

Universidade do Minho

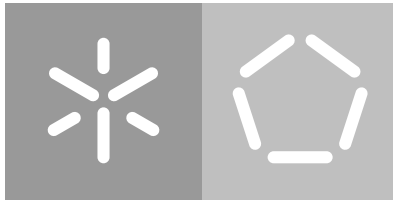
Escola de Engenharia

Departamento de Informática

Stéphane Alexandre Alves Fernandes

**Sistema de Recomendação de Rotas
em Cidades Inteligentes**

December 2018



Universidade do Minho

Escola de Engenharia

Departamento de Informática

Stéphane Alexandre Alves Fernandes

Sistema de Recomendação de Rotas em Cidades Inteligentes

Master dissertation

Master Degree in Computer Science

Dissertation supervised by

Paulo M. Martins de Carvalho

Maria Solange P.F. Rito Lima

December 2018

AGRADECIMENTOS

Gostaria de expressar os meus agradecimentos a todas as pessoas, que contribuíram para a realização e conclusão desta dissertação. No entanto, gostaria de um deixar um agradecimento especial às seguintes pessoas.

Ao meu orientador Professor Paulo Carvalho e à minha coorientadora Professora Solange Lima, pela orientação, disponibilidade e ajuda fornecida durante a realização desta dissertação.

À minha família, nomeadamente aos meus pais, pelas oportunidades que me proporcionam ao longo de todo o meu percurso académico, pela confiança e compreensão depositada em mim durante a licenciatura e o mestrado.

Por fim, também gostaria de deixar os meus agradecimentos a todos os meus amigos que me acompanharam ao longo do percurso académico.

ABSTRACT

Nowadays urban areas suffer from overpopulation so it is essential that their resources are managed in the best way, with the purpose of providing quality of life for the population and providing a sustainable development. In this context, the concept of Smart Cities emerges, focusing on the use of new technologies, whether in transport, health or even in the environment domain, opens the opportunity for improving the quality of life of urban dwellers.

With this work we intend to develop a route recommendation system oriented to assist citizens based on a specific objective function, such as the definition of a sports activity route. A prototype of the system was built and tested using a simulation tool called CupCarbon. Thus, as a case study, it is considered an urban zone where a network of wireless sensors are defined for monitoring environmental parameters such as luminosity or Wi-Fi coverage, in order to assist the planning of routes according to user preferred parameters.

RESUMO

Hoje em dia as zonas urbanas sofrem de sobrepopulação, por isso, é essencial que os seus recursos sejam geridos da melhor forma com o intuito de proporcionar uma melhor qualidade de vida aos seus habitantes bem como, um desenvolvimento sustentável. Neste contexto, surge o conceito de Cidades Inteligentes que é focado no uso de novas tecnologias quer ao nível dos transportes, da saúde ou mesmo relacionadas com o meio ambiente, dando a oportunidade de melhorar a qualidade de vida dos habitantes das zonas urbanas.

Com este trabalho, pretende-se desenvolver um sistema de recomendação de rotas para auxiliar cidadãos com base numa função objetivo específica, como a definição de um percurso de atividade desportiva. Um protótipo do sistema foi construído e testado usando uma ferramenta de simulação chamada CupCarbon. Assim, como caso de estudo, considera-se uma zona urbana onde é definida uma rede de sensores sem fios para monitorizar parâmetros ambientais como luminosidade ou cobertura Wi-Fi, com o intuito de auxiliar o planeamento de rotas de acordo com os parâmetros pretendidos pelo utilizador.

CONTEÚDO

Lista de Figuras	vi
Lista de Tabelas	viii
Lista de Código-Fonte	ix
Lista de Acrónimos	x
1 INTRODUÇÃO	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Metodologia de Investigação	2
1.4 Estrutura do documento	3
2 ESTADO DE ARTE	4
2.1 Revisão bibliográfica	4
2.2 Trabalho Relacionado	6
2.2.1 PedestrianPal	6
2.2.2 Google Maps	7
2.2.3 Komoot	7
2.3 Tecnologias	8
2.3.1 Localização dos pedestres	8
2.3.2 Sistemas de recomendação	9
2.3.3 Cálculo do caminho mais curto	10
2.4 Ferramenta de simulação	13
2.4.1 CupCarbon	13
2.4.2 UrbanSim	15
2.4.3 InterSCSimulator	16
2.4.4 Escolha da Ferramenta de Simulação	17
2.5 Resumo do capítulo	19
3 ABORDAGEM AO PROBLEMA	20
3.1 Abordagem ao problema	20
3.1.1 Arquitetura do sistema	21
3.2 Arquitetura da solução	22
3.2.1 Tipos de Utilizadores	23
3.2.2 Tipos de Critérios de Escolha	23
3.2.3 Tipos de percursos	25
3.2.4 Requisitos de Infra-estrutura	26

3.2.5	Factos e Pressupostos	27
3.2.6	Contexto de utilização	28
3.2.7	Requisitos funcionais	28
3.2.8	Requisitos não funcionais	30
3.2.9	Modelo de Domínio	31
3.2.10	Limitações da Plataforma de Simulação	31
3.3	Resumo do capítulo	32
4	DESENVOLVIMENTO	33
4.1	Implementação	33
4.1.1	Sensores disponíveis no CupCarbon	33
4.1.2	Criação de rotas	35
4.2	Cenário 1	36
4.2.1	Resultados da simulação	39
4.3	Cenário 2	41
4.3.1	Simulação do cenário	43
4.4	Resumo do Capítulo	46
5	CASOS DE ESTUDO - CENÁRIO FINAL	47
5.1	Atributos simulados	47
5.1.1	Atributos variáveis	47
5.1.2	Atributos Fixos	49
5.2	Perfis de Utilizador	51
5.2.1	Perfil A - Local 1	51
5.2.2	Sensores utilizados no cenário	52
5.2.3	Perfil B - Local 2	60
5.2.4	Perfil C - Local 3	64
5.3	Discussão da solução	68
5.3.1	Dificuldades encontradas	69
5.3.2	Infraestruturas de apoio à implementação	70
5.3.3	Implementação do Modelo	70
5.4	Resumo do capítulo	71
6	CONCLUSÃO	72
6.1	Principais conclusões e contribuições	72
6.2	Tópicos de trabalho futuro	74
A	MATERIAL DE SUPORTE	79
A.1	Manual de utilização	79
A.2	Código-fonte da script Receiver utilizada no cenário final	79
A.3	Código-fonte da script Central utilizada no cenário final	80

LISTA DE FIGURAS

2.1	Uso do Google Maps para gerar uma rota entre 2 pontos para pedestres	7
2.2	Exemplificação do uso da ferramenta Komoot para a geração de rotas para pedestres	8
2.3	Algoritmo Dijkstra	11
2.4	Algoritmo Bellman-Ford - arestas a vermelho equivalem a peso negativo	12
2.5	Algoritmo Bidirecional	12
2.6	Exemplo de implementação de uma rede de sensores	14
2.7	Interface utilizador da ferramenta UrbanSim	16
2.8	Execução do simulador para a cidade de São Paulo	17
3.1	Base de uma solução para o problema usando o estudo de Huang e Gartner	20
3.2	Arquitetura da solução para o problema	22
3.3	Diagrama de percursos	26
3.4	Domínio de trabalho	28
3.5	Modelo de domínio	31
4.1	Sensores disponíveis no CupCarbon	34
4.2	Simulação do envio de mensagens	35
4.3	Script valores da Temperatura	35
4.4	Rota sem marcadores	35
4.5	Rota com marcadores	35
4.6	Simulação do movimento de um nodo móvel ao longo de uma rota	36
4.7	Rota principal (rota ₁)	36
4.8	Rota alternativa (rota ₂)	37
4.9	Comunicação ao sensor mobile da rota recomendada (rota ₁)	40
4.10	Deslocamento do sensor mobile pela rota anteriormente recomendada	40
4.11	Cenário 2	42
4.12	Na imagem da esquerda o sensor mobile recebe a indicação para seguir pela rota ₁ , na imagem da direita o sensor mobile já a meio do caminho na rota ₁	45

4.13	O sensor <i>mobile</i> recebe a indicação para seguir pela rota2 (esquerda). O sensor <i>mobile</i> já se encontra a percorrer a rota2 (direita)	46
5.1	Exemplificação da troca de mensagens entre sensores	50
5.2	Rotas criadas para o Local1 do cenário final	52
5.3	Mapa do primeiro local do cenário com os sensores utilizados	53
5.4	Início do deslocamento do sensor móvel ao longo da rota 1	58
5.5	Envio da rota recomendada para o sensor mobile	59
5.6	Deslocamento do sensor móvel ao longo da rota anteriormente recomendada	59
5.7	Rotas criadas para o segundo local do cenário	60
5.8	Mapa do segundo local do cenário com os sensores utilizados	61
5.9	Deslocamento do sensor móvel pela rota 4 (à esquerda), recebendo a mensagem para seguir pela rota 5 (à direita)	63
5.10	Deslocamento do sensor móvel ao longo da rota5	63
5.11	Deslocamento do sensor móvel ao longo da rota5	64
5.12	Mapa do terceiro local do cenário com os sensores utilizados	65
5.13	Início do deslocamento do sensor móvel ao longo da rota6	67
5.14	Envio da mensagem ao sensor móvel para prosseguir pela rota 6	67
5.15	Deslocamento do sensor móvel ao longo da rota recomendada	68

LISTA DE TABELAS

3.1	Critérios de escolha do utilizador	24
3.2	Atributos inerentes à cidade	24
4.1	Exemplo dos valores gerados na simulação do Cenário 2	41
5.1	Dados relativos ao perfil A	57
5.2	Dados relativos ao Perfil B	61
5.3	Dados relativos ao Perfil C	66

LISTA DE CÓDIGO-FONTE

4.1	Código-fonte do sensor mobile	37
4.2	Código-fonte do Sensor1	38
4.3	Código-fonte do Sensor2	39
4.4	Código-fonte do Sensor_router1	43
5.1	Código-fonte do Sensor Transmissor	54
5.2	Código-fonte do Sensor Mobile	56

LISTA DE ACRÓNIMOS

IoT Internet of things

VANT Veículo aéreo não tripulado

GPS Global positioning system

csv Comma-separated values

Sim-Diasca Simulation of Discrete Systems of All Scales

xml Extensible Markup Language

evt Event Viewer Log File

INTRODUÇÃO

Com a constante migração humana das zonas rurais para as zonas urbanas, surgiu a necessidade das zonas urbanas se transformarem de modo a atender o crescente aumento da população, prestando ao mesmo tempo a qualidade de vida que as pessoas esperam obter ao fazer estas migrações [1].

Atendendo à situação presente e tendências futuras, é necessário que as cidades se reinventem na forma como são geridas e os seus recursos são utilizados pela população. Para isso, é fundamental o uso de novas tecnologias no sentido de modernizar os serviços prestados e garantir o seu desenvolvimento sustentado. Neste sentido, surgiu o conceito das Cidades Inteligentes (*Smart Cities*) que tem o objetivo de melhorar e tornar mais eficientes as infraestruturas urbanas e melhorar a prestação dos serviços aos seus habitantes, que são cada vez mais exigentes, e com isso melhorar a sua qualidade de vida.

Na Europa, as cidades de Barcelona, Amesterdão, Londres e Estocolmo são as que mais se destacam em termos de projetos e soluções de *Smart Cities* [2], apresentando vários projetos em áreas tais como mobilidade e transportes, pois são essenciais para o funcionamento de uma cidade, bem como nas áreas da saúde, das infraestruturas, tecnologias e questões ambientais, com o intuito de melhorar a sustentabilidade da cidade.

Mais concretamente, na cidade de Braga foram recentemente instalados nove sensores, com o intuito de monitorizar em tempo real dados recolhidos por sensores relativamente aos índices de poluição do ar, do som e das águas fluviais [3]. Sabendo da importância para a cidade em investir em iniciativas relacionadas com as *Smart Cities*, pretende-se com este trabalho contribuir com uma solução que aporte informação útil ao cidadãos quando estes se deslocam na cidade.

1.1 MOTIVAÇÃO

Segundo estudos recentes em 2050 cerca de 70% da população irá viver nas cidades [4]. Desta forma, é preciso saber gerir de uma forma sustentável as cidades tendo em atenção a qualidade de vida dos habitantes. Os níveis de poluição são elevados, existe um excesso de utilização de veículos particulares, tendo em vista um descongestionamento das cidades e

a promoção da saúde através das caminhadas é importante estimular as viagens a pé como um meio de transporte, dando a devida atenção à escolha de rotas.

Assim, surgiu a ideia de explorar novas tecnologias baseadas em redes de sensores sem fios para monitorizar certos aspetos tais como, os índices de luminosidade ou de cobertura Wi-Fi, por forma a assistir a definição de rotas numa cidade. Dependendo de cada função objetivo, uma pessoa ao deslocar-se na cidade poderia traçar uma rota com base nos aspetos pretendidos. Por exemplo, se for praticar exercício físico à noite seria útil saber se o local onde planeia ir tem uma boa luminosidade ou se existe um percurso melhor iluminado, por forma a garantir uma maior segurança. Caso pretenda usar aplicações como o Spotify enquanto corre, um dos aspetos a ter em conta na escolha da rota, seria o fato de esta apresentar uma boa cobertura Wi-Fi.

1.2 OBJETIVOS

O objetivo principal deste trabalho foca-se no estudo de novos serviços no contexto das *Smart Cities*. Em particular, pretende-se estudar e testar um modelo de recomendação de percursos com base em diferentes funções objetivo. Assim, o trabalho passa pela simulação num ambiente experimental da geração de rotas tendo por base a satisfação do cidadão face a aspetos tais como luminosidade ou cobertura Wi-Fi adequadas. Para além do objetivo principal, pretende-se cumprir os seguintes objetivos parciais:

- levantamento do estado da arte relacionado com o tema do trabalho;
- estudo de ferramentas de simulação e uma comparação entre elas resultando na escolha da plataforma de simulação mais adequada, pretende-se que esta seja vocacionada para o desenho, desenvolvimento e teste de cenários típicos de cidades inteligentes;
- desenho da arquitetura da solução;
- implementação da solução e teste do conceito;

1.3 METODOLOGIA DE INVESTIGAÇÃO

Relativamente a metodologia de investigação, para a realização deste trabalho será dividida em duas fases. Na primeira será feito um estudo do estado da arte, tendo como objetivo adquirir um maior conhecimento sobre o domínio do tema proposto para dissertação, e avaliar outros trabalhos de IoT diretamente relacionados para além da escolha da ferramenta de apoio ao estudo.

Na segunda fase, depois de ter um conhecimento prévio do domínio da dissertação, deverá ser concebido um modelo que dará suporte ao serviço pretendido, assente num

levantamento de requisitos em termos da infraestrutura de monitorização necessária. De seguida será realizada uma ambientação à ferramenta de simulação escolhida. Posteriormente, serão planeadas e realizadas simulações tendo como objetivo chegar a uma solução adequada para o problema proposto. Ao longo desta tarefa será feita uma análise dos resultados obtidos, e efetuados eventuais ajustes ao modelo de simulação.

1.4 ESTRUTURA DO DOCUMENTO

No capítulo inicial desta dissertação é feito um estudo sobre o estado da arte, onde é apresentada uma revisão bibliográfica, uma procura de trabalhos relacionados bem como tecnologias já existentes e por fim, a escolha da ferramenta de simulação que será usada como suporte. No capítulo 3, é feita uma primeira abordagem ao problema apresentando-se uma possível arquitetura para o sistema, seguindo-se um levantamento de requisitos e identificação dos utilizadores que poderão usar o sistema.

No quarto capítulo é descrito o desenvolvimento e implementação da solução, no capítulo seguinte é apresentado um cenário final de estudo sendo também feita a análise e discussão da solução implementada. Por fim, no último capítulo é efetuada uma reflexão sobre as conclusões deste trabalho.

ESTADO DE ARTE

Neste capítulo será feito um levantamento do estado de arte relacionado com o tema deste trabalho, sendo também apresentado um estudo das tecnologias usadas em sistemas de recomendação de rotas, bem como, uma análise de ferramentas relacionadas com este trabalho.

2.1 REVISÃO BIBLIOGRÁFICA

Hoje em dia encontramos vários estudos e aplicações relacionados com a recomendação de rotas, sendo que a maioria se encontra relacionada com transportes públicos e viaturas. No caso dos pedestres grande parte dos estudos foca-se no cálculo de rotas eficientes [5] ou em atrações turísticas [6].

Nos últimos anos, têm sido realizados estudos dando mais atenção a outros atributos na altura de escolher a rota entre dois pontos. Estes têm como objetivo modelar a escolha da rota pelos pedestres, tendo em conta que as características do ambiente são um importante atributo para a tomada de decisão. Destacando-se os seguintes estudos:

- Godlledge (1995) defende que os pedestres quando pretendem ir até determinado destino e tendo tempo suficiente optam, por rotas com diversas qualidades como simplicidade, número baixo de cruzamentos e aspetos estéticos como a existência de parques em vez de simplesmente escolher o caminho mais curto [7];
- Hoogendoorn e Bovy (2004) destacam os seguintes critérios [8] para a tomada de decisão relativamente a escolha de uma rota:
 - tempo;
 - distância;
 - mudanças de direções;
 - interações com outros pedestres;
 - abrigo em caso de más condições meteorológicas
 - atratividade (presença de espaços verdes e iluminação).

- N.J. van der Zijpp & Fiorenzo-Catalano (2005) [9] usam a maioria dos critérios também usados por Hoogendoorn e Bovy, adicionando outros critérios:
 - segurança da rota;
 - níveis de poluição;
 - número de cruzamentos.
- MTI REPORT (2006) segundo este relatório os critérios mais relevantes são a distância, mudanças de direção, segurança nomeadamente existirem travessias com semáforos e passeios em boas condições. Outro critério importante é a aparência, tal como, edifícios históricos, presença de parques ou espaços abertos e presença de outros pedestres [10];
- Guo & Loo (2009 e 2013) nos seus estudos destacam critérios tais como largura dos passeios, distância, espaços abertos (tais como parques), distância da rota e volume de veículos [11];
- Czogalla & Herrmann (2010) decidiram dividir os critérios em quatro categorias [12],
 - segurança, realçando aspetos como travessias com semáforos e baixo volume de veículos a circular;
 - acessibilidade, realçando a largura dos passeios e o declive da rota;
 - atratividade, onde se incorporam critérios como espaços abertos e iluminação;
 - conforto, tendo em atenção aos níveis de ruído.

Partindo da análise dos estudos já elaborados é possível observar que os critérios mais usados são:

- a distância;
- mudanças de direção;
- largura dos passeios;
- tempo da viagem;
- presença de espaços abertos;
- iluminação;
- segurança;
- declive.

Para além destes, é importante realçar que o uso de uma rede sem fios (ex. rede Wi-Fi) que cubra a cidade vai contribuir para que esta se torne inteligente, trazendo diversos benefícios direta ou indiretamente. Tais como, redução do congestionamento de tráfego, crescimento económico, maior segurança bem como, a prestação de serviços como estacionamento inteligente, iluminação inteligente, aplicações relacionadas com o ambiente, etc [13]. Tendo isto em conta, um critério interessante seria considerar o grau de cobertura Wi-Fi ao longo de um percurso, sabendo que cada vez mais as pessoas usam os seus dispositivos móveis, podendo desta forma, poupar o consumo de dados móveis a utilizadores que possuam pacotes de dados mais restritos. Por exemplo, pessoas de negócios normalmente tem uma necessidade constante de estarem com cobertura de rede, nesse caso, seria interessante no caso de se deslocarem a pé para o trabalho, a rota escolhida usar como critério a cobertura Wi-Fi, sendo que Braga possui Wi-Fi grátis em certas zonas da cidade [14].

Segundo Huang e Gartner (2010), um serviço móvel de recomendação de rotas pode ser dividido em três partes [15]:

- posicionamento - determinar a posição do utilizador usando para isso coordenadas GPS;
- cálculo da rota - encontrar a melhor rota entre a origem e o destino;
- comunicação da rota - comunicar a informação da rota de maneira direta para o utilizador.

2.2 TRABALHO RELACIONADO

Nesta secção é apresentada uma análise de três ferramentas relacionadas com o tema deste trabalho, PedestrianPal, Google Maps e Komoot. Estas ferramentas permitem gerar e recomendar rotas a pedestres, sendo feito um estudo das suas funcionalidades.

2.2.1 *PedestrianPal*

Em 2011 foi desenvolvido uma aplicação Android para a cidade de Missoula nos Estados Unidos chamada Pedestrian Pal [16]. Esta aplicação oferece recomendação de rotas baseadas em especificações desejadas, sendo que estas não são necessariamente eficientes, mas sim baseadas nas características desejadas pelos utilizadores.

Esta aplicação surgiu com o intuito de prestar auxílio a turistas que visitassem a cidade pela primeira vez, indicando rotas mais seguras para dar um passeio, rotas até um parque para a prática de exercício físico ou rotas locais para simplesmente esticar as pernas.

Já existem diversas aplicações que recomendam a turistas numa nova cidade pontos de interesse como museus ou até mesmo restaurantes, existem também aplicações como o

Google Maps que indicam as rotas mais curtas até ao destino. Contudo, é importante para um turista numa nova cidade saber que, quando se desloca até ao destino, uma rua é mais segura do que outra, por exemplo. Então esta aplicação permite aos seus utilizadores escolherem alguns parâmetros de entrada como a distância da rota, localização na cidade e declive da rota e, no final, o utilizador avalia a rota que lhe foi sugerida.

2.2.2 Google Maps

Existem também ferramentas como o Google Maps [17] que possuem sistemas de recomendações de rotas, baseando-se sobretudo na escolha de rota mais curta ou mais eficiente até ao destino, não tendo em atenção aspetos estéticos ou segurança na altura de recomendar as rotas. Normalmente o Google Maps é mais usado por condutores mas apresenta também a funcionalidade de recomendação de rotas para pedestres (Figura 2.1) ou mesmo ciclistas.

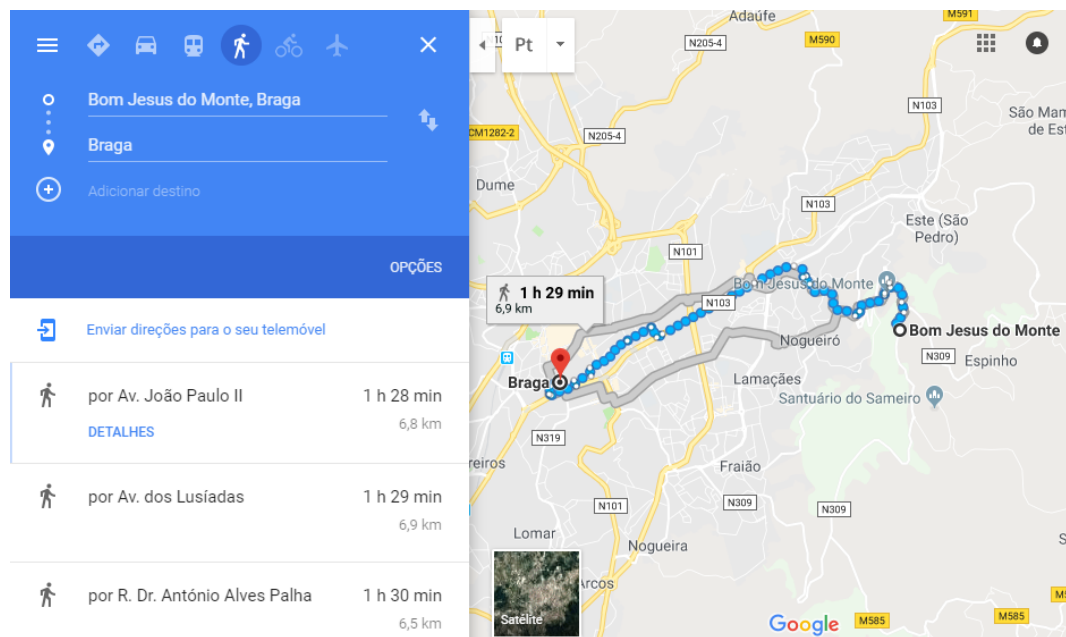


Figura 2.1: Uso do Google Maps para gerar uma rota entre 2 pontos para pedestres

2.2.3 Komoot

É uma aplicação para a prática de desportos ao ar livre usada para pedestres ou para ciclistas [18]. O utilizador insere o ponto de partida e o destino e a aplicação vai gerar a melhor rota apresentando detalhes relativamente às variações de altitude do percurso (Figura 2.2). Permite ainda uma funcionalidade agradável e que vai de acordo ao tema deste trabalho, poder definir o que o utilizador quer ver durante a viagem. É possível, por

exemplo, começar numa estação de comboios, passar por um parque e terminar o percurso num lago.

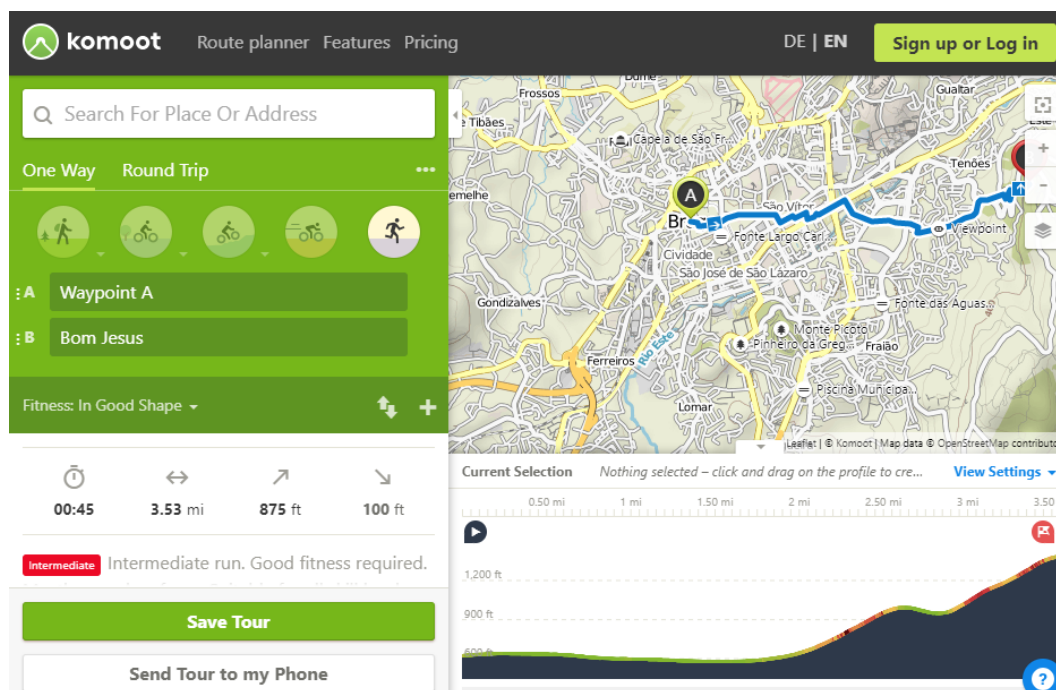


Figura 2.2: Exemplificação do uso da ferramenta Komoot para a geração de rotas para pedestres

2.3 TECNOLOGIAS

2.3.1 Localização dos pedestres

Hoje em dia, é possível obter grandes quantidades de dados relativamente a localização ou trajetórias de indivíduos graças a avanços no uso de sensores, na comunicação sem fios e a popularidade de tecnologias como Wi-Fi e GPS. É importante para uma cidade inteligente fazer uma análise desses dados e relacionando com a recomendação de rotas tentar extrair, por exemplo, trajetórias usadas frequentemente pelos pedestres.

Num sistema móvel de recomendação de rotas é necessário ter conhecimento da localização dos utilizadores. Tendo em conta que na atualidade a maioria dos indivíduos possui dispositivos móveis cada vez mais sofisticados, pode recorrer-se a eles para a obtenção da localização. Existindo para isso duas técnicas, localização relativa e localização absoluta [19]:

Localização relativa – esta técnica calcula a distância entre o dispositivo móvel e o ponto de acesso (torres de rede ou pontos de acesso Wi-Fi). Tem como desvantagem o facto que

a precisão das torres é baixa, normalmente entre 2 a 3km comparado com a posição GPS. Para além disso, as operadoras não costumam fornecer acesso a dados de localização por serem privados.

Localização absoluta – esta técnica indica a localização de um dispositivo móvel com base nas suas coordenadas, utilizando para isso a posição GPS. Esta é considerada a melhor forma para modelar a mobilidade, porém tem desvantagens em termos de privacidade dos dados, pois desta forma é possível saber onde uma pessoa mora, onde trabalha ou onde janta.

2.3.2 Sistemas de recomendação

Para uma adequada sugestão de rotas é conveniente o uso de sistemas de recomendação [20], cujo objetivo é a recomendação de itens (neste caso rotas) aos utilizadores. Os métodos mais utilizados são os seguintes:

Content Based - sugere itens com base em escolhas similares feitas no passado pelo utilizador, sendo usado hoje em dia, por exemplo, no Youtube onde é sugerido aos utilizadores vídeos "Recomendado para si", com base nos vídeos previamente assistidos. Direcionando para sistemas de recomendação de rotas, seria sugerido ao utilizador uma rota com base em escolhas feitas anteriormente pelo mesmo.

Collaborative filtering – recomenda itens que pessoas com gostos similares gostaram no passado, incorporando em sistemas de recomendação de rotas os utilizadores podem basear também a escolha da sua rota em experiências de outros utilizadores. *Collaborative filtering* é considerada a técnica mais popular e implementada no que diz respeito a sistemas de recomendação, sendo usada, por exemplo, pela Amazon.

- **Huang, H. and Gartner, G.,2010** foca o seu estudo na introdução de *Collaborative Filtering* em serviços de localização móveis [21] com o intuito de auxiliar os utilizadores na escolha de rota. Hoje em dia as pessoas tomam, por vezes, decisões com base na opinião de outras pessoas que passaram pela mesma experiência.

O processo de *Collaborative Filtering* inclui três etapas:

- obtenção de dados para a criação de perfis de utilizador;
- cálculo da semelhança entre os utilizadores;
- agregação das avaliações anteriores dos N utilizadores mais parecidos para determinar a posterior recomendação.

Hybrid recommender systems - resulta da combinação dos métodos *Collaborative filtering* e *Content Based filtering*, resultando num melhor desempenho, bem como, recomendações mais precisas.

Este método é usado, por exemplo, na Netflix. Sendo indicado ao utilizador filmes com base no que utilizadores similares viram, bem como filmes com características similares a filmes que o utilizador previamente avaliou positivamente. Traduzindo para a recomendação de rotas, permitiria obter recomendação de rotas com base nas rotas previamente recebidas bem como as rotas que pessoas com perfis similares receberam como recomendação.

2.3.3 Cálculo do caminho mais curto

Hoje em dia estamos habituados ao uso do sistema GPS quando queremos obter uma rota até um determinado destino. Este sistema tem como base dois componentes, mapas digitais e um algoritmo que calcula o caminho mais curto.

O cálculo do caminho mais curto tem como objetivo, dado um grafo $G(V,A)$, onde V representa o conjunto de vértices e A o conjunto de arestas ligadas aos vértices, a procura do caminho com custo mínimo entre um vértice origem e um vértice destino, tendo em atenção os pesos das arestas. Existem diversos algoritmos que calculam o caminho mais curto, destacam-se os seguintes:

Algoritmo de Dijkstra [22] [23] - soluciona o problema do caminho mais curto num grafo dirigido ou não dirigido, sendo que o peso das arestas não pode ser negativo (Figura 2.3). Este algoritmo executa em tempo computacional $O([A+V]\log V)$ onde A é o número de arestas e V é o número de vértices.

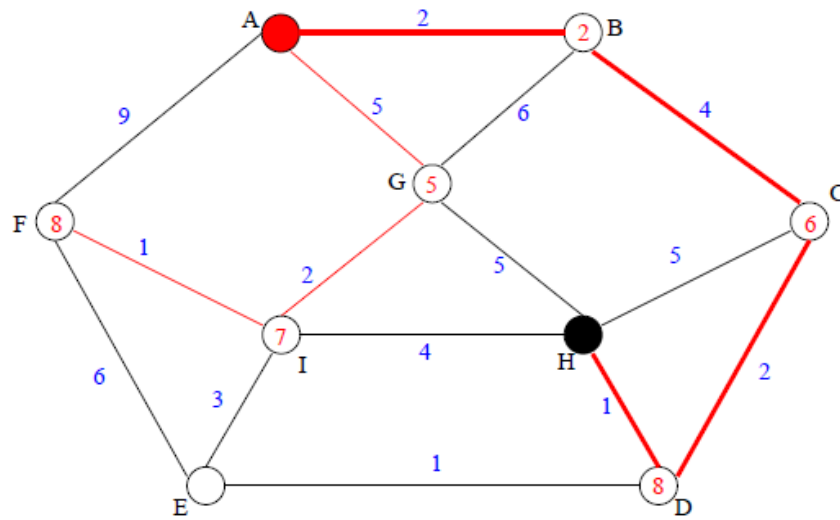


Figura 2.3: Algoritmo Dijkstra [24]

Algoritmo A* [22] [23] - este algoritmo vai encontrar o caminho num grafo do vértice origem até um vértice destino usando aproximações heurísticas. Para isso vai procurar entre todos os caminhos possíveis aquele que possui um menor custo, considerando primeiramente aquele que levar mais rapidamente até ao destino. Para isso começa no vértice origem e constrói uma árvore de caminhos a partir do vértice de origem, expandido o caminho a cada iteração até chegar ao vértice destino. Em cada iteração o algoritmo tem de determinar qual o caminho que vai expandir recorrendo para isso a seguinte função (1):

$$f(n) = g(n) + h(n) \quad (1)$$

sendo que n corresponde ao último nó do caminho, $g(n)$ traduz o custo do caminho que se inicia no vértice origem até n e $h(n)$ é uma heurística que vai estimar o custo mínimo do caminho de n até ao vértice destino.

Bellman-Ford [23] - este algoritmo soluciona o problema do caminho mais curto, sendo que para isso, vai gerar iterativamente uma melhor solução com base na solução obtida anteriormente. Este algoritmo é mais lento que o algoritmo de Dijkstra executando em tempo computacional $O(A \cdot V)$, mas possui uma vantagem pois as arestas podem possuir peso negativo (Figura 2.4).

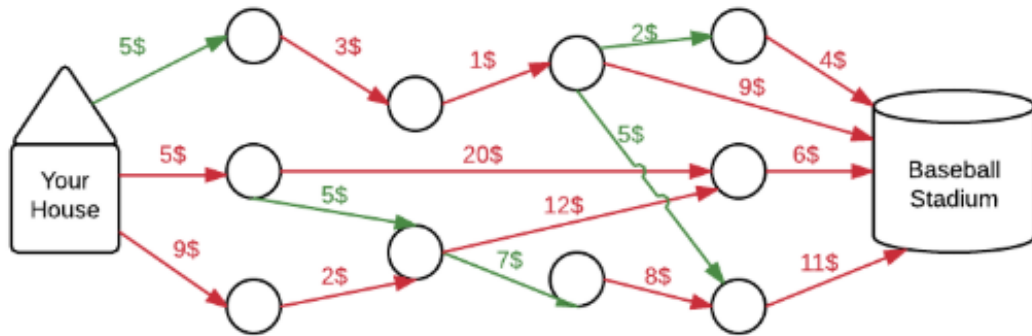


Figura 2.4: Algoritmo Bellman-Ford, arestas a vermelho equivalem a peso negativo [25]

Algoritmo Bidirecional[22] - para obter o caminho mais curto este algoritmo vai simultaneamente procurar a partir do vértice origem em frente e a partir do vértice destino para trás, até que os dois caminhos se encontrem (Figura 2.5) obtendo um caminho da origem até ao destino.

De seguida, é realizada uma comparação entre o peso desse caminho e, no caso de existir mais que um caminho entre os dois vértices, o peso desses caminhos de forma a verificar qual é verdadeiramente o mais curto.

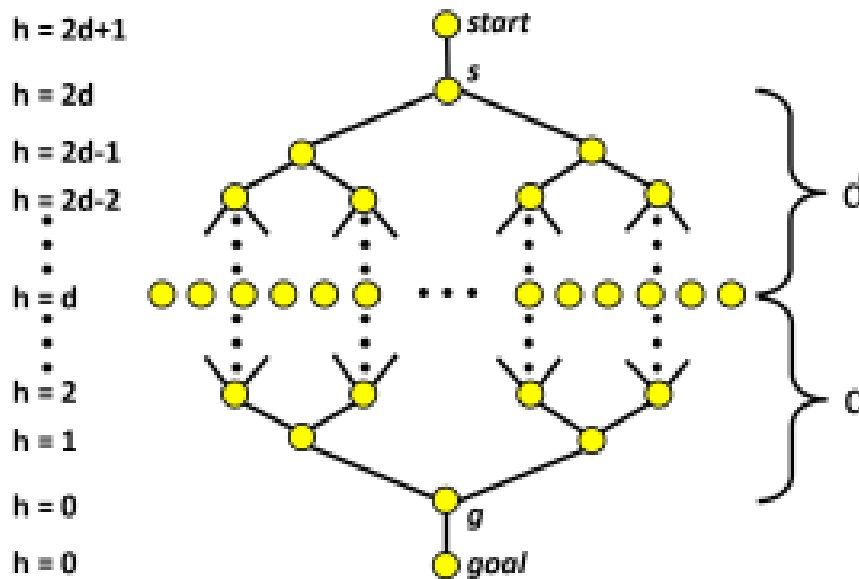


Figura 2.5: Algoritmo Bidirecional [22]

2.4 FERRAMENTA DE SIMULAÇÃO

Um dos objetivos parciais corresponde a encontrar uma ferramenta de simulação que permita a simulação de um meio urbano e diversos fatores como a luminosidade, declive, cobertura Wi-Fi e a criação de rotas. Nesta secção é feita uma análise de três ferramentas de simulação nomeadamente, CupCarbon, UrbanSim e InterSCSimulator, sendo consideradas as suas funcionalidades, bem como as suas vantagens e desvantagens, culminando na escolha da ferramenta mais vocacionada para este projeto.

2.4.1 CupCarbon

A primeira ferramenta estudada é o CupCarbon, esta ferramenta faz parte de um projeto de pesquisa denominado *PERSEPTEUR*, tendo sido criada com o intuito de estudar aspetos relacionados com *Smart Cities*. A versão atual é a versão 3.8 sendo esta uma ferramenta de *software* grátis que permite visualizar, compilar e validar algoritmos de monitorização, bem como a criação de cenários ambientais e o armazenamento dos resultados obtidos [27].

O CupCarbon permite a criação de dois ambientes de simulação:

- o primeiro permite o desenho de cenários de mobilidade e a geração de eventos naturais, bem como a simulação de veículos e objetos voadores;
- o segundo corresponde a uma simulação de eventos numa rede de sensores sem fios tendo em consideração o cenário do primeiro ambiente.

As redes de sensores são de simples implementação, através de uma interface bastante intuitiva, é possível colocar diretamente os sensores no mapa, sendo este obtido a partir do uso da estrutura OpenStreetMap [28] (Figura 2.6).

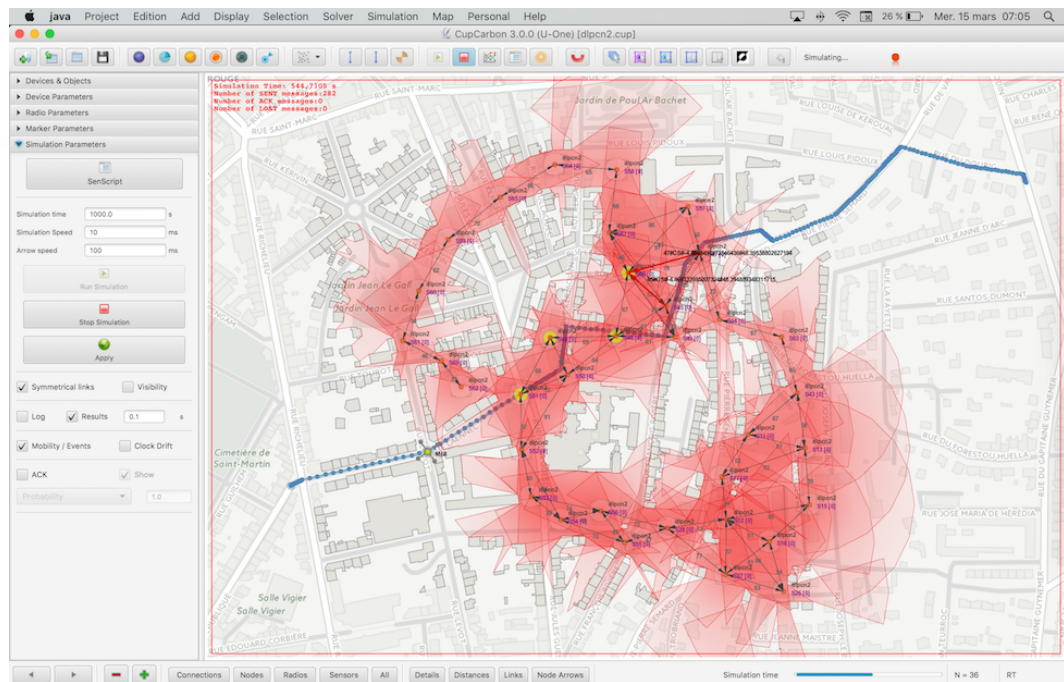


Figura 2.6: Exemplo de implementação de uma rede de sensores

Relativamente aos sensores a ferramenta suporta uma linguagem de programação, *SenScript*, usada para criar *scripts* que são posteriormente associadas aos sensores. Esta linguagem de programação apenas permite inicializar as variáveis através do uso do comando *set*.

Através do uso das *scripts* é possível programar o comportamento de cada sensor individualmente, permitindo, por exemplo, ter uma rede de sensores que comunicam entre si sendo que cada um dos sensores tem um comportamento específico.

O CupCarbon permite a criação de rotas usando *markers*, sendo estes colocados ao longo do mapa de forma a criar a rota pretendida pelo utilizador. Cada *marker* está ligado ao *marker* anterior formando desta forma uma rota, após a criação da rota esta é armazenada num ficheiro GPS. Esta funcionalidade vai de acordo ao pretendido com este trabalho, visto ser possível criar vários ficheiros GPS contendo rotas para posteriormente, através de cálculos, ser escolhida a rota recomendada.

Esta ferramenta permite ainda associar um sensor móvel a uma rota, desta forma, o sensor móvel percorre uma rota previamente criada. Esta funcionalidade permite simular, por exemplo, o deslocamento de um pedestre ao longo de uma rota recomendada.

Outras das funcionalidades permitidas é a exportação e armazenamento dos resultados obtidas num ficheiro csv.

2.4.2 *UrbanSim*

Esta plataforma de simulação foi desenvolvida por Paul Waddell na Universidade da Califórnia. Tem como objetivo, a simulação e análise do desenvolvimento urbano [29], sendo útil para entender as diferentes interações entre políticas de transporte (acessibilidade motorizada e não motorizada), economia e o meio ambiente numa cidade (emissões de gases e efeito de estufa).

Esta ferramenta foi projetada para ser usada por organizações que pretendam fazer um planeamento Metropolitano de cidades ou municípios. *UrbanSim* é essencialmente uma representação computacional dos mercados imobiliários, que correspondem a famílias ou empresas que interagem com os mercados de transporte, permitindo modelar as escolhas feitas em termos de mudanças localizações.

Na sua génese o *UrbanSim* faz uso de modelos (Figura 2.7) que representam as condições locais de uma região, podendo estes ser divididos em duas partes:

- *Census Block* - esta funcionalidade apenas se encontra disponível nos Estados Unidos, corresponde a modelos pré-construídos das áreas metropolitanas dos Estados Unidos tendo sido criados com base em dados públicos obtidos através de censos;
- *Parcel* ou *Zone* - estas duas opções permitem aos utilizadores de qualquer parte do mundo criar modelos *UrbanSim*, possui a mesma estrutura que o modelo *Census Block* sendo neste caso necessário o utilizador fornecer os dados locais relativos à região em questão, de forma a auxiliar a criação do modelo.

A primeira versão desta ferramenta foi desenvolvida em *Java* tendo passado por reformulações sendo que a versão atual tem por base o uso de bibliotecas de *Python*. Foi também desenvolvido a *UrbanSim Cloud Platform* com o objetivo de correr várias simulações simultaneamente em nuvem para escalabilidade. Em termos de simulações, esta plataforma fornece em tempo real informação ao utilizador, sobre o progresso da simulação que se encontra a ser executada. No final este tem acesso aos resultados, podendo estes ser visualizados como mapas 2D ou 3D, gráficos e tabelas.

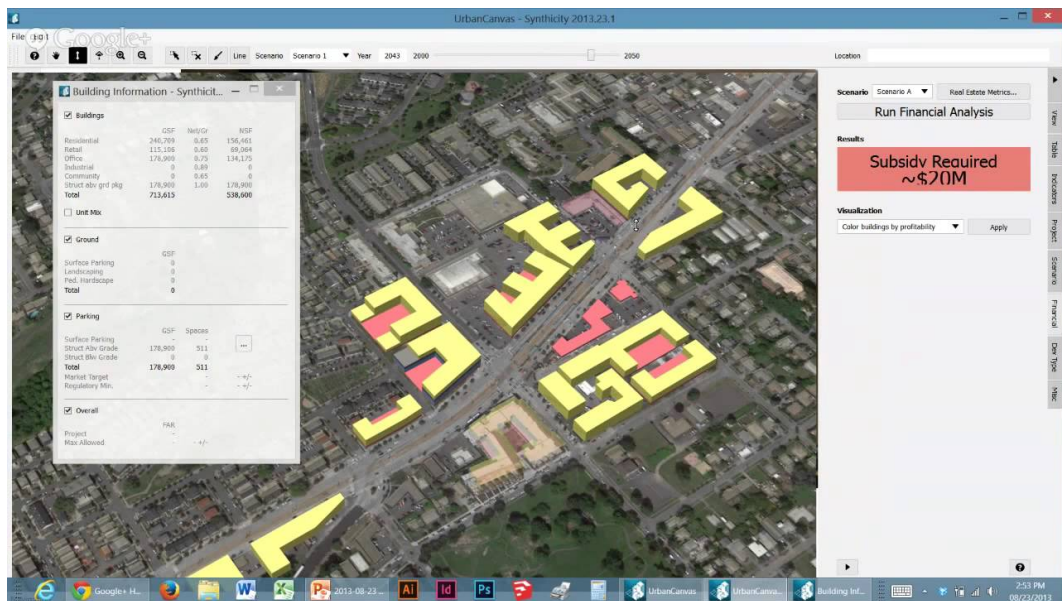


Figura 2.7: Interface utilizador da ferramenta UrbanSim

2.4.3 InterSCSimulator

O InterSCSimulator é um simulador de cidades inteligentes, esta plataforma faz parte de um projeto de pesquisa patrocinado pelo Instituto Nacional de Ciência e Tecnologia, envolve várias instituições brasileiras e foca-se no estudo e desenvolvimento de aplicações para as Cidades Inteligentes do futuro.

Tal como a ferramenta CupCarbon, também é uma ferramenta de *software* grátis e tem como objetivo a simulação de cenários complexos de uma cidade inteligente (Figura 2.8), como gerir o tráfego urbano, gerir os resíduos e redes inteligentes [30]. Este simulador tem como base o *Sim-Diasca*, um simulador de eventos gerais, de grande escala e de eventos discretos e usa o *Erlang* como linguagem de programação permitindo a simulação de milhões de agentes fazendo uso de um mapa real de uma cidade.

Trata-se de uma ferramenta escalável, visto que lida com milhões de agentes, como carros, edifícios e sensores, sendo necessário que estes agentes sejam simulados simultaneamente para isso são executadas simulações paralelas e distribuídas.

O InterSCSimulator tem quatro componentes principais:

- o primeiro é a definição de cenário, que recebe os arquivos de entrada, nomeadamente três ficheiros xml com as configurações, as viagens e o mapa e com essa informação cria o gráfico da cidade e os primeiros veículos;
- mecanismo de simulação, vai executar os algoritmos e modelos de simulação e guardar os resultados da simulação num ficheiro xml;

- visualização do Mapa, que recebe o resultado da simulação (ficheiro xml) e cria uma visualização visual do mapa da cidade e do movimento dos veículos;
- visualização de Gráficos, que também recebe o resultado de simulação e gera vários gráficos com informações sobre o cenário simulado.

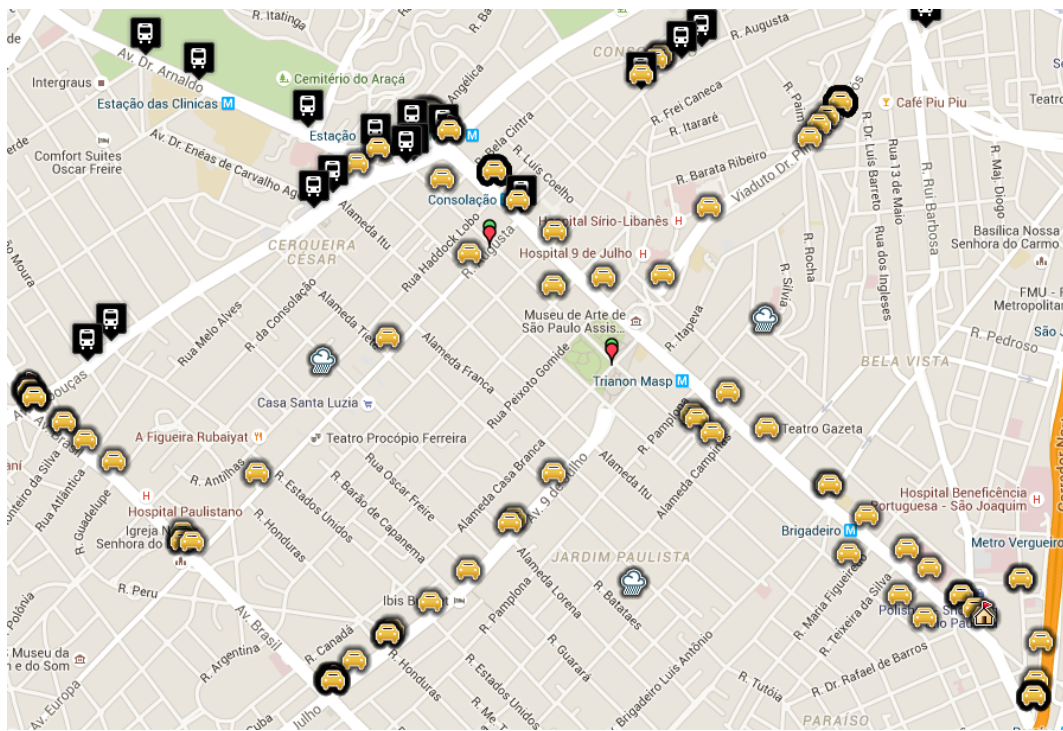


Figura 2.8: Execução do simulador para a cidade de São Paulo

2.4.4 Escolha da Ferramenta de Simulação

Após o estudo das três ferramentas e as suas funcionalidades foi necessário tomar uma decisão relativamente a escolha da ferramenta de simulação, para isso fez-se uma análise e comparação das mesmas.

Numa primeira análise, verifica-se que as ferramentas CupCarbon e InterSCSimulator permitem o seu uso de forma gratuita o que é uma vantagem, a ferramenta UrbanSim apresenta uma pequena desvantagem, visto a ferramenta ter modelos pré-definidos apenas para as zonas metropolitanas dos Estados Unidos. Seria necessário usar o modelo *Zone* para a cidade de Braga e atualmente, é necessário um contrato de consultoria para usar esse modelo.

O aspeto seguinte que foi tido em conta, foi a existência de material de apoio para quem está a trabalhar com a ferramenta pela primeira vez, neste aspeto as ferramentas CupCar-

bon e UrbanSim apresentam um maior número de material de suporte como tutoriais e manuais de utilização relativamente à ferramenta InterSCSimulator.

De seguida partiu-se para uma análise mais crítica a cada uma das funcionalidades inerentes a cada uma das ferramentas:

- UrbanSim, esta ferramenta encontra-se mais focada para a simulação e estudo de interações entre mercados imobiliários e mercados de transporte, tendo em atenção políticas de transporte e economia.

Olhando para estas funcionalidades, estas não se enquadram nem vão ao encontro do pretendido com este trabalho que é a possibilidade de criação de rotas, monitorização de atributos e por fim a recomendação de rotas. Tendo este aspeto em conta e o fato de que esta ferramenta necessitaria de um contrato de consultoria, optou-se por descartar esta ferramenta.

- InterSCSimulator, esta ferramenta permite gerir o tráfego urbano, sendo esta uma funcionalidade que vai ao encontro do pretendido com este trabalho, que é, a simulação de tráfego pedestre num meio urbano. Outro aspeto importante é a escalabilidade, permitindo correr várias simulações de forma paralela e distribuída. Em termos negativos, como dito anteriormente, é a falta de material de apoio e suporte a aprendizagem do uso da ferramenta.
- CupCarbon, tal como o InterSCSimulator vai permite gerir o tráfego urbano contudo apresenta algumas funcionalidades que sobressaem relativamente ao InterSCSimulator, a primeira é hipótese de criar vários sensores (responsáveis por monitorizar os diversos critérios para a escolha da rota) e associar a cada um deles, através da criação de *scripts*, um comportamento específico.

Esta ferramenta apresenta uma interface gráfica bastante simples e fácil de usar, nomeadamente na vertente de criação de rotas e implementação de sensores. Outras das vantagens é o fato de ser uma ferramenta recente, tendo bastante material de apoio a aprendizagem.

Comparando as duas ferramentas, optou-se por escolher a ferramenta CupCarbon como ferramenta utilizada nesta trabalho por apresentar um maior número de vantagens.

Concluída a escolha da plataforma de simulação, uma das funcionalidades que tomou relevância foi a criação de sensores móveis. Nos últimos anos tem havido uma exploração dos veículos aéreos não tripulados (VANT) [31], vulgarmente conhecidos como *drones*, pois estes apresentam diversas aplicações. Estas aplicações passam pela entrega de serviços como, por exemplo, o Amazon Prime Air que faz a entrega de pacotes aos clientes recorrendo a VANTs, podendo ainda ser usados em operações de resgate, bem como para a proteção/segurança pública estando estes equipados com sensores, câmaras e atuadores.

Focando-nos na entrega de serviços, num futuro próximo os drones serão usados para o fornecimento de Internet, o Facebook tem desenvolvido testes no sentido de ter disponível *drones* que forneçam Internet [32] a comunidades remotas ou partes subdesenvolvidas do planeta. Com a criação de sensores móveis seria interessante nas simulações realizadas, ter em atenção o caso em que diversos utilizadores optem por uma rota com base no atributo cobertura Wi-Fi e havendo um pico de utilizadores, usar um nodo móvel representando um *drone* que disponibilizaria Internet para os utilizadores.

2.5 RESUMO DO CAPÍTULO

Ao longo deste primeiro capítulo, fez-se um estudo dos atributos mais usados pelos pedestres quando se trata de tomar uma decisão relativamente à escolha de um percurso. Realizou-se também um estudo relativamente ao trabalho relacionado com esta dissertação, nomeadamente das ferramentas PedestrianPal, Google Maps e Komoot. De seguida, foram analisadas as tecnologias necessárias e mais usadas para a criação de um sistema de recomendação de rotas, tendo estas sido divididas em localização dos pedestres, sistemas de recomendação e cálculo do caminho mais curto.

No final deste capítulo foi ainda realizado um estudo das ferramentas CupCarbon, UrbanSim e InterSCSimulator. Com base na comparação das suas funcionalidades, selecionou-se a ferramenta CupCarbon como plataforma de simulação, pois é aquela que apresenta funcionalidades que vão ao encontro do que é pretendido simular neste trabalho.

No capítulo seguinte, será apresentada uma primeira abordagem ao problema indicando uma arquitetura inicial, sendo de seguida feito um levantamento dos requisitos funcionais e não funcionais.

ABORDAGEM AO PROBLEMA

Este capítulo inicia-se com uma primeira aproximação à resolução do problema proposto e delineada a respetiva arquitetura, tendo por base a revisão bibliográfica e a escolha da ferramenta de simulação feita no capítulo anterior.

De seguida, será apresentado um levantamento de requisitos funcionais e não funcionais tendo em vista um maior esclarecimento sobre as funcionalidades inerentes a este projeto, bem como, a criação do modelo de domínio.

3.1 ABORDAGEM AO PROBLEMA

Tendo por base o estudo de Huang e Gartner realizado na Secção 2.1, um sistema de recomendação de rotas pode ser dividido em posicionamento, cálculo de rota e comunicação da rota (Figura 3.1) segue-se uma possível solução:

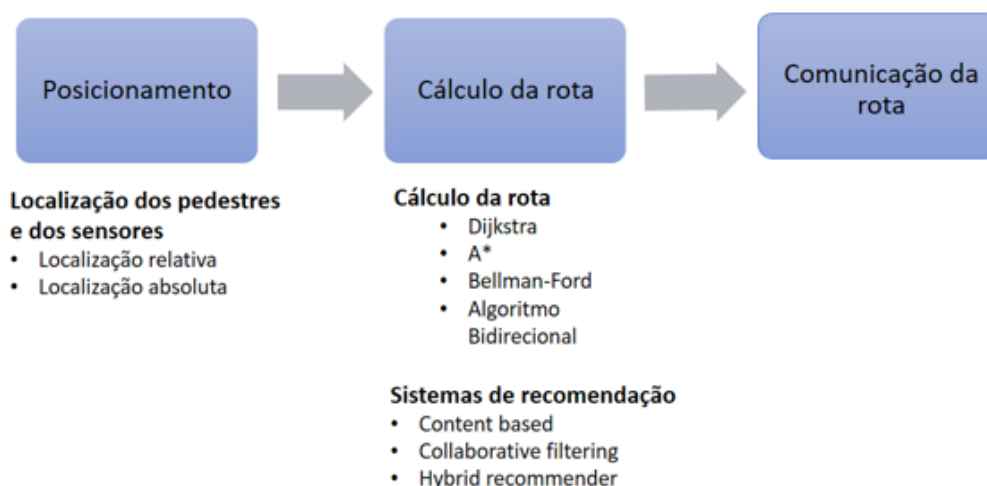


Figura 3.1: Base de uma solução para o problema usando o estudo de Huang e Gartner

- Como discutido na Secção 2.3.1, existem duas possibilidades de obter o **posicionamento**, através de localização relativa ou localização absoluta. Sendo a localização absoluta a forma mais precisa de obter o posicionamento. Como a ferramenta de simulação CupCarbon faz uso da estrutura *OpenStreetMap*, é possível obter a localização absoluta dos intervenientes da simulação. Sendo assim, é possível obter as coordenadas GPS dos sensores utilizados, desta forma pode ser criada uma rede de sensores de monitorização. Os dados recolhidos através da monitorização são posteriormente analisados para ajudar no cálculo da rota.
- A tarefa responsável pelo **cálculo da rota** pode ser dividida em sub-tarefas:
 - atribuir um peso a cada um dos atributos de acordo com a preferência do utilizador, quanto maior o peso maior a relevância dada a esse atributo;
 - definir uma posição inicial e final, sendo que a posição final é mais abrangente, desta forma é possível obter um maior número de rotas que culminem nessa posição;
 - monitorizar e analisar os dados recolhidos a partir dos sensores;
 - fazer uso dos sistemas de recomendação estudados na Secção 2.3.2, permitindo desta forma o utilizador receber recomendação de rotas com base em experiências anteriores ou com base em rotas que utilizadores similares usaram e avaliaram de forma positiva;
 - para o cálculo da rota recorrer a uma função de custo, ou seja, somando os pesos dos atributos pretendidos pelo utilizador que se verificarem ao longo da rota. Quanto maior o valor mais critérios pretendidos essa rota possui.
- Por fim, para a **comunicação da rota**, deve ser apresentado ao utilizador um mapa com as rotas que verificarem o pretendido pelo mesmo. Sendo necessário também a comunicação dos dados monitorizados pelos sensores e o resultado da função de custo aplicada a cada uma das rotas. Desta forma, será comunicado ao utilizador a rota recomendada.

A ferramenta de simulação CupCarbon permite a criação de várias rotas, bem como implementar sensores que monitorizem os atributos. Sendo também capaz de trocar mensagens entre sensores, esta funcionalidade permite comunicar ao sensor móvel a rota que vai mais ao encontro do pretendido.

3.1.1 Arquitetura do sistema

Tendo em atenção ao estudo anterior, chegou-se a uma arquitetura como solução para o problema (Figura 3.2). Esta arquitetura surge como uma expansão da solução anterior.

O primeiro interveniente é o **Utilizador**, este tem associado a si um perfil que lhe permite acesso aos dados de utilizações anteriores bem como os seus atributos preferidos. O utilizador é também responsável por indicar o posicionamento, fornecendo as coordenadas do ponto de partida e chegada.

Para o **cálculo da rota** como referido anteriormente, é tido em conta os seguintes aspetos:

- implementação de uma rede de sensores ao longo da cidade, sendo esta responsável por monitorizar os vários atributos. Os dados recolhidos a partir da monitorização, são armazenados numa base de dados para posterior acesso e uso;
- uso de sistemas de recomendação com base no perfil do utilizador e no histórico de utilizações anteriores;
- uso de algoritmos de caminho mais curto, de forma a calcular com base no peso das rotas a rota que vai ao encontro do pretendido pelo utilizador;
- acesso a percursos pré-definidos.

Por fim, é comunicado ao utilizador a(s) rota(s) recomendada(s).

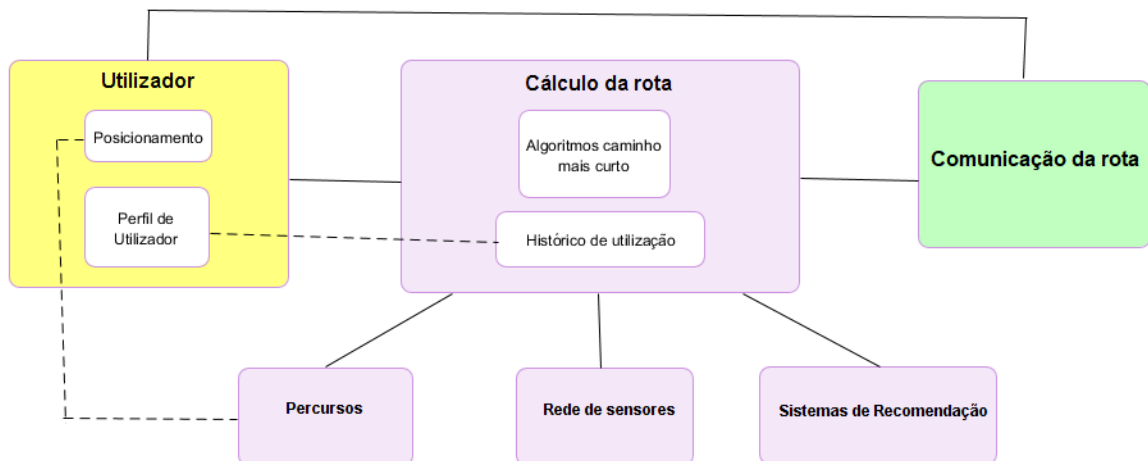


Figura 3.2: Arquitetura da solução para o problema

3.2 ARQUITETURA DA SOLUÇÃO

Tendo por base a arquitetura anteriormente apresentada como solução para o problema, fez-se um levantamento detalhado dos requisitos funcionais e não funcionais do sistema de recomendação de rotas. Inicialmente, é identificado o tipo de utilizadores que irão interagir com a aplicação, bem como, as infraestruturas necessárias para a sua implementação, nomeadamente:

- uso de sensores para a monitorização de alguns atributos dinâmicos;
- um servidor central que comunica com os sensores recebendo os dados e guardando-os numa base de dados.

Para além destas infraestruturas, será necessário ainda a disponibilização em diversas zonas da cidade de pontos agregadores, como por exemplo, ecrãs onde serão disponibilizados os valores dos sensores. Desta forma, os pedestres ao deslocarem-se pela cidade e pelas suas rotas tem acesso a informações atualizadas.

3.2.1 *Tipos de Utilizadores*

Pretende-se que o sistema a ser implementado possa ser utilizado por dois tipos de utilizadores:

- o primeiro corresponde ao grupo de indivíduos que utilizam a aplicação para uso pessoal, utilizando a aplicação de forma a obter recomendação de rotas recorrendo a um conjunto de critérios dinâmicos e estáticos, passíveis de ser ajustados às suas preferências (ver Tabela 3.1).
- o segundo corresponde ao administrador, que utiliza o sistema de forma administrativa. Este utilizador deve ter algumas noções tecnológicas básicas, de forma a manusear a aplicação. O administrador será também responsável por controlar os diversos sensores que serão espalhados pela cidade, tendo em vista a monitorização dos atributos dinâmicos (ver Tabela 3.2).

Para além destas funções, será também responsável por gerir um servidor central, para onde serão enviados os dados recolhidos pelos sensores, sendo posteriormente guardados numa base de dados também gerida pelo administrador.

3.2.2 *Tipos de Critérios de Escolha*

Tendo por base os critérios mais usados pelos pedestres no momento de escolher uma rota, foi necessário fazer uma divisão dos mesmos em duas classes, os critérios relacionados com o utilizador e os relacionados com a cidade:

- **Utilizador**
Este tipo de critério inclui atributos dependentes de cada utilizador. Enquanto para alguns utilizadores será mais relevante um percurso seguro, para outros será mais interessante saber o grau de dificuldade do percurso (ligeiro, médio, difícil).

Tabela 3.1: Critérios de escolha do utilizador

Atributos
percurso de interesse turístico
percurso com espaços abertos
percurso circular
grau de segurança (baixo, médio, alto)
distância
mudanças de direção
tempo médio do percurso
declive
grau de dificuldade do percurso (ligeiro, médio, difícil)

- **Cidade**

Este tipo de critério inclui:

- **atributos dinâmicos**, variam ao longo do tempo sendo por isso necessário recorrer a sensores para fazer a monitorização dos mesmos, permitindo aos utilizadores terem acesso a informação atualizada;
- **atributos estáticos**, estes atributos são fixos e previamente conhecidos.

Tabela 3.2: Atributos inerentes à cidade

Dinâmicos	Estáticos
iluminação	espaços abertos
cobertura Wi-Fi	praças e parques
condições meteorológicas	largura dos passeios
densidade de veículos	zonas comerciais
segurança	locais com interesse histórico
	travessia com semáforo
	declive
	infraestruturas para pessoas com limitações físicas

3.2.3 Tipos de percursos

Tendo por base a Tabela 3.1, foram considerados cinco tipos de percurso disponíveis para o utilizador escolher, associando-se a cada um deles os atributos relacionados (Figura 3.3):

- **percurso seguro**, neste percurso os utilizadores estão preocupados com a sua segurança, quer seja uma pessoa que pratique desporto a noite e pretenda usar ruas bem iluminadas, quer seja um pai que passeie com os seus filhos e pretenda saber se o percurso tem uma baixa densidade de veículos a circular, se as travessias tem semáforos e se a largura dos passeios é grande ou não;
- **percurso com espaços abertos**, este percurso é orientado para utilizadores que prezem o ar livre, dando destaque a presença de espaços verdes, parques e praças;
- **percurso com interesse turístico**, este percurso é mais focado para turistas que estejam a visitar a cidade tendo assim acesso a uma rota onde são indicadas zonas comerciais bem como as zonas de maior interesse histórico na cidade, tais como igrejas, monumentos, etc;
- **grau de dificuldade do percurso**, esta escolha de percurso é dividida em 5 sub-percursos:
 - **ligeiro**, para utilizadores que pretendam dar uma simples caminhada em que a distancia é curta e o declive também é baixo;
 - **difícil**, para utilizadores que pretendam um percurso com uma distância maior, um maior tempo de viagem e um declive mais acentuado;
 - **médio**, para utilizadores que pretendam um percurso intermédio entre o percurso ligeiro e difícil;
 - **dificuldade motora**, este percurso é direcionado para pessoas que tenham algum tipo de dificuldade motora, permitindo desta forma saberem uma rota mais adequada para se deslocarem pela cidade;
 - **percurso circular**, em que os utilizadores pretendem que o ponto de partida do seu percurso seja o mesmo que o ponto de chegada.

Simultaneamente a cada um destes tipos de percurso estão também associados os seguintes atributos/critérios:

- distância;
- declive;
- tempo de viagem;

- condições meteorológicas;
- cobertura Wi-Fi.

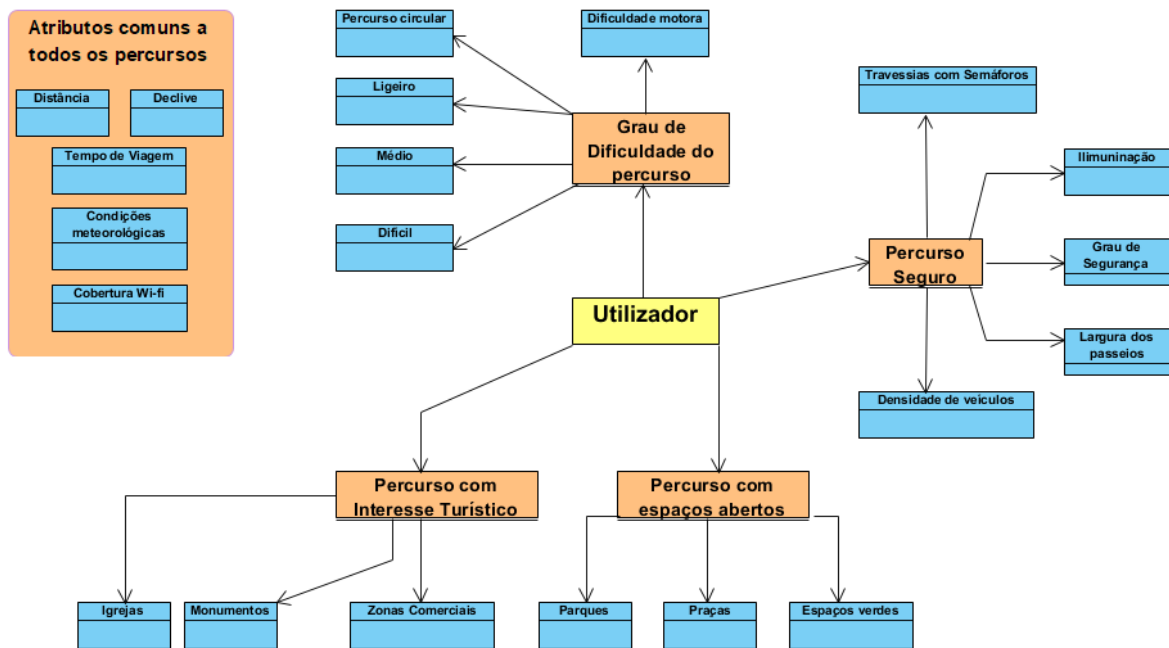


Figura 3.3: Diagrama de percursos

3.2.4 Requisitos de Infra-estrutura

Indicam-se, de seguida, as restrições associadas ao sistema em consideração, especificando, para cada uma delas, uma descrição, um motivo e a forma de aplicação associados.

Descrição: O sistema deve ser destinado a dispositivos móveis.

Motivo: O utilizador deve poder utilizar o sistema em qualquer lugar.

Aplicação : O sistema deverá funcionar em *Android* ou *iOS*, efetuando comunicação via Internet.

Descrição: O sistema deve utilizar sensores colocados em diversas zonas da cidade, responsáveis por monitorizar fatores ambientais tais como a luminosidade, a temperatura, etc, bem como a cobertura de sinal.

Motivo: Para auxiliar o utilizador na escolha da rota é precisa estar ciente dos valores dos atributos dinâmicos verificados em cada zona da cidade (ver Secção 3.2.2).

Aplicação: O sistema irá receber valores dos sensores, disponibilizando-os aos utilizadores de duas formas, através da aplicação móvel e em pontos agregadores (ecrãs) colocados em

pontos estratégicos da cidade.

Descrição: O sistema deve interagir com pontos agregadores de informação situados pela cidade

Motivo: É importante os utilizadores não estarem restritos a uma única rota fixa. Deste modo, ao deslocarem-se pela sua rota inicial poderão encontrar um ponto agregador onde estará disponibilizada informação recolhida pelos sensores podendo o utilizador optar por criar uma nova rota com base na informação que lhe foi apresentada.

Aplicação: O sistema irá comunicar com ecrãs espalhados pela cidade, enviando-lhes os valores obtidos pelos sensores, disponibilizando aos utilizadores a informação relativa à situação atual.

3.2.5 Factos e Pressupostos

Nesta secção serão apresentados alguns factos relevantes relacionados com o sistema a ser desenvolvido bem como pressupostos importantes na fase de levantamento de requisitos.

Factos:

- Os sensores estão distribuídos pelas várias zonas da cidade e cada um destes dispositivos é responsável por monitorizar diversos atributos dinâmicos.
- O sistema envia dados de monitorização e notificações para um servidor central que posteriormente disponibilizará informação relacionada em pontos agregadores colocados na cidade.

Pressupostos:

- A cidade disponibilizará sensores responsáveis pela monitorização de dados ambientais com capacidade de comunicação via Wi-Fi.
- A cidade possuirá um servidor central que será utilizado para o envio dos dados de monitorização para os utilizadores, bem como para os pontos agregadores.
- A cidade disponibilizará ecrãs, que permitirão a visualização constante de informação decorrente do sistema de monitorização.
- Os pontos agregadores possuirão uma base de dados para poderem guardar os valores recebidos e disponibiliza-los até que esses valores sejam novamente atualizados.

3.2.6 Contexto de utilização

Apresenta-se, na seguinte imagem, um diagrama ilustrativo das entidades que interagem com o sistema e os diferentes dados transmitidos entre os mesmos.

O sistema recebe leituras dos sensores, enviando esses valores para dois locais, uma base de dados onde serão armazenados esses valores e para os pontos agregadores. O administrador é responsável por controlar os sensores, o sistema e a base de dados (Figura 3.4).

O utilizador pode interagir de duas formas com o sistema, usando a aplicação mobile onde lhe serão recomendadas rotas ou através da informação disponível nos pontos agregadores espalhados pela cidade.

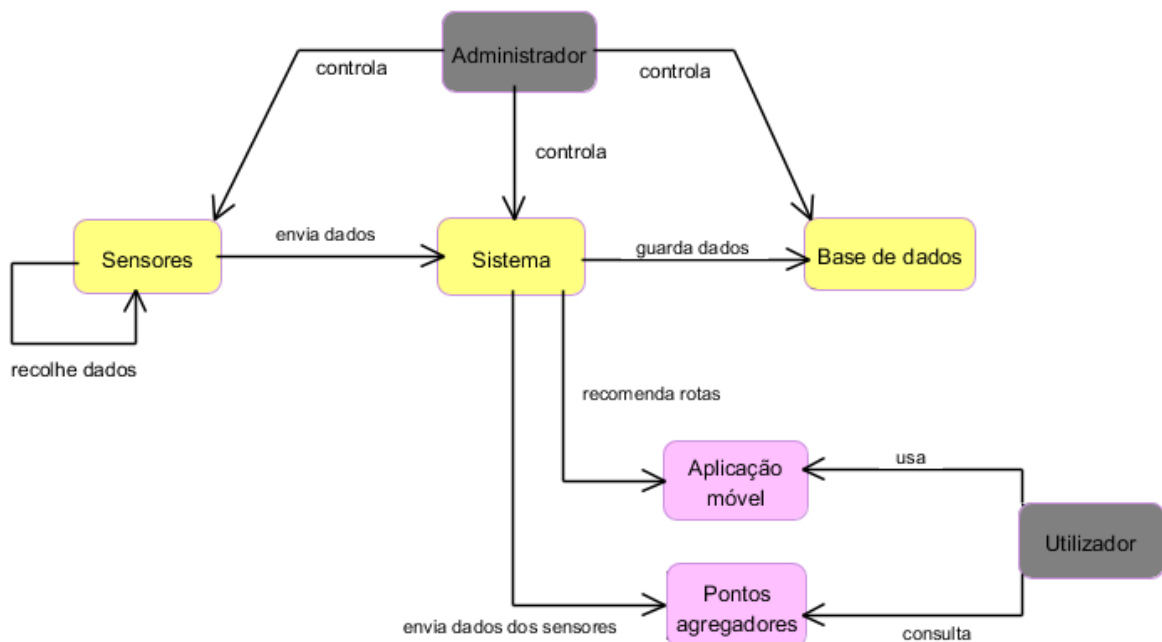


Figura 3.4: Domínio de trabalho

3.2.7 Requisitos funcionais

De forma a perceber quais são as funcionalidades inerentes ao sistema que se pretende desenvolver procedeu-se ao levantamento de requisitos, seguido da documentação dos mesmos.

O formato utilizado para a definição de cada requisito compreende, para além de uma descrição do requisito: um identificador numérico, a sua finalidade e uma pós-condição que permite verificar a correção da implementação do mesmo.

Número do requisito: 1

Descrição: Os utilizadores deverão poder registar-se no sistema, criando uma conta.

Finalidade: Possibilitar aos utilizadores a capacidade de utilizar a aplicação, permitindo ao sistema distingui-los entre si e manter o registo das suas atividades.

Pós-condição: Os utilizadores realizam o registo e o sistema possui a informação dos utilizadores registados e respetivos dados, contendo o registo da atividade dos primeiros.

Número do requisito: 2

Descrição: Um utilizador deve autenticar-se no sistema.

Finalidade: Permitir ao utilizador o acesso às funcionalidades de criação de rota bem como acesso ao seu historial de rotas.

Pós-condição: O utilizador tem permissão criar uma nova rota.

Número do requisito: 3

Descrição: Os utilizadores podem consultar remotamente as condições dos vários atributos dinâmicos.

Finalidade: Saber quais as condições ambientais da cidade de modo a auxiliar a escolha da rota.

Pós-condição: O utilizador conhece as condições ambientais atuais da cidade.

Número do requisito: 4

Descrição: O utilizador deve indicar os atributos desejados para a escolha da rota.

Finalidade: Obter a rota cujos atributos vão ao encontro do pretendido pelo utilizador.

Pós-condição: Comunicação ao utilizador do grupo de rotas mais parecidas com o pretendido.

Número do requisito: 5

Descrição: O sistema deve calcular a rota com base nos atributos fornecidos pelo utilizador, nos dados obtidos pelos sensores e usando Sistemas de recomendação e Algoritmos caminho mais curto.

Finalidade: Obter o máximo de dados para usar numa função de custo tendo em vista a obtenção de rota cujos atributos sejam os mais similares possíveis aos desejados pelo utilizador.

Pós-condição: Obtenção de uma rota.

Número do requisito: 6

Descrição: O sistema deve sugerir algumas rotas pré-definidas com base em escolhas feitas anteriormente pelo utilizador.

Finalidade: Usar informações do utilizador de modo a facilitar a escolha da rota.

Pós-condição: Comunicação ao utilizador de um grupo de rotas recomendadas.

Número do requisito: 7

Descrição: O sistema deve atualizar a informação exibida nos pontos agregadores da cidade.

Finalidade: Permitir a monitorização das condições ambientais em tempo real.

Pós-condição: Os dados são atualizados nos pontos agregadores com base num intervalo de tempo pré-definido.

Número do requisito: 8

Descrição: O administrador do sistema pode configurar, no sistema, os diversos sensores existentes na cidade.

Finalidade: Permitir a monitorização dos atributos dinâmicos.

Pós-condição: O sistema conhece os sensores a partir dos quais recebe leituras.

3.2.8 *Requisitos não funcionais*

Os requisitos não funcionais encontram-se relacionados com o uso do sistema, tendo sido abordado os requisitos operacionais e de aparência.

Requisitos operacionais

- Os sensores vão ser responsáveis por monitorizar atributos específicos.
- Os dados recolhidos dos sensores são enviados para um servidor central, que os regista numa base de dados criada para o efeito.
- O sistema deverá guardar os dados pessoais dos utilizadores numa base de dados, incluindo rotas criadas previamente.
- O sistema deve enviar leituras recebidas dos sensores para os pontos agregadores (ecrãs) espalhados pela cidade.

Aparência

- As informações contidas nos pontos agregadores devem ser facilmente visíveis
- Associar a cada um dos atributos um ícone ilustrativo desse mesmo atributo

3.2.9 Modelo de Domínio

De seguida apresenta-se o modelo de domínio (Figura 3.5) que resume todos os intervenientes do sistema bem como a forma como os utilizadores interagem com o sistema. Como referido na Secção 3.2.1, existem dois tipos de utilizadores, em particular o administrador vai ser responsável por gerir dois aspetos:

- o primeiro corresponde aos sensores colocados ao longo da cidade, este vão monitorizar diversos atributos que serão tidos em conta no momento da geração de uma rota;
- o segundo corresponde a gerir um servidor central que recebe os dados dos sensores e os armazena numa base de dados.

Os dados estarão desta forma disponíveis em pontos agregadores, que o utilizador pode consultar para se inteirar das condições dos diversos atributos. Além disso, o utilizador pode ainda usar uma aplicação móvel que lhe recomendará uma ou mais rotas com base nos atributos pretendidos.

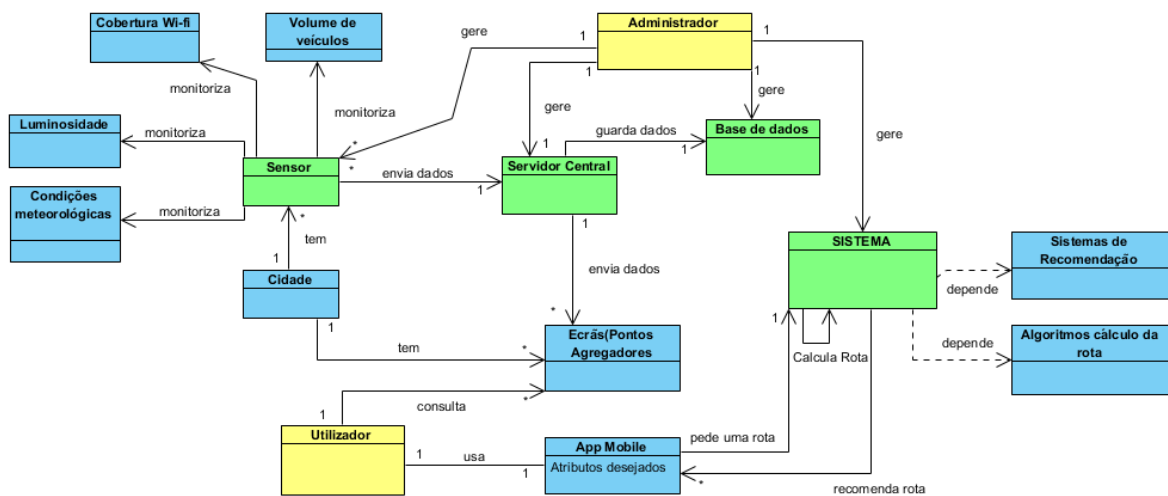


Figura 3.5: Modelo de domínio

3.2.10 Limitações da Plataforma de Simulação

Após concluir o levantamento de requisitos, o passo seguinte foi planear a implementação do modelo criado na plataforma de simulação, tendo por base as funcionalidades disponíveis. Contudo, a plataforma CupCarbon apresenta algumas limitações não sendo possível implementar todos os requisitos, por exemplo, não é possível ter acesso a uma base de da-

dos bem como, qualquer outra entidade externa ao simulador. Desta forma, não é possível simular os seguintes aspetos:

- armazenamento dos dados monitorizados pelos sensores numa base de dados;
- utilização de Sistemas de Recomendação com base no histórico de utilizações, pois não é possível armazenar um histórico de utilizações de cada Utilizador.

Assim sendo, em termos do cálculo da rota na plataforma de simulação, não será tido em conta o uso de sistema de recomendação, nem a implementação de algoritmos de caminho mais curto. Pois o que se pretende é a recomendação de rotas com base na escolha de atributos feita pelo utilizador e não no caminho mais curto.

3.3 RESUMO DO CAPÍTULO

Este capítulo iniciou-se com uma primeira abordagem ao problema, resultando numa solução possível e a sua arquitetura, tendo por base os estudos de Huang e Gartner. De seguida, fez-se um levantamento dos requisitos tendo em vista a implementação de um sistema de recomendação de rotas, iniciando-se com um estudo dos tipos de utilizadores. Seguiu-se uma recolha e divisão dos atributos entre atributos que variam consoante cada utilizador e os atributos inerentes à cidade, sendo estes divididos entre atributos estáticos e dinâmicos.

Por fim, fez-se um levantamento de requisitos funcionais e não funcionais, bem como a criação de um modelo de domínio representativo das várias entidades e as suas interações.

No próximo capítulo, será feita uma ambientação à ferramenta de simulação bem como realizados cenários de testes preliminares em ambiente simulado.

DESENVOLVIMENTO

Ao longo deste capítulo é abordada a parte prática deste trabalho, começando por uma ambientação à plataforma de simulação, sendo analisados os sensores disponíveis no CupCarbon, bem como, ilustrada a criação de rotas. Posteriormente, apresenta-se a implementação e simulação de dois cenários de testes usando o CupCarbon.

4.1 IMPLEMENTAÇÃO

Como abordado anteriormente, a ferramenta CupCarbon permite o uso de sensores para simular os vários atributos, bem como, a criação de rotas. Nesta secção é apresentada uma primeira interação com a ferramenta de simulação, apresentando os sensores inerentes ao CupCarbon. Após tomar conhecimento das funcionalidades de cada sensor, realizaram-se simulações envolvendo o uso de sensores e a comunicação entre os mesmos. Posteriormente, é apresentado o processo de criação de rotas.

4.1.1 Sensores disponíveis no CupCarbon

O Cupcarbon permite o uso de sensores de vários tipos, sendo que, ao longo deste trabalho os sensores usados são os seguintes:

- Sensor *Node* (primeiro a contar da esquerda na Figura 4.1), este sensor contém três partes principais: módulos de rádio que são responsáveis pela comunicação entre sensores, uma unidade de deteção e uma bateria;
- Sensor Gás (quarto a contar da esquerda na Figura 4.1), corresponde a um sensor de volumetria. Este permite ter associado um gerador de eventos naturais que, tendo por base uma distribuição Gaussiana, vai gerar valores aleatórios dentro de um intervalo definido pelo utilizador. Este sensor permite ainda escolher a periodicidade com que os valores são gerados;

- Sensor *Mobile* (sensor mais à direita na Figura 4.1), este sensor está dependente de uma rota a partir de um arquivo GPS, sendo esta criada usando marcadores, como é explicado na próxima secção. Durante a simulação, o sensor *mobile* vai se deslocar para cada próximo ponto de um em um segundo.



Figura 4.1: Sensores disponíveis no CupCarbon

Na implementação destes sensores pretende-se que cada um deles tenha as suas próprias funcionalidades. Como referido na Secção 2.4.1, a ferramenta Cupcarbon possui uma linguagem de programação chamada *SenScript* que permite a criação de *scripts* que posteriormente são associadas aos sensores. Permitindo que estes tenham comportamentos com base na funcionalidade pretendida.

Numa primeira ambientação, realizou-se uma simulação usando dois sensores *Node*, sendo associado a cada um dos sensores uma *script*, onde é descrito o comportamento que o sensor deve seguir. Assim, associa-se ao primeiro sensor uma *script* para transmitir mensagens e ao segundo uma *script* para ser o recetor das mensagens, imprimindo-as de seguida.

Pretende-se que os sensores simulem valores reais variáveis com o tempo. Assim associou-se ao sensor transmissor um Sensor Gás, para simular a geração de valores de temperatura (Figura 4.3), sendo necessário fornecer os seguintes parâmetros para a criação deste evento natural:

- média = 25°C;
- desvio padrão = 5°C;
- periodicidade = 1 segundo.

Colocando o sensor de volumetria no raio de um sensor transmissor, este vai receber os valores gerados aleatoriamente e de seguida enviar os mesmos para o sensor recetor (Figura 4.2). Desta forma, aproximamo-nos da funcionalidade desejada, pois pretende-se que sejam gerados valores para diversos atributos e de seguida comunicados a um recetor.

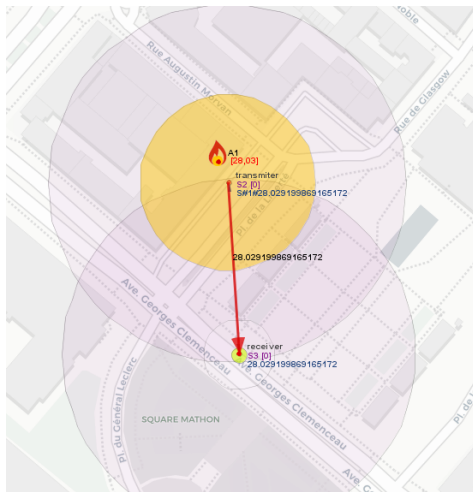


Figura 4.2: Simulação do envio de mensagens

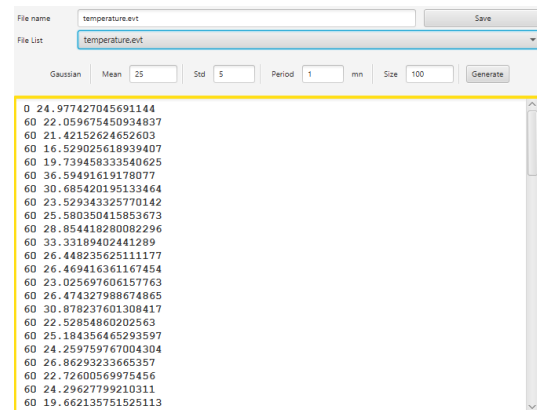


Figura 4.3: Script valores da Temperatura

4.1.2 Criação de rotas

Após testar a definição de sensores e a comunicação entre eles, o passo seguinte foi a criação de rotas. No Cupcarbon, a funcionalidade *Marker* permite criar uma rota entre dois marcadores (Figura 4.4), traduzindo-se numa reta entre os dois pontos. Porém, esta opção não tem em atenção se essa rota atravessa edifícios ou se existem ruas entre esses dois pontos. Assim sendo, existe a opção *Route from markers* que a partir dos dois pontos definidos vai proceder a criação de uma rota com base nas ruas que efetivamente existem no mapa (Figura 4.5), permitindo contudo que sejam feitos ajustes à rota para corresponder ao desejado.

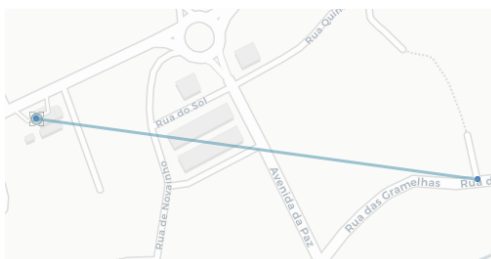


Figura 4.4: Rota sem marcadores

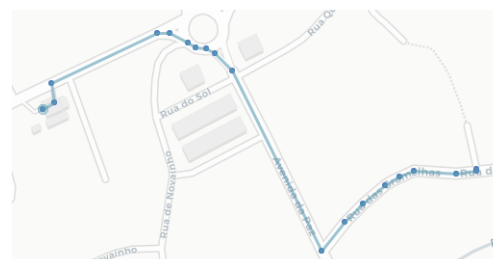


Figura 4.5: Rota com marcadores

Depois de ter a rota criada, é necessário criar um nodo móvel que corresponde ao utilizador que percorre a sua rota. Ao nodo móvel é associado um sensor garantindo, desta forma, a comunicação entre o nodo móvel e os sensores (Figura 4.6), bem como um ficheiro GPS no qual se encontra guardado a rota.

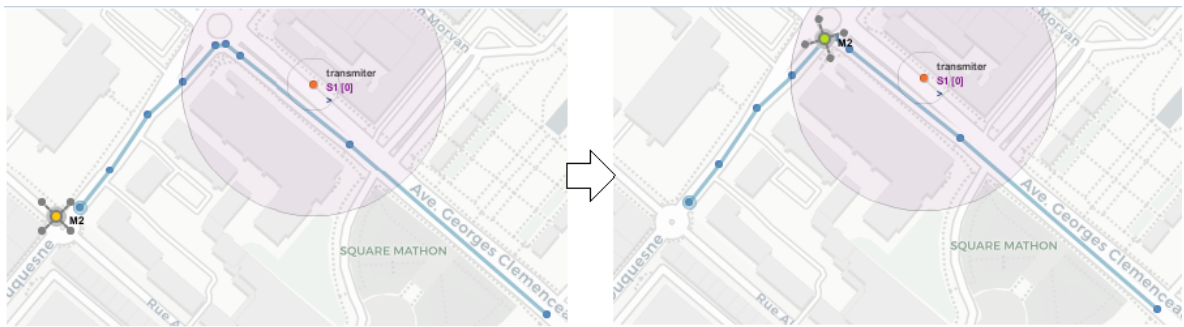


Figura 4.6: Simulação do movimento de um nodo móvel ao longo de uma rota

4.2 CENÁRIO 1

Concluída a ambientação ao CupCarbon e, em particular, como são programados os diversos sensores, criou-se um primeiro cenário de teste. Com este primeiro cenário pretende-se simular a tomada de decisão por um utilizador relativamente à escolha da rota a seguir com base no valor de um dado atributo. Neste primeiro caso, o valor do atributo simulado corresponde ao valor da temperatura, para isso foram criadas duas rotas:

- uma rota principal denominada rota1 (Figura 4.7), pela qual o utilizador (sensor móvel) inicia o seu percurso:

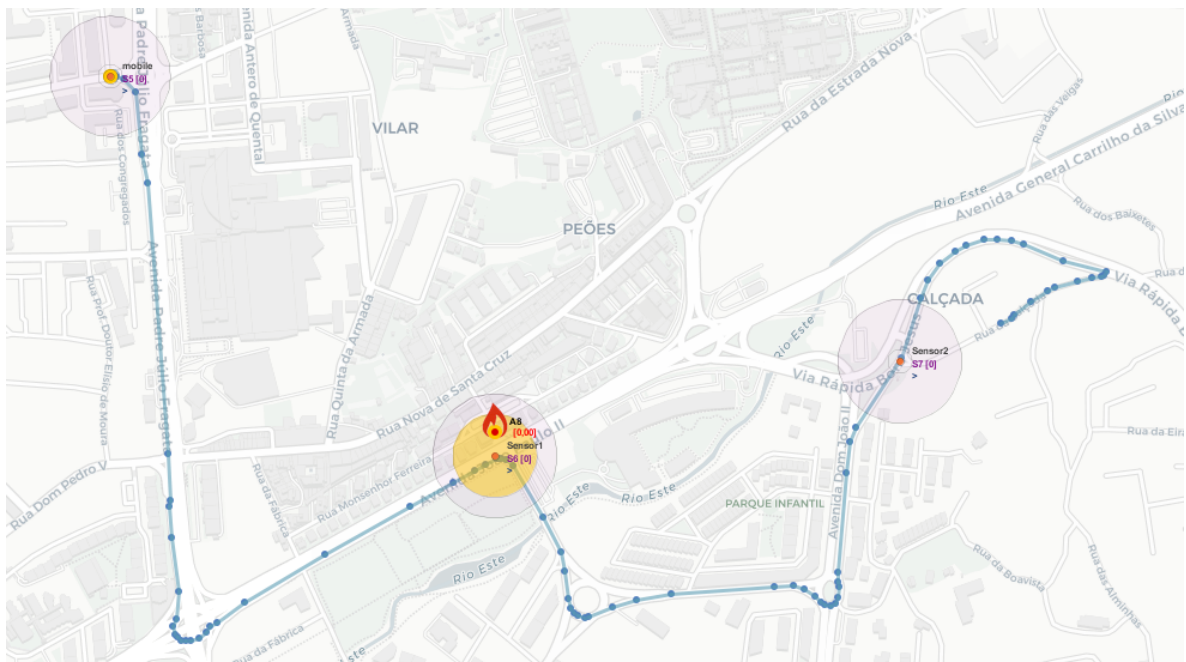


Figura 4.7: Rota principal (rota1)

- e uma rota alternativa denominada rota2 (Figura 4.8), para quando o utilizador chegar à bifurcação poder optar por qual rota prosseguir com base no valor do atributo considerado:

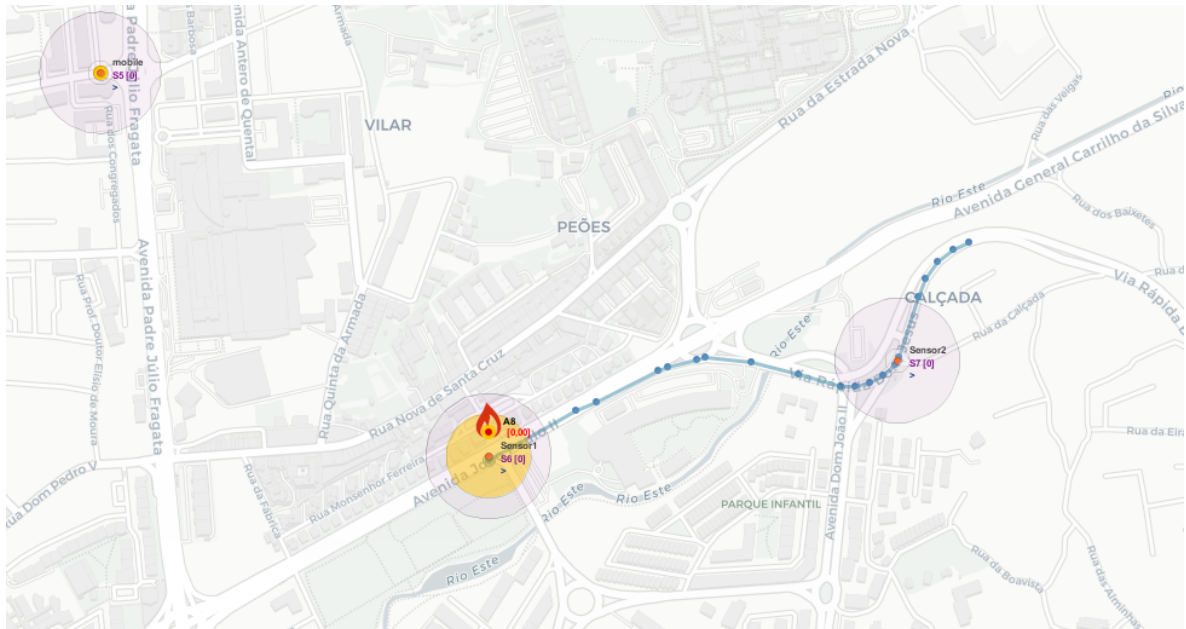


Figura 4.8: Rota alternativa (rota2)

Para além das duas rotas, foram adicionados ao cenário 1 três sensores:

- sensor *mobile/móvel*, sensor representativo de um utilizador que vai percorrer o cenário. Este sensor quando inicia o percurso tem associado um ficheiro GPS que corresponde à rota1, isto significa que vai-se deslocar ao longo dessa rota.

Para além disso, tem uma *script* associada (Código 4.1) permitindo assim a comunicação com outros sensores. Quando este sensor aproxima-se do raio do sensor **Sensor 1** recebe uma mensagem (Código 4.1, linha 4) que contém o *id* do sensor que recebe a mensagem bem como a rota pela qual deve seguir, sendo esse valor imprimido de seguida (Código 4.1, linha 6). Caso o valor de *\$rota* seja diferente do valor da rota que se estava a percorrer até esse momento, é necessário trocar o ficheiro GPS pela qual o sensor se estava a deslocar. Isto significa que se receber uma mensagem a dizer *route2* o sensor *mobile* deve trocar o ficheiro GPS da rota1 para a rota2.

```

1   loop
2   wait
3   read msg
4   rdata $msg x id rota
5   if($x==B)

```

```

6         print $id : $rota
7         set last B
8         route $rota
9     end

```

Código 4.1: Código-fonte do sensor mobile

- sensor **Sensor1**, este sensor é o responsável por tomar a decisão sobre qual rota recomendar ao utilizador. Tem associado a si um sensor Gás, responsável por gerar valores aleatórios para a temperatura com:

- média = 20°C;
- desvio padrão = 4;
- periodicidade = 1 segundo.

A cada segundo é lido um novo valor para a temperatura, quando o sensor *mobile* se aproxima deste sensor é lido o valor da temperatura, se este for inferior a 20°C vai indicar ao sensor *mobile* para prosseguir pela rota2, caso contrário indica ao sensor *mobile* para continuar na rota1.

De seguida, apresenta-se o código Senscript associado a este sensor (Código 4.2). Inicialmente é verificado se algum valor foi monitorizado, *if(\$v!=X)*, onde *v* é uma variável do tipo *areadsensor* que representa um sensor gás ao qual foi associado o atributo da temperatura. Caso tenha sido monitorizado algum valor, ou seja, *\$v* é diferente de *X*, é lido o valor monitorizado (*temp*) e efetuado o arredondamento às unidades.

Caso o valor lido para a temperatura seja inferior a 20 (Código 4.2 linha 7), é criada e enviada uma mensagem para o sensor *mobile* para este seguir pela rota2, caso contrário é recomendado seguir pela rota1. Por fim, como este código está a ser executado num ciclo é esperado um segundo (*delay 1000*), até voltar a executar.

```

1     loop
2     areadsensor v
3     if($v!=X)
4         rdata $v a b temp
5         int temp $temp
6         print $temp
7     if($temp < 20)
8         data msg B SENSOR25 route2
9         send $msg 5
10    else
11        data msg B SENSOR25 route1
12        send $msg 5
13    end
14    delay 1000

```

Código 4.2: Código-fonte do Sensor1

- Sensor **Sensor2**, este sensor é usado quando é recomendado ao utilizador para prosseguir pela rota2, sendo responsável por reconduzir o sensor *mobile* para a rota principal, rota1 (Código 4.3 linha 5), quando chegar ao fim da rota2.

```

1      loop
2      wait
3      read m
4      if($m==A)
5          data d C SENSOR25 route1
6          send $d
7      end
8      delay 1000

```

Código 4.3: Código-fonte do Sensor2

4.2.1 Resultados da simulação

Com a realização de uma simulação preliminar na ferramenta Cupcarbon, pretende-se verificar o processo de tomada de decisão e de recomendação de rotas. Pretende-se simular o deslocamento de um utilizador e, de seguida, uma tomada de decisão com base num atributo pretendido. Assumindo que o valor gerado para a temperatura seja superior a 20°C, é comunicado ao sensor mobile para seguir pela rota1 (Figura 4.9), o sensor mobile recebe a mensagem e vai prosseguir pela rota1 (Figura 4.10).

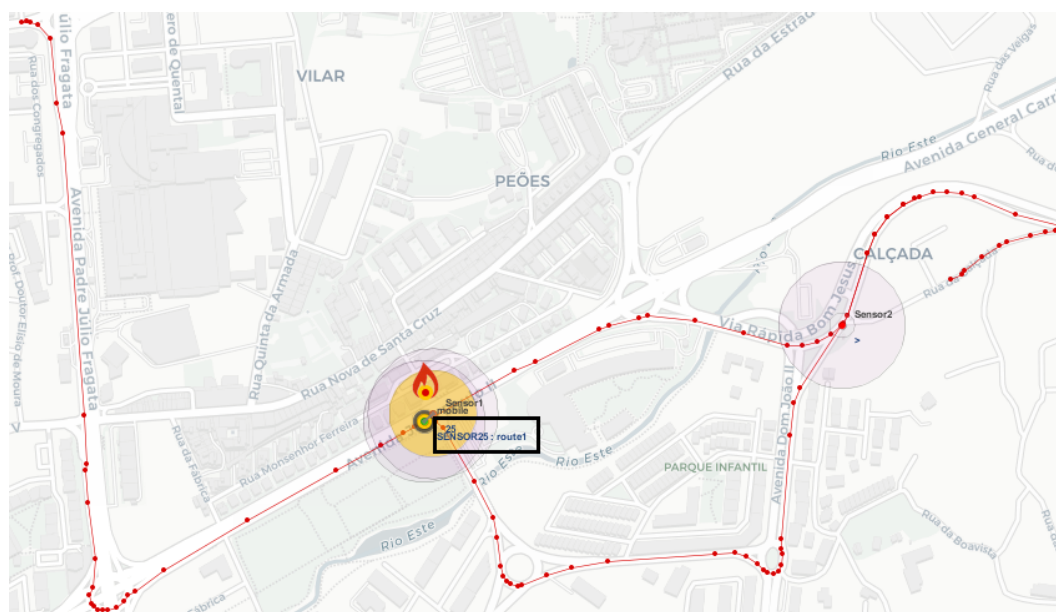


Figura 4.9: Comunicação ao sensor mobile da rota recomendada (rota1)

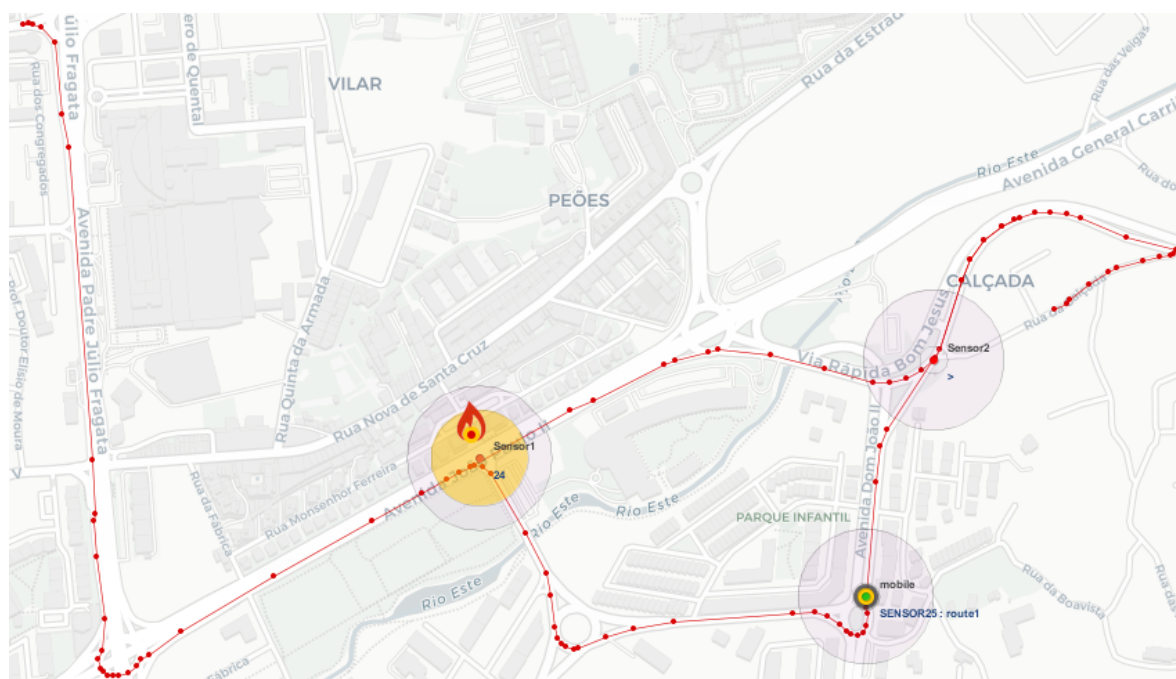


Figura 4.10: Deslocamento do sensor mobile pela rota anteriormente recomendada

O cenário de simulação construído e os resultados obtidos permitiram verificar a adequabilidade do processo inicial para a tomada de decisão. Por forma a aumentar a versatilidade da solução, introduziu-se a definição pelo utilizador de pesos associados aos atributos

considerados. Assim, um utilizador pode decidir que atributos são da sua preferência na escolha da rota, atribuindo-lhes um valor em conformidade.

4.3 CENÁRIO 2

Foi desenvolvido um segundo cenário para simular a escolha de rota realizada por um utilizador que, ao deparar-se com um cruzamento no seu percurso, tem de tomar uma decisão com base nos atributos pretendidos. Para isso, foram criadas novamente duas rotas e a cada uma das rotas foi associado um sensor responsável por gerar os valores dos atributos para essa rota. Esses valores são comunicados através do envio de uma mensagem a cada segundo para um sensor central. Cada um dos sensores associados às rotas é responsável por gerar os valores relativos aos seguintes três atributos:

- **temperatura**, continua a considerar-se um evento natural que pretende simular o valor da temperatura sentida;
- **iluminação**, trata-se de outro evento natural que pretende indicar se a rota possui ou não iluminação, podendo tomar os seguintes valores:
 - zero - não tem iluminação;
 - um - tem iluminação.
- **cobertura de sinal Wi-Fi**, este atributo é simulado através da geração de um número aleatório que indica a qualidade do sinal recebido, podendo o seu valor variar entre:
 - zero, má qualidade;
 - um, fraca qualidade;
 - dois, qualidade aceitável;
 - três, boa qualidade;
 - quatro, ótima qualidade.

Na tabela 4.1 são apresentados alguns dos valores gerados para cada um dos atributos usando a ferramenta de simulação CupCarbon:

Tabela 4.1: Exemplo dos valores gerados na simulação do Cenário 2

Sinal Wi-Fi	Temperatura	Iluminação
1	18,911	1
2	17,855	1
0	14,244	0
1	24,454	0

Tabela 4.1 – continuação da página anterior

Sinal Wi-Fi	Temperatura	Iluminação
0	19,755	0
2	27,890	0
2	24,587	1
0	12,599	1
4	16,683	0
1	22,838	1
3	27,465	0
1	16,611	0
2	20,423	1

Para a simulação deste cenário foram criadas duas rotas e usados quatro sensores (Figura 4.11):

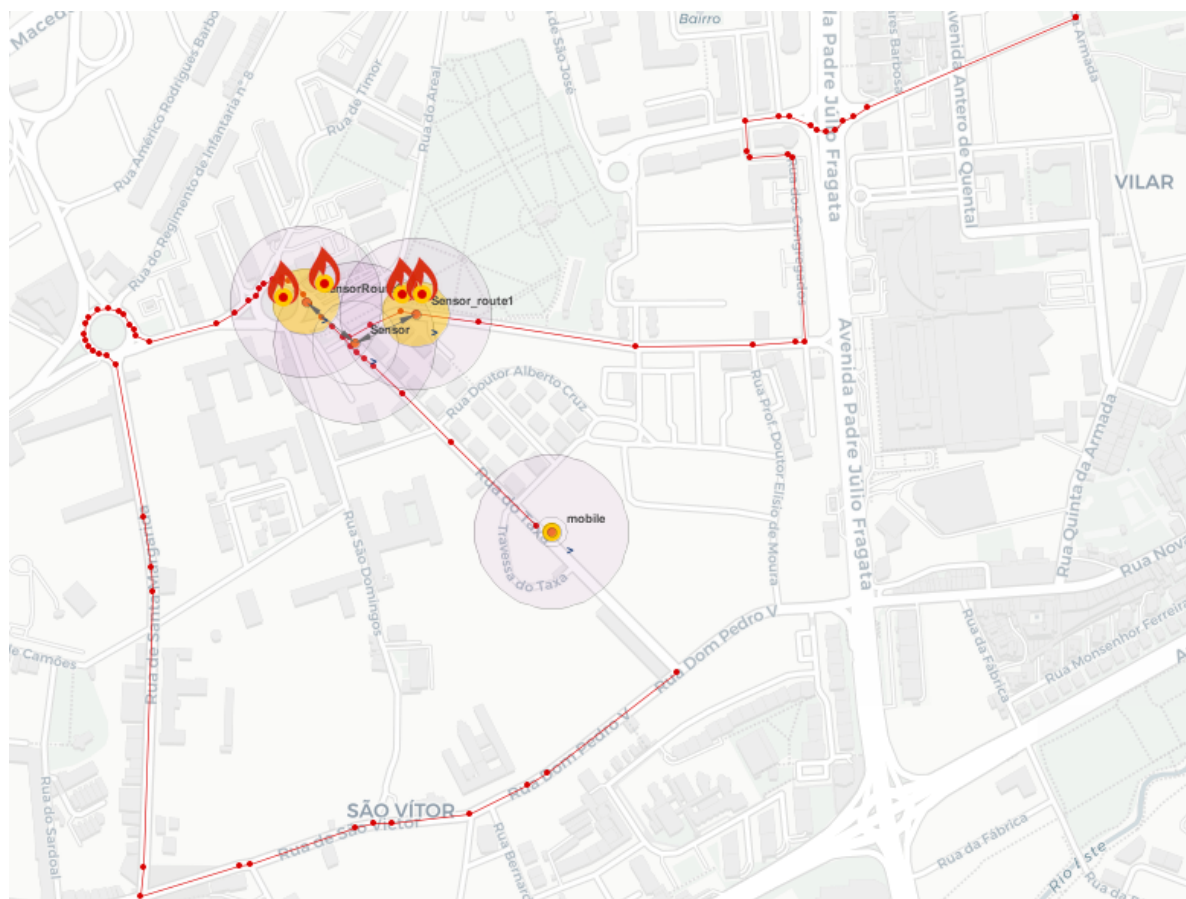


Figura 4.11: Cenário 2

- rota 1, corresponde à rota que vira à direita na bifurcação;
- rota 2, rota que continua em frente na bifurcação e seguindo posteriormente para a esquerda;
- sensor *mobile/móvel*, sensor representativo do utilizador, sendo responsável pelo deslocamento ao longo do cenário. Este sensor usa o mesmo código-fonte que o sensor *mobile* do cenário 1;
- sensor **Sensor_route1**, sensor associado a rota1 responsável por gerar valores para os atributos simulados, comunicando-os a cada segundo ao sensor central. Como dito anteriormente, este sensor tem associado dois eventos naturais que simulam a temperatura (Código 4.4 linha 6) e a luminosidade (Código 4.4 linha 7), sendo o valor da cobertura Wi-Fi gerado aleatoriamente (Código 4.4 linha 2);

```

1      loop
2      randb wifi 0 4
3      areadsensor v
4      if($v!=X)
5          rdata $v a b temp d ilum
6          int temp $temp
7          int ilum $ilum
8          print $wifi $temp $ilum
9          data p $wifi $temp $ilum
10         send $p 4
11     end
12     delay 1000

```

Código 4.4: Código-fonte do Sensor_route1

- sensor **Sensor_route2**, sensor associado a rota2 possuindo a mesma função que o sensor anterior;
- sensor **Sensor**, sensor central do cenário responsável por receber as mensagens dos sensores associados a cada uma das rotas e comparar os valores recebidos com os valores pretendidos pelo utilizador. Desta forma, percebe qual a rota mais adequada ao pretendido pelo utilizador, por fim comunica a rota ao sensor *mobile*.

4.3.1 Simulação do cenário

Em termos de simulação, o sensor móvel vai começar a percorrer a rota até chegar ao sensor central, onde a cada segundo são recebidos os valores de temperatura, cobertura Wi-Fi e iluminação (Tab.4.1) dos dois sensores associados a cada uma das rotas. De seguida, o

sensor móvel vai receber a recomendação de uma rota com base nas escolhas feitas pelo perfil do utilizador.

Desta forma, foram criados dois perfis de utilizadores pedestres que pretendam receber uma recomendação de rota:

- Perfil A
Representa um jovem que pretende fazer jogging à noite enquanto ouve música online. Sendo assim, pretende escolher como atributos preferenciais a presença de iluminação ao longo da rota, atribuindo-lhe um peso de 60% e pretende ainda que a cobertura Wi-Fi seja superior a 2 (qualidade aceitável) para que possa ouvir música online, atribuindo um peso de 20% . Por fim, pretende que a temperatura seja superior a 16°C, atribuindo um peso de 20%.
- Perfil B
Representa um senhor de idade que pretenda dar um passeio matinal pela cidade. Neste caso não tem interesse em que haja iluminação, pretende que a cobertura Wi-Fi seja superior a 1 (fraca qualidade) e que a temperatura não esteja elevada, ou seja, inferior a 14°C.

Para simular a execução dos dois perfis no Cupcarbon, na script do sensor central foi definida uma variável *perf* que representa o perfil que se está a simular. Esta variável tomará os valores A ou B, que resultará na execução do perfil do utilizador correspondente.

- Em termos de execução, quando o sensor móvel aproxima-se da área do sensor central, este vai processar a mensagem recebida de cada um dos sensores associados a cada rota. Isto significa que o sensor central recebe duas mensagens a cada segundo com a seguinte estrutura, **#Wi-Fi#Temperatura#Iluminação**, procedendo da seguinte forma:
 - caso o valor recebido para a cobertura de sinal Wi-Fi seja superior a 2 é somado o valor 20 (correspondente ao peso respetivo) a uma variável global chamada *total1* (*total2* para a mensagem recebida do sensor associado à rota2), sendo que esta variável foi anteriormente inicializada a zero;
 - caso o valor recebido para a iluminação seja um, significa que a rota possui iluminação, é somado o valor 60 à variável *total1* (*total2* para a rota2), correspondendo ao peso do atributo em questão;
 - caso o valor para a temperatura seja superior a 16°C, de igual forma, é somado o valor 20 à variável *total1* (*total2* para a rota2).

Os mesmos cálculos são realizados para a mensagem recebida relativa ao sensor associado à rota2.

No final, são comparados os valores das duas variáveis *total1* e *total2*. No caso de o valor da variável *total1* ser superior ao da variável *total2*, o nodo móvel prossegue

pela rota1 pois é a que mais satisfaz os requisitos pretendidos pelo utilizador, caso contrário, segue pela rota2.

Na Figura 4.12 é ilustrada uma das várias simulações realizadas. Neste caso, o sensor móvel ao chegar à bifurcação recebe a recomendação para seguir pela rota1, prosseguindo por essa rota.



Figura 4.12: Na imagem da esquerda o sensor mobile recebe a indicação para seguir pela rota1, na imagem da direita o sensor mobile já a meio do caminho na rota1

- No caso do perfil B, este dá 65% de importância da temperatura ser inferior a 14°C e 35% da cobertura de sinal Wi-Fi ser superior a 1. Em termos de aplicação é realizado o mesmo que para o perfil A apenas variando o peso dado aos atributos recebidos em cada mensagem,
 - caso o valor recebido para a cobertura de sinal Wi-Fi seja superior a 1, soma-se 35 à variável global chamada *total1* (*total2* para a mensagem recebida do sensor associado à rota2) estando esta previamente inicializada a zero;
 - é ignorado o valor recebido para a iluminação;
 - caso o valor para a temperatura seja inferior a 14°C , soma-se 65 à variável *total1*.

No final são comparados os valores das duas variáveis *total1* e *total2*, a variável que tiver valor superior indica a rota pela qual o nodo móvel prossegue.

Tal como para o perfil A, foram realizadas várias simulações para o perfil B. A Figura 4.13 ilustra uma das várias simulações realizadas, em que o sensor móvel ao chegar à bifurcação recebe a recomendação para seguir pela rota2, prosseguindo por essa rota. De realçar que, em ambos os perfis, foram realizadas várias simulações havendo casos em que para o perfil B também recebia recomendação para seguir pela rota1. Contudo como uma forma de mostrar o deslocamento do sensor móvel por ambas as rotas existentes, optou-se por mostrar o sensor móvel a deslocar-se pela rota2.



Figura 4.13: O sensor *mobile* recebe a indicação para seguir pela rota2 (esquerda). O sensor *mobile* já se encontra a percorrer a rota2 (direita)

4.4 RESUMO DO CAPÍTULO

Ao longo deste capítulo foi descrita a ambientação à plataforma de simulação CupCarbon, foram apresentadas as funcionalidades disponíveis tais como os sensores disponíveis e a forma como associar um determinado comportamento a esses sensores. Sendo também ilustrado como é feita a criação de rotas no mapa da cidade e por fim simulados dois cenários de teste e analisados os resultados para esses cenários.

No próximo capítulo, será apresentado um caso de estudo mais completo, envolvendo um maior número de atributos e efetuada uma discussão dos resultados obtidos.

CASOS DE ESTUDO - CENÁRIO FINAL

Ao longo deste penúltimo capítulo é apresentado o cenário final implementado na ferramenta Cupcarbon. Este cenário contempla três sub-cenários, tendo cada um deles associado um perfil de utilizador. Desta forma, pode-se simular um caso específico de um utilizador que pretende obter uma recomendação de rota, sendo também apresentados os comportamentos que foram associados aos sensores para este cenário. No final do capítulo, apresenta-se uma discussão dos resultados obtidos. Por fim, é abordado o que seria necessário para implementar este modelo na realidade e o seu desenvolvimento.

5.1 ATRIBUTOS SIMULADOS

Quando se realizou uma observação mais cuidadosa do cenário elaborado anteriormente, verificou-se que a tomada de decisão por parte do utilizador tinha a ver com os valores gerados de forma aleatória num ponto específico da rota, nesse momento. Por exemplo, se a temperatura num dado momento é 17°C isso não significa que a temperatura seja a mesma ao longo de toda a rota ou se a cobertura Wi-Fi for 2 não existe garantia que ao longo da rota hajam zonas sem cobertura.

Para tornar o cenário mais realista e versátil, optou-se por considerar uma rede de sensores colocados ao longo da rota capazes de enviar as suas leituras para o próximo sensor, de forma a existir comunicação ao longo da rota dos valores gerados.

Desta forma, para o terceiro e último cenário, foram tidos em conta uma maior variedade de atributos com o pretexto de fornecer um melhor auxílio ao utilizador na altura de escolher uma rota. De seguida, descrevem-se os atributos simulados e a forma como estes são transmitidos ao longo da rota.

5.1.1 *Atributos variáveis*

- **cobertura Wi-Fi** - representa o valor da cobertura Wi-Fi num sensor sendo gerado aleatoriamente. Este valor corresponde a um número entre zero, representando uma

pior qualidade do sinal recebido e cinco, representando uma melhor qualidade do sinal. Este valor é transmitido ao próximo sensor que gera o seu próprio valor de cobertura Wi-Fi e calcula a média com o valor previamente recebido, passando para o sensor seguinte o valor da média obtido;

- **percentagem da rota com cobertura Wi-Fi** - representa a percentagem da rota que possui cobertura Wi-Fi, pois para alguns casos pode ser importante saber se toda a rota possui cobertura ou não. Desta forma, quando é gerado o valor da cobertura Wi-Fi, se esse valor for igual a zero, é incrementada uma variável chamada *n_wifi*, previamente inicializada a zero, sendo o seu valor enviado para o próximo sensor que procederá de igual modo. Com isto, pretende-se saber o número de sensores ao longo da rota que não possuem cobertura Wi-Fi;
- **média da temperatura** - indica a média da temperatura ao longo da rota. Foram gerados ficheiros *evt*, sendo que cada um contém valores gerados aleatoriamente para a temperatura, variando apenas o intervalo da mesma. Assim sendo, para cada rota os valores gerados para a temperatura encontram-se num intervalo similar, evitando desta forma discrepâncias.

Desta forma, haverá rotas com temperaturas mais altas e rotas com temperaturas mais baixas. Em cada sensor é lido o valor do evento natural correspondente ao valor da temperatura e enviado para o sensor seguinte. Este lê o seu valor da temperatura e faz a média com o valor recebido enviando o valor da média para o sensor seguinte;

- **intervalo de temperatura** - tem como objetivo indicar ao utilizador o valor mínimo e máximo da temperatura ao longo da rota, por forma a complementar a informação sobre a temperatura média. Note-se que, o valor da média pode ser subjetivo, pois pode ocultar amplitudes térmicas não desejadas pelo utilizador. De facto, podem existir pessoas que são incapazes de caminhar se a temperatura for superior a um determinado valor. Desta forma, para auxiliar o utilizador para além de apresentar o valor da média da temperatura ao longo da rota, é apresentado o intervalo de valores para a temperatura sendo mais um atributo a ter em conta.

Assim sendo, no momento em que é gerado o valor da temperatura, este valor é comparado com o atual valor máximo e mínimo da rota, caso o valor seja superior ao máximo então essa temperatura passa a ser a máxima, caso seja inferior à temperatura mínima, essa passa a ser a temperatura mínima enviada para o próximo sensor.

- **iluminação** - indica a percentagem da rota que possui iluminação. Para isso, em cada sensor é lido o valor gerado para a iluminação, podendo tomar os seguintes valores:
 - zero - zona onde se encontra localizado o sensor não possui iluminação;
 - um - zona onde se encontra localizado o sensor possui iluminação;

Tal como os valores da cobertura de sinal Wi-Fi, em cada sensor é gerado um valor aleatório para representar o valor da iluminação. De seguida, é verificado se esse valor é igual a um, caso o seja, é incrementada uma variável que conta o número de sensores em que existe iluminação e enviado o valor dessa variável para o sensor seguinte. No final, divide-se o valor da variável correspondente ao número de sensores em que existe luminosidade pelo número de sensores na rota obtendo-se a percentagem de rota com iluminação.

5.1.2 Atributos Fixos

Para além dos atributos variáveis, existem cinco atributos fixos associados a cada rota:

- **tamanho da rota** - indica o número de metros da rota sendo importante para os utilizadores possuírem uma ideia inicial do tamanho do percurso para ajudar na sua decisão;
- **número de sensores** - indica o número de sensores usados para cada rota, sendo usado para o cálculo da percentagem de rota que possui cobertura Wi-Fi e para a percentagem de rota com iluminação. Como abordado anteriormente, existe uma variável que indica o número de sensores ao longo da rota cujo valor de cobertura Wi-Fi é zero, no final esse valor é subtraído ao número de sensores da rota e dividido esse valor pelo número de sensores presentes na rota obtendo-se a percentagem da rota com cobertura Wi-Fi;
- **percurso circular** - é usado para os utilizadores que pretendem começar uma rota num determinado ponto e saber se conseguem terminar a rota no ponto onde a rota foi iniciada;
- **espaços abertos** - indica se a rota escolhida possui ou não a presença de espaços abertos;
- **interesse turístico** - indica se ao longo da rota existem pontos de interesse turístico ou mesmo zonas comerciais.

Como explicado anteriormente, os valores destes atributos vão sendo transmitidos ao longo dos vários sensores, sendo que entre cada sensor é enviada uma mensagem (Figura 5.1) que toma a seguinte forma:

Wifi#n_Wifi#temp#temp_Min#temp_Max#iluminacao#id

- *Wifi* indica a cobertura de sinal Wi-Fi;
- *n_Wifi* indica o número de sensores sem sinal Wi-Fi;

- *temp* indica o valor da média da temperatura;
- *temp_Max* e *temp_Min* indicam a temperatura mínima e máxima respetivamente;
- *iluminação* indica o número de sensores com iluminação ao longo da rota;
- *id*, indica o *id* do sensor que vai receber esta mensagem.

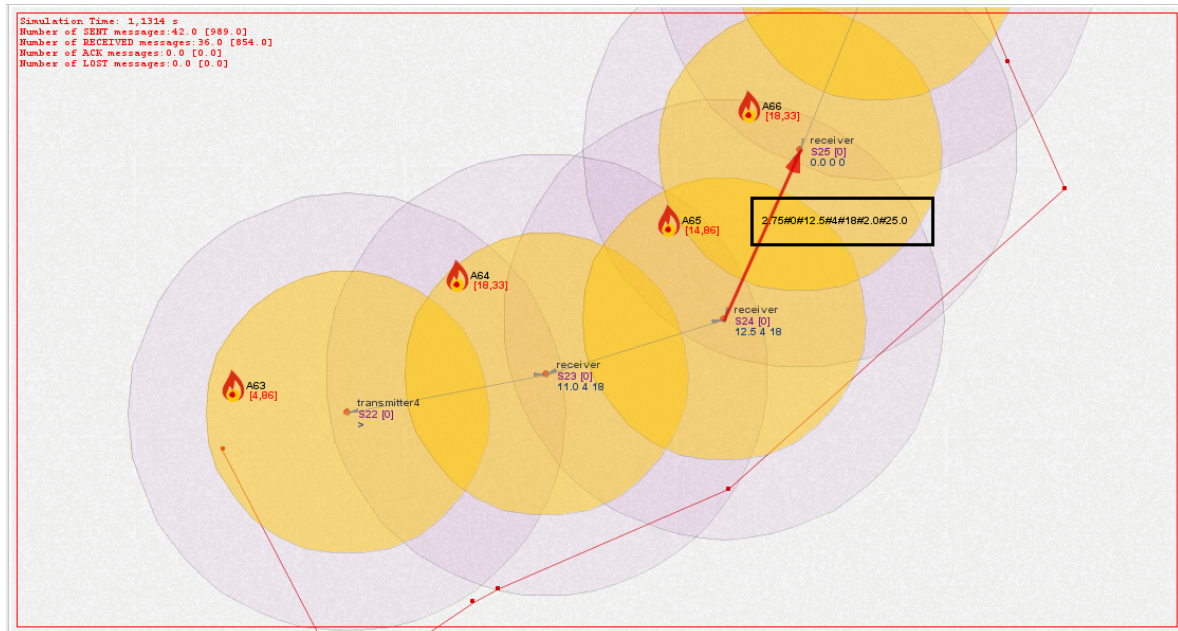


Figura 5.1: Exemplificação da troca de mensagens entre sensores

Existem nos cenários sensores denominados "Centrais", colocados em locais de bifurcação de rotas, onde o utilizador tem de decidir qual a rota pela qual pretende seguir com base nos atributos desejados. Este sensor vai receber duas mensagens (no caso em que existem duas rotas), a primeira contém todos os valores monitorizados relativos a uma rota e a segunda todos os valores monitorizados relativamente a rota alternativa. Seguidamente, é feita uma análise a cada um dos atributos contidos na mensagem, com base nas escolhas do utilizador, que previamente define o peso que pretende dar a cada um dos atributos. Para esse efeito foi definido que cada utilizador possui uma percentagem de 100%, sendo esta distribuída por cada um dos atributos dando naturalmente mais percentagem aos atributos preferenciais.

No sensor central são analisados os valores recebidos para cada um dos atributos e comparados com os valores definidos pelo utilizador previamente. Se os valores verificarem o pretendido pelo utilizador, suponhamos que pretende uma temperatura média superior a 15°C e a temperatura média da rota1 é 18°C, é adicionado a uma variável *Total_rota1* associ-

ada, neste caso à rota₁, a percentagem que o utilizador especificou anteriormente. Se tiver dado uma percentagem de 45% à temperatura, a variável *Total_rota1* passa a ser 45.

No final, são comparados os valores das variáveis *Total_rota1* e *Total_rota2*, no caso em que existe duas rotas. Se o valor da variável *Total_rota1* for superior ao da variável *Total_rota2* é enviada uma mensagem para o sensor móvel, que representa o utilizador, a indicar para este seguir pela Rota₁. Caso contrário, é enviada uma mensagem ao utilizador para seguir pela rota₂.

5.2 PERFIS DE UTILIZADOR

Para a simulação foram criados três perfis, com o objetivo de representar três indivíduos que pretendem obter uma recomendação de rota, associados a diferentes locais.

5.2.1 Perfil A - Local 1

O primeiro perfil simula um turista que pretende caminhar de manhã na cidade de Braga e conhecer melhor os pontos de interesse da cidade. Além disso, deseja que ao longo da rota existam espaços abertos e espaços de interesse turístico como monumentos ou zonas comerciais. Para esta simulação foram criadas três rotas (Figura 5.2), tirando partido da capacidade de inclusão do Google Maps na ferramenta CupCarbon para ser possível ver e associar zonas com espaços abertos, bem como monumentos e zonas comerciais na cidade a essas rotas. De seguida apresentam-se as rotas criadas:



Figura 5.2: Rotas criadas para o Local1 do cenário final

- A rota 1, na Figura 5.2 a cor verde, possui uma distância total de 829m e passa junto ao Castelo de Braga e a espaços verdes, terminando no centro comercial BragaShopping;
- A rota 2, na Figura 5.2 a cor vermelha, é uma alternativa à rota 1 e possui uma distância de 937m, passando pela avenida Central de Braga onde existem várias zonas abertas com zonas verdes, passando também pelo Museu Nogueira da Silva e ficando perto do centro comercial BragaShopping;
- rota 3, na Figura 5.2 a cor preta, possui uma distância total de 1047m, passa perto da Sé de Braga e do Jardim de Santa Bárbara dando oportunidade ao utilizador de conhecer esses locais.

5.2.2 Sensores utilizados no cenário

Para este cenário final foram criados quatro *scripts*, tendo sido posteriormente associadas aos sensores permitindo que estes apresentem funcionalidades distintas. Estes sensores são aplicados para os três locais criados para simular cada um dos perfis. Na Figura 5.3 é

apresentado o mapa e os sensores para este primeiro perfil, sendo de seguida apresentada uma explicação das funcionalidades de cada uma das *scripts* criadas. De realçar, que para os restantes locais foram usadas as mesmas quatro *scripts*, procedendo apenas a pequenos ajustes consoante cada um dos perfis:

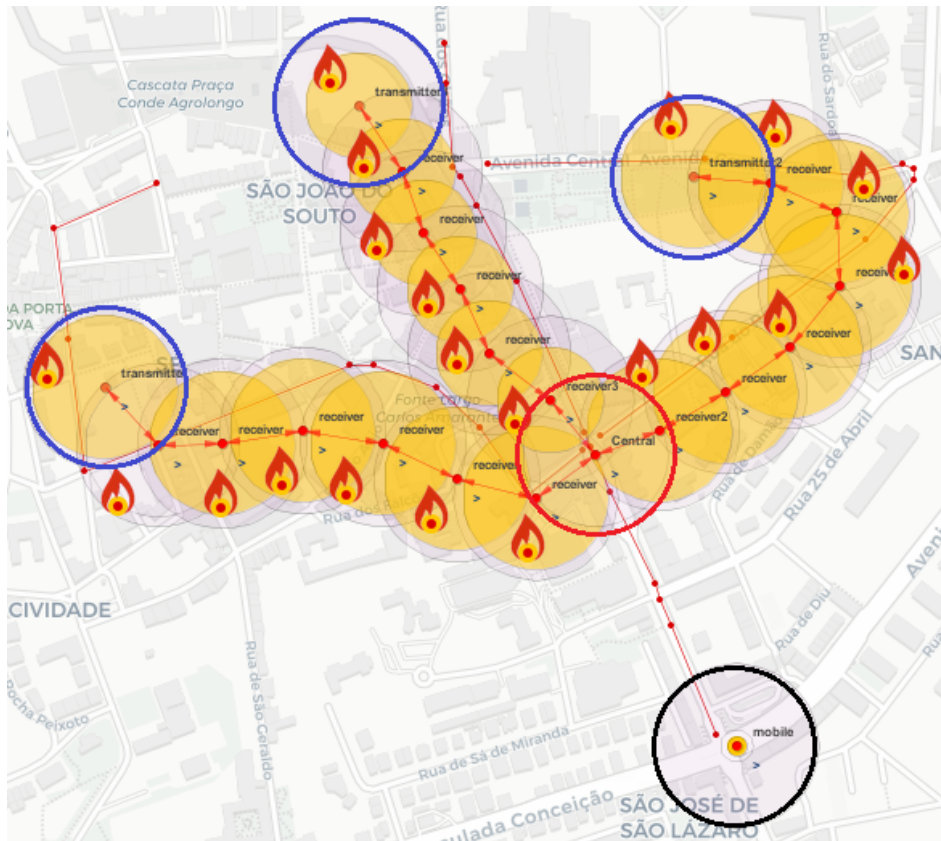


Figura 5.3: Mapa do primeiro local do cenário com os sensores utilizados

Para uma melhor distinção dos sensores e respetivas funcionalidades foram usadas cores para cada funcionalidade, nomeadamente:

- cor preta, representa o sensor *mobile/móvel* que representa o indivíduo que se desloca ao longo do cenário;
- cor vermelha, representa os sensores *centrais* responsáveis pela tomada de decisão relativamente a recomendação da rota;
- cor azul, representa os sensores *transmissores* onde são gerados os valores iniciais dos atributos;
- os restantes sensores são os sensores *recetores* que vão propagar as mensagens com os valores dos atributos.

De seguida são explicadas as funcionalidades de cada um dos sensores bem como as *scripts* correspondentes, criadas usando a linguagem *SenScript*.

Sensor Transmissor

Para este perfil foram usados três sensores transmissores, sendo o ponto de partida para os dados que são gerados e transmitidos ao longo da rota à qual estão associados. O código-fonte associado a estes sensores (Código 5.1), funciona da seguinte forma:

```

1      loop
2      randb wifi 0 5
3      set n_wifi 0
4      randb iluminacao 0 1
5      set per_iluminacao 0
6      areadsensor v
7      if($v!=X)
8          rdata $v a b temp
9          int temp $temp
10         set d 0
11         if($wifi == 0)
12             set n_wifi 1
13         end
14         if($iluminacao == 1)
15             set per_iluminacao 1
16         end
17         data e $wifi $n_wifi $temp $temp $temp $per_iluminacao 2
18         send $e
19         end
20         delay 1000

```

Código 5.1: Código-fonte do Sensor Transmissor

- 1º - são gerados valores aleatórios para a cobertura de sinal Wi-Fi (Código 5.1 linha 2) e para a iluminação (Código 5.1 linha 4). É também definido que a variável *n_wifi* (número de sensores cujo o valor da cobertura Wi-Fi é zero) é iniciada a zero.
- 2º - é lido o valor do evento natural associado à temperatura recorrendo se para isso ao *areadsensor*, sendo feito o arredondamento do mesmo para um valor inteiro (Código 5.1 linha 9).
- 3º - é analisado se o valor da variável *wifi* é zero (Código 5.1 linha 11), caso o seja significa que nesse ponto não existe cobertura de sinal Wi-Fi e a variável *n_wifi* é incrementada.
- 4º - é criada uma mensagem contendo o valor de todos os atributos (Código 5.1 linha 17). Tratando-se do primeiro sensor a gerar os dados o valor da temperatura máxima

e mínima é igual à temperatura média. O valor "2" no final da mensagem corresponde ao *id* do sensor que irá receber a mensagem, neste caso a mensagem é transmitida do *sensor1* para o *sensor2*.

- 5° - é esperado um segundo e o código volta a executar a partir do início (Código 5.1 última linha).

Sensor Recetor

Entre os sensores transmissores e o sensor Central, responsável pela tomada de decisão, existem vários sensores denominados *receiver*/recetores responsáveis pela transmissão de mensagens ao longo de toda a rota. Devido a quantidade de código ser bastante extensa o mesmo encontra-se disponível no Anexo A.2. De seguida, explica-se o funcionamento da *script* associada a estes sensores:

- 1° - é recebida uma mensagem com o seguinte formato "**rdata \$s1 wifi n_wifi temp tempMin tempMax iluminacao id**", contendo os dados recebidos do sensor anterior;
- 2° - é lido o valor do evento natural associado à temperatura recorrendo-se para isso ao *areadsensor*. Seguidamente, é comparado esse valor com o valor recebido para a temperatura mínima e caso o valor lido seja menor que o valor mínimo recebido é atualizado o valor da temperatura mínima, o mesmo é feito para a temperatura máxima.

Depois de comparar os dois valores é feita a média da temperatura lida com a temperatura média recebida passando a ter a temperatura média atual para enviar para o próximo sensor;

- 3° - é gerado um valor aleatório para a cobertura de sinal Wi-Fi e para a iluminação. Posteriormente é feita a média entre os valores gerados e os valores recebidos do sensor anterior. Para a cobertura de sinal Wi-Fi, verifica-se se o valor é igual a zero e caso o seja é incrementado o valor da variável *n_wifi* que contém o número de sensores cujo sinal Wi-Fi é zero;
- 4° - é criada a seguinte mensagem, *data p \$wifi2 \$n_wifi \$temp2 \$tempMin \$tempMax \$iluminacao2 \$id*, sendo que o valor da variável *id* é incrementado antes da mensagem ser enviada para o sensor seguinte.

Por fim, a mensagem é enviada, *send \$p \$id*, e o sensor fica a espera até receber uma nova mensagem.

Sensor Mobile

O sensor *mobile/móvel* representa o utilizador que se desloca ao longo da rota, possui um código extremamente simples (Código 5.2). Este sensor vai-se deslocando ao longo da rota até chegar ao raio de mensagem do sensor Central. Nesse momento, vai receber uma mensagem que lhe indica qual a rota pela qual ele deve prosseguir (Código 5.2 linha 3). Para isso é executado o código *route \$rota* em que a rota recebida na mensagem, passa a ser a rota pela qual prossegue. O valor de rota pode ser "route1", "route2", "route3" para este primeiro perfil.

```

1      loop
2      wait
3      read d
4      rdata $d x n rota
5      if ($x==B)
6          print $n : $rota
7          set last B
8          route $rota
9      end

```

Código 5.2: Código-fonte do Sensor Mobile

Sensor Central

Neste primeiro local existe um sensor Central. Este é o componente mais importante da simulação pois é o sensor responsável pela recomendação das rotas. Nesta tarefa, tem em conta os atributos definidos pelo utilizador e os seus respetivos pesos para tomar a decisão sobre qual a melhor rota para o utilizador prosseguir. Para isso, são realizados os seguintes passos:

- 1º - são recebidas três mensagens cada uma contendo os dados relativos a uma das rotas;
- 2º - são criadas e inicializadas nove variáveis:
 - *total1*, *total2* e *total3* estas variáveis vão acumular os pesos dos atributos que se verifiquem para cada uma das rotas;
 - *numSenR1*, indica o número de sensores presentes na rota 1 para o caso do primeiro sensor central o seu valor é 5;
 - *numSenR2*, indica o número de sensores presentes na rota 2 (alternativa à primeira rota), neste caso o valor é 6;
 - *numSenR3*, indica o número de sensores presentes na rota 3 (alternativa à primeira rota), para o caso do primeiro sensor central o seu valor é 6;

- *distance1*, indica a distância da rota1 o seu valor é 829m;
 - *distance2*, indica a distância da rota2 o seu valor é 937m;
 - *distance3*, indica a distância da rota3 o seu valor é 1047m;
- 3º - são analisados os valores de todos os atributos recebidos na mensagem relativa a uma rota e comparados os valores dos atributos com o pretendido pelo utilizador. Caso o valor desses atributos verifique o pretendido, é adicionado à variável *total* relativa a essa rota o peso que inicialmente o utilizador lhe deu, realiza-se o mesmo para a mensagem recebida para as restantes rotas;
 - 4º - é comparado o valor das três variáveis *total* de cada rota, sendo enviada uma mensagem ao sensor móvel para seguir pela rota cuja variável correspondente tenha o maior valor. Em caso de igualdade entre a rota1 e uma das alternativas, é dado prioridade à rota principal neste caso, rota1. Se a rota2 e rota3 tiverem o mesmo valor é dado prioridade à rota2.

Depois de feita uma explicação sobre os tipos de sensores usados no cenário, considere-se de novo o perfil A. É sabido que este utilizador pretende conhecer a cidade e visitar pontos de interesse turístico incluindo espaços abertos como parques ou jardins. Além destes atributos, este utilizador pretende que a temperatura média seja superior ou igual a 13°C e que preferencialmente o intervalo de temperaturas esteja entre os [12°C, 17°C]. Adicionalmente, pretende que a cobertura de sinal Wi-Fi seja superior a 2 para poder consultar informações sobre a cidade e que pelos menos 70% da rota possua cobertura Wi-Fi. Tratando-se de um passeio matinal não dá qualquer importância à iluminação. Pretende ainda que a distância seja inferior a 1,5km.

Na Tabela 5.1 encontram-se resumidos os valores pretendidos bem como o peso correspondente atribuído.

Tabela 5.1: Dados relativos ao perfil A

Atributos	Valores pretendidos	Peso atribuído
Temperatura média	13°C	15%
Intervalo de temperatura	12°C – 17°C	25%
Cobertura Wi-Fi	2	12%
Percentagem da rota com Wi-Fi	70%	8%
Iluminação	—	—
Distância	<=1,5km	10
Espaços abertos	Sim	15
Interesse Turístico	Sim	15
Percurso Circular	—	—

Para simular este cenário no Cupcarbon foi necessário no sensor central adequar os valores aos pretendidos pelo utilizador e fazer a atribuição de pesos em conformidade. Comparando as três rotas criadas é possível observar que todas elas vão de encontro ao pretendido pelo utilizador apresentando espaços abertos como o "Jardim de Santa Bárbara", locais de interesse turístico como a "Sé de Braga" e a presença de um centro comercial "Bragashopping". Sendo que este indivíduo dá importância à presença de espaços abertos e pontos de interesse turístico, as variáveis *total* que acumulam o peso dos atributos para cada uma das rotas começam com o valor 30, pois todas verificam o critério do utilizador.

Sendo assim, o sensor móvel começa a deslocar-se ao longo da rota chegando à bifurcação, ver Figura 5.4.

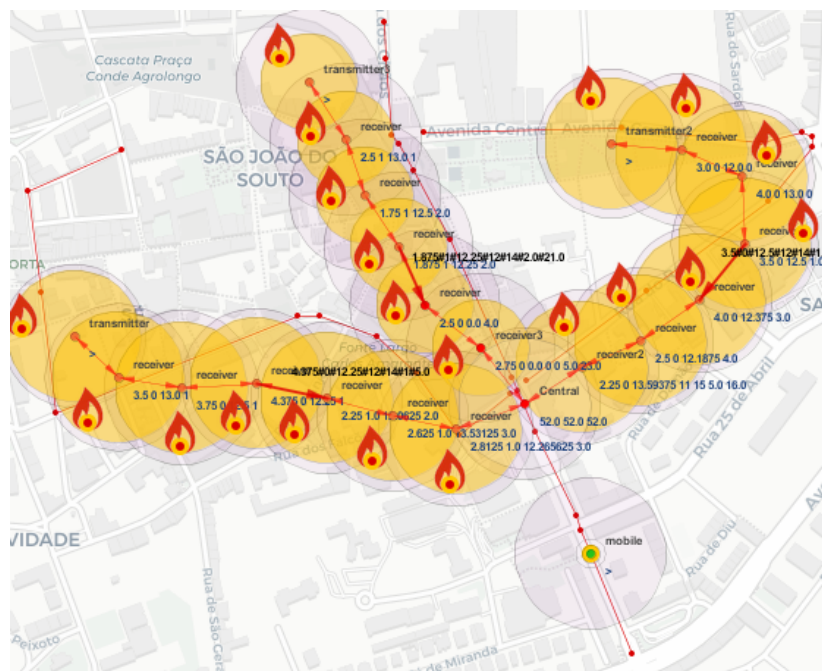


Figura 5.4: Início do deslocamento do sensor móvel ao longo da rota 1

Neste local vai receber uma mensagem do sensor Central a indicar qual a rota para prosseguir. Analisando os totais das três rotas é possível observar que o total da rota 1 e da rota 3 é igual a 40 enquanto que o da rota 2 é 52 (Figura 5.5). Este valor indica que esta rota verifica num maior número os atributos pretendidos pelo utilizador, resultando no envio de uma mensagem ao sensor *mobile* para prosseguir pela rota 2, seguindo esta recomendação recebida (Figura 5.6).

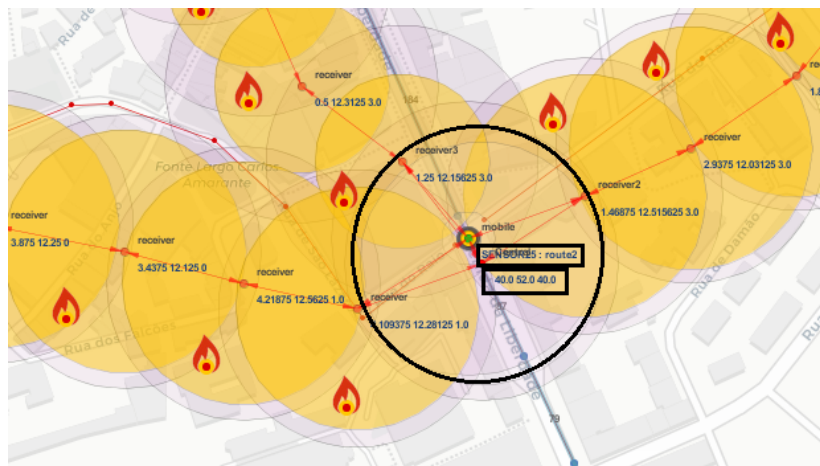


Figura 5.5: Envio da rota recomendada para o sensor mobile

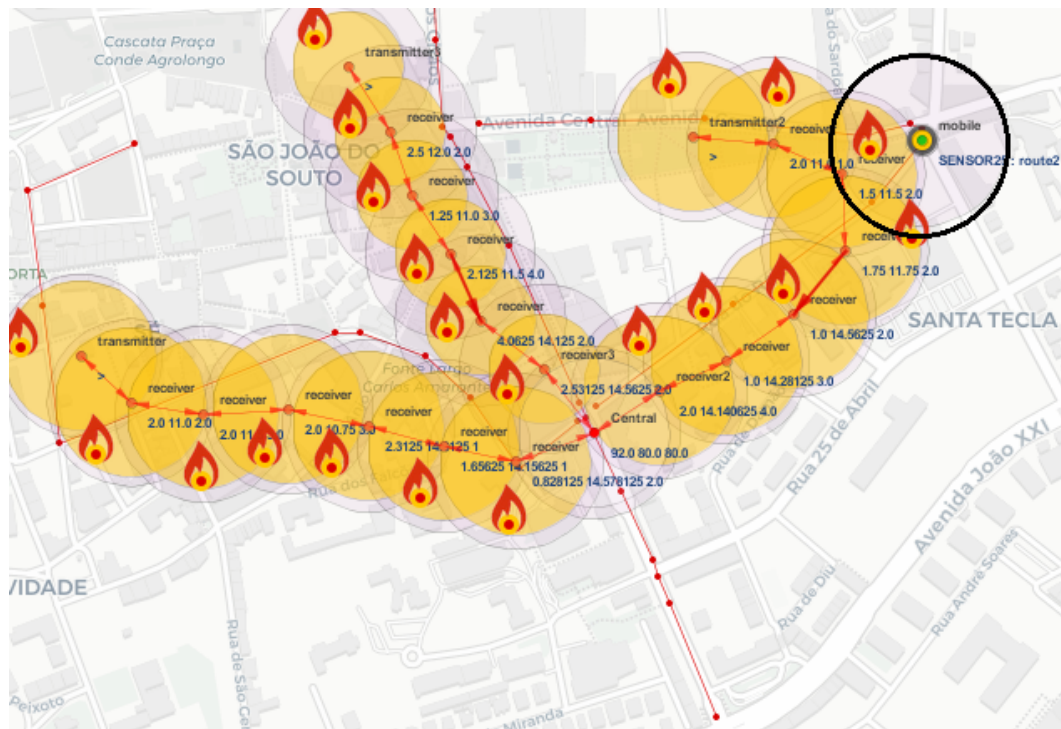


Figura 5.6: Deslocamento do sensor móvel ao longo da rota anteriormente recomendada

Em síntese, é possível observar o correto deslocamento do sensor móvel inicialmente ao percorrer a rota principal e posteriormente seguindo pela rota 2. O comportamento observado foi o esperado face ao inicialmente pretendido, visto que, o sensor móvel seguiu pela rota que perfaz num maior valor os atributos pretendidos pelo utilizador (rota 2).

5.2.3 Perfil B - Local 2

Para o segundo local foram criadas duas rotas para simularem especificamente a tomada de decisão por parte de um utilizador relativamente a duas rotas circulares (Figura 5.7), ou seja o utilizador pretende terminar o percurso perto do ponto onde este foi iniciado. De seguida, apresentam-se as rotas criadas:

- A rota 4 na Figura 5.7 com a cor preta, possui uma distância total de 1906m e inicia-se no círculo preto;
- A rota 5 na Figura 5.7 a cor vermelha, possui 1910m aos quais é acrescentada a distância a partir do ponto inicial (círculo preto) até à rotunda onde é tomada a decisão sobre qual das rotas seguir (813m), perfazendo um total de 2723m.

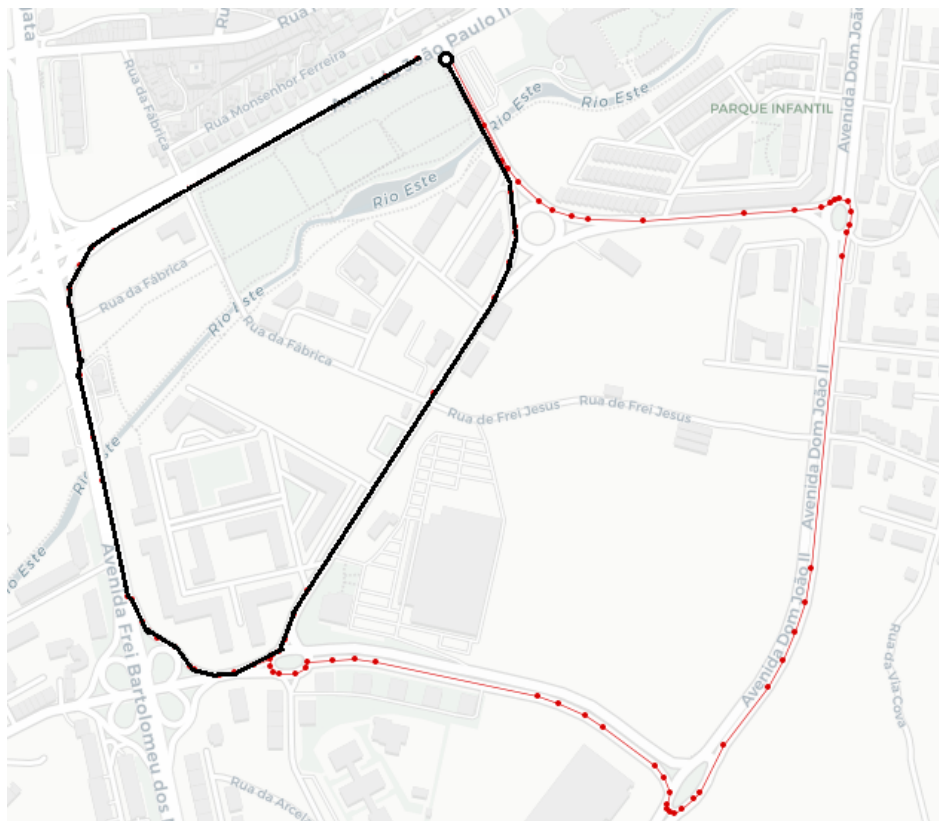


Figura 5.7: Rotas criadas para o segundo local do cenário

Para este local do cenário foram usados os mesmos tipos de sensores, existindo dois sensores transmissores a cor azul, um sensor *mobile* a cor preta e um sensor Central a cor vermelha (Figura 5.8).

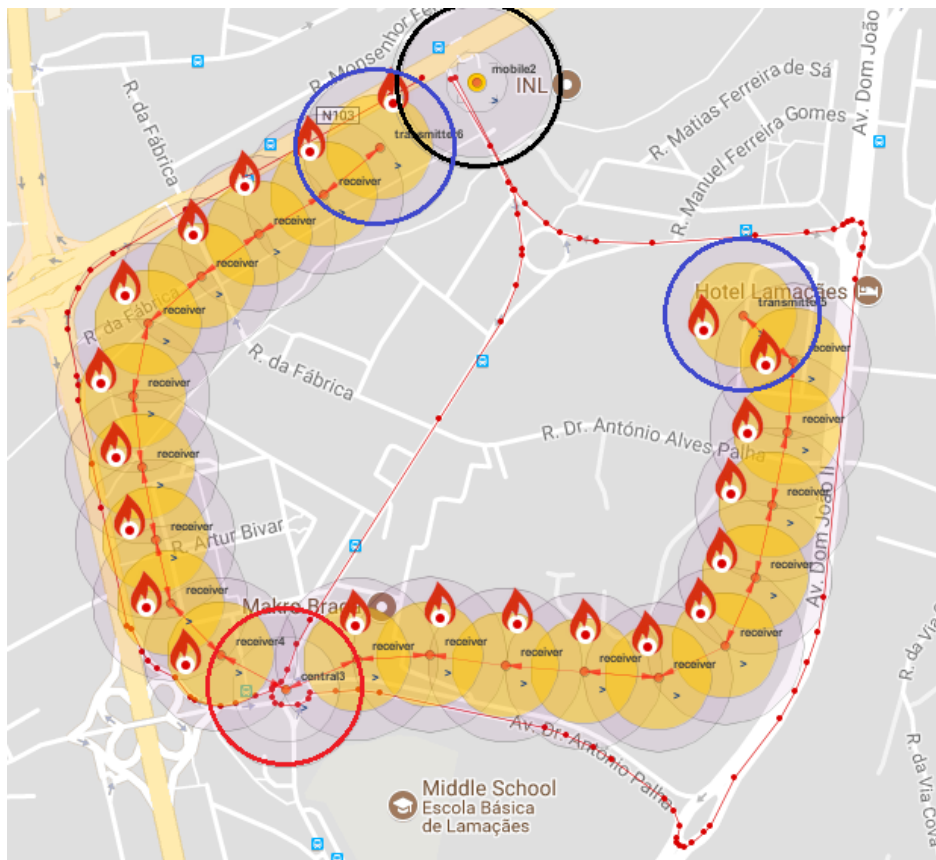


Figura 5.8: Mapa do segundo local do cenário com os sensores utilizados

O perfil B representa então um indivíduo que pretenda caminhar ou praticar jogging ao final da tarde, mas ao mesmo tempo pretende que a rota recomendada seja um percurso circular. Relativamente aos atributos, este pretende que a temperatura média seja superior a 21°C, não dando grande importância ao intervalo de temperatura, requerendo apenas que a temperatura máxima não ultrapasse os 25°C. Além disso, pretende que a cobertura Wi-Fi seja superior a 3 e que pelos menos 80% da rota possua cobertura Wi-Fi. Como o percurso é realizado ao final da tarde não dá qualquer importância à iluminação. Pretende ainda que a distância seja superior a 1,5km. Na Tabela 5.2 encontram-se resumidos os valores pretendidos bem como o peso atribuído correspondente.

Tabela 5.2: Dados relativos ao Perfil B

Atributos	Valores pretendidos	Peso atribuído
Temperatura média	21°C	25%
Intervalo de temperatura	< 25°C	15%

Continua na página seguinte

Tabela 5.2 – continuação da página anterior

Atributos	Valores pretendidos	Peso atribuído
Cobertura Wi-Fi	>3	15%
Percentagem da rota com Wi-Fi	80%	10%
Iluminação	—	—
Distância	>=1,5km	10%
Espaços abertos	—	—
Interesse Turístico	—	—
Percurso Circular	Sim	25%

Tal como realizado para o perfil A para a simulação no Cupcarbon, foi necessário no sensor Central adequar os valores para os pretendidos por este utilizador e ajustar os pesos respetivos. De acordo com os atributos especificados, é possível verificar que este dá importância a que a rota seja circular. Como as rotas 4 e 5 verificam esta condição significa que as variáveis que acumulam os pesos dos atributos considerados pelo utilizador para cada rota, são iniciadas a vinte e cinco correspondendo à percentagem atribuída a este atributo.

Para além disso, foi necessário ajustar os valores das restantes variáveis:

- a variável *numSenR1* representa o número de sensores presentes na rota4, sendo o seu valor 10;
- a variável *numSenR2* representa o número de sensores presentes na rota5 (alternativa à rota4), sendo o seu valor 11;
- o valor da variável *distance1* é 1906m e corresponde ao comprimento da rota6;
- o valor da variável *distance2* é 2723m e corresponde ao comprimento da rota7.

A simulação é iniciada e o sensor móvel começa a deslocar-se ao longo da rota 4 até encontrar uma rotunda onde surge a possibilidade de seguir por duas rotas para chegar ao mesmo destino. A decisão é tomada com base na rota que verifica em maior número os atributos pretendidos, prosseguindo pela rota recomendada (ver Figura 5.9). Foram realizadas várias simulações sendo que, por vezes, o sensor móvel seguia pela rota 4 e noutras pela rota 5. Para ilustrar o funcionamento deste perfil, são apresentadas as Figuras 5.9 e 5.10, que pretendem mostrar o percurso feito pelo nodo móvel e as mensagens trocadas, onde o sensor móvel ao chegar à rotunda recebe mensagem para optar pela rota5, continuando efetivamente por essa rota.

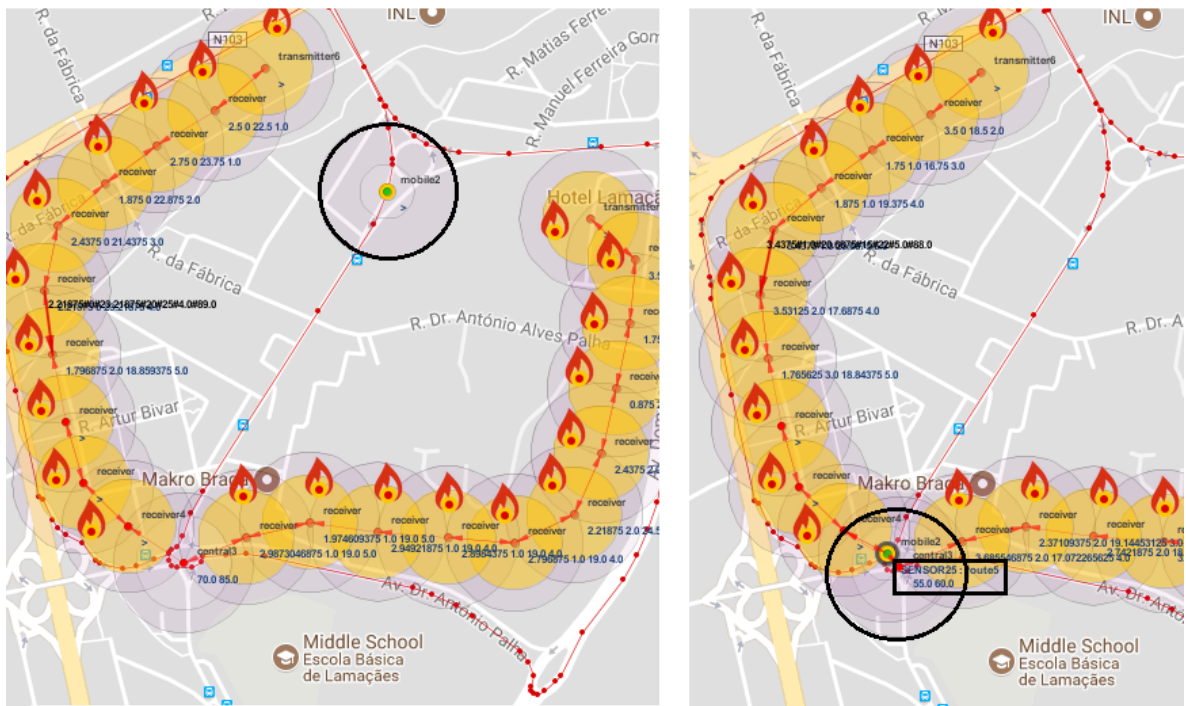


Figura 5.9: Deslocamento do sensor móvel pela rota 4 (à esquerda), recebendo a mensagem para seguir pela rota 5 (à direita)

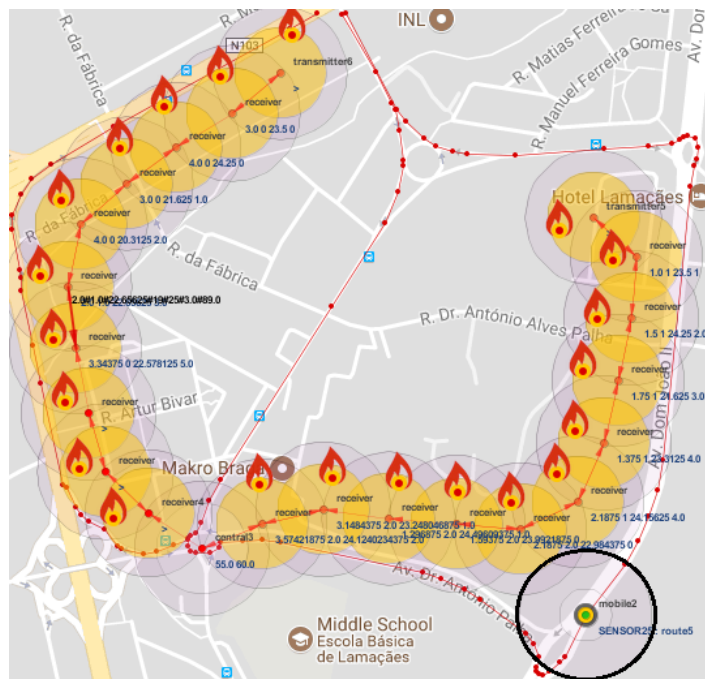


Figura 5.10: Deslocamento do sensor móvel ao longo da rota5

Tal como para o perfil A é possível observar o correto funcionamento do sensor móvel face ao comportamento inicialmente pretendido. Tratando-se ambas as rotas de percursos circulares, o sensor móvel prossegue pela rota que verifica num maior número os restantes atributos desejados.

5.2.4 Perfil C - Local 3

Para o último perfil considera-se um utilizador que deseja deslocar-se até um estabelecimento à noite para tomar café, tratando-se de um ambiente noturno este utilizador tem como atributo chave a presença de iluminação ao longo da rota. Sendo assim, foi simulado um cenário com duas rotas (Figura 5.11),

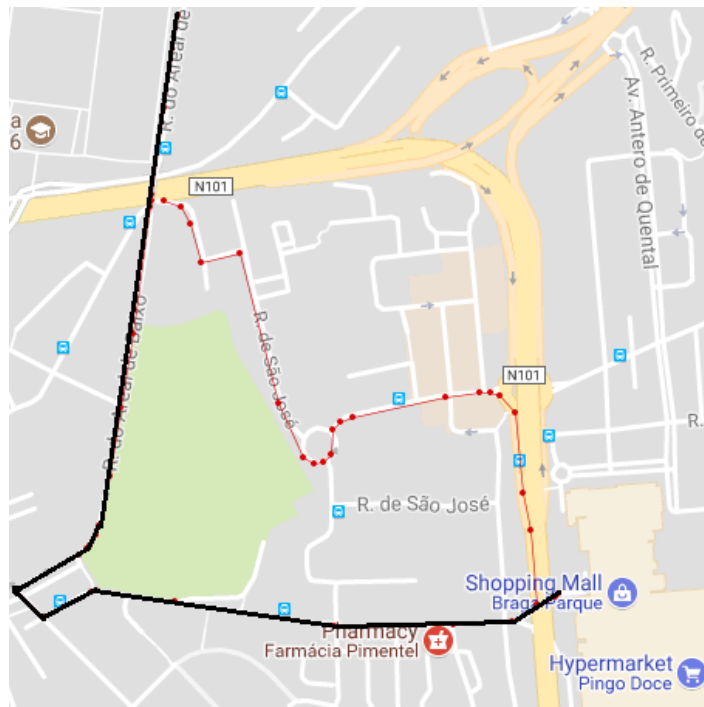


Figura 5.11: Deslocamento do sensor móvel ao longo da rota5

- a rota 6 na Figura 5.11 com cor preta possui uma distância total de 1366m e inicia-se no círculo preto;
- a rota 7 na Figura 5.11 com cor vermelha possui um comprimento de 927m.

Para este local do cenário foram usados os mesmos quatro tipos de sensores, dois sensores transmissores a cor azul, um sensor *mobile* a cor preta e um sensor Central a cor vermelha (Figura 5.12).

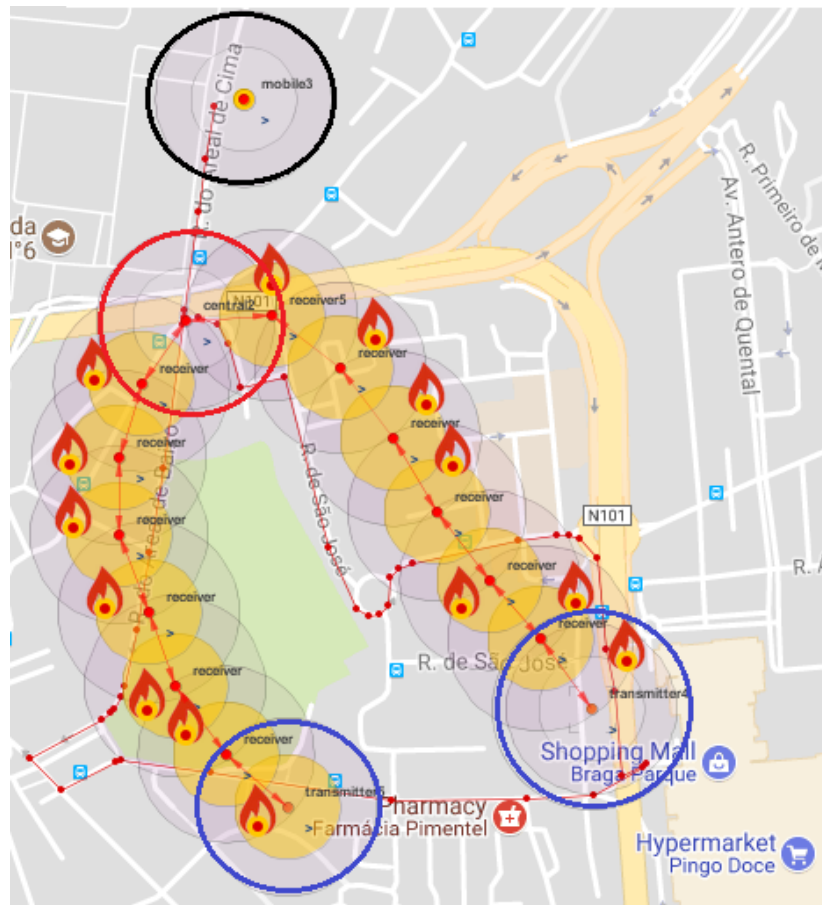


Figura 5.12: Mapa do terceiro local do cenário com os sensores utilizados

Este perfil representa um indivíduo que pretende fazer uma pequena caminhada e, ao mesmo tempo, deslocar-se até um local ao fim de jantar e tomar café. Tratando-se de um empresário que está a tratar de negócios, pretende que 80% da rota possua cobertura de sinal Wi-Fi. Além disso, dá grande importância à presença de iluminação na grande maioria do percurso, contudo não dá grande relevância à temperatura média bastando que o intervalo de temperaturas esteja entre os $[16^{\circ}\text{C}, 24^{\circ}\text{C}]$.

Como dito anteriormente, a cobertura de sinal Wi-Fi é um atributo importante, ou seja, pretende que a cobertura Wi-Fi seja superior a 2. Tratando-se de um horário noturno o utilizador dá grande importância à iluminação pretendendo que pelo menos 80% da rota apresente iluminação. Pretende ainda que a distância seja superior a 700m.

Na Tabela 5.3 encontram-se resumidos os valores pretendidos bem como o seu peso.

Tabela 5.3: Dados relativos ao Perfil C

Atributos	Valores pretendidos	Peso atribuído
Temperatura média	—	—
Intervalo de temperatura	16°C – 24°C	10%
Cobertura Wi-Fi	>2	30%
Percentagem da rota com Wi-Fi	80%	20%
Iluminação	80%	30%
Distância	>700m	10%
Espaços abertos	—	—
Interesse Turístico	—	—
Percurso Circular	—	—

Tal como realizado para os restantes perfis, para a simulação no Cupcarbon foi necessário no sensor central adequar os valores para os pretendidos por este utilizador e ajustar os pesos consoante o que este pretende. Em comparação com os restantes dois perfis, é possível verificar que nenhum dos atributos desejados é já inerente à rota. Isto significa, que ambas as variáveis que acumulam os pesos dos atributos que verifiquem as pretensões do utilizador são iniciadas a zero.

Para além disso foi necessário ajustar os valores das restantes variáveis,

- o valor da variável $numSenR1$ é 7 e corresponde à rota6;
- o valor da variável $numSenR2$ é 7 e corresponde à rota7;
- $distance1$ é 1366m e corresponde ao comprimento da rota6;
- $distance2$ é 927m e corresponde ao comprimento da rota7

A simulação é iniciada e o sensor móvel começa a deslocar-se ao longo da rota 4, encontra uma bifurcação onde tem a possibilidade de seguir por duas rotas que levam ao mesmo destino, neste caso o estabelecimento "Braga Parque"(Figura 5.13).



Figura 5.13: Início do deslocamento do sensor móvel ao longo da rota6

Para a recomendação da rota, a decisão é tomada igualmente com base na rota que verifica num maior número de atributos objetivo. Assim sendo, ao chegar à bifurcação, o sensor móvel vai receber a indicação da rota pela qual deve continuar (Figura 5.14)

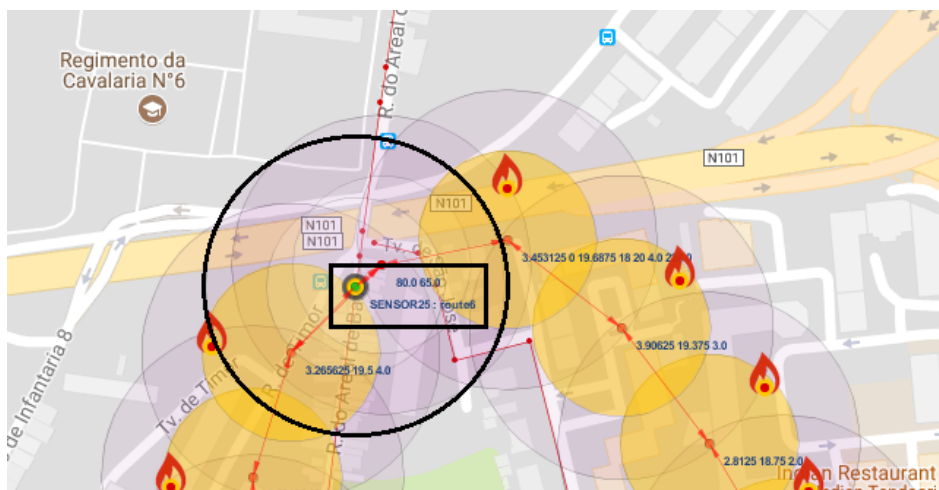


Figura 5.14: Envio da mensagem ao sensor móvel para prosseguir pela rota 6

Tal como para os restantes perfis foram realizadas varias simulações obtendo recomendações para seguir pela rota 6 quer pela rota 7. Contudo, para ilustrar o funcionamento deste perfil, o sensor móvel recebeu a recomendação para seguir pela rota6 verificando-se que segue efetivamente por essa rota (Figura 5.15).



Figura 5.15: Deslocamento do sensor móvel ao longo da rota recomendada

Em linha com os cenários anteriores os resultados obtidos correspondem ao inicialmente pretendido, o sensor móvel prossegue pela rota que lhe é efetivamente recomendada.

5.3 DISCUSSÃO DA SOLUÇÃO

Quando executada a simulação consegue-se observar o deslocamento do nodo móvel (pedestre) a percorrer corretamente a rota inicial até chegar ao local onde é tomada a decisão sobre qual rota prosseguir, normalmente uma bifurcação ou uma rotunda. É também possível observar uma correta troca de mensagens entre os vários sensores ao longo das rotas, bem

como as mensagens enviadas pelos sensores centrais para o nodo móvel, recomendando-lhe a rota pela qual deve prosseguir.

Tratando-se de um sistema de recomendação de rotas num ambiente simulado, não existem resultados de usabilidade palpáveis que possam ser detalhados pois o nodo móvel não é capaz de informar se a rota recomendada foi do seu agrado ou não. Mais ainda, não é possível o utilizador (nodo móvel) fornecer interativamente os atributos pretendidos e os pesos correspondentes. O trabalho de simulação desenvolvido constitui um estudo preliminar que serve como base para o desenvolvimento de uma aplicação de recomendação de rotas a pedestres com funções objetivo distintas e parametrizáveis.

Num cenário de aplicação real, seria possível obter opinião dos utilizadores e fazer uma avaliação da qualidade da experiência, permitindo desta forma melhorar o sistema de recomendação de rota. Saber se a rota recomendada foi ou não do seu agrado fazendo, por exemplo, uma avaliação da mesma num *ranking* ou simplesmente se gostou ou não dessa rota, permitiria afinar o sistema.

Para além disso, outra sugestão seria permitir com base no *feedback* dos utilizadores ajustar e melhorar os algoritmos responsáveis pela recomendação de rotas, por exemplo incorporando Inteligência Artificial, tendo em vista a recomendação de rotas com base em experiências passadas e grupos de utilizadores.

5.3.1 Dificuldades encontradas

Uma das dificuldades encontradas encontra-se relacionada com os atributos simulados no Cupcarbon. Existem alguns atributos importantes mas difíceis de simular como o caso do grau de segurança, largura dos passeios ou o declive das rotas por isso não foram implementados na simulação. No caso de um atributo que indicasse o grau da segurança, quando traduzido este modelo para a realidade, poderiam ser os próprios utilizadores a indicar o grau de segurança da rota através de prévio conhecimento da mesma ou após a percorrerem. Outra hipótese seria recorrer as autoridades locais para indicar um valor de segurança para alguns pontos da cidade, permitindo o conhecimento das zonas mais problemáticas e evitando, por isso, a recomendação de rotas nessas zonas.

Ainda relacionado com os atributos, existe uma vasta quantidade de atributos que poderiam ter sido simulados, contudo os atributos considerados são bastante ilustrativos para um leque alargado de utilizadores. Quando traduzido para um ambiente real, os utilizadores também poderiam dar sugestões de outros atributos para enriquecer a tomada de decisão.

Outra limitação no que diz respeito às simulações realizadas no CupCarbon, diz respeito à não implementação de nenhum sistema de recomendação, nomeadamente, *Content Based*, *Collaborative filtering* e *Hybrid recommender systems*. Como dito anteriormente, não é possível

o sensor móvel informar se a rota que percorreu foi do seu agrado ou não, assim sendo, não é possível guardar um histórico de utilizações anteriores, tendo em vista a utilização de um sistema de recomendação.

5.3.2 *Infraestruturas de apoio à implementação*

Para a implementação deste projeto num contexto real, fez-se um pequeno estudo das infraestruturas que seriam necessárias usar ou reutilizar, ao longo da cidade de Braga. Na cidade existem diversos pontos onde é disponibilizada informação relativa aos autocarros que circulam nesses pontos, bem como a temperatura ambiente no momento. Uma ideia seria, por exemplo, reaproveitar esses pontos para que, além dessas informações, se indique nas zonas onde existe Wi-Fi grátis na cidade, a cobertura do sinal Wi-Fi. Desta forma, seria feita uma reutilização dos pontos agregadores já existentes adicionando mais informação para que as pessoas que circulem a pé tenham disponíveis informações atualizadas.

Relativamente à monitorização dos atributos seria necessário um investimento no sentido de implementar uma rede de sensores sem fios alargada. Estes sensores precisam ser resistentes para suportar condições atmosféricas adversas, necessitam ainda de baterias de grande durabilidade, que permitam o envio dos dados bem como da sua localização para um ou vários pontos centrais. Neste pontos, seriam recebidas as condições dos vários atributos nos diversos pontos da cidade, tendo a aplicação móvel acesso a esses dados tendo em vista a recomendação das rotas.

5.3.3 *Implementação do Modelo*

O desenvolvimento deste modelo passa pela criação de uma aplicação móvel que permita aos utilizadores da cidade de Braga e posteriormente de outras cidades receberem recomendações de rota. Neste contexto, este trabalho constitui uma base e suporte pois foi definido um objetivo, bem como o público-alvo e feita uma análise de soluções existentes para a recomendação de rotas para pedestres. Faz sentido a criação de uma aplicação móvel pois, desta forma, permite que quem utilize a aplicação consiga aceder em qualquer momento e localização. Além disso, para o caso em que foi recomendada uma rota inicial, pode posteriormente receber uma nova recomendação caso pretenda mudar de direção ou tipo de percurso.

O passo seguinte passa por decidir o sistema operativo móvel em que se pretende trabalhar que podem ser:

- *Android*, usando, por exemplo, a linguagem Java;
- *iOS*, usando, por exemplo, a linguagem *Objective C*;

- *Windows Phone*, utilizando a linguagem C#, por exemplo.

Ao longo deste trabalho fez-se no Capítulo 5 um levantamento dos requisitos necessários para a criação de uma aplicação móvel. Desta forma, já existe uma base de requisitos para a implementação do modelo e para a interação dos utilizadores com a aplicação móvel relativamente a escolha dos atributos de maior importância. Ao longo das simulações para cada um dos perfis foram definidos os valores pretendidos para os atributos, bem como o peso que estes davam a cada um dos atributos. Uma forma mais autónoma seria, por exemplo, os utilizadores ordenarem por grau de importância os atributos, tendo o sistema previamente definido que ao atributo mais importante estava atribuído um peso fixo máximo e aos restantes atributos o peso fixo diminuía à medida que o seu grau de importância também diminuía.

Para além dos requisitos, é necessário fazer a incorporação de um sistema de recomendação. A melhor solução passaria por uma sistema híbrido, fazendo uso dos métodos *Content Based* e *Collaborative filtering*. O método *Content Based* seria usado para a recomendação ao utilizador de novas rotas com base num histórico de utilização associado ao perfil do utilizador. O método *Content Based* seria mais focado para o uso por turistas, tendo deste modo disponível rotas que utilizadores ou turistas com perfis similares usaram anteriormente.

5.4 RESUMO DO CAPÍTULO

Ao longo deste capítulo foi apresentado o cenário final simulado no CupCarbon, foram apresentados os três perfis e locais simulados, bem como os quatro comportamentos que foram associados aos sensores:

- Sensor mobile;
- Sensor transmissor;
- Sensor recetor;
- Sensor central.

Foram também apresentados os resultados obtidos e, por fim, apresentada uma discussão da solução implementada e do que seria necessário para implementar este modelo na realidade. Foram ainda identificadas e abordadas as funcionalidades necessárias para a implementação de uma aplicação móvel capaz de recomendar rotas com base nos atributos pretendidos pelos utilizadores. No próximo capítulo são apresentadas as conclusões deste trabalho.

CONCLUSÃO

Neste capítulo são resumidas as principais conclusões e contribuições do trabalho desenvolvido, bem como apontadas linhas de trabalho futuro.

6.1 PRINCIPAIS CONCLUSÕES E CONTRIBUIÇÕES

Desde sempre as caminhadas foram um modo de transporte importante na realização de deslocamentos urbanos do dia a dia. Os países desenvolvidos incentivam cada vez mais o transporte não motorizado desde que, são reconhecidos globalmente os efeitos negativos dos veículos motorizados no clima do planeta e na saúde dos habitantes. Contudo, estes incentivos precisam considerar e entender quais as rotas utilizadas pelos pedestres. Na busca por esse entendimento, este trabalho debruçou-se sobre o problema da escolha de rota em zonas urbanas. Numa primeira fase, fez-se um levantamento do estado da arte sobre o tema sendo este dividido em três partes:

- na primeira parte fez-se uma leitura de estudos relacionados com o tema proposto, nomeadamente identificando os atributos mais usados pelos pedestres para a escolha de uma rota em ambiente urbano;
- numa segunda parte fez-se um estudo de ferramentas relacionadas com o planeamento de rotas, como o Google Maps e o Komoot, tendo sido estudadas as suas funcionalidades. Para além destas aplicações, realizou-se um estudo sobre o sistema de planeamento de rotas PedestrianPal;
- por fim fez-se um levantamento de tecnologias usadas para a recomendação de rotas para pedestres, sendo estas divididas em localização dos pedestres, sistemas de recomendação e algoritmos para o cálculo do caminho mais curto. Relativamente à localização dos pedestres existem dois métodos, localização relativa e absoluta, sendo este último a melhor forma de modelar a mobilidade de um dispositivo móvel. De seguida, estudaram-se os sistemas de recomendação *Content Based*, *Collaborative filtering* e uma solução híbrida que mistura os dois métodos anteriores, verificando-se que

esta apresenta melhor desempenho, bem como recomendações mais precisas. Por fim, foram estudados os algoritmos usados para o cálculo do caminho mais curto.

Concluído o estudo relativo ao estado da arte, seguiu-se um dos objetivos parciais deste trabalho, a escolha da plataforma de simulação que permitisse efetuar a simulação realista da geração de rotas em ambiente urbano. Realizou-se um estudo sobre três ferramentas de simulação, Cupcarbon, UrbanSim e InterSCSimulator, sendo analisadas e comparadas as vantagens e desvantagens de cada uma. Este estudo culminou com a escolha da ferramenta CupCarbon como plataforma de simulação. Esta escolha foi baseada no fato desta ferramenta apresentar bastante material de suporte, permitir gerir tráfego urbano bem como permitir a criação de rotas e redes de sensores.

Numa segunda fase, foi inicialmente estruturada e criada uma primeira arquitetura de uma solução para o problema proposto tendo por base os estudos de Huang e Gartner [18]. Após a definição de uma arquitetura inicial, foi efetuado um levantamento de requisitos tendo em vista a criação de um modelo de domínio representativo da solução. Para isso, foram estudados potenciais tipos de utilizadores e definidos requisitos funcionais e não funcionais.

Depois de efetuado o levantamento de requisitos foi desenvolvida a componente prática deste trabalho, no sentido de simular o processo de recomendação de rotas a utilizadores que se deslocam numa cidade. Concluída a ambientação ao CupCarbon, foram criados dois cenários iniciais de aprendizagem e um cenário final que pretende mostrar um modelo como base de suporte ao serviço pretendido.

Para os cenários simulados foram criados diferentes perfis de utilizador. Desta forma, pretendeu-se simular casos específicos da recomendação de rotas com base nos atributos objetivo de cada um dos perfis. Foram realizadas várias simulações e testes para cada um dos perfis criados com o intuito de obter uma solução para o problema inicialmente proposto, verificando a sua funcionalidade e correto funcionamento.

Comparando o modelo criado com as ferramentas estudadas no estado de arte, é possível verificar um leque de novas funcionalidades que estas não permitiam. A maioria destas ferramentas ainda tem pouco em consideração as pretensões dos utilizadores na tomada de decisão da escolha de uma rota, baseando-se sobretudo na distância, declive e tempo de viagem. As vantagens do modelo criado passam pela disponibilização de um maior número de atributos para basear a decisão, bem como, na atribuição de pesos a cada um dos atributos escolhidos.

Em síntese, o estudo realizado contribuiu para identificar as funcionalidades necessárias para o desenvolvimento de um sistema de recomendação de rotas, culminando com a criação de um modelo funcional em ambiente simulado que permite a recomendação de rotas.

6.2 TÓPICOS DE TRABALHO FUTURO

Trabalhos como este devem ser aprimorados e continuados. Uma sugestão para trabalho futuro passa por acrescentar ao cenário final mais alternativas de rotas para a escolha do utilizador, bem como, incrementar o número de atributos simulados. Para além disso, seria importante realizar inquéritos aos utilizadores do sistema tendo como objetivo aumentar o número de atributos monitorizados, bem como, obter possivelmente novas funcionalidades.

Outra sugestão de trabalho futuro encontra-se relacionada com a recolha de dados ambientais, seria interessante usar os sensores de humidade e temperatura que alguns *smartphones* possuem, desta forma, os próprios utilizadores ajudariam a recolher mais informação relativamente aos vários atributos utilizados para a escolha de uma rota. Assim sendo, os dados recolhidos pelos utilizadores poderiam, por exemplo, ser armazenados numa base de dados *Open Data* para a cidade de Braga, onde seriam armazenados dados de diversas áreas com o intuito, neste caso, de auxiliar a recomendação de rotas.

BIBLIOGRAFIA

- [1] Benevolo C., Dameri R.P., D’Auria B. (2016) Smart Mobility in Smart City. In: Torre T., Braccini A., Spinelli R. (eds) Empowering Organizations. Lecture Notes in Information Systems and Organisation, vol 11. Springer, Cham
- [2] Angelidou, M. (2016). Four European Smart City Strategies. *International Journal of Social Science Studies*, 4(4).
- [3] Bloguedominho.blogs.sapo.pt. (2018). BRAGA TEM SISTEMA INOVADOR PARA MONITORIZAR O AMBIENTE. [online] Available at: <https://bloguedominho.blogs.sapo.pt/braga-tem-sistema-inovador-para-7740617> [Accessed 26 Oct. 2017].
- [4] Unric.org. (2017). Relatório da ONU mostra população mundial cada vez mais urbanizada, mais de metade vive em zonas urbanizadas ao que se podem juntar 2,5 mil milhões em 2050. [online] Available at: <http://www.unric.org/pt/actualidade/31537-relatorio-da-onu-mostra-populacao-mundial-cada-vez-mais-urbanizada-mais-de-metade-vive-em-zonas-urbanizadas-ao-que-se-podem-juntar-25-mil-milhoes-em-2050> [Accessed 28 Oct. 2017].
- [5] CHORIANOPOULOS, K. (2013). Pedestrian navigation and shortest path: Preference versus distance.
- [6] Borràs, J., Moreno, A. and Valls, A. (2014). Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16), pp.7370-7389.
- [7] Reginald G. Golledge (1995). Path Selection and Route Preference in Human Navigation: A Progress Report
- [8] Hoogendoorn, S. and Bovy, P. (2004). Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological*, 38(2), pp.169-190.
- [9] van der Zijpp, N. and Fiorenzo Catalano, S. (2005). Path enumeration by finding the constrained K-shortest paths. *Transportation Research Part B: Methodological*, 39(6), pp.545-563.

- [10] Schlossberg, M. (2007). *How far, by which route, and why?*. San Jose, CA: Mineta Transportation Institute.
- [11] Guo, Z. and Loo, B. (2013). Pedestrian environment and route choice: evidence from New York City and Hong Kong. *Journal of Transport Geography*, 28, pp.124-136.
- [12] Czogalla, O; Herrmann, A. Parameters determining route choice in pedestrian network
- [13] Yucel, S. (2017). *Smart Community Wireless Platforms*.
- [14] Silva, S. (2017). Braga. Internet passou a ter acesso livre em 30 ruas de Braga. [online] PÚBLICO. Available at: <https://www.publico.pt/2014/07/21/local/noticia/internet-passou-a-ter-acesso-livre-em-30-ruas-de-braga-1663750> [Accessed 07 Nov. 2017].
- [15] Huang H., Gartner G. (2009) A Survey of Mobile Indoor Navigation Systems. In: Gartner G., Ortog F. (eds) *Cartography in Central and Eastern Europe. Lecture Notes in Geoinformation and Cartography*. Springer, Berlin, Heidelberg pp.305-319
- [16] Bahls, B. (2011). *Pedestrian pal*. [Missoula, Mont.]: The University of Montana.
- [17] MakeUseOf. (2017). How Does Google Maps Work?. [online] Available at: <https://www.makeuseof.com/tag/technology-explained-google-maps-work/> [Accessed 11 Nov. 2017].
- [18] komoot. (2017). Komoot | Better Planning and Navigation for Hikers and Bikers. [online] Available at: <https://www.komoot.com/> [Accessed 12 Nov. 2017].
- [19] Dempsey, C. (2017). Absolute and Relative Location - GeoLounge: All Things Geography. [online] GeoLounge: All Things Geography. Available at: <https://www.geolounge.com/absolute-relative-location/> [Accessed 15 Nov. 2017].
- [20] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), pp.734-749.
- [21] Huang, H. & Gartner, G., 2010. Collaborative Filtering Based route Recommendation for assisting Pedestrian Wayfinding. In *Proceedings of the 7th International Symposium on*

Location Based Services & TeleCartography. Guangzhou, China.

[22] Sanders, P., Schultes, D.: Engineering fast route planning algorithms. In: Demetrescu, C. (ed.) WEA 2007. LNCS, vol. 4525, pp. 23–36. Springer, Heidelberg (2007).

[23] Ostrowski, D. (2015). Comparative Analysis of the Algorithms for Pathfinding in GPS Systems. ICN 2015 : The Fourteenth International Conference on Networks

[24] Barros, J. (2017). Algoritmos sobre Grafos, Algoritmos e Complexidade. Departamento de Informática, Universidade do Minho, Braga

[25] Robert C. Holte, Ariel Felner, Guni Sharon, and Nathan R. Sturtevant. Bidirectional search that is guaranteed to meet in the middle. In Proceedings of AAAI, 2016.

[26] Brilliant.org. (2018). Bellman-Ford Algorithm | Brilliant Math & Science Wiki. [online] Available at: <https://brilliant.org/wiki/bellman-ford-algorithm/> [Accessed 24 Nov. 2017].

[27] Cupcarbon.com. (2018). CupCarbon - A Smart City & IoT WSN Simulator. [online] Available at: <http://www.cupcarbon.com/> [Accessed 7 Dec. 2017].

[28] OpenStreetMap. (2018). OpenStreetMap. [online] Available at: <https://www.openstreetmap.org/7.839> [Accessed 8 Dec. 2017].

[29] UrbanSim. (2018). UrbanSim. [online] Available at: <http://www.urbansim.com/urbansim/> [Accessed 16 Dec. 2017].

[30] Santana E.F.Z., Lago N., Kon F., Milojicic D.S. (2018) InterSCSimulator: Large-Scale Traffic Simulation in Smart Cities Using Erlang. In: Dimuro G., Antunes L. (eds) Multi-Agent Based Simulation XVIII. MABS 2017. Lecture Notes in Computer Science, vol 10798. Springer, Cham

[31] Hossein Motlagh, N., Taleb, T. and Arouk, O. (2016). Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives. IEEE Internet of Things Journal, 3(6), pp.899-922.

[32] the Guardian. (2017). Facebook drone that could bring global internet access completes test flight. [online]

Available at: <https://www.theguardian.com/technology/2017/jul/02/facebook-drone-aquila-internet-test-flight-arizona> [Accessed 8 Jan. 2018].



MATERIAL DE SUPORTE

A.1 MANUAL DE UTILIZAÇÃO

De seguida apresenta-se os passos necessários a seguir por forma a usar a ferramenta CupCarbon para simular os cenários criados:

- download da plataforma CupCarbon através do link <http://www.cupcarbon.com/> ;
- download dos cenários construídos disponíveis no link: https://github.com/stephanefernandes/Proj_Diss_CupCarbon ;
- abrir a ferramenta e o cenário que se pretende simular, *CupCarbon* -> *Project* -> *Open project* -> Pasta do Cenário 1 (2 ou 3) -> Cenário1.cup;
- seleccionar na aba lateral esquerda a opção *Marker Parameters*, arrastar para cima e seleccionar a opção *Draw all routes* por forma a apresentar todas as rotas do cenário;
- seleccionar na aba lateral esquerda a opção *Simulation Parameters*, seleccionar a *checkbox Mobility/ Events* caso não se encontre seleccionada por forma a permitir a mobilidade do sensor móvel;
- na aba lateral esquerda *Simulation Parameters* ajustar os parâmetros relativos à velocidade da simulação *Simulation Speed* e *Arrow Speed*, quanto mais baixo os seus valores mais rápido é executado a simulação.

Parâmetros de referência:

Simulation Speed = 150 ms e *Arrow Speed* = 100 ms.

- seleccionar na aba superior a opção *Simulation* -> *Run Simulation* para simular o cenário.

A.2 CÓDIGO-FONTE DA SCRIPT RECEIVER UTILIZADA NO CENÁRIO FINAL

Seguidamente, apresenta-se o código-fonte criado para o sensor *Receiver* utilizado no cenário final:

```

1 loop
2 wait
3 read s1
4 rdata $s1 wifi n_wifi temp tempMin tempMax per_iluminacao id
5 areadsensor v
6 if($v!=X)
7 rdata $v a b temp2
8 int temp2 $temp2
9 if($temp2 < $tempMin)
10   set tempMin $temp2
11 end
12 if($temp2 > $tempMax)
13   set tempMax $temp2
14 end
15 plus temp2 $temp2 $temp
16 div temp2 $temp2 2
17 randb wifi2 0 5
18 if($wifi2 == 0)
19   plus n_wifi $n_wifi 1
20 end
21 plus wifi2 $wifi2 $wifi
22 div wifi2 $wifi2 2
23 randb iluminacao2 0 1
24 if($iluminacao2 == 1)
25   plus per_iluminacao $per_iluminacao 1
26 end
27 plus id $id 1
28 print $wifi2 $temp2 $per_iluminacao $n_wifi
29 data p $wifi2 $n_wifi $temp2 $tempMin $tempMax $per_iluminacao $id
30 send $p $id
31 end
32 delay 1000

```

A.3 CÓDIGO-FONTE DA SCRIPT CENTRAL UTILIZADA NO CENÁRIO FINAL

Por fim, apresenta-se o código-fonte criado para o sensor central para o **perfil B**, o código é idêntico para os restantes perfis havendo apenas trocas nos valores usados para comparação e também no peso dado a cada atributo.

```

1 loop
2 set pef B
3 wait
4 read v

```

```

5 rdata $v wifi1 n_wifi1 temp1 tempMin1 tempMax1 per_iluminacao1
6 wait
7 read s1
8 rdata $s1 wifi2 n_wifi2 temp2 tempMin2 tempMax2 per_iluminacao2
9 if($pef == B)
10     set total1 25
11     set total2 25
12     set numSenR1 10
13     set numSenR2 11
14     set distance1 1906
15     set distance2 2723
16     div n_wifi1 $n_wifi1 $numSenR1
17     mult n_wifi1 $n_wifi1 100
18     div n_wifi2 $n_wifi2 $numSenR2
19     mult n_wifi2 $n_wifi2 100
20     div per_iluminacao1 $per_iluminacao1 $numSenR1
21     mult per_iluminacao1 $per_iluminacao1 100
22     div per_iluminacao2 $per_iluminacao2 $numSenR2
23     mult per_iluminacao2 $per_iluminacao2 100
24     if($temp1 > 21)
25         plus total1 $total1 25
26     end
27     if($wifi1 > 3)
28         plus total1 $total1 15
29     end
30     if($distance1 > 1500)
31         plus total1 $total1 10
32     end
33     if($n_wifi1 > 80)
34         plus total1 $total1 10
35     end
36     if($tempMax1 < 25)
37         plus total1 $total1 15
38     end
39     if($temp2 > 21)
40         plus total2 $total2 25
41     end
42     if($wifi2 > 3)
43         plus total2 $total2 15
44     if($distance2 > 1500)
45         plus total2 $total2 10
46     end
47     if($n_wifi2 > 80)
48         plus total1 $total2 10
49     end
50     if($tempMax2 < 25)
51         plus total2 $total2 15

```

```
52     end
53     print $total1 $total2
54     if($total1 > $total2)
55         data d B SENSOR25 route4
56             send $d 225
57     else
58         data d B SENSOR25 route5
59             send $d 225
60
61 end
```

