo de dados de trajetos é guardado em formato GPX e posteriormente convertido num formato compatível com a base de dados, através dos serviços *web* desenvolvidos para o efeito.

Por último, para a execução e manutenção da base de dados e serviços *web*, recorreu-se ao XAMPP [31]. Este pacote de *software* gratuito inclui um conjunto de ferramentas úteis, como o Apache [32], já pré-configuradas, que permitem a execução dos serviços, mencionados anteriormente, numa máquina local.

O código do projeto foi desenvolvido no Bitbucket, um repositório privado, *online* e gratuito, baseado no sistema de controlo de versões Git [33].

3.1.2 Processo de desenvolvimento

O processo de desenvolvimento deste projeto teve por base um modelo incremental com reuniões semanais, com a presença dos orientadores e do aluno. Em cada reunião foi feito um ponto de situação do estado do protótipo e discutidas novas funcionalidades para implementação. Com a finalidade de garantir o bom funcionamento do sistema, vários testes foram realizados à medida que foram sendo adicionadas novas funcionalidades. No final do processo incremental, obteve-se o protótipo atual do sistema de análise de trajetos com rosas dos ventos.

3.2 Requisitos do sistema

Antes de ser iniciado o processo de codificação do sistema, definiu-se um conjunto de requisitos e funcionalidades que se consideraram essenciais, tendo em conta os objetivos do projeto.

O primeiro requisito do sistema, R1, é: oferecer ao utilizador a capacidade de analisar trajetos, à sua escolha, num mapa, através da colocação e interação com gráficos inspirados nas rosas dos ventos. Estes gráficos devem poder ser criados livremente em qualquer ponto do mapa, podendo existir vários em simultâneo, permitindo ao utilizador analisar trajetos em pontos que considere relevantes.

O segundo requisito do sistema, R2, foi definido da seguinte forma: os elementos gráficos, das rosas dos ventos devem ser capazes de mostrar dados sobre os trajetos que agregam, como são exemplo a quantidade total de trajetos, o raio de ação da rosa, os intervalos de horas de ocorrência, a velocidade de circulação na zona, as direções de origem/destino e a velocidade média.

O terceiro e último requisito do sistema, R3, foi estipulado como: possibilitar a manipulação dos gráficos mostrados, através de filtros que permitam ajustar os parâmetros aplicados a cada rosa dos ventos criada no mapa. Estas modificações devem poder ser feitas de forma individual ou conjunta, de forma a que o utilizador consiga rapidamente analisar comparativamente dois ou mais pontos no mapa, sob as mesmas condições. Por outro lado, deve ser disponibilizada uma ferramenta que permita inspecionar as várias rosas criadas, permitindo facilmente analisar a coerência de filtros e descobrir quais os parâmetros que diferem.

Os requisitos R1, R2 e R3 representam as ideias chave das funcionalidades que o sistema de análise de trajetos com rosas dos ventos deve oferecer, para permitir o trabalho de análise do utilizador.

3.3 Desenvolvimento do sistema

Nesta secção detalham-se os vários componentes que constituem o sistema de análise de trajetos com rosas dos ventos, antecidos pela descrição de uma perspetiva geral da arquitetura do sistema. 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 foram utilizadas ao longo do processo de desenvolvimento do sistema.

3.3.1 Arquitetura do sistema

Para que seja dada uma perspectiva geral da arquitetura do sistema foi preparada a Figura 7 na qual são visíveis os principais componentes que constituem o sistema desenvolvido, bem como as dependências que existem entre si.

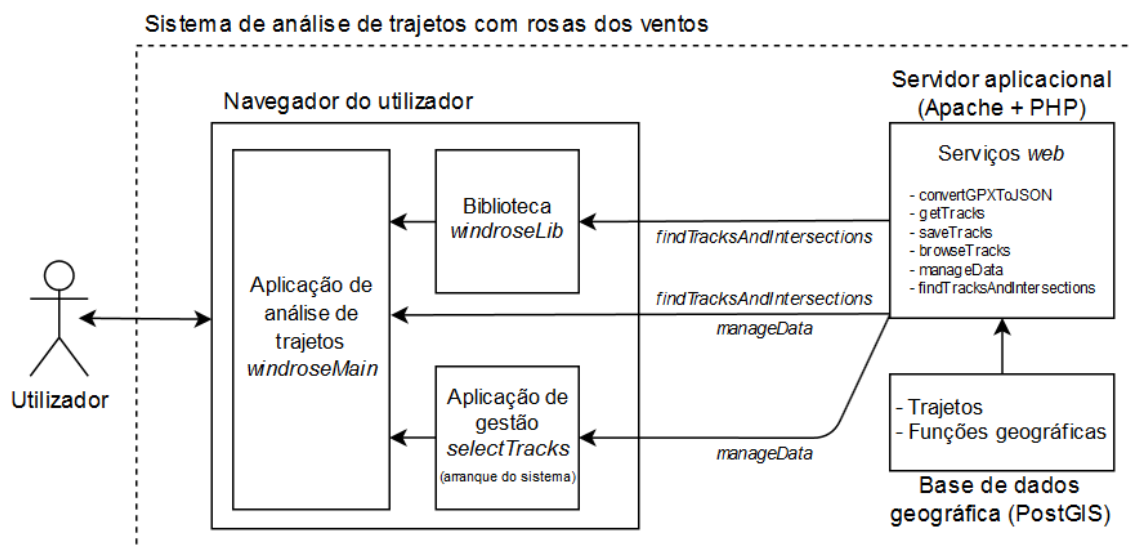


Figura 7 – Arquitetura do sistema de análise de trajetos com rosas dos ventos

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 aplicacional 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.

Esta arquitetura apresenta algumas diferenças relativamente à do primeiro protótipo do sistema [6]. Nesse protótipo, a base de dados é inexistente, não havendo persistência de dados de trajetos. A ausência de serviços *web* afeta o desempenho de processamento de informação caso existam grandes conjuntos de trajetos.

As subsecções seguintes detalham cada uma das componentes de arquitetura mencionadas. Tendo em conta as dependências, mostradas na Figura 7, começa-se por descrever a base de dados geográfica, seguida dos serviços *web*, da biblioteca da rosa dos ventos e, por último, das aplicações de gestão e análise de trajetos. A ordem das componentes da biblioteca e aplicação de análise está alinhada com os objetivos do trabalho, isto é, primeiro detalha-se a biblioteca e só depois a aplicação de análise de trajetos.

3.3.2 Base de dados geográfica

Esta subsecção descreve o modelo de dados elaborado para a criação da base de dados do sistema, bem como o processo de criação das tabelas.

Modelação

Recorrendo ao modelo Entidade-Associação (EA) é possível definir uma relação entre as principais entidades presentes na informação a ser modelada. No modelo apresentado na Figura 8 existem três entidades que se relacionam: *Track*, *Segmento* e *Ponto*.

Uma *Track* é composta por dois atributos, nomeadamente um identificador único que serve de chave primária e uma descrição. Esta entidade comporta um ou mais segmentos que, por sua vez, agregam pontos.

O *Segmento* é uma entidade que comporta um conjunto de Pontos (entidade seguinte), descrita por vários atributos, tendo alguns deles com informação agregada, como são exemplo a latitude máxima e mínima, que detalham informações dos pontos comportados, para além de um identificador único e sequência temporal com que o *Segmento* se insere dentro da *Track*.

A relação “possui”, associa a entidade *Track* com segmentos e define que esta é composta por um ou mais Segmentos, sendo que cada destes pertence a uma única *Track*.

Por último, o Ponto é uma entidade descrita por um identificador único, uma latitude, longitude, elevação, velocidade instantânea, instante de registo e sequência de ocorrência.

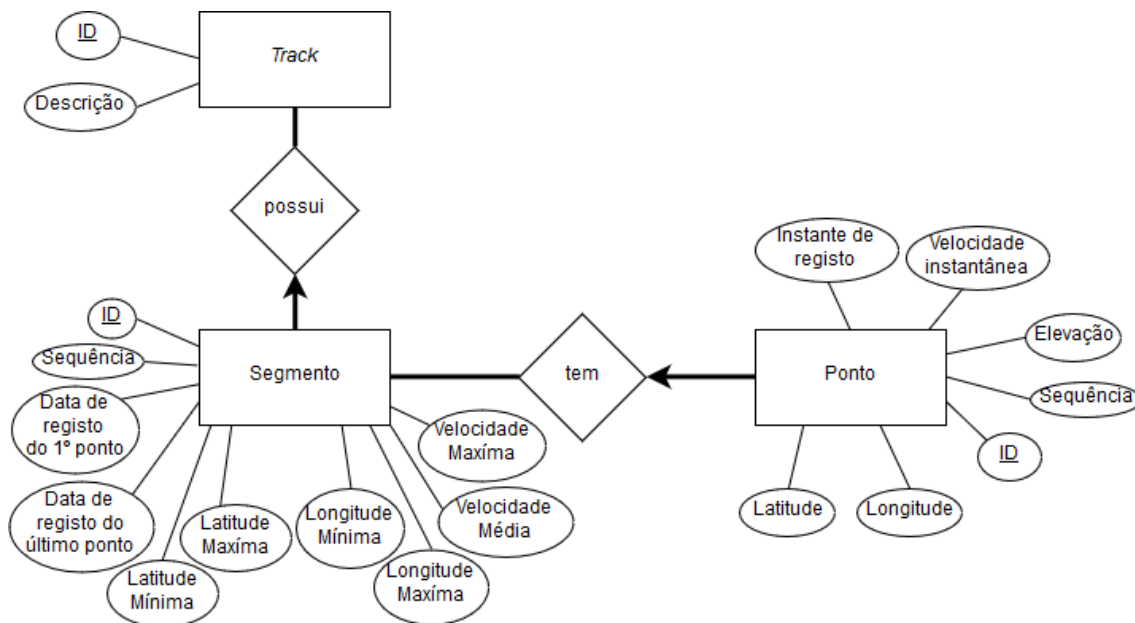


Figura 8 – Modelo de dados em Entidade-Associação (EA)

Um Segmento é obrigatoriamente constituído por um ou mais Pontos, através da relação “tem”. Por outro lado, um Ponto pertence obrigatoriamente a um único segmento.

Em resumo, uma *Track* é constituída por um conjunto de trajetos percorridos por uma determinada entidade como é o caso de veículos, pessoas ou barcos. O Segmento representa cada um dos trajetos que estão associados à *Track*, sendo que este é constituído por pontos que formam o trajeto na totalidade. O Ponto é a entidade responsável por definir registos temporais que, quando agregados, formam um Segmento, pertencente a uma dada *Track*.

É possível adicionar entidades que forneçam um contexto ao tipo de trajetos que se está a registar. Tome-se como possível exemplo uma tabela de táxis, no caso de se pretender registar percursos numa determinada zona de atuação dos veículos. Para tal podem ser incluídos dados das viaturas que efetuam os percursos bem como o número de passageiros.

A criação do modelo de dados, apresentado na Figura 8, representa um passo importante para a persistência de dados do sistema. O conceito de trajeto está sempre presente na base de dados, o que leva a que a informação possa ser reutilizada para vários fins, como por exemplo, a aplicação de filtros sobre os dados.

Criação da base de dados

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 [29]. A decisão de utilização deste sistema de gestão de base de dados permitiu a utilização de funções geográficas predefinidas que melhoraram substancialmente a descoberta dos trajetos que cruzam a área de agregação da rosa comparativamente com o primeiro protótipo do sistema.

A estrutura da base de dados é constituída por três tabelas principais: *tracks*, segmentos e pontos. A informação que cada uma das três tabelas comporta, é semelhante ao que foi definido pelo modelo EA presente na Figura 8.

De seguida, é apresentado o código de criação da tabela de *tracks*, composta por um identificador único e uma descrição:

```
CREATE TABLE Tracks (  
    id          INT PRIMARY KEY,  
    description VARCHAR(100)  
);
```

A criação da tabela de segmentos foi elaborada através do seguinte código:

```
CREATE TABLE Segments (  
    id          SERIAL PRIMARY KEY,  
    sequence    INT NOT NULL,  
    track_id    INT REFERENCES Tracks(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  
);
```

A tabela de segmentos, descrita pelo código acima, possui atributos que guardam dados agregados sobre os pontos comportados, como por exemplo, as latitudes e longitudes máximas e mínimas, permitindo criar uma *bounding box* que delimita a área de ocorrência do trajeto.

Por último, é mostrado o código de criação da tabela de pontos da base de dados:

```
CREATE TABLE Points(  
    id          SERIAL PRIMARY KEY,  
    sequence    INT                                NOT NULL,  
    segment_id  INT REFERENCES Segments(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  
);
```

De todos os atributos apresentados no código acima, destaca-se o *wkb_geometry*. Este atributo possui um valor fundamental para a utilização de filtros geográficos do PostGIS, tal como aquele que é utilizado no serviço *web getTracks*, descrito mais adiante na subsecção 3.3.3, para a filtragem de segmentos com pontos contidos dentro de um determinado círculo.

O preenchimento destas três tabelas é da responsabilidade do serviço *web saveTracks*, descrito na subsecção 3.3.3. Existe uma pequena limitação referente à tabela que comporta todas as *tracks*: uma vez que os identificadores desta tabela são inteiros, o ficheiro GPX original deve conter um identificador inteiro em cada *track*. Caso contrário, será retornado um erro por incoerência de tipo de campos da base de dados.

3.3.3 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. De seguida são descritos todos os serviços *web* desenvolvidos em PHP e utilizados no sistema.

Conversão de GPX para JSON

Este serviço *web*, designado por *convertGPXToJson*, constitui o primeiro passo para o tratamento da informação de trajetos provenientes de ficheiros GPX. O formato GPX é muito usado para registar trajetos, tornando o sistema compatível com um vasto leque de dispositivos GPS. Por outro lado, o formato JSON (*JavaScript Object Notation*) [34] permite um processamento dos dados mais eficiente com a linguagem JavaScript.

Foi criada uma página HTML com um formulário que permite ao utilizador fazer *upload* de um ficheiro GPX para o serviço. Uma vez que se faz uso do método HTTP POST, não é possível efetuar pedidos, manualmente, através da barra de endereço do navegador *web*.

Relativamente ao formato GPX, foram feitos testes com duas aplicações populares: o Pocket Earth [35] e o Geo Tracker [15]. A decisão de suporte do formato de ambas as aplicações é importante, uma vez que cada uma está disponível apenas para apenas um de dois sistemas operativos: iOS ou Android. Permite-se assim a utilização do serviço por parte de utilizadores de *smartphones* com estes sistemas operativos.

O algoritmo de conversão é constituído pelos seguintes passos:

1. O código PHP que corre no servidor recebe o ficheiro GPX enviado pelo utilizador. De seguida, esse mesmo ficheiro é aberto com uma função apropriada para a leitura de um ficheiro XML, uma vez que o GPX é um tipo de XML.
2. A navegação do conteúdo do ficheiro é feita a partir de nós, cujos nomes são iguais às *tags* utilizadas no GPX. Inicialmente é criada uma estrutura de dados em que existe um *array* com todos os segmentos dos percursos registados. Este *array* possui um objeto que contém o nome descritivo do segmento e um *array* com objetos do tipo *Ponto*. Por sua vez, um ponto é um objeto que contém 5 atributos: latitude, longitude, elevação, velocidade instantânea e instante de registo.
3. O último passo do algoritmo consiste na conversão da estrutura de dados descrita acima, no formato JSON.

Eis um excerto de um ficheiro JSON gerado pelo serviço *web* responsável pela conversão:

```
[ { "name": "Casa - FCUL",
  "segmentPoints": [
    { "latitude": "39.086399",
      "longitude": "-9.259854",
      "elevation": "117",
      "speed": "4.67",
      "time": "2016-09-30T08:03:06.687Z"
    }, ... ]
} ]
```

A estrutura do ficheiro segue o padrão descrito no exemplo acima. Existe um primeiro nível de dados que contém dois atributos referentes aos segmentos: o nome do segmento e os pontos que estão contidos no mesmo. No nível de dados abaixo, encontram-se descritos os pontos que fazem parte do segmento, sendo que existem 5 atributos distintos que os caracterizam: latitude, longitude, elevação, velocidade instantânea e o instante em que foram registados.

A conversão para JSON é conseguida através da utilização da função *json_encode()* do PHP. O resultado é guardado num ficheiro JSON e armazenado numa pasta do servidor. Os ficheiros GPX originais enviados para o serviço são também armazenados. No final de todo o processo é devolvido um endereço que permite, ao utilizador, aceder ao ficheiro JSON gerado. Existe ainda a possibilidade de se navegar por todos os ficheiros JSON contidos no servidor, através de um serviço *web* auxiliar, denominado *browseTracks*.

No primeiro protótipo do sistema [6], a tarefa de tratamento de trajetos é feita integralmente no *browser* que executa a aplicação *web*. Embora esta solução funcione para quantidades limitadas de trajetos, em cenários com um conjunto de dados de dimensões consideráveis, como por exemplo, a movimentação de táxis numa cidade, o navegador acaba por ter dificuldades em processar tamanha quantidade de informação. Com a implementação do serviço *convertGPXtoJSON*, a tarefa de conversão de dados, provenientes de ficheiros GPX, é efetuada no servidor. A escalabilidade do sistema é fortalecida com esta implementação, permitindo que sejam analisados conjuntos de trajetos com maiores dimensões.

Carregamento de dados de trajetos para a base de dados

Este serviço *web*, denominado *saveTracks*, permite enviar os trajetos guardados em JSON, obtidos a partir do serviço *web convertGPXtoJSON* para a base de dados geográfica.

Foi criada uma página HTML com um formulário que permite o envio de um ficheiro JSON para o serviço em questão. É usado o método HTTP *POST*, pelo que não é possível formular pedidos através da barra de endereço do navegador *web*.

A operação de leitura do ficheiro JSON é feita através da função *json_decode()* do PHP, que gera uma estrutura de dados de *arrays* semelhante à que foi utilizada na

conversão do ficheiro GPX. O preenchimento da base de dados é feito com comandos INSERT do SQL.

De uma forma resumida, para todas as *tracks* e segmentos presentes no ficheiro JSON, são inseridas novas entradas nas respetivas tabelas da base de dados. De seguida, por cada ponto presente num segmento, é adicionada uma entrada na tabela de pontos da base de dados com uma referência para o segmento a que este pertence. No final da execução, as tabelas de *tracks*, segmentos e pontos estão devidamente preenchidas com o conteúdo do ficheiro JSON fornecido.

Pesquisa de trajetos na base de dados

Uma das principais vantagens de se utilizar uma base de dados geográfica, prende-se com a quantidade 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.

Este serviço *web*, denominado *getTracks*, tira partido de funções disponibilizadas pelo PostGIS. Foi criada uma página HTML com um formulário que pode ser preenchido com os parâmetros da interrogação a ser feita à base de dados, através do serviço em questão. Uma vez que se recorre ao método HTTP *GET*, é possível especificar os parâmetros através do endereço do navegador *web*, sem que seja necessário recorrer ao formulário da página HTML.

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

```
http://localhost/webServices/getTracks/getSegments.php?  
latitude=38.758939&longitude=-9.156492&radius=10&  
submit=Get+Segments%21
```

No exemplo acima, estão definidos três parâmetros distintos: a latitude e longitude do ponto central do círculo e o raio do mesmo, sendo que os valores de cada campo são 38.758939, -9.156492 e 10 (metros), respetivamente. Com esses dados, o serviço *web* descobre todos os pontos contidos dentro da área geográfica delimitada pelo círculo. De seguida, são encontrados todos os segmentos aos quais esses pontos pertencem. Por fim, é devolvida a informação de todos os segmentos que intersejam o círculo com o raio definido pelo utilizador.

Para a verificação dos segmentos que interseccionam o círculo são utilizadas duas funções geográficas, disponibilizadas pelo PostGIS: *ST_MakePoint* e *ST_Distance_Sphere*. A função *ST_MakePoint* permite criar o ponto central do círculo na base de dados. De seguida, recorre-se à função *ST_Distance_Sphere* para calcular a distância de um dado ponto, até ao centro do círculo, de raio igual ao que foi definido pelo utilizador. No final, são devolvidos detalhes de todos os segmentos que possuam pelo menos um ponto com uma distância até ao centro do círculo igual ou inferior ao valor do raio.

Este serviço constitui uma forma de filtrar trajetos registados, que pode ser útil em situações de elevada densidade dos mesmos, ou até mesmo em cenários que se pretenda avaliar o fluxo pedonal, ou veicular, presente numa determinada área. Este serviço *web* constitui uma versão preliminar da filtragem de segmentos, tendo dado origem ao serviço *findTracksAndIntersections*, utilizado pela biblioteca *windroseLib* para calcular todas as interseções de trajetos com os raios de agregação das rosas dos ventos, sendo descrito mais à frente no documento.

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

Para que fosse possível enviar o conteúdo de trajetos entre a aplicação de gestão e a aplicação de análise de trajetos, criou-se um serviço intermediário designado por *manageData*, cujo objetivo passa por guardar temporariamente trajetos provenientes da aplicação *selectTracks*, de forma a que estes possam ser lidos pela aplicação *windroseMain* (ver arquitetura na Figura 7).

O funcionamento deste serviço assenta na utilização de sessões do PHP em que é aberta uma sessão por cada utilizador que procede ao carregamento de trajetos através da aplicação de gestão de trajetos. Cada sessão possui uma variável que guarda o conteúdo dos trajetos provenientes do serviço de seleção dos mesmos. Quando a aplicação de análise arranca, executa um pedido de acesso à informação dos trajetos guardada na variável de sessão do utilizador. Os dados dos trajetos são guardados no formato JSON, para que a sua manipulação seja facilitada.

Em ambas as aplicações, *selectTracks* e *windroseMain*, são utilizados pedidos AJAX (*Asynchronous JavaScript and XML*) [36] para o envio e consulta dos trajetos. De seguida, são mostrados dois exemplos de pedidos AJAX feitos pela aplicação de gestão (coluna da esquerda) e análise (coluna da direita), para a colocação e consulta dos dados dos trajetos na variável de sessão do utilizador.

Tabela 2 – Exemplo de pedidos AJAX para o serviço *manageData*

| Pedido de envio dos trajetos | Pedido de acesso aos trajetos |
|--|---|
| <pre>\$.ajax({ url: webServiceURL, type: "POST", data: { value: gpxString }, async: false }).done(function(msg) { console.log("PHP output: " + msg); });</pre> | <pre>\$.ajax({ url: serviceURL, type: "GET", data: {check: "Do it!"}, dataType: 'JSON', async: false, success: function (response) { console.log("XML response successfully received!"); serviceResponse = response; }, error: function (response) { console.log("PHP error: "+ response); } });</pre> |

Consulta de trajetos na base de dados

Para que se tirasse partido das funcionalidades da base de dados geográfica, era conveniente criar um serviço que permitisse fazer consultas dos dados dos trajetos guardados e utilizar as funções geográficas. O serviço *findTracksAndIntersections* foi construído para dar resposta a estas duas necessidades.

Existem quatro tipos de pedido que o serviço *findTracksAndIntersections* suporta. O tipo de pedido deve ser definido com cabeçalho *request* quando se efetua o acesso ao serviço. Os tipos de pedidos são listados de seguida:

- *pointsIDs* – Efetua a listagem dos identificadores de todos os pontos pertencentes a um dado segmento. A mensagem que invoca este tipo de pedido deve ser acompanhada de um campo “*data*”, que deve comportar o identificador do segmento sobre o qual se pretende obter os identificadores dos pontos pertencentes. A resposta é dada no formato JSON com um *array* que segue a estrutura [*IDponto1*, *IDponto2*, ...];

- *segmentsIDs* – O funcionamento é semelhante ao do pedido “*pointsIDs*”, com a diferença que são listados identificadores de segmentos pertencentes a uma dada *Track*, o que leva a que seja necessário enviar o identificador da mesma na mensagem de pedido (igualmente com o campo “*data*”). A resposta é dada no formato JSON com um *array* que segue a estrutura [*IDsegmento1*, *IDsegmento2*, ...];
- *pointsSegmentsIDs* – Pedido que devolve os identificadores de segmentos e respetivos pontos, pertencentes a um dado conjunto de *Tracks*. É necessário enviar na mensagem de pedido as *Tracks* sobre as quais se pretende consultar os segmentos e pontos pertencentes. A resposta é dada no formato JSON com um *array* que segue a estrutura [[*IDsegmento1*, [*IDponto1*, *IDponto2*, ...]], ...];
- *findIntersections* – Efetua um conjunto de chamadas a funções geográficas da base de dados com o intuito de descobrir os pontos em que ocorrem intersecções de trajetos com um círculo definido na mensagem de pedido. A mensagem de pedido ao serviço deve possuir quatro campos:
 - *tracks* – identificadores das *Tracks* que se pretendem testar na ocorrência de intersecções com o círculo;
 - *centerLat* – latitude do centro do círculo;
 - *centerLon* – longitude do centro do círculo;
 - *radius* – raio do círculo, em metros.

A resposta é dada em formato JSON com um *array* de intersecções dos trajetos com o círculo, com a estrutura [[*IDtrack*, *IDsegmento*, *PrimeiroPontoDeInterseção*, *ÚltimoPontoDeInterseção*], ...].

A chamada das funções geográficas para a determinação das intersecções pode ser resumida da seguinte forma:

1. Para cada um dos segmentos de cada uma das *Tracks*, enviadas no campo *tracks*, é criado um objeto na base de dados, através do comando *ST_MakePoint* com os pontos do segmento;
2. Cria-se um círculo 2D na base de dados, através do comando *ST_Buffer*. O círculo é centrado no par de coordenadas fornecido no pedido e toma o raio definido pelo respetivo campo;

3. Para cada segmento criado sob a forma de trajeto na base de dados, é testada a interseção com o círculo, através do comando *ST_Intersection*;
4. Caso a interseção não seja vazia:
 - 4.1. Uma vez que a interseção retornada contém todos os pontos que intersectam o círculo em formato objeto, o resultado é convertido para texto para facilitar a sua manipulação, através da função *ST_AsText*;
 - 4.2. Extraí-se o primeiro e último ponto da interseção, através do comando *ST_StartPoint* e *ST_EndPoint*;
 - 4.3. O resultado é colocado na mensagem de resposta, com o formato mencionado anteriormente.
5. Caso a interseção a interseção seja vazia, é descartada e analisada a interseção seguinte até todas serem processadas.

O pedido *findIntersections* é importante na construção dos elementos da rosa dos ventos, visto que fornece a informação crucial para a orientação, tamanho das pétalas e cor das várias secções.

3.3.4 Biblioteca da rosa dos ventos

Nesta subsecção é descrita a biblioteca de funções do sistema de análise de trajetos com rosas dos ventos, designada por *windroseLib*, responsável por criar os elementos gráficos das rosas dos ventos.

O principal propósito da biblioteca da rosa dos ventos é permitir, de uma forma simples, a construção de rosas dos ventos com dados agregados de trajetos. Para alcançar este objetivo foram definidas responsabilidades que a biblioteca assume, nomeadamente:

- Permitir a criação de rosas dos ventos a partir de dados de trajetos;
- Descobrir as interseções dos trajetos com a área de agregação da rosa;
- Desenhar todos os elementos gráficos associados à rosa dos ventos;
- Permitir a aplicação de filtros sobre os trajetos agregados;
- Definir os eventos de interação com o utilizador suportados pela biblioteca para a manipulação das rosas;
- Disponibilizar as rosas dos ventos num formato fácil de manipular;
- Oferecer flexibilidade e independência do sistema.

A criação de rosas dos ventos com a biblioteca foi concretizada em JavaScript através de um construtor que aceita parâmetros relacionados com a rosa que se pretende criar, tais como propriedades gráficas (número de círculos da grelha da rosa e escala de cores, por exemplo), filtros sobre os trajetos (como são exemplos o raio de agregação, a velocidade média máxima e a distância mínima à origem), parâmetros do posicionamento da rosa (a latitude e longitude do ponto central da rosa dos ventos) e por último os dados de todos os trajetos que estão desenhados no mapa. Após a chamada do construtor, a biblioteca executa automaticamente todo o processo de criação de uma rosa, sem que seja necessária qualquer intervenção do programador.

Uma vez fornecidos os dados dos trajetos, como parâmetro no construtor, a biblioteca determina os pontos de interseção com o raio de agregação da rosa. Este procedimento é feito com o auxílio do serviço *web findTracksAndIntersections*, descrito anteriormente. Os dados das interseções devolvidos por esse serviço são então utilizados no processo de construção dos elementos gráficos da rosa.

O processo de criação dos vários elementos gráficos da rosa é conseguido através de um conjunto de métodos de desenho SVG (*Scalable Vector Graphics*) [37], que recebem argumentos, como por exemplo, o valor hexadecimal da cor de uma linha.

A manipulação dos dados de trajetos agregados na rosa foi conseguida através da implementação de funções de filtragem de trajetos. Foi implementado um método na biblioteca que efetua uma cópia do conjunto original das interseções dos trajetos com a periferia da área de agregação da rosa. A partir desta cópia gera-se um subconjunto com as interseções que respeitem os valores dos filtros. Um exemplo possível passa pela aplicação do filtro de horas para trajetos compreendidos entre as 12h e as 24h, sendo gerada uma cópia das interseções iniciais e excluídas todas aquelas que estejam fora dos intervalos de horas definidos.

A biblioteca é responsável por definir um conjunto de eventos que devem ser utilizados para a manipulação das rosas, suportando funcionalidades como a seleção/deseleção das rosas e a aplicação de filtros. Os eventos devem ser acompanhados de campos obrigatórios que informem os detalhes da ação a que se destinam. Estes eventos são descritos em detalhe na subsecção 3.3.5, referente à aplicação de análise de trajetos, uma vez que são despoletados através da interface gráfica da mesma.

É conveniente que toda a informação referente à rosa seja disponibilizada num formato que facilite a sua manipulação. Esta responsabilidade da biblioteca é cumprida através da construção de um objeto *WindRose* que contém todos os elementos associados a uma rosa, tais como trajetos, elementos gráficos em objetos SVG e métodos que permitem aceder à informação guardada na rosa, como é o caso das interseções dos trajetos com a área de agregação.

Por último, pretendeu-se que a biblioteca da rosa dos ventos seja uma ferramenta flexível, capaz de integrar qualquer outro sistema ou aplicação que pretenda utilizar gráficos inspirados na rosa dos ventos para a visualização de trajetos agregados. Para que o processo de aprendizagem da biblioteca seja agilizado, utilizou-se o JSDoc 3 [38], uma ferramenta que permite gerar documentação JavaScript de forma automática, para documentar todos os métodos implementados na biblioteca. A documentação da biblioteca pode ser consultada no Apêndice B, onde estão detalhados todos os métodos que fazem cumprir as responsabilidades da biblioteca, descritas anteriormente.

É importante referir que a biblioteca foi desenvolvida com o intuito de ser aumentada futuramente, ou seja, novos métodos poderão ser implementados, disponibilizando novas funcionalidades, sem que seja comprometido o atual funcionamento. Podemos tomar como exemplo a inclusão de um novo método que permita identificar em que pétala da rosa está contido um determinado trajeto desenhado no mapa, o que faria que com que a lista de funcionalidades fosse estendida.

Na subsecção 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 subsecção 3.3.3 .

3.3.5 Aplicação *web* de análise de trajetos

A aplicação de análise de trajetos, denominada *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. Esta subsecção está focada no funcionamento da aplicação *windroseMain*, sendo dada uma explicação geral das informações que são dispostas na rosa dos ventos, dos aspetos relacionados com o painel de filtros e inspetor e do sistema de eventos para a

manipulação de rosas. Existe ainda uma subsecção dedicada à aplicação de gestão *selectTracks*, que serve de utilitário de arranque à aplicação *windroseMain*.

Seleção de trajetos

Com o intuito de se conferir flexibilidade ao sistema desenvolvido, criou-se uma aplicação para possibilitar a seleção preliminar de trajetos contidos num ficheiro GPX, para análise. Esta aplicação de gestão de trajetos, denominada por *selectTracks*, serve de auxiliar de arranque à aplicação *windroseMain*, fornecendo-lhe os trajetos seleccionados pelo utilizador. A seleção é feita através de uma página *web* contruída para o efeito.

A Figura 9 mostra um exemplo de funcionamento da aplicação *selectTracks*, com o carregamento do ficheiro *individualTaxis.gpx*. Embora apenas estejam visíveis 3 *tracks* seleccionadas, o ficheiro GPX era composto na totalidade por 8, sendo que as restantes não estavam marcadas na lista do lado esquerdo da figura. Esta é a razão pela qual no mapa, apresentado no lado direito da figura, estão desenhados apenas três retângulos.

À medida que vão sendo marcadas/desmarcadas *tracks*, o mapa ajusta-se de acordo com as zonas de circulação das *tracks* seleccionadas. As *bounding boxes* são desenhadas de acordo com os pontos extremos de cada uma das *tracks*, nomeadamente o par de coordenadas máximas a sudoeste (SW) e nordeste (NE), sendo que o ajuste de posição e *zoom* do mapa é feito de acordo com o posicionamento destas caixas.

Cada entrada da lista de *tracks* é acompanhada por uma lista de detalhes. Esta informação é composta pelo nome da *track*, descrição e data de início e fim do registo. No final da lista, existem dois botões que permitem rapidamente marcar ou desmarcar todas as *tracks*, bem como enviar para a aplicação *web* do sistema o conteúdo dos itens seleccionados.

Simplificando, quando o utilizador carrega no botão de envio de trajetos para a aplicação de análise, é feita uma filtragem de acordo com a seleção de itens efetuada, sendo criada uma cópia do conteúdo do ficheiro para a memória de armazenamento local do *browser* e removidas todas as *tracks* que não apresentem identificadores iguais aos escolhidos na seleção.

Após a criação do conjunto de dados final, é enviada uma cópia no formato JSON para o serviço *manageData*. Tal como detalhado anteriormente, este serviço abre uma sessão de armazenamento por cada utilizador (através da utilização de sessões do PHP)

onde são colocados os trajetos recebidos. O conteúdo é posteriormente consultado pela aplicação de análise *windroseMain*, durante a sua fase de arranque.

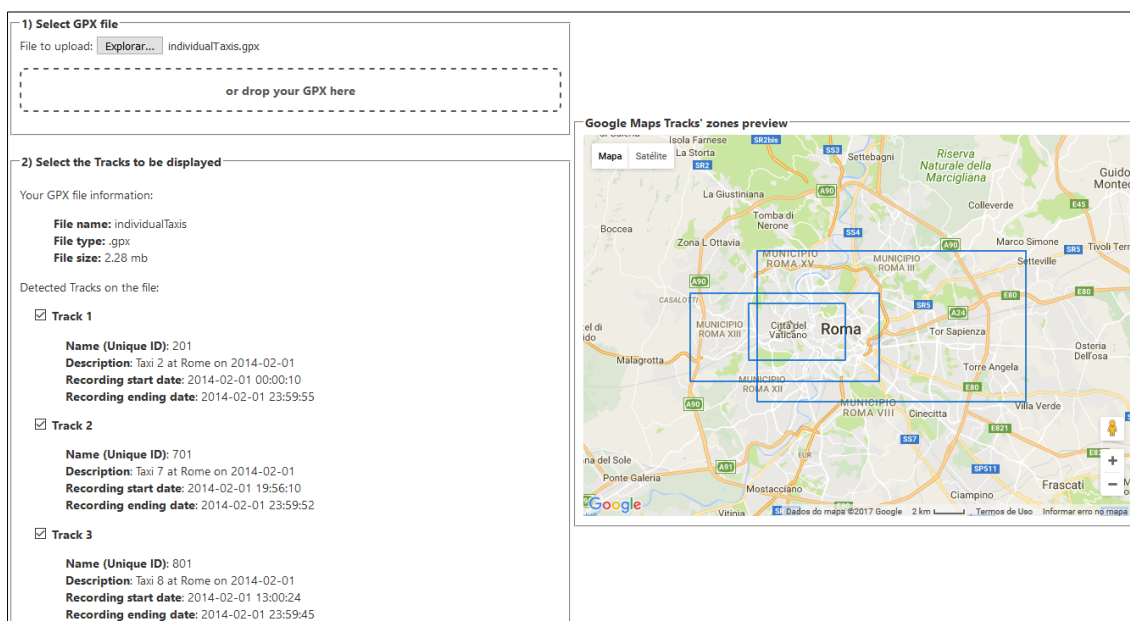


Figura 9 – Exemplo de funcionamento da aplicação de gestão *selectTracks*

A aplicação de gestão *selectTracks* constitui uma ferramenta auxiliar de arranque à aplicação de análise *windroseMain*, tornando-se o ponto de partida para o processo de análise de trajetos do sistema desenvolvido.

Agregação de trajetos com rosas dos ventos

As rosas dos ventos constituem a base para a representação de dados agregados de trajetos. A Figura 10 mostra um cenário em que foi criado um marcador no mapa, previamente carregado com trajetos de táxis em Roma. Cada trajeto é assinalado por uma cor diferente, atribuída aleatoriamente na fase de carregamento dos dados.

A criação de cada marcador é feita com um clique no botão direito do rato em qualquer zona do mapa. Cada marcador possui um número, que serve de identificador único, e em seu redor, está delimitada a área de agregação de trajetos a cinzento. Esta área delimita o espaço onde são analisados os trajetos, sendo que aqueles que não interseam a área a cinzento, são ignorados.

No lado direito do marcador 1, está desenhada uma caixa que contém a rosa dos ventos correspondente. O número apresentado no canto superior esquerdo da caixa da rosa dos ventos representa o número do marcador correspondente.

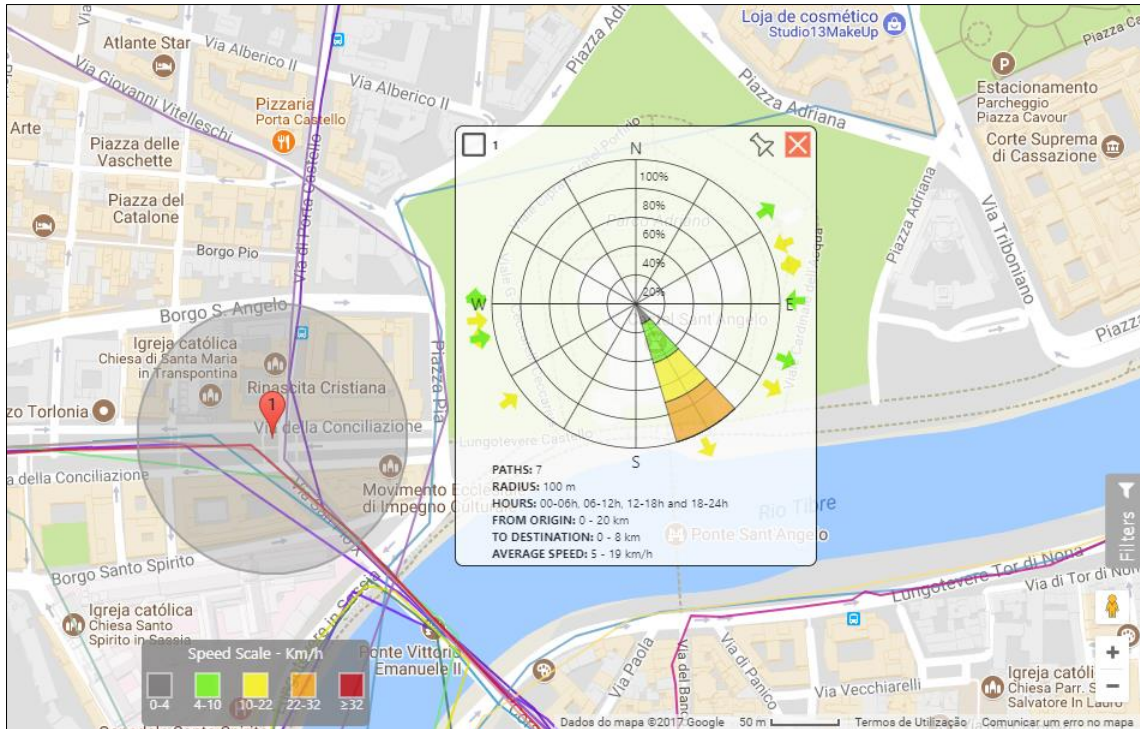


Figura 10 – Exemplo de funcionamento da aplicação de análise *windroseMain*, com uma rosa dos ventos

No cabeçalho da caixa existem três botões distintos, da esquerda para a direita: o botão de seleção da rosa, o botão de *pin* e o botão de fecho. O botão de seleção serve para indicar se a rosa pode ser modificada ou não, uma vez que só podem ser alteradas rosas que estejam selecionadas. O botão *pin* permite fixar a posição da rosa no mapa, o que permite a seleção do texto no seu interior, mas impede que a caixa da rosa possa ser arrastada com o rato. O botão de fecho serve para apagar a caixa da rosa, juntamente com o marcador e área de agregação correspondentes.

A rosa dos ventos possui uma grelha com uma escala de ocorrência relativa, que se ajusta consoante o máximo de trajetos que uma pétala pode conter. No caso da rosa da Figura 10, a escala atinge o valor de 100%, uma vez que existe uma única pétala que contém todos os trajetos que interseitam o raio de agregação (ou seja, todos os trajetos vêm de sudeste). Cada um dos valores da escala, está associado a um dos círculos da grelha da rosa dos ventos, sendo que a escala cresce do círculo próximo do centro até ao círculo exterior.

A pétala que está desenhada na rosa, possui três características distintas:

- A **direção**, que aponta para sudeste (SE), representa a direção de entrada com que os trajetos intersestaram, pela primeira vez, o raio de agregação do marcador da rosa.
- O **comprimento** da pétala atinge o tamanho máximo, uma vez que vai desde o centro da mesma, até ao círculo exterior da grelha. Esta propriedade traduz a quantidade total de trajetos que a pétala agrega relativamente ao total de trajetos apresentado no campo *PATHS* do rodapé da rosa. Visto que a pétala atinge o círculo dos 100% da escala, conclui-se que a pétala possui a totalidade dos trajetos que intersestam a área de agregação da mesma.
- As **secções coloridas** da pétala definem as diferentes velocidades com que os trajetos intersestam o raio de agregação. No caso da pétala apresentada, existem 4 velocidades de interseção distintas. A correspondência entre cores e velocidades é feita na caixa, apresentada no rodapé da janela da aplicação *web*, com o título *Speed Scale*. O tamanho de cada uma das secções está relacionado com a quantidade de trajetos que cada uma delas agrega. No exemplo, as secções verde, amarela e laranja, comportam dois trajetos cada, sendo que a cinzenta possui apenas um. A soma dos trajetos presentes em cada uma das secções, perfaz o total de trajetos agregados pela pétala. Embora não seja visível na imagem, caso o utilizador deixe o rato sobre uma secção de pétala, é mostrada uma *tooltip* com o total de trajetos correspondente à secção apontada pelo rato.

Na fronteira exterior da zona das pétalas da rosa são colocadas setas de origem e destino de cada trajeto. A orientação da seta de origem é definida pela direção que aponta do início de um trajeto para o marcador da rosa no mapa. A orientação da seta de destino é dada pela direção do marcador para o último ponto do trajeto. A cor do par de setas representa a velocidade média de todo o trajeto correspondente, enquanto que a cor nas secções das pétalas corresponde à velocidade aquando da interseção com a área de agregação da rosa.

Por último, o rodapé da rosa dos ventos contém informações sobre os parâmetros da rosa e os trajetos que esta agrega. Os primeiros três campos (*PATHS*, *RADIUS* e *HOURS*), apresentam a quantidade de trajetos que intersestam a área de agregação da rosa e os intervalos de hora permitidos para a visualização de trajetos, respetivamente. Os últimos três campos (*FROM ORIGIN*, *TO DESTINATION*, *AVERAGE SPEED*),

representam mínimos e máximos relativos à distância, em Km, à 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.

Painel de Filtros e Inspetor

O painel de filtros da aplicação apresenta-se na margem direita do ecrã da aplicação *web*, como mostrado na Figura 11, e permite a configuração dos parâmetros das rosas dos ventos existentes no mapa, bem como dos parâmetros de criação de novas rosas.

No exemplo da Figura 11 foram criadas três rosas em posições distintas do mapa. As rosas 2 e 3 encontram-se selecionadas, o que é evidenciado pela cor de fundo cinzenta das caixas e pela *check mark* do botão de seleção no canto superior esquerdo das mesmas.

Quando o utilizador procede à criação de um novo marcador no mapa, a rosa que é criada ao lado do marcador toma os valores que estão definidos em cada um dos campos do painel de filtros. A criação de novas rosas com parâmetros inválidos origina uma mensagem de erro. As várias caixas de rosas dos ventos que vão sendo criadas não podem ser minimizadas, só podendo ser fechadas. Por outro lado, a posição do marcador é fixa, o que implica que, caso um utilizador pretenda mudar a posição de uma da rosa, terá de apagar a mesma e criar uma nova na posição que pretende.

A aplicação dos filtros está sujeita à utilização do botão *Apply*, o que motivou a notificação de alterações não guardadas. Sempre que estejam rosas selecionadas, e existam alterações nos filtros relativamente aos que estão aplicados às rosas, é mostrado um símbolo * em frente ao campo, como sinalização de que o valor foi alterado mas não foi aplicado. Caso voltem a ser definidos os parâmetros aplicados nas rosas ou utilizado o botão *Apply*, a notificação desaparece. Para conveniência do utilizador, foram criados dois botões (*Select All* e *Unselect All*) que permitem fazer rapidamente uma marcação/desmarcação de todas as rosas presentes no mapa. A modificação de parâmetros das rosas só ocorre no caso destas estarem selecionadas.

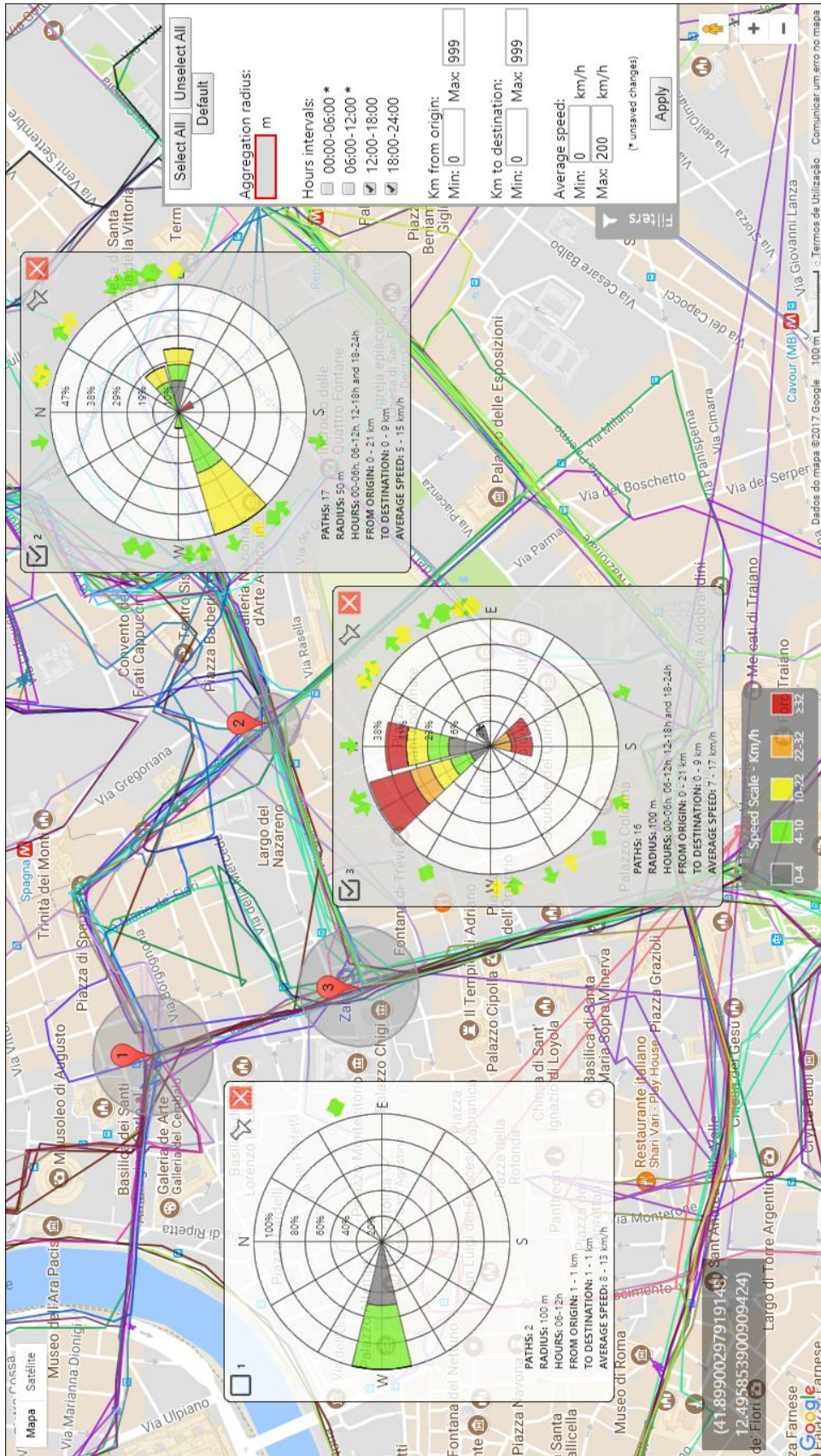


Figura 11 – Rosas dos ventos criadas em três pontos distintos do mapa para análise de trajetos

Na Figura 11 está visível a funcionalidade do Inspetor existente no painel de filtros. Sempre que é selecionada uma rosa, o painel de filtros ajusta os vários campos para os valores que estão definidos nas rosas selecionadas. Caso seja encontrada alguma incoerência, é colocado um rebordo vermelho e retirado o valor do campo em questão no painel de filtros. A incoerência desaparece com a desmarcação de uma ou mais rosas que estejam a gerar o conflito ou simplesmente com a aplicação de um valor comum ao campo que possui a incoerência. Sempre que existam incoerências, a criação de rosas é impossibilitada com uma mensagem de erro.

Atendendo à Figura 11, observa-se que raio de agregação das rosas 2 e 3 é diferente, o que motivou o aparecimento da notificação de incoerência de valores. Para os intervalos de horas, caso se procedesse à marcação da rosa 1, seriam assinaladas a fundo cinzento, as caixas desmarcadas referentes aos dois primeiros intervalos de horas no painel de filtros, uma vez que estão definidos mas ainda não aplicados às rosas 2 e 3. A sinalização de incoerência da *checkboxes* das horas é ligeiramente diferente devido a limitações de manipulação de estilo para este tipo de campo.

Eventos para manipulação de rosas

Toda a manipulação de rosas dos ventos assenta num conjunto de eventos. No sistema desenvolvido, os eventos podem ser lançados por parte da aplicação de análise *windroseMain* ou pelas instâncias *WindRose*, criadas a partir da biblioteca. A cada evento, está associado um *listener* que o processa, executando um conjunto de ações adequadas. Existe um ficheiro onde estão declarados todos os eventos utilizados no sistema, servindo de guia para o desenvolvimento de futuras aplicações.

De seguida, é apresentada uma lista com todos os eventos utilizados entre a aplicação *windroseMain* e a biblioteca *windroseLib*. Para cada evento, é explicado qual o momento em que o mesmo é despoletado, qual a sua finalidade, onde é que o seu *listener* está implementado e qual o comportamento do mesmo.

- ***applyFilters*** – Lançado pela aplicação *windroseMain* sempre que o utilizador pressiona o botão *Apply* no menu de filtros. Este evento comporta vários campos associados, nomeadamente todos os filtros definidos no painel, aquando da utilização do botão. O *listener* deste evento está implementado em todas as instâncias *WindRose* e sempre que deteta a ocorrência do evento,

consulta os dados dos filtros a aplicar, contidos no evento, e trata de chamar a função de aplicação de filtros da rosa;

- ***selectRose*** – Lançado pela aplicação *windroseMain*, sempre que o botão *Select All*, no painel de filtros, é pressionado. Este evento não comporta quaisquer campos. O *listener* deste evento está implementado em todas as instâncias *WindRose*, sendo que perante a ocorrência do evento, trata de alterar a rosa para o estado ‘selecionada’, caso esta se encontre no estado ‘desseleccionada’;
- ***unselectRose*** – O princípio é semelhante ao do evento *selectRose*, com a diferença de servir para mudar as rosas para o estado ‘desseleccionada’;
- ***roseCreated*** – Lançado por uma instância *WindRose*, sempre que a mesma acaba de criar uma rosa. O evento comporta um campo com o identificador único da rosa que acabou de ser criada. O *listener* deste evento está implementado na aplicação *web* e sempre que deteta a ocorrência do evento, trata de imprimir, na consola do navegador *web*, uma mensagem informativa da criação da rosa e verifica se é necessário, ou não, mostrar a caixa com a escala de cores da aplicação. De notar que esta caixa é mostrada sempre que existe pelo menos uma rosa criada no mapa;
- ***roseDeleted*** - Lançado por uma instância *WindRose*, sempre que a mesma termina o seu processo de destruição. O evento comporta um campo com o identificador único da rosa que acabou de ser destruída. O *listener* deste evento está implementado na aplicação *web* e sempre que deteta a ocorrência do evento, imprime, na consola do navegador *web*, uma mensagem informativa da destruição da rosa;
- ***roseSelected*** – Lançado por uma instância *WindRose*, sempre que a rosa acaba de passar para o estado ‘selecionada’. O evento comporta um campo com o identificador único da rosa que acabou de ser selecionada. O *listener* deste evento está implementado na aplicação *web* e sempre que deteta a ocorrência do evento, caso existam mais rosas já selecionadas, 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;

- *roseUnselected* – O princípio é o mesmo que o do evento *roseSelected*, com a diferença que este evento ocorre sempre que uma rosa é desseleccionada;
- *deleteRose* – Lançado por uma instância *WindRose*, sempre que o utilizador carrega no botão de fecho da rosa. O evento comporta um campo com 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).

A utilização de eventos entre a aplicação *windroseMain* e a biblioteca *windroseLib* constitui um mecanismo importante para a manipulação de rosas, uma vez que os eventos servem de canal de comunicação. Outra vantagem associada à utilização de eventos, prende-se com a possibilidade de se manipular várias rosas dos ventos em simultâneo, o que não era possível no protótipo anterior do sistema.

3.3.6 Bibliotecas externas

Ao longo do desenvolvimento do sistema recorreu-se a várias bibliotecas externas com o intuito de facilitar o desenvolvimento de determinadas funcionalidades, como descrito de seguida:

- Google Maps JavaScript API [14] – Utilizada em todas as funcionalidades que estejam relacionadas com a manipulação do mapa ou elementos associados ao mesmo, como o desenho de trajetos, cálculo de *bounding boxes*, criação de marcadores e respetivos círculos;
- jQuery [39] – Utilizado de forma geral na escrita do código JavaScript. Esta biblioteca tornou-se bastante útil pela facilidade com que permite executar determinadas tarefas que, em JavaScript puro, seriam mais elaboradas de se fazer, como é o caso dos pedidos AJAX;
- jQuery UI [40] – Tornou a funcionalidade de arraste das caixas das rosas muito simples de implementar, uma vez que bastou associar a cada objeto *WindRose* a propriedade *draggable* desta biblioteca, tornando o arraste suave e funcional;
- tabSlideOut [41] – Permitiu a criação do painel de filtros lateral, juntamente com o botão de acesso. Para além da criação deste elemento, a biblioteca

disponibiliza um conjunto de propriedades que permitem ajustar o comportamento do painel (por exemplo, configurar o fecho do painel para automático ou manual);

- Font Awesome [42] – Permitiu a inserção de ícones, com licença de utilização gratuita, em alguns elementos da aplicação *windroseMain*, como é o caso do ícone presente no botão do painel de filtros;
- qTip2 [43] – Possibilitou a rápida criação das *tooltips*, mostradas quando o rato está colocado sobre uma secção de pétala. O elemento acompanha a posição do rato enquanto este estiver sobre alguma secção da pétala, sendo mostrado o total de trajetos que estão agregados na secção a que o rato se sobrepõe.

Embora o uso destas bibliotecas tenha sido opcional, verificou-se que ofereceram uma clara vantagem no desenvolvimento de *software* devido à sua facilidade de utilização. Na maioria dos casos, caso não se tivesse recorrido às bibliotecas mencionadas, teria sido difícil executar todas as funcionalidades em tempo útil.

3.4 Preparação e análise de um conjunto de trajetos

Nesta secção 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 a preparação do conjunto de trajetos, até à sua análise com a criação de rosas na aplicação *web* do sistema.

3.4.1 Dados de trajetos de táxis

É importante testar o desempenho do sistema com um conjunto de trajetos grande. Para tal, utilizou-se a plataforma CRAWDAD, cujo objetivo passa por disponibilizar conjuntos de dados obtidos a partir de equipamentos com acesso a comunicações sem fios. Em concreto, considerou-se um conjunto de dados que representa a movimentação de 320 táxis na cidade de Roma durante o mês de Fevereiro de 2014 [44].

O conjunto de dados original, com o registo das movimentações de todos os veículos, constitui um ficheiro de extensão “.txt” com milhões de linhas e com um tamanho aproximado de 1.5GB. Para efeitos de teste, considerou-se uma amostra

consideravelmente mais reduzida, tendo-se optado por filtrar os registos para o primeiro dia do mês de fevereiro de 2014.

Por outro lado, o conteúdo original dos dados possuía um formato que não era favorável para uma análise posterior. De seguida, é apresentada uma linha do ficheiro original, que representa um registo de uma posição de um táxi no dia 1 de fevereiro às 00:00:00:

```
156;2014-02-01 0:00:00.739166+01;POINT(41.8836718276551 12.4877775603346)
```

Os vários campos, separados por “;” incluem o ID do táxi, o instante de recolha da posição e as coordenadas de latitude e longitude em forma de ponto.

Para se aproximar o conteúdo de cada linha ao convencional formato “CSV” (*Comma Separated Values*), aplicou-se um conjunto de transformações por forma a facilitar o seu carregamento para aplicações que permitam o tratamento da informação. Após o processamento do ficheiro, a linha apresentada anteriormente ficou com o seguinte aspeto:

```
156;2014-02-01 00:00:00;41.883672;12.487778
```

No novo formato da linha a data passa a ter o formato HH:MM:SS; a sintaxe do ponto foi removida, passando os valores de latitude e longitude a ter apenas 6 casas decimais e a ser separados por “;”. Com este pré-processamento dos dados, obteve-se um ficheiro com um tamanho de 12.2MB, com 293295 linhas.

Toda esta atividade de pré-processamento, foi realizada através de um programa escrito na linguagem Java, denominado *taxiProcessing*, que realiza a leitura do ficheiro original e escreve novas linhas devidamente tratadas num novo ficheiro. Os dias e meses que se pretendem incluir no ficheiro de *output* são parametrizáveis.

Após a obtenção do ficheiro resultante da atividade de pré-processamento, procedeu-se ao carregamento do mesmo para o RStudio [28]. O carregamento dos dados consistiu na criação de um *data frame* com 4 colunas distintas, em que cada uma corresponde a cada uma das entradas separadas por “;”.

Recorrendo à função *summary()*, obteve-se um sumário referente aos 293295 registos carregados (para o dia 1 de fevereiro de 2014): os IDs dos táxis em circulação variaram entre 2 e 368, sendo que existem registos de 162 táxis; o intervalo de registos de posição das viaturas variou entre as 00:00:00 e as 23:59:59; o valor mínimo de latitude registado foi 41.65899 e o máximo 42.23961; o valor mínimo de longitude registado foi 12.04925, sendo que o máximo foi 12.85367.

Na subsecção seguinte detalha-se o processo de geração de um ficheiro GPX a partir do conjunto de dados CSV que foi detalhado nesta subsecção.

3.4.2 Criação do ficheiro GPX com trajetos de táxis

Tendo em conta que os dados de entrada do sistema têm de estar no formato GPX, foi necessário fazer a conversão dos trajetos de táxis em Roma, descritos anteriormente. Para a criação do ficheiro GPX, escreveu-se uma aplicação utilitária na linguagem Java, denominada *gpxGenerator*, que aceita como entrada o conjunto de dados gerado pela aplicação *taxiProcessing*.

O primeiro passo da aplicação passa por ordenar as entradas do ficheiro de texto, por identificador de táxi e ordem ascendente de registo temporal. Esta tarefa é realizada através da execução de um *script* escrito em R, que carrega os dados num *data frame* e efetua o seu processamento. Para além da tarefa de ordenação, foi também reduzido o número total de táxis para oito. Esta decisão assentou no facto do ficheiro com os 162 táxis dar origem a um GPX muito grande para carregamento, o que fazia com que a quantidade de trajetos desenhados no mapa fosse excessiva e provocasse quebras de desempenho no sistema.

No final da execução do *script* R, é gerado um ficheiro CSV ordenado, pronto a ser processado pelo programa Java.

Tendo em conta a existência de erros frequentes entre a posição fornecida pelo GPS e a real, incluiu-se um pré-processamento para tratamento de situações pouco reais. Esta tarefa constitui o segundo passo da aplicação *taxiProcessing*.

A Figura 12 mostra um exemplo de um cenário em que ocorreu um erro de leitura do dispositivo GPS, o que origina uma oscilação irrealista no registo do trajeto. O tratamento deste tipo de erros foi feito a partir de um parâmetro, ajustável, que define a aceleração máxima que pode existir entre o registo de dois pontos, *maximumAccelerationThreshold*, tendo sido utilizado 10 metros por segundo. Caso a aceleração não seja respeitada, o ponto é descartado e é avaliado o ponto seguinte, até que seja encontrado um ponto que satisfaça esta condição. Este processo constitui a parte de filtragem de pontos do algoritmo.

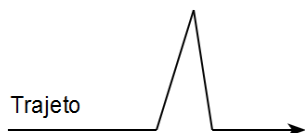


Figura 12 – Exemplo de um erro de leitura com sinal GPS



Figura 13 – Exemplo do erro de leitura quando um veículo se encontra parado num dado intervalo de tempo e volta a retomar a circulação



Figura 14 – Exemplo de uma situação em que ocorre uma incoerência temporal num dado trajeto, devido a perdas ou desligar de sinal

Por outro lado, achou-se conveniente diferenciar trajetos feitos pelo mesmo táxi quando ocorre uma paragem. Estas situações podem acontecer quando um determinado veículo está parado numa praça de táxis à espera de recolher um passageiro. Na maioria dos casos, devido à leitura imprecisa do sinal GPS, ocorrem oscilações em torno do ponto real onde o veículo se encontra parado. A espera por um passageiro numa praça de táxis constitui um possível cenário de exemplo para a ocorrência desta situação. Convencionou-se que quando ocorrem oscilações de registo prolongadas, tal como exemplificado na Figura 13, deve ser considerado o início de um novo trajeto percorrido pelo mesmo táxi, a partir do momento que o registo do trajeto volta a normalizar. A diferenciação de trajetos para o mesmo táxi, constitui o terceiro e último passo da aplicação *taxiProcessing*.

A Figura 13, apresenta um exemplo do aspeto de um trajeto registado quando um veículo se encontra parado por alguns instantes. Para cenários como este, tiveram-se em conta três parâmetros, que podem ser ajustados à medida de cada utilizador:

- Distância mínima que o táxi tem de percorrer, num dado intervalo de tempo, para que não seja considerado parado, *movingDistance*, tendo sido utilizado o valor de 50 metros;
- Intervalo de tempo que o táxi dispõe para percorrer a distância mínima definida, *movingInterval*, tendo-se utilizado 5 minutos;
- Intervalo de tempo máximo entre dois pontos, *maximumTimeInterval*, tendo sido definido 30 minutos.

Os parâmetros *movingDistance* e *movingInterval* servem para detetar se o táxi se encontra parado em algum local ou se está a circular muito lentamente no trânsito. O parâmetro *maximumTimeInterval*, permite descobrir se existiu alguma perda prolongada de sinal GPS que possa ter causado alguma incoerência temporal no registo de um trajeto, como exemplificado na Figura 14.

Caso algum dos três parâmetros não seja respeitado, é criado um novo segmento para o táxi que está a ser analisado.

É importante referir que o algoritmo utilizado, embora elimine estas situações presentes nos dados de trajetos do ficheiro original de táxis, não produz um resultado completamente isento de erros, tendo sido bastante difícil encontrar um *threshold* que minimize a eliminação excessiva de pontos.

No final da execução do algoritmo é gerado um ficheiro GPX com a seguinte estrutura: cada *track* representa um táxi, com um identificador único, que efetuou um conjunto de trajetos. Dentro da *track*, existe um conjunto de segmentos, em que cada um representa um trajeto percorrido pelo táxi da *track* em questão. Por fim, cada segmento possui um conjunto de pontos, registados ao longo do tempo, que compõe o trajeto. O ficheiro GPX final com os trajetos dos oito táxis, ficou com um tamanho total de 2.27 MB.

Na subsecção seguinte são explicados os passos para a conversão num formato compatível e carregamento do ficheiro GPX para a base de dados geográfica do sistema.

3.4.3 Conversão e carregamento dos trajetos

Após a obtenção do ficheiro GPX, é necessário convertê-lo num formato compatível para carregamento na base de dados geográfica.

A Figura 15 mostra, através de um diagrama de sequência, que a origem dos dados dos trajetos surge a partir de um ficheiro com a extensão GPX do utilizador. Por sua vez, o ficheiro GPX é enviado ao serviço *web convertGPXToJSON*, que o converte para o formato JSON, sendo o conteúdo guardado num ficheiro com essa extensão. Posteriormente, o ficheiro JSON é enviado, pelo utilizador, para o serviço *saveTracks*, onde o conteúdo é lido e convertido para linhas de informação que são inseridas na base de dados geográfica.

No final, é utilizado o serviço *web getTracks* sendo passado um par de coordenadas que definem o centro do círculo e o raio da área de agregação de uma hipotética rosa dos ventos. O serviço em questão efetua uma interrogação à base de dados sendo obtida uma resposta no formato JSON com os dados de todos os segmentos que intersejam o círculo definido pelo utilizador.

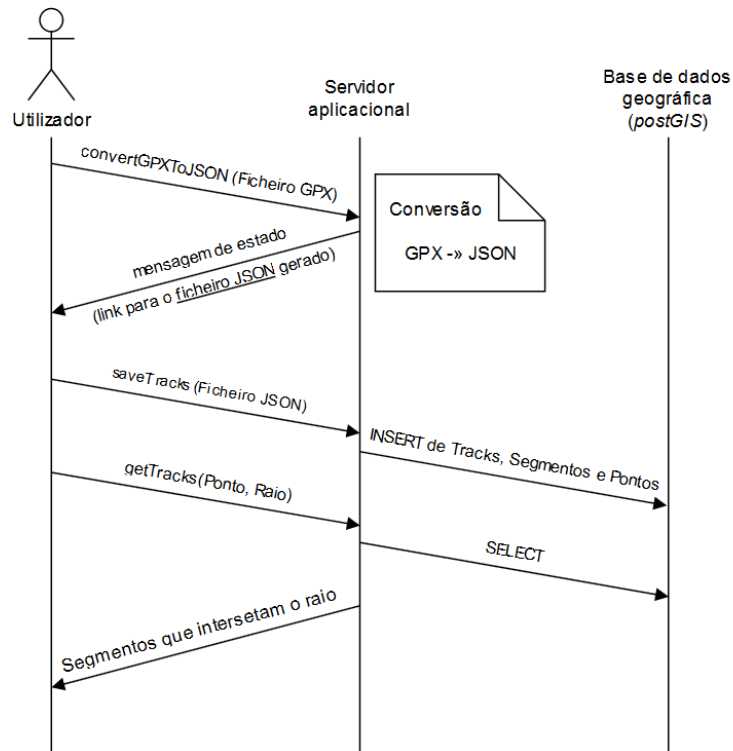


Figura 15 – Cenário de conversão e carregamento de um GPX para a base de dados, com posterior consulta de trajetos

Uma vez carregados os trajetos para a base de dados, o utilizador está apto a efetuar a sua análise com a aplicação *web* do sistema. A subsecção seguinte mostra os passos necessários para a seleção dos trajetos a partir de um ficheiro GPX e carregamento para a aplicação de análise *windroseMain*.

3.4.4 Seleção de trajetos para a aplicação de análise

Antes do carregamento de qualquer conjunto de trajetos para a aplicação de análise *windroseMain*, é necessário proceder ao carregamento dos mesmos para a base de dados geográfica, tal como descrito anteriormente. Idealmente, este processo deveria ser automático e incluído na aplicação de gestão *selectTracks*. No entanto, optou-se por se deixar o processo de carregamento de trajetos à responsabilidade do utilizador.

O diagrama de sequência apresentado na Figura 16 mostra os passos para a construção e carregamento de um conjunto de trajetos de táxis em Roma. Estes dados foram utilizados ao longo de todo o processo de desenvolvimento do sistema e na aplicação de testes para as sessões de avaliação com utilizadores, descrita no Capítulo 4.

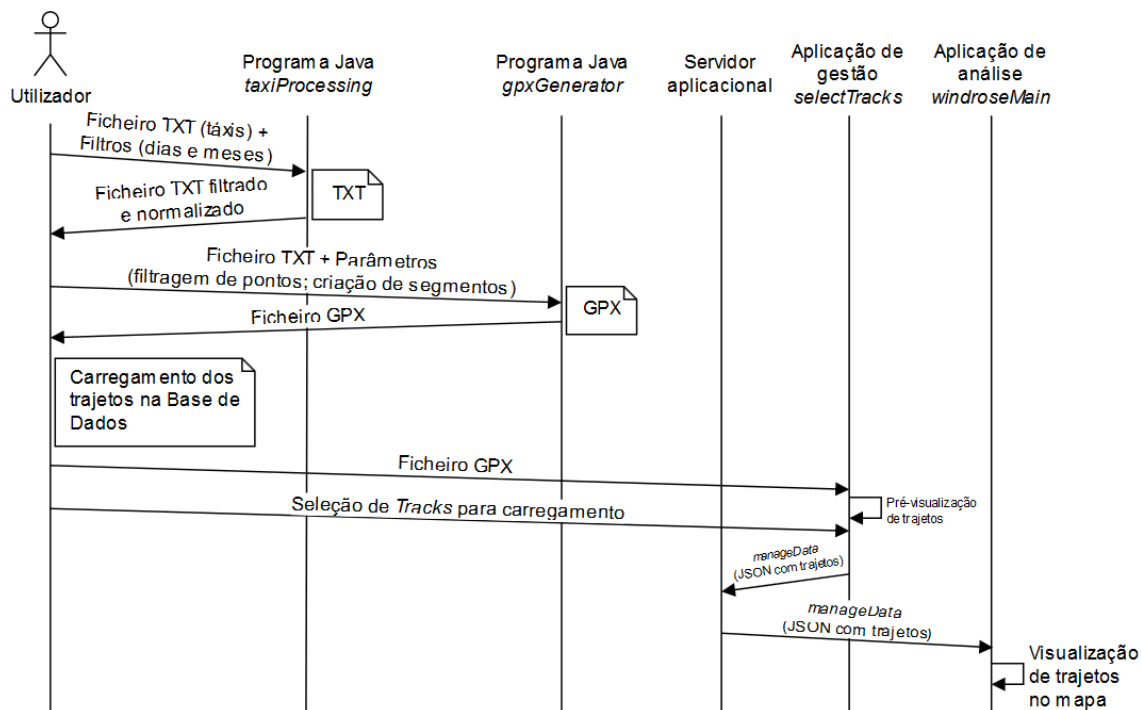


Figura 16 – Cenário de criação de um ficheiro GPX com trajetos de táxis e carregamento para a aplicação de análise de trajetos do sistema

No diagrama foi omitido o processo de carregamento dos trajetos na base de dados geográfica, uma vez que este é idêntico ao descrito na Figura 15, excluindo a utilização do serviço *getTracks*.

3.4.5 Criação de uma rosa dos ventos na aplicação de análise

Depois de carregados para a base de dados geográfica, os trajetos estão prontos a ser analisados usando a aplicação *windroseMain*, através da criação de elementos gráficos inspirados em rosas dos ventos.

O cenário de criação de uma rosa apresentado na Figura 17 assume que o utilizador já procedeu ao carregamento dos trajetos na base de dados com o serviço *saveTracks*. Nos primeiros passos, é acionado o serviço *manageData*, com a transferência dos trajetos escolhidos na aplicação *selectTracks* para a aplicação *windroseMain*.

Durante o processo de carregamento dos trajetos em *windroseMain*, existe uma sincronização de identificadores que é feita, a partir dos identificadores que a base de dados atribuiu aos segmentos e pontos. Este passo é importante, para que exista coerência de dados entre a base de dados e a aplicação de análise.

Após a sincronização de identificadores e desenho dos trajetos, o utilizador procede à criação de um novo marcador no mapa. Esta ação despoleta a invocação do

método *createWindrose* em *windroseMain*, que cria uma rosa dos ventos através do construtor da biblioteca *windroseLib*. Nesta invocação são passados alguns parâmetros relacionados com as propriedades da rosa a criar, tais como o raio, o centro do marcador da rosa, os trajetos do mapa, entre outros.

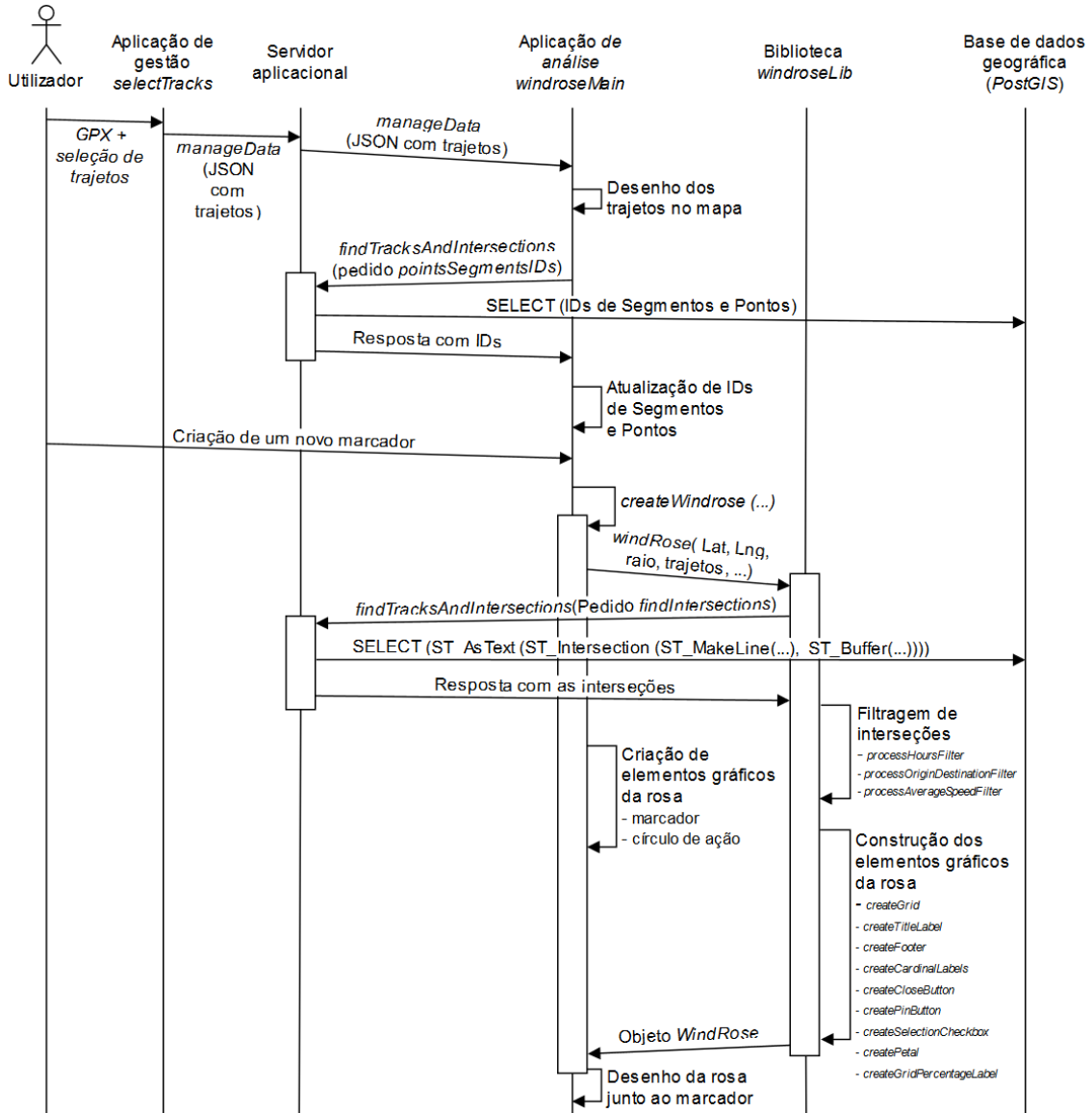


Figura 17 – Cenário de criação de um marcador na aplicação de análise *windroseMain*

Na biblioteca, o primeiro passo consiste na descoberta das interseções dos trajetos do mapa com a área de agregação da rosa que está a ser criada. Esta tarefa é feita com o auxílio do serviço *findTracksAndIntersections* com um pedido do tipo *findIntersections*. Após a obtenção das interseções, são aplicados filtros de acordo com as propriedades que foram definidas em *windroseMain*. Este passo é importante, uma vez que apenas devem

ser mostrados dados de interseções cujos trajetos respeitem os parâmetros da rosa em questão.

Obtidas as interseções com a área de agregação, a biblioteca da rosa dos ventos constrói os elementos gráficos da rosa, tais como as pétalas e os botões de seleção, *pin* e fecho. O objeto JavaScript que representa uma rosa tem acesso aos trajetos que foram carregados para a aplicação *web*, com o intuito de facilitar a geração dos elementos gráficos que a constituem.

Por último, após concluída a criação de todos os elementos gráficos da rosa num *SVG Canvas*, é devolvida à aplicação *web* um objeto do tipo *WindRose* que contém todos os elementos da rosa criada, sendo o grafismo da rosa colocado junto do respetivo marcador. A aplicação *windroseMain* permite que o utilizador altere aspetos das rosas, como o raio de agregação de trajetos ou a aplicação de filtros de tempo para, por exemplo passar de 100 para 200 metros de raio e considerar apenas trajetos realizados durante a manhã.

3.5 Sumário

Neste capítulo foi apresentado o sistema que permite a análise de trajetos com rosas dos ventos. Explicou-se como foram concretizados os dois primeiros objetivos definidos para o trabalho, nomeadamente a reestruturação e enriquecimento da biblioteca da rosa dos ventos e a construção de uma nova aplicação *web*. Também foi descrita a arquitetura e funcionamento dos seus componentes. Foram ainda apresentados cenários para detalhar a forma de interação entre os vários componentes do sistema quando são desempenhadas determinadas tarefas.

No capítulo seguinte é descrito o processo de avaliação do sistema com utilizadores com o objetivo de aferir o grau de usabilidade do mesmo.

Capítulo 4

Avaliação do sistema

Neste capítulo é descrito um estudo de usabilidade do sistema, efetuado com utilizadores, começando com uma explicação do propósito da avaliação, seguido do *design* da experiência, apresentação de resultados e, por último, discussão de resultados e avaliação feita por um perito.

4.1 Propósito

Após o desenvolvimento do atual protótipo do sistema de análise de trajetos com rosas dos ventos, foi feita a avaliação da usabilidade da aplicação *web*, com testes com utilizadores, focada nos seguintes objetivos:

- Perceber a facilidade que os utilizadores têm em avaliar cenários de excesso de informação num mapa, recorrendo a rosas dos ventos;
- Quantificar o nível de usabilidade da aplicação *web* do sistema através do método SUS [45];

A seguir é explicado como é que as avaliações foram concretizadas.

4.2 Estudo com utilizadores

Nesta secção está detalhada a caracterização dos participantes nas sessões de avaliação do sistema, o aparato usado nos testes, incluindo uma aplicação de testes que serviu de ferramenta para a concretização dos mesmos, as tarefas desempenhadas por cada participante e por último, o procedimento seguido em cada sessão de avaliação.

4.2.1 Participantes

Voluntariaram-se 21 pessoas para participar nas sessões de avaliação do sistema, nomeadamente 11 homens e 10 mulheres, com idade entre os 17 e os 27 anos. Todos os participantes foram selecionados por facilidade de contacto em meio social.

A maioria dos participantes não possuía qualquer experiência de análise de trajetos sobre mapas. No entanto, 3 das 21 pessoas já tinham utilizado aplicações para

efetuar análise de trajetos em dispositivos móveis inseridas na temática de saúde, nas quais os utilizadores registam trajetos que efetuam a pé ou de bicicleta e posteriormente visualizam-nos num mapa, juntamente com informação como a velocidade, elevação e calorias gastas, entre outras. É ainda importante referir que a maioria das pessoas considerou que um mapa com um elevado conjunto de trajetos desenhados (mostrado no Apêndice C na página 108), constitui excesso de informação, dificilmente gerível para análise.

4.2.2 Aparato

Cada sessão de avaliação foi feita com um utilizador, tendo decorrido numa sala fechada, sem interferências externas e com baixos níveis de ruído. Foi utilizado um computador portátil com um ecrã de 15.6” e resolução de 1366x768 pixéis, juntamente com um rato para facilitar a interação com a interface do sistema. O uso do *touchpad* do portátil não foi permitido, uma vez que este podia interferir nos testes, prejudicando o desempenho dos utilizadores.

Aplicação de testes

Para a concretização dos testes com utilizadores, desenvolveu-se uma aplicação de testes da aplicação *web* de análise do sistema, tendo o intuito de recolher dados de desempenho humano em vários cenários de análise com rosas dos ventos. Esta aplicação permite cobrir as tarefas definidas para o estudo com utilizadores, explicadas mais à frente, e capturar dados da atividade do utilizador, de uma forma que interfere pouco na interface da aplicação *windroseMain*. Cada cenário é acompanhado por uma questão, cuja resposta deve ser dada através de cliques em elementos gráficos que são mostrados na aplicação.

Ao longo do decorrer de um teste são guardados valores de um conjunto de variáveis, contemplando:

- *UserID* – ID do utilizador, inserido num campo de input para o efeito, antes de ser iniciado o teste;
- *QuestionID* – ID da questão;
- *QuestionOrder* – Ordem da questão;

- *Errors* – Total de erros na resposta, sendo que um erro representa um clique num elemento gráfico da interface do sistema que não corresponde à resposta certa da questão que está a ser respondida;
- *ExecutionTime* – Tempo de resposta, em segundos, desde que a pergunta é mostrada, até ao instante em que é despoletada a mensagem de resposta certa;
- *Skipped* – Se a pergunta foi respondida ou passada à frente. Este campo assume o valor *normal* em caso de não ter sido forçado o carregamento da pergunta seguinte. O valor *skipped* é utilizado para o caso contrário. De notar que o número de tentativas de resposta por questão é ilimitado;
- *Date* – Data da resposta, no formato “dia/mês/ano”;
- *Hour* – Hora da resposta, no formato “hora:minutos:segundos”. É importante referir que os campos referentes a instantes temporais (*Date* e *Hour*), registam o momento em que surgiu a mensagem de resposta certa. Caso seja forçado o carregamento da pergunta seguinte, é registado o instante em que foi solicitada esta operação.

O *log* é gerado num elemento *div* de HTML escondido no rodapé da interface da aplicação de testes. A visibilidade desta *div* pode ser alterada com um botão invisível colocado no canto superior direito da caixa de texto com o enunciado da questão atual. Quando a *div* é tornada visível, surge uma *scroll bar* na margem direita do ecrã que permite revelar todo o conteúdo do *log*. Existe ainda um segundo botão, colocado no canto inferior direito da caixa de questão, que permite saltar (fazer *skip*) para a questão seguinte do teste. Esta opção foi criada para eventuais situações em que o utilizador esteja com dificuldades em chegar à resposta correta. Estes dois botões constituem ferramentas escondidas na interface da aplicação, devendo ser utilizadas apenas por quem conduz a sessão de avaliação, de forma a não distrair o utilizador do foco do teste.

Para cada execução de uma tarefa pelo participante, fica registada uma linha no *log*. De seguida, apresenta-se um exemplo de um *log* gerado a partir de um teste com um utilizador, onde cada variável é separada por “,”.

```
UserID, QuestionID, QuestionOrder, Errors, ExecutionTime, Skipped, Date, Hour
1, 2, 1, 0, 49.1, normal, 28/5/2017, 12:15:05
1, 4, 2, 0, 114.7, normal, 28/5/2017, 12:17:01
1, 1, 3, 0, 88.0, normal, 28/5/2017, 12:18:34
1, 10, 4, 0, 116.1, normal, 28/5/2017, 12:20:32
```

1, 6, 5, 0, 60.5, normal, 28/5/2017, 12:21:35
1, 3, 6, 0, 19.7, normal, 28/5/2017, 12:21:59
1, 5, 7, 0, 203.3, normal, 28/5/2017, 12:25:23
1, 7, 8, 0, 66.9, normal, 28/5/2017, 12:26:32
1, 9, 9, 0, 79.5, normal, 28/5/2017, 12:27:53
1, 8, 10, 0, 10.7, normal, 28/5/2017, 12:28:07

A título de exemplo, a partir do registo anterior é possível saber que a questão com o identificador 2 foi a primeira a aparecer no teste do utilizador com identificador 1. Este não errou a resposta e demorou 49.1 segundos a responder à questão. A questão decorreu normalmente, sem que tenha sido necessário forçar o carregamento da pergunta seguinte. Por último, esta questão foi respondida no dia 28 de maio de 2017, às 12:15.

4.2.3 Tarefas

Foram criados 4 tipos de questão, que correspondem a tarefas plausíveis e frequentes de análise de trajetos, nomeadamente:

- SEMA (*Select marker*) – Selecionar o marcador que corresponde à rosa dos ventos mostrada no ecrã;
- SEPE (*Select petal*) – Selecionar a pétala com o maior número de trajetos e velocidade ou a pétala com maior velocidade, da rosa que está a ser mostrada no ecrã;
- SERO (*Select rose*) – Selecionar, de entre várias, a rosa dos ventos que corresponde ao marcador mostrado no ecrã;
- INSE (*Inspect and select*) – Inspeccionar uma determinada zona com trajetos no mapa, delimitada por linhas a tracejado, e assinalar por onde passam os trajetos que respeitem as características apresentadas na questão, como por exemplo a zona por onde passam veículos com o destino mais distante.

Na Figura 18 encontra-se um exemplo de uma questão do tipo INSE, em que se pede que o utilizador se foque na região delimitada a tracejado no mapa, onde se pretende que seja testada a capacidade de análise de trajetos através da colocação de rosas dos ventos em pontos estratégicos. Após a visualização das rosas, o utilizador deve assinalar a região por onde passam veículos com a velocidade média mais baixa entre as 18 e as 24 horas, fazendo duplo clique num marcador de rosa. Caso o marcador se encontre na zona considerada correta, a aplicação valida a ação como uma resposta certa e mostra uma

caixa de aviso (*alert box*) com essa informação. O mesmo sucede para uma resposta errada. A questão seguinte é carregada após o clique no botão ‘Ok’ da *alert box*, sempre que é dada a resposta certa.

Para as questões do tipo SEMA, pretende-se que o utilizador interprete os trajetos agregados numa rosa dos ventos desenhada do mapa e consiga associar o marcador que lhe corresponde. São criados três marcadores no mapa com as letras A, B e C, juntamente com uma rosa dos ventos. O utilizador deverá validar a resposta com duplo clique no marcador que considera ser correspondente à rosa mostrada no mapa.

As questões do tipo SERO seguem a lógica inversa à do tipo SEMA. Neste tipo de questão é desenhado um marcador no mapa, juntamente com três rosas dos ventos. O utilizador deve analisar os trajetos que intersectam a zona de agregação em torno do marcador e escolher qual a rosa dos ventos correspondente, através de um duplo clique naquela que considera correta.

Existem ainda as questões do tipo SEPE, em que se pretende que o utilizador analise o conteúdo das pétalas e selecione uma secção que satisfaça a condição do enunciado da questão, como por exemplo, a seleção de uma secção de pétala que contenha o maior número de trajetos e maior velocidade. A validação da resposta é feita através de um duplo clique na secção de pétala que o utilizador considera correta.

A Tabela 3 mostra a correspondência entre os vários indentificadores das questões dos testes e a descrição correspondente.

Tabela 3 – Identificadores das questões do teste e respetivas descrições

| Identificador de questão | Descrição |
|---------------------------------|--|
| SEMA 1, 2 | Tendo em conta a rosa dos ventos no mapa, faça duplo clique no marcador correspondente. |
| SEPE 1 | Atendendo à rosa dos ventos no mapa, faça duplo clique na secção de pétala com maior número de trajetos e maior velocidade. |
| SEPE 2 | Atendendo à rosa dos ventos no mapa, faça duplo clique na secção de pétala com maior velocidade. |
| SERO 1,2 | Tendo em conta o marcador do mapa, faça duplo clique na rosa dos ventos correspondente. |
| INSE 1, 3 | Dentro da região delimitada a tracejado no mapa, escolha a zona por onde passam mais veículos entre as 18h e as 24h. |
| INSE 2 | Dentro da região delimitada a tracejado no mapa, escolha a zona por onde passam veículos com destino mais distante, a qualquer hora. |
| INSE 4 | Dentro da região delimitada a tracejado no mapa, escolha a zona por onde passam veículos com velocidade média mais baixa, entre as 18h e as 24h. |

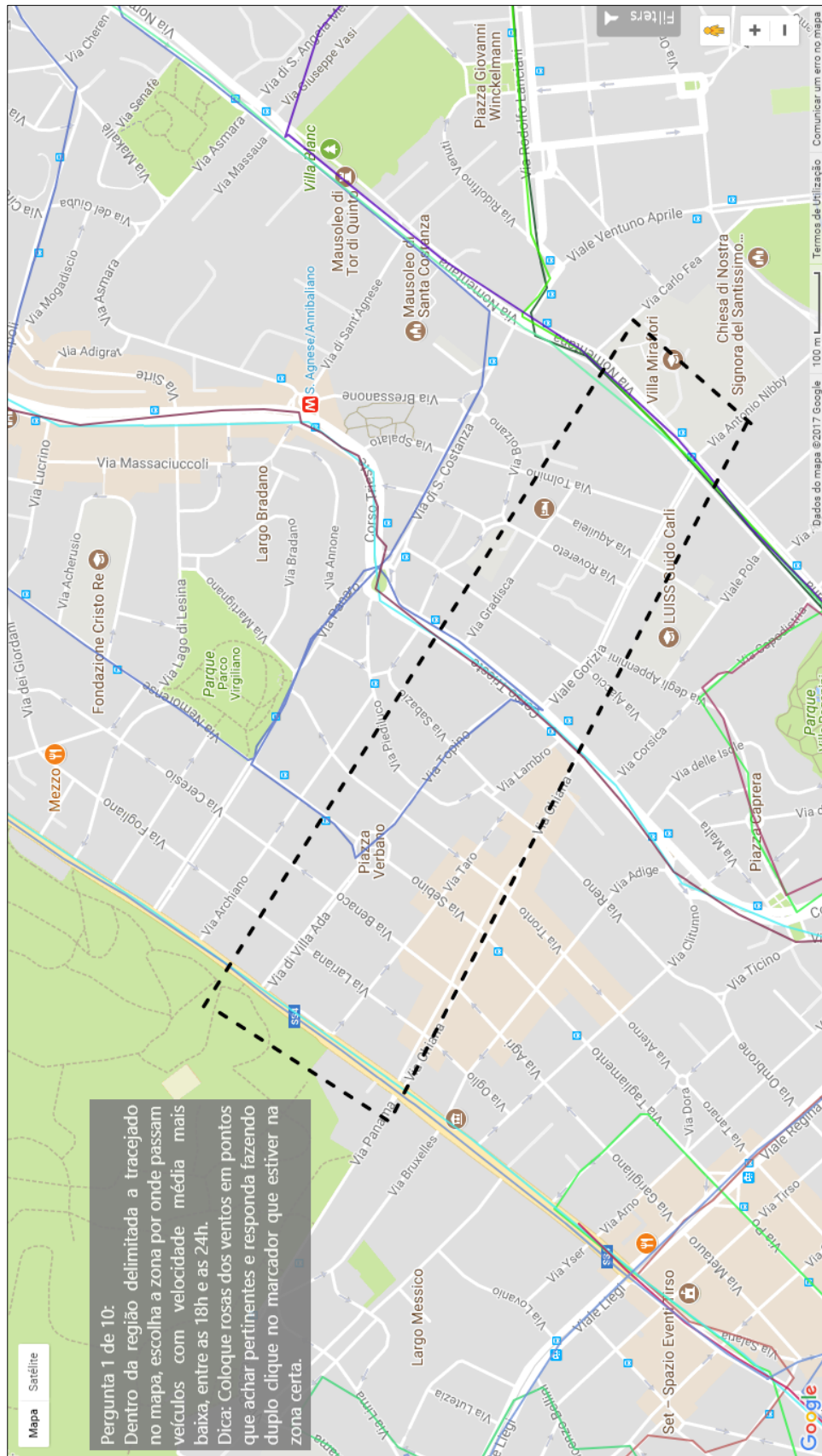


Figura 18 – Uma questão do tipo INSE mostrada durante os testes do sistema

As perguntas do tipo INSE foram acompanhadas de uma dica no enunciado, com o intuito de guiar o utilizador: “Coloque rosas dos ventos em pontos que achar pertinentes e responda fazendo duplo clique”.

Para cada um dos tipos de questão, existiram 2 cenários associados, com exceção para as questões INSE, em que existem 4 cenários. A criação de diferentes cenários foi pensada com o intuito de testar o efeito de aprendizagem entre as várias questões, ou seja, pretende-se que o utilizador faça uso dos mesmos elementos gráficos em duas situações de análise distintas. Com esta aproximação, tenta-se perceber se a resposta dada foi mecanizada ou com base na informação obtida através da análise dos elementos gráficos das rosas e dos trajetos nos mapas.

Existiram no total 10 questões no banco de perguntas da aplicação. Cada utilizador respondeu às mesmas questões, sendo que estas foram ordenadas de forma aleatória no início de cada teste. Isto fez com que cada utilizador tivesse uma probabilidade elevada de responder a um teste com uma sequência de questões diferente dos restantes.

4.2.4 Procedimento

Cada teste realizado por cada participante foi composto por um conjunto de passos, listados por ordem, de seguida:

- Realização de um questionário inicial;
- Treino com a aplicação *web* de análise de trajetos (*windroseMain*);
- Realização do teste com a aplicação de testes;
- Realização de um questionário final de usabilidade.

Nos passos de treino e realização do teste foi utilizado o conjunto de dados de trajetos de táxis em Roma, detalhado anteriormente na secção 3.4.

O primeiro passo da sessão de avaliação consistiu na realização de um questionário *online*, elaborado em Google Forms (ver Apêndice C), que permitiu obter dados sobre cada utilizador, nomeadamente: género, idade, e experiência com trajetos. Existiram ainda duas questões de opinião pessoal sobre quais os atributos considerados importantes na análise de trajetos e, dado um mapa com trajetos, se existia excesso de informação caso fosse necessário proceder à análise dos mesmos.

O segundo passo da sessão de avaliação foi a realização de um treino com a aplicação *windroseMain* do sistema, composto por duas partes. Na primeira, foi mostrada

uma imagem, impressa numa folha A4 (ver Apêndice D), da síntese da aplicação de análise, com uma rosa dos ventos criada num ponto em que passavam trajetos no mapa. Esta imagem era acompanhada de uma legenda para os principais elementos gráficos da aplicação, permitindo que o participante recebesse um sumário das funcionalidades da interface da aplicação de análise do sistema. Na segunda parte, o participante utilizou a aplicação livremente com um conjunto reduzido de trajetos, com a finalidade de facilitar a aprendizagem. Eventuais dúvidas que surgissem ao longo do treino eram prontamente esclarecidas, sendo que também não existia qualquer limite de tempo durante esta fase da avaliação.

O próximo passo da avaliação foi a resposta a questões usando a aplicação de testes. Durante a execução, o participante pôde ser esclarecido quanto a eventuais dúvidas relacionadas com o enunciado da questão. Este procedimento não afetou os resultados, uma vez que era feito antes da resposta poder ser dada. Caso surgissem dificuldades na resposta às questões, eram fornecidas algumas dicas, sem comprometer as soluções. Embora nunca tenha sido preciso, caso o participante ficasse muito tempo sem conseguir resolver a questão, era forçado o carregamento da seguinte.

Para as questões INSE, nos casos em que os marcadores foram colocados num ponto muito próximo da zona correta, mas fora da área de validação, o erro detetado pela aplicação foi ignorado e 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. Estes tempos foram designados de tempos de recuperação.

O último passo da sessão de avaliação foi o preenchimento de um questionário de usabilidade (ver Apêndice E). Este questionário foi também elaborado em Google Forms e permitiu quantificar o grau de usabilidade do sistema, tendo seguido a estrutura do teste SUS [45]. No final do questionário, foi dada oportunidade aos participantes para sugerirem melhorias à interface e novas funcionalidades, bem como apontarem dificuldades que tenham encontrado durante a utilização do sistema.

Após a resposta às questões, foi copiado o *log* gerado para um ficheiro CSV. Este ficheiro foi sendo construído à medida que foram realizados mais testes com participantes. No final, o ficheiro ficou com o registo dos 21 testes realizados. Para a compreensão de eventuais dificuldades nas respostas às questões e validação de registos

que suscitasse algumas dúvidas, procedeu-se à gravação do ecrã do computador onde foram realizados os testes, sem captura de som. Para este efeito, utilizou-se a ferramenta OBS Studio [46]. No final, obtiveram-se 21 ficheiros de vídeo, com o registo de cada um dos testes dos participantes.

4.3 Resultados

Esta secção descreve os principais resultados da avaliação com utilizadores. Os dados foram recolhidos do *log* da aplicação de testes e dos questionários inicial e de usabilidade. Todo o seu processamento e construção de gráficos foi realizado com o RStudio [28].

4.3.1 Tempos de resposta

Os gráficos apresentados na Figura 19 e Figura 20, mostram os *box plots* dos tempos de resposta dos utilizadores às questões dos testes, em segundos.

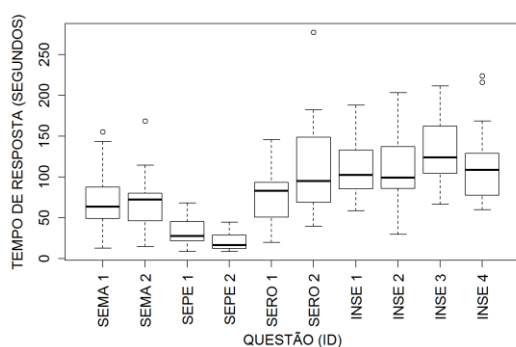


Figura 19 – Tempos de resposta dos utilizadores, por questão

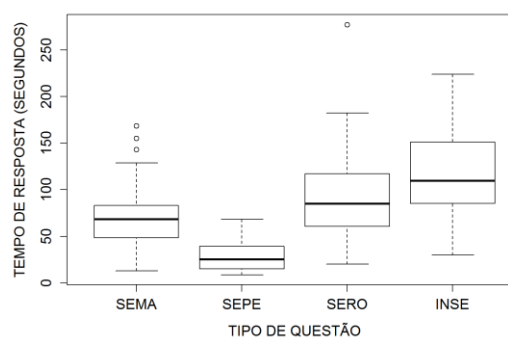


Figura 20 – Tempos de resposta dos utilizadores, por tipo de questão

Na Figura 19, os identificadores das questões foram atribuídos relativamente ao cenário do tipo da questão. Tome-se como exemplo as perguntas do tipo SEMA, em que os identificadores SEMA 1 e SEMA 2 representam os cenários 1 e 2 da pergunta, respetivamente. Embora os identificadores sejam fixos, a ordem dos cenários foi sempre aleatória para cada participante.

Na Figura 20 verifica-se que o maior tempo de resposta, por parte dos utilizadores, está associado às perguntas do tipo INSE. Por outro lado, as perguntas do tipo SEPE apresentaram os tempos de resposta mais baixos. Os tipos SEMA e SERO, tiveram tempos de resposta intermédios.

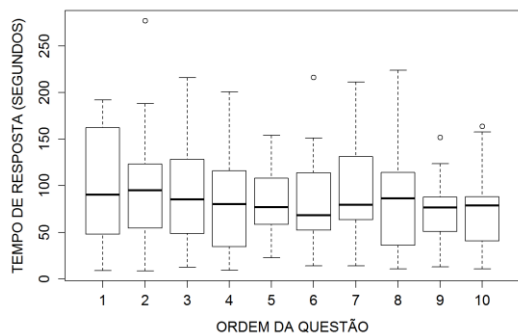


Figura 21 – Tempos de resposta dos utilizadores, por ordem de questão

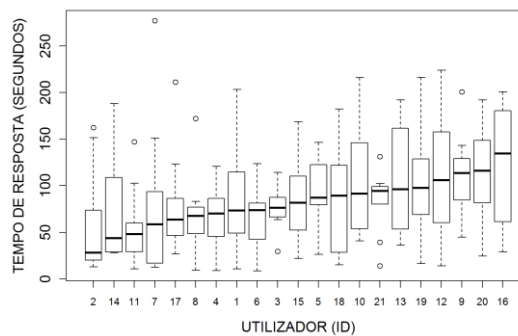


Figura 22 – Tempos de resposta, por utilizador

A Figura 21 mostra que as medianas do tempo de resposta em função da ordem das perguntas dos testes são bastante próximas. Por último, na Figura 22 é apresentado, por ordem crescente de mediana, a variação do tempo de resposta por utilizador.

4.3.2 Erros

Esta subsecção é destinada aos resultados dos erros dados ao longo dos testes. Estes representam a quantidade de vezes que os utilizadores clicaram em elementos gráficos da interface do sistema que não correspondiam à resposta certa da questão a que estavam a responder.

A Figura 23 evidencia que os erros ao longo dos testes foram praticamente inexistentes, independentemente do tipo de questão a que os utilizadores foram submetidos.

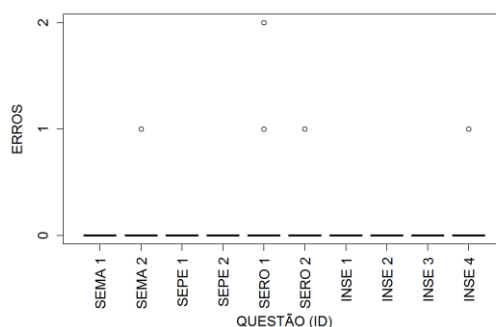


Figura 23 – Erros ao longo dos testes, por tipo de questão

Tendo em conta que a ocorrência de erros foi praticamente nula, optou-se por não se incluir mais gráficos relativamente a este parâmetro, visto que a quantidade total de erros não pode ser considerada significativa.

4.3.3 Usabilidade do sistema

Nesta subsecção apresentam-se os gráficos das pontuações de usabilidade do sistema, atribuídas pelos utilizadores, com base no questionário SUS (ver Apêndice E).

Na Figura 24 estão representadas as classificações de usabilidade totais, elaboradas conforme um artigo que faz uma correspondência entre intervalos de pontuação e as mesmas [45]. As pontuações totais de usabilidade foram calculadas de acordo com fórmulas do SUS, onde cada pergunta possui um determinado peso para o cálculo da pontuação de usabilidade final. A maioria dos utilizadores considerou que o sistema é fácil de usar, e por isso, a sua pontuação correspondeu à classificação máxima da escala.

Na Figura 25 é visível a distribuição da pontuação de usabilidade atribuída por género dos utilizadores. As medianas de ambos os géneros são praticamente as mesmas, existindo uma variação ligeiramente superior nos utilizadores do sexo feminino.

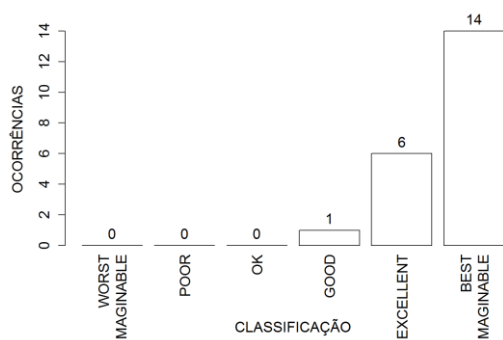


Figura 24 – Classificações de usabilidade do sistema, atribuídas pelos utilizadores

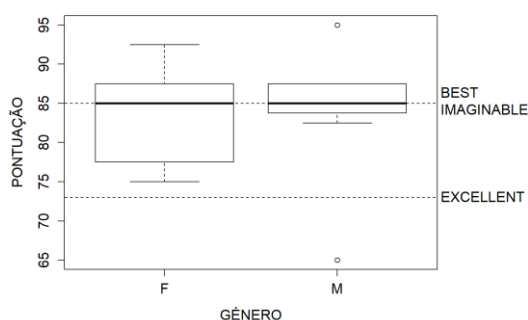


Figura 25 – Classificações de usabilidade do sistema, por género de utilizador

A Figura 26 mostra as classificações atribuídas pelos utilizadores às 10 perguntas do questionário SUS. Cada pergunta possuía uma afirmação referente à utilização do sistema e foi classificada numa escala de 1 a 5, relativamente ao grau de concordância do utilizador com a afirmação, em que o valor 1 representa discordância total, e o 5 concordância total. De seguida, são apresentadas as 10 afirmações do questionário SUS:

1. Gostaria de utilizar esta aplicação com frequência;
2. Achei a aplicação desnecessariamente complexa;
3. Achei a aplicação fácil de se usar;
4. Precisaria da ajuda de uma pessoa com conhecimentos técnicos para utilizar a aplicação;
5. As várias funcionalidades da aplicação estão muito bem integradas;

6. A aplicação apresenta um elevado grau de incoerência;
7. Acredito que a maioria das pessoas facilmente aprenderá a usar a aplicação;
8. Achei a aplicação muito estranha de se usar;
9. Senti-me muito confiante ao usar a aplicação;
10. Precisei de aprender muitas coisas antes de conseguir usar a aplicação.

Durante a execução dos primeiros 17 questionários de usabilidade, a questão 3 foi omitida por lapso, tendo o problema sido corrigido para os últimos 4. Uma vez que a obtenção da pontuação de usabilidade depende obrigatoriamente das 10 questões do teste SUS, procedeu-se a uma recuperação dos dados em falta, cerca de uma semana depois. Tendo em conta os identificadores e datas de registo de cada teste, elaborou-se uma reconstituição da sequência de participação dos utilizadores. Para uma segunda verificação, perguntou-se a cada um dos 17 utilizadores em causa, se ainda tinham presente o identificador que lhes tinha sido atribuído. Todos responderam afirmativamente, tendo os identificadores fornecidos correspondido à reconstituição feita. Por último, recolheu-se a classificação, de 1 a 5, à pergunta que estava em falta, tendo a mesma sido adicionada aos questionários de usabilidade desses utilizadores.

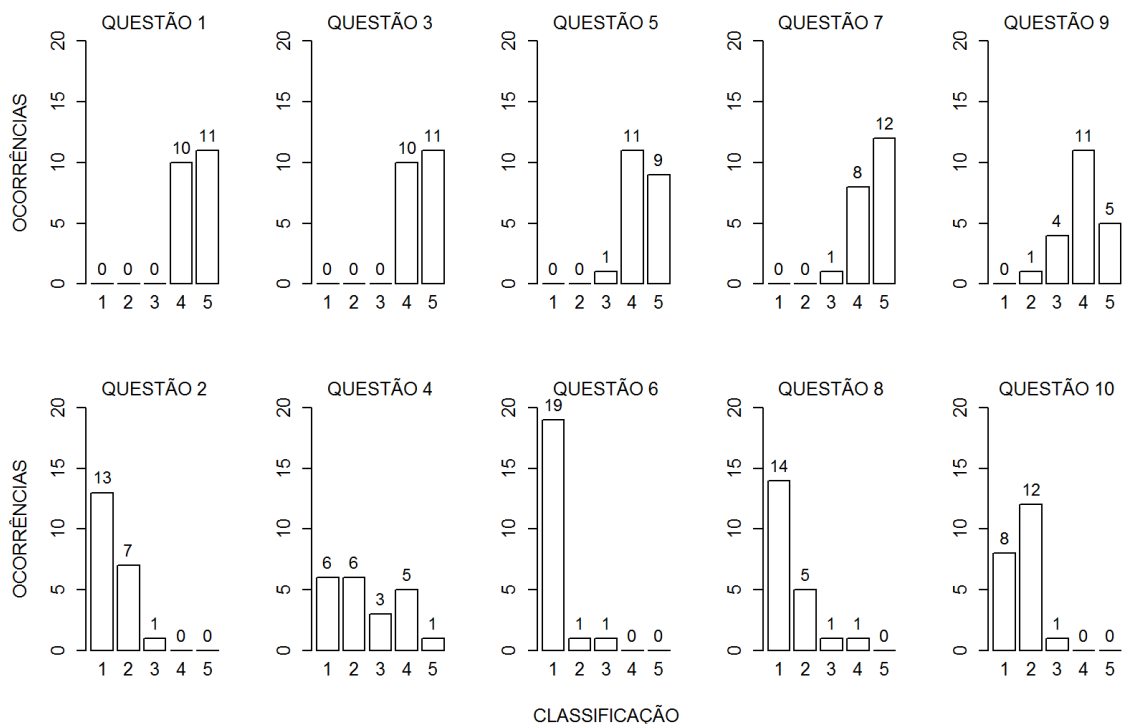


Figura 26 – Pontuações atribuídas em cada pergunta do questionário SUS

Na primeira linha da Figura 26 estão dispostas as questões que possuem as frases construídas num sentido positivo, sendo que na segunda linha estão dispostas as frases

formuladas de forma negativa. O grau de concordância dos utilizadores, relativamente às questões da primeira linha foi moderadamente elevado. Por outro lado, nas questões da segunda linha, a questão 4 registou níveis de concordância e discordância semelhantes, sendo que nas restantes questões da linha, a discordância foi uma classificação comum.

4.4 Discussão

Nesta secção é apresentada uma discussão dos resultados obtidos, incluindo diversas ilações sobre o processo de avaliação com os utilizadores, sendo também divulgada a opinião de um perito, relativamente à utilização do sistema no seu contexto profissional.

4.4.1 Tempos de resposta

No que diz respeito aos tempos de resposta dos utilizadores, considera-se que as perguntas mais fáceis de responder foram as do tipo SEPE, em que as respostas foram quase imediatas após a leitura do enunciado da questão, e as do tipo SEMA, embora os tempos destas últimas tenham sido ligeiramente mais elevados. Por outro lado, registaram-se tempos de resposta algo elevados nas questões do tipo SERO. Nos cenários deste tipo de questão, criaram-se, propositadamente, rosas dos ventos que suscitassem dúvidas quanto à orientação das pétalas e quantidade de trajetos.

Um dos problemas nas respostas às questões SEMA foi a utilização de cores aleatórias para a representação de diferentes caminhos no mapa. Foi frequente existirem cores muito semelhantes, o que levou alguns utilizadores a pensar que se tratavam do mesmo trajeto a interseção o raio de agregação da rosa. Na maioria dos casos, isto fez com que fosse gasto mais tempo na diferenciação de cores do que propriamente na análise das rosas dos ventos.

Por último, como era esperado, as questões do tipo INSE têm os tempos de resposta mais elevados, uma vez que requerem um fluxo de trabalho mais elaborado, relativamente aos outros tipos de questão.

Os testes para a existência de eventuais efeitos de aprendizagem revelaram que esse efeito foi inexistente. Este resultado pode ser constatado na Figura 21, onde existe uma mediana muito semelhante da primeira à última questão colocada a cada utilizador. Uma vez que a ordem das questões foi alterada a cada teste, pode-se admitir que nos

vários cenários dos 4 tipos de pergunta, os utilizadores responderam raciocinando sobre a informação mostrada no ecrã. Caso as respostas tivessem ocorrido de forma mecanizada, seria verificada uma descida da mediana nas questões finais, o que não aconteceu.

4.4.2 Erros

A ausência de erros foi uma constante no decorrer dos testes, como evidenciado na Figura 23. Curiosamente, esta situação já havia ocorrido nas avaliações com utilizadores no primeiro protótipo do sistema [6].

Com a avaliação do atual protótipo, desenvolveram-se questões com um grau de complexidade mais elevado, como é o caso das questões do tipo INSE, que obrigam a um fluxo de trabalho elaborado que podia ser suscetível à ocorrência de erros (algo que se esperava nos resultados). De uma forma surpreendente, e bastante positiva, os utilizadores continuaram a não errar e a única consequência da maior complexidade parece ter sido o tempo de resposta mais longo. Este resultado leva a crer que a aplicação de análise de trajetos é fácil de perceber.

4.4.3 Usabilidade do sistema

No geral, o questionário de usabilidade mostrou que os utilizadores sentiram que o sistema é fácil de utilizar, tal como sugere a Figura 24, onde apenas um utilizador atribuiu uma classificação final ao sistema intermédia (*Good*), sendo que os restantes atribuíram o nível seguinte (*Excellent*) ou máximo (*Best Imaginable*).

O lapso que levou a questão 3 ter sido classificada uma semana depois dos testes, pelos primeiros 17 utilizadores, podia ter levado a resultados duvidosos. No entanto, todos esses utilizadores afirmaram ter presente os testes que realizaram e que ainda conseguiam dar uma classificação válida de acordo com a sua prestação. Esta questão acabou por não se revelar problemática, visto que existiu uma concordância geral.

Atendendo à Figura 26, destaca-se a questão 4 (“Precisaria da ajuda de uma pessoa com conhecimentos técnicos para utilizar a aplicação”) por não ter uma concordância geral entre os utilizadores, ao contrário das restantes. Estes resultados podem advir do facto de ter existido alguma subjetividade na interpretação da afirmação. A maior parte das pessoas fez questão de perguntar qual o contexto em que a pergunta surgia,

nomeadamente se existia algum manual de apoio da aplicação que pudesse ser consultado, ou se simplesmente a pessoa tinha que tentar utilizar a aplicação sem qualquer instrução prévia. Foi-lhes dito para classificarem a questão assumindo que existia o manual de apoio ao funcionamento do sistema.

Provavelmente, os utilizadores que concordaram com a questão 4 assumiram que a documentação seria suficiente para que pudessem aprender a utilizar o sistema. Por outro lado, os utilizadores que utilizaram a discordância como classificação, tiveram em conta que o manual não seria suficiente para que conseguissem fazer uma utilização correta do sistema, sendo que alguns reportaram que, como não tinham por hábito efetuar a leitura dos manuais, acabariam por explorar a aplicação por tentativa e erro.

4.4.4 Tempos de recuperação

Cada um dos 7 erros ocorridos durante os testes, possui um valor de tempo de recuperação associado, que representa a quantidade de segundos, desde o instante em que ocorreu o erro até ao momento em foi dada a resposta certa. De uma forma geral, os tempos de recuperação foram reduzidos, tendo variado entre 5 e 101 segundos (com mediana de 17 segundos), o que mostra que os utilizadores conseguiram rapidamente ajustar o seu raciocínio para chegar à resposta certa da questão.

4.4.5 Influência da experiência prévia com análise de trajetos

O número de participantes com experiência prévia de análise de trajetos em mapas foi muito reduzido, tendo sido contabilizadas apenas três pessoas. Estes conhecimentos não se revelaram vantajosos relativamente ao desempenho nos testes, pois dois destes utilizadores deram um erro ao longo dos seus testes. Estes erros, embora não sendo muito significantes, comprovam que com ou sem experiência prévia, os utilizadores obtiveram um desempenho semelhante no decorrer dos testes.

4.4.6 Fluxos de trabalho

A forma de interagir com o sistema pode variar de utilizador para utilizador, e isso verificou-se nas questões INSE.

A maioria dos utilizadores preferiu fazer uma criação de rosas com os parâmetros pré-definidos, aplicando os filtros posteriormente. Uma minoria dos utilizadores ajustou

os filtros, de acordo com o que era pedido no enunciado da questão, antes de procederam à criação das rosas dos ventos nos pontos que achavam pertinentes. Acredita-se que esta ocorrência se deva ao facto de os utilizadores terem tido pouco tempo de contacto com o sistema antes da realização do teste, mesmo que não tenha existido qualquer limite de tempo no treino com a aplicação. Talvez também devido ao facto de se tratar de uma funcionalidade que não está assinalada de forma notável (como um botão, por exemplo), a maioria das pessoas não se lembrou que podia ajustar os parâmetros antes da criação das rosas.

Duas outras funcionalidades quase nunca utilizadas foram os botões de marcação/desmarcação de todas as rosas dos ventos no mapa, ou seja, a maioria dos participantes optou por marcar as rosas dos ventos uma a uma. Após a realização do teste, os participantes foram questionados sobre o fluxo de trabalho para o ajuste das propriedades das rosas dos ventos. Com as respostas obtidas, chegou-se à principal causa: os botões não eram suficientemente elucidativos na função que desempenhavam, o que causava alguma insegurança quanto à sua utilização.

Por último, alguns utilizadores tiveram tendência para usar o inspetor, em vez de olharem diretamente para a informação no rodapé de cada rosa. Este cenário registou-se, por exemplo, na pergunta INSE que pedia para ser assinalada a zona onde passavam veículos com o destino mais longínquo. No final, os participantes acabaram por utilizar a informação mostrada no rodapé de cada rosa e responderam com sucesso à questão. Estas ocorrências podem estar relacionadas com o facto de as informações mostradas no rodapé não possuírem um título que esclareça que são relativas aos trajetos que são agregados pela rosa dos ventos em questão.

4.4.7 Opinião de um perito

A obtenção de *feedback* por parte de um profissional numa área com aplicabilidade para o sistema desenvolvido é importante, uma vez que se avalia o sistema na ótica de um analista de trajetos.

Após terem sido efetuados alguns contactos, surgiu a oportunidade de se marcar uma entrevista com o diretor do Centro de Atividades de Transporte de Bucelas da Rodoviária de Lisboa, tendo este mostrado disponibilidade para observar, experimentar e avaliar o sistema desenvolvido. A primeira parte da reunião consistiu numa demonstração

prática do sistema, com o mesmo conjunto de trajetos utilizado nos testes com utilizadores. O *feedback*, obtido na segunda parte do encontro, teve em conta um possível cenário de análise de trajetos de autocarros pelas rotas da rede concessionada pela empresa.

Visto que se trata de uma aplicação de análise de trajetos registados no passado, a sua utilidade foi considerada importante no que toca ao planeamento e gestão das rotas dos veículos. Dado que este tipo de trabalho é feito por analistas, o sistema deveria ser capaz de produzir relatórios de informação das várias rotas que são criadas no ecrã, com um resumo dos dados mostrados pelas mesmas. Desta forma, deixaria de ser necessário que a interpretação do sistema estivesse dependente da familiarização com os vários componentes gráficos da interface.

Relativamente à área de agregação das rotas, foi mencionado que é muito importante fornecer uma análise sobre o que se passa dentro do raio de cada marcador, em vez de fora, como está no protótipo atual. Um exemplo disso, seria perceber por onde é que cada trajeto entra no raio de ação e por onde sai, em vez de existir uma agregação geral desses dados. Esta análise individual poderia ser acompanhada de propriedades dentro do raio de agregação, tais como a velocidade média e distância percorrida. Foi também referido que seria interessante perceber qual a velocidade média desde a origem dos trajetos até ao centro do marcador, bem como daí até ao destino. A necessidade de todos estes detalhes foi justificada com a possível análise de rotas alternativas para os autocarros, baseadas no fluxo de tráfego dentro das áreas de agregação de rotas criadas em pontos estratégicos.

A flexibilidade de configuração de parâmetros foi um dos aspetos mais apreciados, visto que o sistema que a empresa utiliza para monitorizar os seus veículos, não permite a reconfiguração de certos filtros. No entanto, foi apontado que seria útil a possibilidade de reconfiguração dos intervalos de horas em que se pretendem observar trajetos tendo em conta, por exemplo, os horários de ponta.

Por último, a possibilidade de utilizar o sistema com dados em tempo real foi apontada como uma ideia inovadora que permitiria melhorar a forma de monitorização dos veículos da empresa. A inclusão de alertas de atrasos de circulação nas várias rotas criadas no mapa, permitiria perceber que veículos estariam afetados dentro do raio de ação de um determinado marcador. O redirecionamento de autocarros através de rotas

alternativas mais rápidas seria facilitado com a criação de rosas dos ventos noutros pontos do mapa que permitissem verificar a rapidez com que determinados veículos da frota os atravessaram.

Em conclusão foi mencionado que o sistema desenvolvido constitui um protótipo muito interessante, com uma forma inovadora de analisar trajetos, podendo ser melhorado e vir a mostrar-se útil em várias áreas profissionais.

4.5 Sumário

Neste capítulo foram apresentados aspetos relacionados com a avaliação do sistema de análise de trajetos, nomeadamente os procedimentos utilizados nas sessões de avaliação, a aplicação utilizada para os testes, os resultados obtidos e uma discussão sobre aquilo que foi observado. A parte da discussão conta também com a opinião de um perito sobre o funcionamento do sistema.

Capítulo 5

Conclusões

Este capítulo destina-se à apresentação das principais contribuições do projeto, competências adquiridas, desafios encontrados no decorrer do trabalho e, por último, ideias para desenvolver trabalho futuro.

5.1 Principais contribuições

Tendo em conta o primeiro objetivo do sistema – evolução da biblioteca da rosa dos ventos – desenvolveram-se novas funcionalidades que deram origem à criação de uma nova versão, atualizada, da biblioteca, com maior capacidade de análise de trajetos. O código da biblioteca foi reorganizado, conferindo uma melhor separação de responsabilidades, tendo também sido definido um conjunto de eventos para a manipulação de rosas. A flexibilidade também foi tida em conta, sendo possível utilizar a biblioteca em qualquer outra aplicação, para além da que foi descrita neste documento. Foi ainda gerada documentação (no Apêndice B) com o intuito de agilizar o processo de aprendizagem das funcionalidades da biblioteca por parte de programadores.

O segundo objetivo do trabalho – construção de uma aplicação *web* – foi cumprido através de uma reformulação total da aplicação *web* existente. A nova aplicação *web* possui um conjunto de características que permite a criação, manipulação e análise de rosas dos ventos sobre um mapa com trajetos. O novo painel de filtros e inspetor, auxilia todas estas tarefas de uma forma eficaz, permitindo a consulta e alteração de parâmetros de uma ou mais rosas em simultâneo e sinalização de incoerências entre valores de filtros de rosas selecionadas.

Adicionalmente, foi criado um conjunto de serviços *web* que auxiliam o funcionamento do sistema, com um conjunto de tarefas que são realizadas do lado do servidor. Estes serviços contribuem para o bom funcionamento do sistema, bem como para um ganho na capacidade de análise grandes conjuntos de dados de trajetos. A sua implementação permitiu: a possibilidade de análise de conjuntos de trajetos maiores, relativamente ao primeiro protótipo do sistema, o suporte a qualquer ficheiro GPX e

possibilidade de consulta e utilização de funções geográficas sobre os trajetos guardados na base de dados geográfica.

O terceiro, e último, objetivo – avaliação do sistema – foi concretizado através de sessões com utilizadores. Este aspeto foi importante, uma vez que permitiu perceber que o grau de usabilidade do sistema é elevado e que as rosas dos ventos permitem aos utilizadores cumprir várias tarefas de análise de trajetos quase sem errarem.

Por último, e não menos importante, foi publicado e apresentado um artigo sobre o anterior e atual protótipo do sistema na 12^o Conferência Ibérica de Sistemas e Tecnologias de Informação, que decorreu em Lisboa, no ISCTE, durante o mês de junho de 2017 [7].

5.2 Competências adquiridas

O desenvolvimento deste trabalho levou a que fossem desenvolvidas várias competências, tanto a nível de programação como ao nível de *soft skills*.

Ao nível da programação, destaca-se a aprendizagem de várias bibliotecas em JavaScript. Estas bibliotecas, apesar de possuírem um vasto leque de funcionalidades, requerem alguma habituação na forma como são integradas no código da aplicação *web* e da biblioteca do sistema. A utilização da ferramenta RStudio também permitiu o enriquecimento dos conhecimentos com a linguagem R.

O contacto com um sistema de gestão de base de dados geográfica, PostGIS em PostgreSQL, permitiu uma primeira familiarização com bases de dados deste tipo, nomeadamente com a utilização de funções geográficas sobre os dados de trajetos.

A construção dos serviços *web* obrigou a que existisse uma familiarização com as principais bibliotecas do PHP e com os protocolos usados para a formulação de pedidos e respostas entre o cliente e os serviços *web*. A configuração do servidor aplicacional, bem como a instalação e configuração da base de dados geográfica, foi algo novo que contribuiu para a experiência com estas tecnologias.

A utilização de um repositório *online* com controlo de versões para guardar o código do projeto permitiu o contacto com este tipo de tecnologia, mais concretamente o Git, que era escasso antes da realização deste projeto.

A produção e apresentação do artigo permitiu adquirir competências na escrita e capacidade de comunicação. A flexibilidade foi outro dos aspetos desenvolvidos ao longo

da realização do trabalho, uma vez que perante alguns imprevistos, foi exigida alguma capacidade de adaptação para que o trabalho prosseguisse normalmente.

5.3 Desafios encontrados

Uma das maiores dificuldades encontradas 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. Embora o formato JSON tivesse facilitado a troca de informação, a implementação das estruturas de dados teve que ser cuidada para os dados ficarem num formato coerente e de fácil compreensão.

As diferenças entre ficheiros de trajetos foi outro problema encontrado, uma vez que nem todos os ficheiros GPX seguem um conjunto de atributos adicionais idênticos. Os serviços *web* de processamento e carregamento de trajetos para a base de dados tiveram de ser construídos tendo em conta algumas destas diferenças, o que fez que com que esta tarefa não fosse trivial.

Outra dificuldade encontrada relacionou-se com o uso das funções geográficas do PostGIS. Embora exista documentação com exemplos práticos, foi necessário investir algum tempo a aprender a sintaxe e forma de uso das mesmas.

5.4 Trabalho futuro

Os aspetos relacionados com o trabalho futuro deste projeto foram compilados a partir do *feedback* obtido com as avaliações dos utilizadores, da avaliação de um perito e da lista de funcionalidades futuras pensadas ao longo da fase de codificação do sistema.

A possibilidade de efetuar análises aos trajetos, dentro da área de agregação da rosa, é um dos aspetos mais interessantes. 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.

Por outro lado, poderiam ser aplicados outros tipos de filtros aos trajetos, com maior liberdade de ajuste de parâmetros, como por exemplo intervalos de horas personalizados.

A possibilidade de ser gerado um relatório com informação sobre os dados mostrados nas várias rosas do mapa também é uma ideia de implementação, uma vez que

facilitaria bastante a tarefa de análise de trajetos a quem não está totalmente familiarizado com a interface gráfica do sistema.

Ainda ao nível da 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 *sliders* com um limite mínimo e máximo para os campos de velocidade média e distância à origem e destino. A possibilidade de utilização de novas formas geométricas para a área de agregação de trajetos de uma rosa é algo que pode ser explorado, uma vez que pode facilitar a seleção, sobre o mapa, de determinados trajetos em concreto.

O suporte de atualizações dos trajetos em tempo real nas rosas dos ventos é outra funcionalidade que pode ser implementada. Com esta funcionalidade, as rosas seriam atualizadas, de uma forma periódica, com a chegada de novos dados de trajetos à base de dados. Esta atualização também seria válida caso o utilizador pretendesse mover o marcador da rosa para outro ponto do mapa, o que atualmente não é possível.

Bibliografia

- [1] M. J. Eppler e J. Mengis, «The concept of information overload: A review of literature», *The Information Society*, vol. 20, n. 5, pp. 325–344, 2004.
- [2] J. Thomas e J. Kielman, «Challenges for visual analytics», *Information Visualization*, vol. 8, n. 4, pp. 309–314, 2009.
- [3] S. Upoor, O. Trullols-Cruces, M. Fiore, e J. M. Barcelo-Ordinas, «Generation and analysis of a large-scale urban vehicular mobility dataset», *IEEE Transactions on Mobile Computing*, vol. 13, n. 5, pp. 1061–1075, 2014.
- [4] J. Yuan *et al.*, «T-Drive: Driving directions based on taxi trajectories», em *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*, 2010, pp. 99–108.
- [5] Meteoblue, «Climate Lisbon Airport», 2016. [Em linha]. Disponível em: https://www.meteoblue.com/en/weather/forecast/modelclimate/lisboa-%2F-portela_portugal_6301792. [Acedido: 18-Out-2016].
- [6] M. Carreira, «Agregação de dados usando a metáfora da rosa dos ventos», (Dissertação de Mestrado em Engenharia Informática, ainda não avaliada) Universidade de Lisboa.
- [7] R. Vaz, M. Carreira, A. Ferreira, e A. P. Afonso, «Track analysis system with wind roses», em *Proceedings of the 12th Iberian Conference on Information Systems and Technologies (CISTI)*, 2017, pp. 806–810.
- [8] H. L. Crutcher, «On the standard vector-deviation wind rose», *Journal of Meteorology*, vol. 14, n. 1, pp. 28–33, 1957.
- [9] R. Conceição, M. Melgão, H. G. Silva, K. Nicoll, R. G. Harrison, e A. H. Reis, «Transport of the smoke plume from Chiado’s fire in Lisbon (Portugal) sensed by atmospheric electric field measurements», *Air Quality, Atmosphere and Health*, vol. 9, n. 3, pp. 275–283, 2015.
- [10] J. W. Winchester e G. D. Nifong, «Water pollution in Lake Michigan by trace elements from pollution aerosol fallout», *Water, Air, and Soil Pollution*, vol. 1, n. 1, pp. 50–64, 1971.
- [11] X. Jia, D. Chung, J. Huang, M. Petrilli, e L. The, «ARO: Geographic information systems-based system for optimizing airport runway orientation», *Journal of*

- Transportation Engineering*, vol. 130, n. 5, pp. 555–559, 2004.
- [12] N. M. Al-Abadi, «Wind energy resource assessment for five locations in Saudi Arabia», *Renewable Energy*, vol. 30, n. 10, pp. 1489–1499, 2005.
- [13] Microsoft, «Bing map shapes: Polylines and polygons». [Em linha]. Disponível em: <https://msdn.microsoft.com/en-us/library/mt712696.aspx>. [Acedido: 22-Fev-2017].
- [14] Google, «Google Maps JavaScript API: Simple polylines». [Em linha]. Disponível em: <https://developers.google.com/maps/documentation/javascript/examples/polyline-simple>. [Acedido: 22-Fev-2017].
- [15] I. Bogdanovich, «Geo Tracker - GPS tracker», 2016. [Em linha]. Disponível em: https://play.google.com/store/apps/details?id=com.ilyabogdanovich.geotracker&hl=pt_PT. [Acedido: 27-Out-2016].
- [16] Y. Zheng, «Trajectory data mining: An overview», *ACM Transactions on Intelligent Systems and Technology*, vol. 6, n. 3, pp. 1–41, 2015.
- [17] T. M. Green, W. Ribarsky, e B. Fisher, «Building and applying a human cognition model for visual analytics», *Information Visualization*, vol. 8, n. 1, pp. 1–13, 2009.
- [18] World Wide Web Consortium (W3C), «Web services architecture», 2004. [Em linha]. Disponível em: <https://www.w3.org/TR/ws-arch/>. [Acedido: 21-Jul-2017].
- [19] J. R. Harger e P. J. Crossno, «Comparison of open-source visual analytics toolkits», *Proceedings of SPIE*, vol. 8294, n. 1, p. 82940E–82940E–10, 2012.
- [20] «Birdeye information visualization and visual analytics library». [Em linha]. Disponível em: <https://code.google.com/archive/p/birdeye/>. [Acedido: 21-Jul-2017].
- [21] Google, «Visualization API Reference». [Em linha]. Disponível em: <https://developers.google.com/chart/interactive/docs/reference>. [Acedido: 21-Jul-2017].
- [22] M. Bostock e J. Heer, «Protovis: A graphical toolkit for visualization», *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, n. 6, pp. 1121–1128, Nov. 2009.
- [23] L. Zhang, A. Stoffel, M. Behrisch, e S. Mittelstädt, «Visual analytics for the big data era: A comparative review of state-of-the-art commercial systems», em *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*

- (VAST), 2012, pp. 173–182.
- [24] Lakes Environmental, «WRPLOT View», 2016. [Em linha]. Disponível em: <http://www.weblakes.com/products/wrplot/index.html>. [Acedido: 28-Jul-2017].
 - [25] PHP development team, «PHP: Hypertext Preprocessor», 2016. [Em linha]. Disponível em: <http://php.net/manual/en/index.php>. [Acedido: 02-Nov-2016].
 - [26] W3C, «W3C - HTML», 2016. [Em linha]. Disponível em: <https://www.w3.org/html/>. [Acedido: 02-Nov-2016].
 - [27] JetBrains, «PhpStorm», 2016. [Em linha]. Disponível em: <https://www.jetbrains.com/phpstorm/>. [Acedido: 02-Nov-2016].
 - [28] «RStudio – Open source and enterprise-ready professional software for R», 2016. [Em linha]. Disponível em: <https://www.rstudio.com>. [Acedido: 02-Nov-2016].
 - [29] «PostGIS: Spatial and geographic objects for PostgreSQL», 2016. [Em linha]. Disponível em: <http://www.postgis.net>. [Acedido: 02-Nov-2016].
 - [30] «PostgreSQL: The world’s most advanced open source database», 2016. [Em linha]. Disponível em: <https://www.postgresql.org>. [Acedido: 02-Nov-2016].
 - [31] «XAMPP installers and downloads for Apache Friends», 2016. [Em linha]. Disponível em: <https://www.apachefriends.org/index.html>. [Acedido: 02-Nov-2016].
 - [32] The Apache Software Foundation, «Apache HTTP server project», 2016. [Em linha]. Disponível em: http://httpd.apache.org/ABOUT_APACHE.html. [Acedido: 19-Jul-2017].
 - [33] Atlassian, «Bitbucket: The Git solution for professional teams», 2017. [Em linha]. Disponível em: <https://bitbucket.org>. [Acedido: 22-Jul-2017].
 - [34] «JSON - JavaScript object notation». [Em linha]. Disponível em: <http://www.json.org/>. [Acedido: 15-Set-2017].
 - [35] Geomagik, «Pocket Earth: Offline maps and travel guides for iPhone and iPad», 2012. [Em linha]. Disponível em: <https://pocketearth.com>. [Acedido: 02-Nov-2016].
 - [36] «Ajax - Asynchronous JavaScript and XML». [Em linha]. Disponível em: <https://developer.mozilla.org/en-US/docs/AJAX>. [Acedido: 15-Set-2017].
 - [37] W3C SVG Working Group, «SVG: Scalable vector graphics». [Em linha]. Disponível em: <https://www.w3.org/Graphics/SVG>. [Acedido: 25-Jul-2017].

- [38] «JSDoc 3 - API documentation generator for JavaScript», 2017. [Em linha]. Disponível em: <http://usejsdoc.org/about-getting-started.html>. [Acedido: 05-Set-2017].
- [39] The JS Foundation, «jQuery». [Em linha]. Disponível em: <https://jquery.com>. [Acedido: 09-Jul-2017].
- [40] The JS Foundation and the jQuery UI Team, «jQuery UI - jQuery user interface extension», 2012. [Em linha]. Disponível em: <http://jqueryui.com>. [Acedido: 09-Jul-2017].
- [41] M. Fielding, «tabSlideOut - jQuery plugin to create tabs that slide out to show a form», 2016. [Em linha]. Disponível em: <https://github.com/hawk-ip/jquery.tabSlideOut.js>. [Acedido: 09-Jul-2017].
- [42] D. Gandy, «Font Awesome: the iconic font and CSS toolkit». [Em linha]. Disponível em: <http://fontawesome.io/>. [Acedido: 09-Jul-2017].
- [43] C. Thompson, «qTip2 - Pretty powerful tooltips», 2013. [Em linha]. Disponível em: <http://qtip2.com/guides>. [Acedido: 09-Jul-2017].
- [44] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, e A. Rabuffi, «CRAWDAD dataset Roma/taxi (v.2014-07-17)», *CRAWDAD wireless network data archive*, 2014. [Em linha]. Disponível em: <http://crawdad.org/roma/taxi/20140717>. [Acedido: 07-Nov-2016].
- [45] A. Bangor, P. Kortum, e J. Miller, «Determining what individual SUS scores mean: Adding an adjective rating scale», *Journal of Usability Studies*, vol. 4, n. 3, pp. 114–123, 2005.
- [46] H. Bailey, «Open Broadcaster Software Studio (OBS Studio)», 2017. [Em linha]. Disponível em: <https://obsproject.com>. [Acedido: 09-Jul-2017].

Apêndice A

Artigo apresentado na 12^o CISTI

Sistema de Análise de Trajetos com Rosas dos Ventos

Track Analysis System with Wind Roses

Ricardo Vaz¹, Mauro Carreira¹, António Ferreira², Ana Paula Afonso²

LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
¹{fc41939,fc35991}@alunos.fc.ul.pt, ²{amferreira,apafonso}@fc.ul.pt

Resumo — Neste artigo apresentamos um sistema de análise de trajetos recorrendo a gráficos inspirados em rosas dos ventos. Os elementos da rosa incluem pétalas que agregam trajetos em função da direção, mostrando também as suas velocidades médias em redor de um ponto marcado num mapa, bem como setas que apontam para a origem e destino desses trajetos. O utilizador pode interagir com a rosa aplicando filtros de tempo e ajustando o raio de agregação, permitindo, por exemplo, visualizar movimentos pendulares da população a entrar de manhã e sair à noite de uma cidade. Uma avaliação com utilizadores revelou a quase ausência de erros na resposta a questões de interpretação das agregações dos trajetos, e tempos de resposta que não diferiram significativamente com o género nem com o escalão etário, mas que sugerem melhorias na interface gráfica. O sistema oferece uma nova forma de analisar trajetos, baseada numa técnica clássica de agregar ventos, contribuindo para aumentar o corpo de conhecimento na área da visualização analítica de dados.

Palavras Chave — *visualização analítica; rosa dos ventos; agregação de trajetos.*

Abstract — In this paper we present a system for analyzing tracks using graphics inspired in wind roses. The elements of the rose include petals that aggregate tracks given their direction and that also show their average speeds within an area around a point in a map, as well as arrows that point to the origin and destination of the tracks. The user can interact with the rose by applying time filters and by adjusting the aggregation radius, allowing, for instance, the visualization of commuting patterns. An evaluation with users revealed that almost no errors were made regarding the answers to questions about the interpretation of track aggregation data, and that the response times did not differ significantly with gender or age band and suggested improvements in the graphical interface. The system offers a new way of analyzing tracks, based on a classic technic for aggregating wind data, contributing to the body of knowledge in the area of visual analytics.

Keywords — *visual analytics; wind rose; track aggregation.*

I. MOTIVAÇÃO

É comum as organizações possuírem grandes quantidades de dados, que excedem a capacidade humana de processamento, havendo a necessidade de os agregar e filtrar para possibilitar a descoberta de padrões e tendências [1]. A representação gráfica dos dados agregados e a interação com os mesmos são aspetos importantes para o bom desempenho dos decisores, estando na essência da visualização analítica de dados [2].

Os dados em causa neste artigo são trajetos, que podem ser percorridos por entidades, tais como pessoas, automóveis, aviões, e barcos, e que se presume terem sido registados por dispositivos de posicionamento global (GPS), ubíquos nos dias de hoje. A análise de trajetos é usada em tarefas de planeamento, por exemplo, urbano, em que a agregação dos trajetos permite visualizar movimentos pendulares da população e reorganizar o fluxo de tráfego, como foi feito na cidade de Colónia, na Alemanha [3]. Neste caso, os trajetos foram agregados com linhas de diferentes espessuras, e a passagem do tempo foi visualizada com mapas animados, técnicas também usadas na demonstração de um estudo com trajetos em Pequim, na China [4].

Neste artigo procuramos ir além da informação disponibilizada pelos mapas animados: enquanto estes permitem aplicar sucessivamente filtros temporais que revelam, por exemplo, onde há mais circulação automóvel em cada hora do dia, descrevemos a seguir um sistema capaz de mostrar, de forma agregada, de onde vieram e para onde foram esses condutores. Desta forma, em vez dos decisores terem de interpretar múltiplas imagens em busca de padrões para a acumulação de automóveis, passam a dispor de explicações sobre o trânsito.

O sistema integra uma aplicação *web* e serviços suportados por uma base de dados geográfica para oferecer ao decisor a possibilidade de analisar trajetos num mapa colocando e interagindo com gráficos inspirados nas rosas dos ventos. Uma rosa dos ventos é um gráfico radial convencionalmente usado na representação de ventos predominantes numa zona, tal como o aeroporto de Lisboa [5]. Aplicando a rosa dos ventos aos dados dos trajetos, a pétala maior é aquela cuja direção agrega mais trajetos e a maior região colorida dentro de cada pétala representa o intervalo de velocidades mais comum. A estes elementos gráficos, acrescentámos filtros para, por exemplo, agregar trajetos realizados só de manhã ou de tarde, e setas de origem e destino dos trajetos no rebordo da rosa, sendo que, do nosso conhecimento, estas últimas são uma ideia inovadora.

O artigo está estruturado como segue: na Secção II apresentamos o trabalho relacionado; na Secção III descrevemos as componentes e funcionalidades do sistema desenvolvido; na Secção IV mostramos os resultados de uma avaliação preliminar com utilizadores, e discutimos, na Secção V as limitações existentes; na Secção VI apresentamos as conclusões e ideias de trabalho futuro.

II. TRABALHO RELACIONADO

Existem na literatura várias aplicações de rosas dos ventos em cenários de análise e tomada de decisão, bem como trabalhos sobre visualização de trajetos, que descrevemos a seguir.

A. Aplicações das rosas dos ventos

As pétalas das rosas dos ventos permitem visualizar qual a direção ou direções dos ventos dominantes numa zona em redor de um ponto, sendo tanto maiores quanto maior for a frequência acumulada de vento no ângulo coberto por cada pétala. Por exemplo, numa rosa com quatro pétalas seriam agregados ventos cobrindo 45° de um lado e doutro dos pontos cardeais norte, este, sul e oeste. Dentro de cada pétala é comum existir uma escala de velocidades, em que os ventos mais fracos estão na parte da pétala mais próxima do centro da rosa [6].

A aplicação WRPLOT [7] permite desenhar rosas a partir de ficheiros com dados de ventos, contendo datas e horas, direções, velocidades, e localizações geográficas. O aspeto gráfico da rosa pode ser ajustado, incluindo a escala de cores das velocidades e o número de pétalas, e os dados podem ser filtrados para, por exemplo, mostrar só os ventos em dias úteis. Devido ao formato específico usado nos ficheiros, uma limitação desta aplicação é cingir-se a dados de ventos, não se adaptando a outros tipos de dados, como trajetos.

Para além da caracterização estrita dos ventos, as rosas dos ventos têm-se mostrado úteis em cenários de análise de poluição atmosférica e decisões sobre infraestruturas.

Recentemente, foi publicado um estudo de simulação do movimento das nuvens de fumo nos dois dias após o incêndio do Chiado em 1988, suportado por rosas dos ventos com o centro definido no aeroporto da Portela (devido à disponibilidade de dados nesse período), através das quais foi possível descrever a evolução da orientação dos ventos, bem como da sua intensidade [8]. Noutro estudo, foi usada uma rosa para agregar ventos durante um período de dez anos, tendo sido determinante para concluir que a poluição atmosférica ao longo da costa sudeste do lago Michigan contribuiu para a presença de elementos químicos poluentes na água [9].

No que toca a decisões de planeamento de infraestruturas, o sistema ARO [10] procura facilitar a tarefa de orientar as pistas de um aeroporto, recorrendo a rosas dos ventos para permitir a análise da predominância dos ventos, em particular se estes são favoráveis à descolagem e aterragem de aviões. Noutro trabalho, conduzido na Arábia Saudita, as rosas dos ventos foram úteis para caracterizar os ventos dominantes em cinco zonas, permitindo decidir sobre a instalação de moinhos eólicos [11].

B. Visualização de trajetos

Relativamente aos dados de trajetos, uma visualização comum consiste em unir com uma linha pontos geográficos registados sucessivamente pelo dispositivo GPS, sendo estas linhas projetadas num mapa. Esta é a abordagem dos serviços de mapas da Microsoft [12] ou da Google [13], sendo que a aplicação *web* do sistema aqui proposto tira partido deste último.

Apesar de ser intuitivo visualizar trajetos num mapa, quando estes são muitos ou se sobrepõem, passa a ser adequado fazer agregações. Uma técnica habitual consiste em juntar vários trajetos semelhantes numa só linha com traço mais grosso, como

no já mencionados estudos sobre o trânsito de Colónia [3] e Pequim [4]. Alternativamente, a agregação pode não ser feita nas linhas dos trajetos, mas sim numa estrutura própria, como a rosa dos ventos, desenhada acima do mapa e dos trajetos.

Um estudo comparativo de aplicações de visualização de trajetos [14] confirma que as técnicas mencionadas nos parágrafos anteriores são comuns, acrescentando mapas animados, cartogramas, e representação de trajetos sobre um globo terrestre em 3D. A maior parte das visualizações serve de suporte à análise de dados estatísticos, com pouca vertente de interação com o utilizador para, por exemplo, serem aplicados filtros de tempo, ou para considerar elementos específicos dos trajetos, como sejam a sua origem e destino, a distância percorrida, o tempo total decorrido, e outros. São precisamente alguns destes elementos, bem como os filtros temporais, que procurámos integrar no sistema que descrevemos na secção seguinte.

III. DESCRIÇÃO DO SISTEMA

Organizámos a descrição do sistema de análise de trajetos com rosas dos ventos em três partes: primeiro apresentamos as principais componentes da arquitetura; depois mostramos uma rosa dos ventos para trajetos gerada sobre um ponto num mapa, contendo filtros acessíveis ao utilizador; e, por fim, abordamos algumas otimizações de desempenho.

A. Arquitetura

O sistema possui quatro componentes principais: 1) serviços *web* programados em PHP; 2) uma aplicação *web* escrita em JavaScript que permite a colocação de rosas dos ventos sobre um mapa com trajetos; 3) uma biblioteca com a estrutura de dados e funções das rosas dos ventos para trajetos, em JavaScript; e 4) uma base de dados geográfica, em PostGIS [15].

Os serviços *web* servem de apoio à aplicação *web*, oferecendo funcionalidades tais como leitura de trajetos de ficheiros GPX, carregamento de dados de trajetos para a base de dados geográfica, e consulta dos mesmos. O formato GPX (*GPS Exchange Format*) é muito usado para registar trajetos, tornando o sistema compatível com um vasto leque de dispositivos GPS.

Um dos serviços *web* transforma dados de trajetos em formato GPX para JSON (*JavaScript Object Notation*), permitindo um processamento mais eficiente com a linguagem JavaScript. Os trajetos em JSON podem ser enviados para a base de dados geográfica usando um segundo serviço, sendo este invocado aquando do arranque da aplicação *web*.

Existe também um serviço *web* responsável por criar uma sessão partilhada por uma página de seleção preliminar de trajetos contidos num ficheiro GPX e a aplicação *web* de análise de trajetos. Após o envio de um ficheiro GPX, são listadas as suas *tracks*, com nome e datas de início e fim dos pontos registados, e é mostrado um mapa com as *bounding boxes* de cada trajeto selecionado. A seleção preliminar de trajetos está separada da aplicação de análise, podendo vir a ser integrada nesta.

A aplicação *web* de análise de trajetos é a principal componente do sistema do ponto de vista do utilizador, comportando-se como um cliente da base de dados geográfica e da biblioteca de rosas dos ventos para trajetos. A janela da aplicação é dominada por um mapa com os trajetos previamente selecionados, podendo o utilizador escolher pontos onde pretende criar rosas

para agregar e analisar os trajetos próximos desses pontos. A criação de uma rosa é feita com um clique no botão direito do rato num ponto do mapa, sendo desenhado um marcador e um círculo com a área de agregação. Inicialmente, o raio de agregação é de 100 metros, podendo ser alterado posteriormente.

Quando é colocado um marcador no mapa, a aplicação *web* inicia a descoberta dos trajetos que interseitam o círculo da área de agregação da rosa, tirando partido de funções do PostGIS. Assim, usámos *ST_MakePoint* para guardar pontos de trajetos e *ST_MakeLine* para os trajetos em si. A função *ST_Buffer* serviu para criar um círculo com um dado centro e raio, sendo estes parâmetros idênticos aos usados para desenhar o círculo no mapa da aplicação *web*. Por fim, recorreremos à função *ST_Intersection* para obter o primeiro e último ponto da interseção de cada trajeto com um círculo, úteis para saber em que pétala da rosa cada trajeto deve constar.

Obtidas as interseções com o círculo, a aplicação *web* usa a biblioteca da rosa dos ventos para desenhar os elementos gráficos de uma rosa, tais como as pétalas e a escala de velocidades, sendo esta colocada junto do respetivo marcador. O objeto JavaScript que representa uma rosa tem acesso aos trajetos que foram carregados para a aplicação *web*, com o intuito de facilitar a geração dos elementos gráficos que a constituem.

A aplicação *web* permite que o utilizador altere aspetos de cada rosa individualmente, como o raio de agregação de trajetos ou a aplicação de filtros de tempo para, por exemplo passar de 100 para 200 metros de raio e considerar apenas trajetos realizados durante a manhã. Contudo, também está previsto o controle de todas as rosas em simultâneo, em especial num mapa que contenha vários locais estratégicos a analisar como um todo, para facilitar a comparação dos dados mostrados nas rosas.

B. Exemplo de rosa dos ventos para trajetos

A Figura 1 mostra, no lado esquerdo, uma rosa dos ventos para um ponto com marcador 1 no mapa do lado direito, onde aparecem trajetos feitos a pé, registados através de um telemóvel com a aplicação PocketEarth [16].

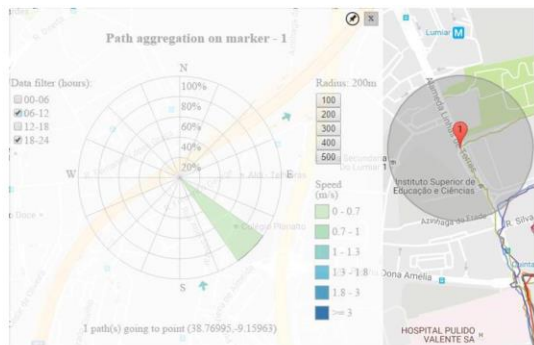


Figura 1. Rosa dos ventos para trajetos em torno de um ponto no mapa

Na rosa da Figura 1 estão agregados três trajetos, que interseitam o círculo de 200 metros de raio em torno do marcador 1. Este raio pode ser modificado com os botões à direita da rosa, compreendendo valores pré-definidos entre 100 e 500 metros.

Neste caso, verifica-se que apenas um dos três trajetos está representado na rosa, com uma pétala a verde, em consequência da aplicação de um filtro de tempo no lado esquerdo da rosa, que determinou a seleção só dos trajetos que tenham sido registados das 6h às 12h e das 18h às 24h.

Na fronteira exterior da zona das pétalas da rosa são colocadas setas de origem e destino de cada trajeto, sendo que, neste caso, apenas aparece um par de setas, uma vez que existe um único trajeto a ser mostrado na rosa. A orientação da seta de origem é definida pela direção que aponta do início de um trajeto para o marcador da rosa no mapa, permitindo perceber que o trajeto teve início a sul. A orientação da seta de destino é dada pela direção do último ponto do trajeto e o marcador, sendo, neste caso, nordeste.

No que toca às pétalas da rosa, existe uma escala percentual que mostra o número relativo de trajetos correspondentes a cada pétala. No caso da Figura 1, visto que está escolhido apenas um trajeto, o valor máximo da escala é 100%, equivalente à proporção de um trajeto nessa pétala de um total de um trajeto.

Por último, à direita das pétalas existe uma escala colorida para mostrar gamas de velocidades, sendo usada tanto para as pétalas como para as setas de origem e destino, embora com significados distintos. No exemplo da Figura 1, um decisor pode perceber que dentro da área de agregação da rosa, isto é, no raio de 200 metros em torno do marcador 1 no mapa, a velocidade média foi de 0 a 0.7 m/s (cerca de 3 Km/h, com pétala de cor verde claro), tendo sido inferior, nesse troço, à velocidade média do trajeto completo, entre 1 e 1.3 m/s (aproximadamente 4.5 Km/h), mostrada pela cor mais azulada das setas de origem e destino. Com esta dupla codificação de velocidades médias, um decisor pode rapidamente perceber se em torno de um ponto no mapa existe uma grande diferença de velocidades face aos trajetos completos, bastando comparar as cores dominantes das pétalas e das setas de origem e destino.

C. Otimizações de desempenho

O sistema de análise de trajetos com rosas dos ventos tem um protótipo funcional, usado no exemplo da Figura 1, estando em desenvolvimento um segundo protótipo que contempla algumas melhorias e otimizações que descrevemos a seguir.

A decisão de passar a ser utilizada uma base de dados geográfica PostGIS permitiu a invocação de funções predefinidas que melhoraram substancialmente a descoberta dos trajetos que cruzam a área de agregação da rosa, bem como dos pontos de interseção dos trajetos com a periferia dessa área.

Existem ainda colunas com valores pré-calculados na base de dados de trajetos, como por exemplo as coordenadas da *bounding box* de cada trajeto, que permitem excluir, à partida, trajetos que não sejam possíveis candidatos de interseção com o raio de agregação da rosa. Outro exemplo de colunas derivadas é o das datas de início e fim de cada trajeto, obtida a partir do mínimo e máximo das datas nos respetivos pontos, que tornam mais eficiente a aplicação de filtros de tempo.

Outra alteração face ao primeiro protótipo do sistema, motivada em parte pela necessidade de acesso à base de dados geográfica, teve que ver com o desenvolvimento de serviços *web*,

permitindo também retirar parte das responsabilidades da aplicação *web*, facilitando o seu desenvolvimento e permitindo o acesso a um leque mais alargado de funções geográficas.

IV. AVALIAÇÃO COM UTILIZADORES

O primeiro protótipo do sistema foi avaliado por 12 mulheres e 7 homens, com idades entre os 15 e os 65 anos (mediana de 42 anos), tendo sido integrado na aplicação *web* um gerador de cenários de análise de trajetos com rosas dos ventos. Em cada cenário foi colocada uma de cinco questões ao utilizador, descritas mais à frente. As respostas foram dadas posicionando o cursor com o rato e fazendo clique com o botão esquerdo. Para cada resposta, foi registado o tempo e o número de erros.

Cada sessão de teste decorreu numa sala fechada, sem interferências externas, com a aplicação *web* a ser executada num computador portátil com rato. Foi explicado o propósito do teste, bem como os conceitos em estudo, tendo os utilizadores tido tempo para explorar a aplicação de forma livre antes de responderem às questões.

As questões colocadas foram as seguintes, cada uma com uma abreviatura identificativa: perante três marcadores no mapa, faça clique no marcador correspondente à rosa dos ventos mostrada (MAROS); faça clique na pétala que agrega mais trajetos (PMAXT); faça clique na pétala que agrega trajetos com velocidades médias superiores às das outras pétalas (PMAXV); e faça clique na seta de origem/destino do trajeto com maior velocidade média (OMAXV/DMAXV). Cada uma destas questões foi apresentada aos utilizadores por ordem aleatória, e registaram-se duas respostas para cada.

Aplicámos o teste estatístico de Shapiro-Wilk aos tempos e erros das respostas às questões, tendo este revelado que os dados não seguem uma distribuição normal ($p \leq 3.387 \times 10^{-4}$), pelo que nas comparações seguintes foi usado o teste não-paramétrico de Wilcoxon para duas amostras.

A. Tempos de resposta às questões

As medianas dos tempos de resposta às questões variaram entre 15.4 segundos em PMAXT e 22.9 segundos em OMAXV. DMAXV teve mediana de 22.3 segundos, que não diferiu significativamente da de OMAXV ($V=428$, $p=0.408$). Em MAROS a mediana foi de 17.6 segundos e em PMAXV foi de 18.5 segundos. As únicas diferenças significativas de tempos de resposta existiram entre PMAXT e OMAXV/DMAXV ($V=174$, $p=0.003$ e $p=0.023$, respetivamente).

Relativamente a eventuais efeitos de aprendizagem entre a primeira e a segunda vez que os utilizadores responderam a cada questão, apenas num caso encontramos uma diferença significativa: em PMAXV a mediana na primeira resposta foi de 22.1 segundos enquanto na segunda resposta foi de 13.1 segundos ($V=163$, $p=0.005$). Nos restantes casos, $p \geq 0.332$.

Também analisámos possíveis efeitos do género dos utilizadores nos tempos de resposta, com os resultados mostrados na Figura 2. Em nenhuma questão foi encontrada uma diferença significativa entre homens e mulheres ($p \geq 0.105$).

Por fim, comparámos os tempos de resposta em função de três escalões etários, 15-29, 30-44, e 45-65, tendo os resultados mostrado que não houve diferenças significativas ($p \geq 0.084$).

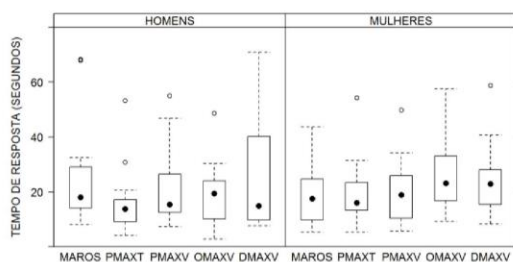


Figura 2. Tempos de resposta às questões por género de utilizador

B. Erros nas respostas às questões

Os utilizadores acertaram quase sempre nas respostas às questões, tendo a mediana dos erros sido sempre zero. Naturalmente, não existiram diferenças significativas entre a primeira e a segunda vez que responderam a cada questão ($p \geq 0.705$), nem entre homens e mulheres ($p \geq 0.184$), nem entre escalões etários ($p \geq 0.084$).

V. DISCUSSÃO E LIMITAÇÕES

Nesta secção discutimos os principais resultados obtidos na avaliação com utilizadores e descrevemos algumas limitações atuais do sistema de análise de trajetos com rosas dos ventos.

A. Discussão dos resultados da avaliação

A quase ausência de erros nas respostas às questões pode levar a crer que estas eram demasiado fáceis. No entanto, correspondem a interpretações de dados que consideramos plausíveis e essenciais para a utilização de uma rosa dos ventos adaptada à análise de trajetos: perante uma rosa, foi bem identificado o correspondente marcador num mapa (MAROS), exigindo a comparação de dados em vários níveis de detalhe (agregados na rosa e detalhados no mapa); e, focando na rosa em si, as pétalas foram escolhidas corretamente quanto ao número de trajetos (PMAXT) e velocidades (PMAXV), e as velocidades também foram bem percebidas nas setas de origem e destino (OMAXV/DMAXV). Assim, julgamos que os elementos gráficos da rosa são fáceis de entender e que a introdução de setas em redor das pétalas não levou a dificuldades extra.

Relativamente aos tempos de resposta, começamos por mencionar que a interpretação das velocidades médias nas setas de origem foi semelhante às de destino (OMAXV/DMAXV), o que seria desejável. As respostas mais rápidas foram na identificação da pétala com mais trajetos agregados (PMAXT), ou seja, da maior pétala de todas, o que intuitivamente é mais fácil de fazer do que analisar regiões coloridas dentro das pétalas (PMAXV) ou, em particular devido às diferenças significativas obtidas, setas coloridas (OMAXV e DMAXV). A escala de cores pode ter afetado os resultados, não tendo havido oportunidade para testar com outras. A diferença entre os tempos da primeira e segunda resposta em PMAXV foram uma surpresa indesejada, que julgamos dever-se a um efeito de aprendizagem que não conseguimos detetar e controlar atempadamente. Por fim, a ausência de diferenças entre homens e mulheres e entre pessoas dos vários escalões etários, juntamente com o reduzido número de erros cometidos, sugerem que as rosas dos ventos são adequadas para a análise de trajetos em mapas.

B. Limitações do sistema

O protótipo atual do sistema não tem ainda implementada a funcionalidade de aplicação de filtros a várias rosas em simultâneo num mesmo mapa. A interface destes filtros estará num painel específico da aplicação *web*, estando previsto o envio de eventos, tais como só trajetos de manhã, a serem processados por *listeners* associados a cada rosa. Desta forma, seria facilitada a comparação das várias rosas, pois todas mostrariam trajetos agregados em condições semelhantes, e seria permitida uma análise holística de vários locais importantes num mapa. Atualmente, só podem ser aplicados filtros a uma rosa de cada vez. Este modo confere uma maior liberdade ao decisor que utiliza o sistema, embora possa trazer alguns problemas de coerência entre as várias rosas mostradas.

Outra limitação que pretendemos ultrapassar está relacionada com a impossibilidade de filtrar trajetos em função da distância desde a origem destes até que seja alcançada a interseção com a área de agregação de uma rosa. Neste caso, pretendemos criar uma interface que permita definir a distância mínima e máxima dos trajetos a incluir na agregação. Esta ideia pode ser estendida à duração e velocidade dos trajetos, conferindo uma maior capacidade de análise a um decisor. A título de exemplo, pode ser útil para o planeamento de estradas perceber que trajetos vindos da periferia passam por um centro urbano.

VI. CONCLUSÕES E TRABALHO FUTURO

Neste artigo apresentámos um sistema que permite a análise de trajetos em mapas recorrendo a gráficos inspirados em rosas dos ventos. Este tipo de gráfico é muito usado para representar ventos num dado local, sendo nova a sua aplicação a trajetos. O utilizador pode visualizar as direções e velocidades dos trajetos que passam em redor de um local, bem como interagir com as rosas definindo filtros de tempo ou ajustando o raio de agregação, podendo colocar várias num mesmo mapa.

O sistema foi avaliado por utilizadores de várias faixas etárias tendo sido registada um reduzido número de erros nas respostas a questões de interpretação dos dados mostrados na rosa dos ventos adaptada à análise de trajetos. Os tempos de resposta revelaram algumas diferenças em questões sobre aspetos específicos da rosa, podendo ser pertinente redesenhar elementos gráficos tais como a escala de cores usada nas velocidades.

No futuro próximo, pretendemos aumentar as funcionalidades do sistema, incluindo novos filtros de agregação de trajetos, como é o caso das distâncias mínima e máxima até ao local de colocação da rosa, bem como a possibilidade de controlar de

forma síncrona várias rosas dos ventos para facilitar a comparação dos trajetos agregados em diferentes locais num mapa.

AGRADECIMENTOS

Agradecemos o apoio material da unidade de I&D LaSIGE, ref. UID/CEC/00408/2013. Agradecemos também aos voluntários que participaram na avaliação do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] M. J. Eppler and J. Mengis, "The concept of information overload: A review of literature," *Inf. Soc.*, vol. 20, no. 5, pp. 325–344, 2004.
- [2] J. Thomas and J. Kielman, "Challenges for visual analytics," *Inf. Vis.*, vol. 8, no. 4, pp. 309–314, 2009.
- [3] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Trans. Mob. Comput.*, vol. 13, no. 5, pp. 1061–1075, 2014.
- [4] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-Drive: Driving directions based on taxi trajectories," in *Proc. of the 18th SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (GIS '10)*, 2010, pp. 99–108.
- [5] Meteoblue, "Climate Lisbon Airport." Disponível: https://www.meteoblue.com/en/weather/forecast/modelclimate/lisboa-%252F-portela_portugal_6301792. [Acedido: 15-Fev-2017].
- [6] H. L. Crutcher, "On the standard vector-deviation wind rose," *J. Meteorol.*, vol. 14, no. 1, pp. 28–33, 1957.
- [7] Lakes Environmental, "WRPLOT View." Disponível: <https://www.weblakes.com/products/wrplot>. [Acedido: 15-Fev-2017].
- [8] R. Conceição, M. Melgão, H. G. Silva, K. Nicoll, R. G. Harrison, and A. H. Reis, "Transport of the smoke plume from Chiado's fire in Lisbon (Portugal) sensed by atmospheric electric field measurements," *Air Qual. Atmos. Heal.*, vol. 9, no. 3, pp. 275–283, 2015.
- [9] J. W. Winchester and G. D. Nifong, "Water pollution in Lake Michigan by trace elements from pollution aerosol fallout," *Water. Air. Soil Pollut.*, vol. 1, no. 1, pp. 50–64, 1971.
- [10] X. Jia, D. Chung, J. Huang, M. Petrilli, and L. The, "ARO: Geographic Information Systems-Based System for Optimizing Airport Runway Orientation," *J. Transp. Eng.*, vol. 130, no. 5, pp. 555–559, 2004.
- [11] N. M. Al-Abbad, "Wind energy resource assessment for five locations in Saudi Arabia," *Renew. Energy*, vol. 30, no. 10, pp. 1489–1499, 2005.
- [12] Microsoft, "Bing map shapes: Polylines and polygons." Disponível: <https://msdn.microsoft.com/en-us/library/mt712696.aspx>. [Acedido: 15-Fev-2017].
- [13] Google, "Google Maps JavaScript API: Simple polylines." Disponível: <https://developers.google.com/maps/documentation/javascript/examples/polyline-simple>. [Acedido: 15-Fev-2017].
- [14] J. R. Harger and P. J. Crossno, "Comparison of open-source visual analytics toolkits," in *Proc. of SPIE*, 2012, vol. 8294, no. 1.
- [15] "PostGIS: Spatial and geographic objects for PostgreSQL." Disponível: <http://postgis.net/documentation>. [Acedido: 15-Fev-2017].
- [16] Geomagik, "PocketEarth: Offline maps and travel guides for iPhone and iPad." Disponível: <https://pocketearth.com>. [Acedido: 15-Fev-2017].

Apêndice B

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 |
| outerCircleRadius | | 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 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 C

Questionário inicial dos testes

Questionário inicial

Este questionário destina-se a cada utilizador que efetuar a avaliação da aplicação de visualização analítica de trajetos recorrendo a rosas dos ventos. Todos os dados pessoais recolhidos serão utilizados meramente para fins estatísticos.

*Obrigatório

Identificador de utilizador: *

Sua resposta

Sexo: *

Masculino

Feminino

Idade: *

Sua resposta

Está habituado a visualizar trajetos sobre mapas? *

Sim

Não

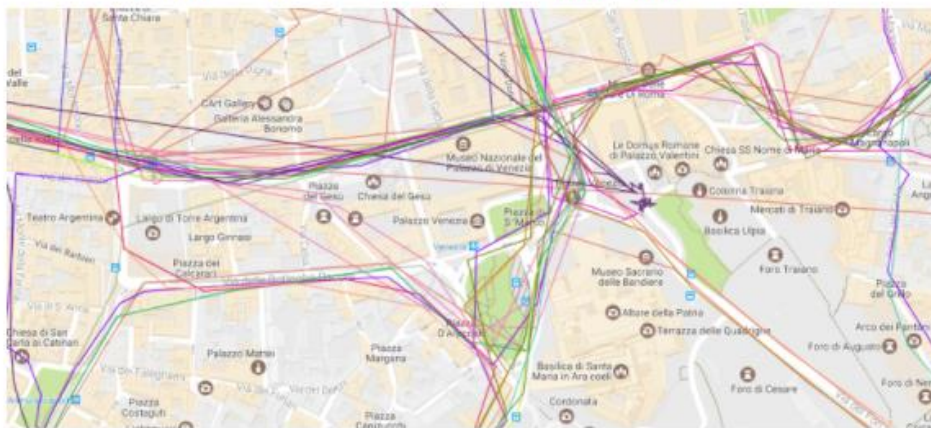
Se sim, quais as aplicações que utiliza?

Sua resposta

Considerando a análise de trajetos, que atributos considera importantes? *

- Velocidade
- Distância
- Tempo decorrido
- Direção de origem/destino
- Período de ocorrência
- Outro: _____

Classifique a existência, ou não, de excesso de informação se tivesse de analisar os trajetos da seguinte figura *



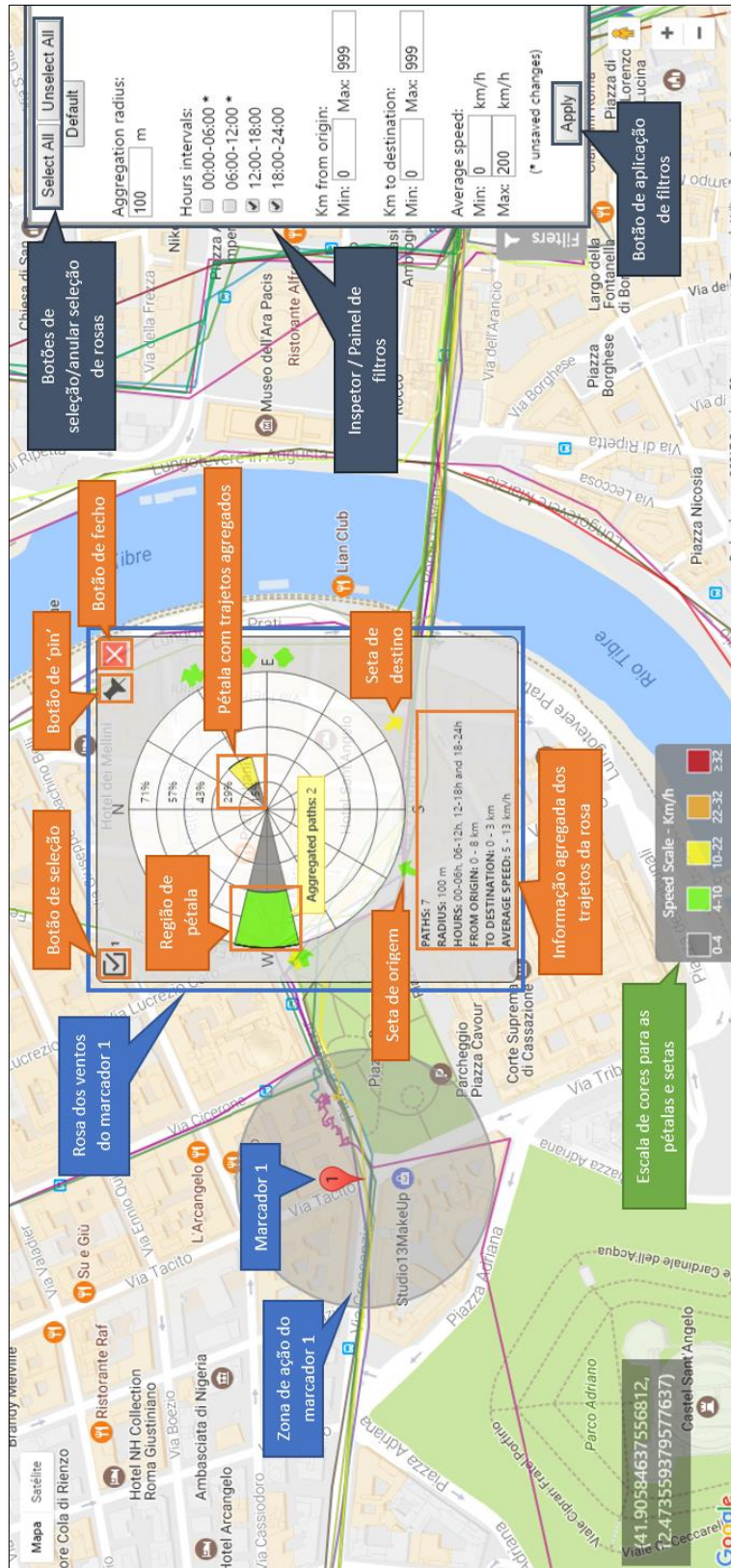
- 1 - Informação razoável e gerível
- 2 - Volume considerável de informação e pouco gerível
- 3 - Excesso de informação e dificilmente gerível

ENVIAR

Nunca envie senhas pelo Formulários Google.

Apêndice D

Guia rápido da aplicação web *windroseMain*



Apêndice E

Questionário de usabilidade (SUS)

Questionário de Usabilidade (SUS)

Este é um questionário anônimo que tem a finalidade de inquirir o utilizador sobre a usabilidade da aplicação de visualização de dados de trajetos agregados, recorrendo a 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

Identificador de utilizador: *

Sua resposta

Gostaria de utilizar esta aplicação com frequência *

| | 1 | 2 | 3 | 4 | 5 | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

Achei a aplicação desnecessariamente complexa *

| | 1 | 2 | 3 | 4 | 5 | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

Achei a aplicação fácil de se usar *

| | | | | | | |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

Precisaria da ajuda de uma pessoa com conhecimentos técnicos para utilizar a aplicação *

| | | | | | | |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

As várias funcionalidades da aplicação estão muito bem integradas *

| | | | | | | |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

A aplicação apresenta um elevado grau de incoerência *

| | | | | | | |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

Acredito que a maioria das pessoas facilmente aprenderão a usar a aplicação *

| | | | | | | |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

Achei a aplicação muito estranha de se usar *

| | | | | | | |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

Senti-me muito confiante ao usar a aplicação *

| | 1 | 2 | 3 | 4 | 5 | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

Precisei de aprender muitas coisas antes de conseguir usar a aplicação *

| | 1 | 2 | 3 | 4 | 5 | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|
| Discordo Totalmente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Concordo Totalmente |

Sentiu alguma dificuldade ao usar a aplicação? *

- Sim
- Não

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

Sua resposta

Sugestões para melhorar a aplicação:

Sua resposta

ENVIAR

Nunca envie senhas pelo Formulários Google.