



TÉCNICO
LISBOA

Iluminação Inteligente para Espaços Amplos

João António Cruz Correia

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Orientador: Prof. Renato Jorge Caleira Nunes

Júri

Presidente: Prof. Paolo Romano

Orientador: Prof. Renato Jorge Caleira Nunes

Vogal: Prof. Alberto Manuel Ramos da Cunha

Outubro de 2016

Agradecimentos

Gostaria de agradecer em especial aos meus pais por me terem sempre apoiado e suportado, não só nesta fase académica, mas também no resto dos meus estudos.

Quero igualmente agradecer ao meu orientador, o Prof. Renato Nunes, por ter sugerido este tema para a minha tese e pelo apoio e orientação científica.

Resumo

O consumo de energia a nível mundial tem vindo a aumentar nos últimos anos e como muita da energia provém de fontes não-renováveis, este aumento do consumo poderá trazer consequências ambientais graves num futuro próximo. A eletricidade é a forma de energia cujo consumo mais tem crescido, pelo que é importante encontrar tecnologias mais eficientes. Este trabalho propõe um sistema de iluminação inteligente para espaços fechados de grande dimensão, como por exemplo parques de estacionamento subterrâneos. O sistema será constituído por luminárias inteligentes, que funcionarão de forma autónoma, e uma aplicação de configuração. A solução proposta segue uma abordagem distribuída e tem a particularidade de apenas iluminar as áreas onde se encontram pessoas, sendo capaz de determinar a direção de deslocação dos utilizadores e iluminar essas áreas antecipadamente. O objetivo é reduzir o consumo de energia, sem diminuir a qualidade da iluminação, oferecendo uma solução de fácil instalação e de elevada escalabilidade. Como não é viável implementar um protótipo deste sistema, devido ao elevado número de luminárias necessárias, optou-se por desenvolver e testar o sistema com recurso a um simulador. Nesta dissertação é descrita a arquitetura da solução proposta e são descritos os desenvolvimentos efetuados, de que se destacam o simulador, o algoritmo presente nas luminárias e a aplicação de configuração. Apresentam-se também os testes efetuados e os resultados obtidos que demonstram que a solução proposta é eficaz e permite poupanças energéticas superiores a outros sistemas.

Palavras-chave: Luminárias inteligentes, Sistema de Iluminação, Poupança de Energia, Sistema Distribuído

Abstract

The worldwide energy consumption has been increasing in recent years and since much of the energy comes from non-renewable sources, this increase in consumption may have serious environmental consequences in the near future. Electricity is the form of energy whose consumption has grown the most, so it is necessary to find more efficient technologies. This paper proposes an intelligent lighting system for indoor large-scale spaces, such as underground car parks. The system will consist of intelligent lighting luminaires, which will operate autonomously, and a configuration application. The proposed solution follows a distributed approach and has the particularity to only illuminate the areas where people are present, being able to determine the direction of movement of users and illuminate those areas in advance. The goal is to reduce energy consumption without decreasing the quality of lighting, offering a solution that is easy to install and offers high scalability. Since it is not feasible to implement a prototype of this system due to the high number of necessary luminaires, it was decided to develop and test the system using a simulator. In this thesis it is described the architecture of the proposed solution and are described the developments made, with emphasis on the simulator, the algorithm present in the luminaires and the configuration application. We also present the tests performed and their results, which demonstrate that the proposed solution is efficient and allows greater energy savings than other systems.

Keywords: Intelligent Luminaires, Lighting System, Energy Saving, Distributed System

Índice

| | |
|---|------------|
| Agradecimentos | i |
| Resumo | ii |
| Abstract | iii |
| Lista de Tabelas | vi |
| Lista de Figuras | vii |
| 1. Introdução | 1 |
| 1.1 Motivação | 1 |
| 1.2 Objetivos..... | 1 |
| 1.3 Estrutura do Documento..... | 2 |
| 2.Trabalho Relacionado | 3 |
| 2.1 Sistema de iluminação baseado numa rede de sensores | 3 |
| 2.2 Tvilight | 6 |
| 2.3 SSL (Smart Street Lighting)..... | 8 |
| 2.4 Sistema de controlo de iluminação baseado numa rede sensores..... | 11 |
| 2.5 Controlo de iluminação baseado em sistema de vídeo | 13 |
| 2.6 Discussão | 16 |
| 3. Descrição da Solução Proposta | 17 |
| 3.1 Descrição Funcional | 17 |
| 3.2 Principais componentes do sistema | 18 |
| 3.3 Implementação | 19 |
| 3.3.1 Simulador | 19 |
| 3.3.2 Abordagem ao Algoritmo | 27 |
| 3.3.3 Aplicação de configuração..... | 28 |
| 4. Algoritmo | 36 |
| 4.1 Tipos de Mensagens | 36 |
| 4.2 Envio de Mensagens | 36 |
| 4.3 Estados | 39 |
| 4.4 Descrição detalhada..... | 41 |
| 4.5 Limitações | 46 |
| 4.6 Melhoramentos..... | 47 |
| 5. Avaliação | 49 |
| 5.1 Método de Avaliação | 49 |
| 5.2 Testes..... | 52 |
| 5.2.1 Teste 1 – Percursos semelhantes, utilização baixa | 52 |
| 5.2.2 Teste 2 - Percursos semelhantes, utilização média | 54 |
| 5.2.3 Teste 3 – Percursos semelhantes, utilização alta..... | 56 |
| 5.2.4 Teste 4 – Percursos diferentes, utilização baixa | 58 |
| 5.2.5 Teste 5 - Percursos diferentes, utilização média | 60 |

| | |
|---|-----------|
| 5.2.6 Teste 6 – Percursos diferentes, utilização alta | 62 |
| 5.3 Avaliação dos resultados..... | 64 |
| 6. Conclusão | 66 |
| 7. Referências | 68 |

Lista de Tabelas

| | |
|--|----|
| 1. Desempenho do sistema | 15 |
| 2. Resumo das características | 16 |
| 3. Correspondência entre as direções e os seus identificadores | 28 |
| 4. Tabela de transição de estados | 39 |
| 5. Correspondência entre estado e saída..... | 41 |
| 6. Estatísticas da poupança de energia no teste 1 | 53 |
| 7. Estatísticas do envio de mensagens no teste 1 | 53 |
| 8. Estatísticas da poupança de energia no teste 2..... | 55 |
| 9. Estatísticas do envio de mensagens no teste 2 | 55 |
| 10. Estatísticas da poupança de energia no teste 3..... | 57 |
| 11. Estatísticas do envio de mensagens no teste 3 | 57 |
| 12. Estatísticas da poupança de energia no teste 4..... | 59 |
| 13. Estatísticas do envio de mensagens no teste 4 | 59 |
| 14. Estatísticas da poupança de energia no teste 5..... | 61 |
| 15. Estatísticas do envio de mensagens no teste 5 | 61 |
| 16. Estatísticas da poupança de energia no teste 6..... | 63 |
| 17. Estatísticas do envio de mensagens no teste 6 | 63 |
| 18. Estatísticas da poupança de energia nos testes efetuados..... | 64 |
| 19. Estatísticas do envio de mensagens nos testes efetuados | 66 |

Lista de Figuras

| | |
|---|----|
| 1. Exemplo do sistema de iluminação inteligente..... | 4 |
| 2. Exemplo do SSL..... | 9 |
| 3. Máscara de visibilidade e exemplos de alvos humanos..... | 14 |
| 4. Foi detetado movimento..... | 18 |
| 5. O utilizador moveu-se da esquerda para a direita..... | 18 |
| 6. Ecrã inicial do simulador..... | 21 |
| 7. Janela onde podemos ver a poupança de energia..... | 22 |
| 8. Janela onde podemos ver informações do envio de mensagens..... | 23 |
| 9. Deteção de um utilizador, com raio de iluminação igual a 3..... | 28 |
| 10. Ecrã inicial da aplicação de configuração..... | 29 |
| 11. Janela onde podemos criar um novo <i>map script</i> | 29 |
| 12. Janela onde se removem os nós..... | 30 |
| 13. Janela onde se define os vizinhos de cada nó..... | 31 |
| 14. Janela onde se define o raio de iluminação..... | 31 |
| 15. Janela onde se define a distância de antecipação..... | 32 |
| 16. Janela onde se define a intensidade em percentagem de cada uma das intensidades..... | 32 |
| 17. Janela onde se adiciona e remove paredes..... | 33 |
| 18. Janela onde se define a intensidade base de cada nó..... | 33 |
| 19. Janela onde se define as direcções especiais de qualquer nó..... | 34 |
| 20. Janela onde se define o valor do <i>timeout</i> | 35 |
| 21. Janela onde se simula a falha de um ou mais nós..... | 35 |
| 22. Envio de uma mensagem RI a partir do nó que detetou o movimento..... | 37 |
| 23. Nó que recebeu uma mensagem RI..... | 37 |
| 24. Envio de uma mensagem RI (setas azuis) e de uma mensagem DA (seta castanha) a partir do nó que detetou o movimento..... | 38 |
| 25. Envio de uma mensagem RI a partir do nó que recebeu uma mensagem DA..... | 38 |
| 26. Envio de uma mensagem RI a partir do nó que recebeu uma mensagem DA..... | 38 |
| 27. Máquina de estados..... | 40 |
| 28. Fluxograma do ciclo principal algoritmo..... | 42 |
| 29. Fluxograma quando existe uma deteção de movimento..... | 43 |
| 30. Primeira deteção de movimento de um utilizador..... | 44 |
| 31. Segunda deteção de movimento de um utilizador..... | 45 |
| 32. Dois utilizadores a movimentarem-se..... | 45 |
| 33. Dois utilizadores cruzaram-se..... | 46 |
| 34. Cálculo da direcção errada..... | 47 |
| 35. Exemplo das direcções especiais..... | 48 |
| 36. Mapa utilizado nos testes..... | 50 |

| | |
|--|----|
| 37. <i>Map script</i> do mapa utilizado nos testes..... | 51 |
| 38. Caminhos percorridos pelos utilizadores no teste 1 | 52 |
| 39. <i>User script</i> utilizado no teste 1 | 52 |
| 40. Caminhos percorridos pelos utilizadores no teste 2 | 54 |
| 41. <i>User script</i> utilizado no teste 2 | 55 |
| 42. Caminhos percorridos pelos utilizadores no teste 3 | 56 |
| 43. <i>User script</i> utilizado no teste 3 | 57 |
| 44. Caminhos percorridos pelos utilizadores no teste 4 | 58 |
| 45. <i>User script</i> utilizado no teste 4 | 58 |
| 46. Caminhos percorridos pelos utilizadores no teste 5 | 60 |
| 47. <i>User script</i> utilizado no teste 5 | 61 |
| 48. Caminhos percorridos pelos utilizadores no teste 6 | 62 |
| 49. <i>User script</i> utilizado no teste 6 | 63 |

1. Introdução

1.1 Motivação

O consumo de energia mundial tem vindo a crescer e estima-se que aumente em 46% até 2040. Isto deve-se, em grande parte, aos países em desenvolvimento como a China, a Índia e o Brasil [1]. Com as questões ambientais que enfrentamos atualmente e com 87% da energia gerada mundialmente proveniente de fontes de energia não-renováveis [2], é necessário encontrar alternativas. As energias renováveis são uma das alternativas, no entanto, o seu crescimento não é rápido o suficiente para substituir as energias não-renováveis, pelo menos num futuro próximo [1]. Podemos, portanto concluir que estamos a assistir a um crescimento energético insustentável, que poderá ter graves consequências ambientais.

A eletricidade é a forma de energia que tem crescido mais rapidamente [3]. Nos últimos anos tem havido um maior foco na procura de novas tecnologias que permitam utilizar a eletricidade de forma mais eficiente. Uma das aplicações da eletricidade é a iluminação de espaços fechados de grandes dimensões. Estes espaços, normalmente, não têm iluminação natural, pelo que são obrigados a usar iluminação artificial.

Muitos dos espaços de grandes dimensões utilizam sistemas de iluminação bastante ineficientes. Estes sistemas, como no caso dos parques de estacionamento dos centros comerciais, normalmente estão ligados desde a hora de abertura até hora de fecho. Noutras situações, podem ser usados sensores de presença que acendem grandes áreas por períodos de tempo significativos, como sucede, por exemplo, nas garagens dos edifícios de habitação. Por haver pouco ou nenhum controlo das lâmpadas, é iluminada uma área muito superior à necessária. Isto faz com que haja desperdício de energia e maior desgaste das lâmpadas e consequentemente, os custos com a energia e a manutenção são superiores.

Para reduzir o desperdício, uma possível solução seria a utilização de lâmpadas LED de grande eficiência, em vez das lâmpadas fluorescentes, que são normalmente utilizadas nesses espaços. Mas isso apenas reduz o problema, não sendo a solução ideal. Para abordar este problema, este trabalho propõe um sistema de iluminação inovador que utiliza luminárias inteligentes, que iluminam apenas a área em que se encontram pessoas ou para onde estas se dirigem, reduzindo assim o desperdício de energia.

1.2 Objetivos

Tendo em conta a situação energética atual, este trabalho propõe um sistema de iluminação inteligente para espaços fechados de grande dimensão, como por exemplo parques de estacionamento

subterrâneos, que permita obter poupanças de energia consideráveis sem comprometer a qualidade da iluminação. Para tal, são utilizadas luminárias inteligentes. As luminárias são constituídas por lâmpadas LED de intensidade regulável, um sensor de movimento, um módulo de comunicação rádio e um microprocessador. Para além das luminárias inteligentes, também existe uma aplicação de configuração, que permite gerir e configurar os vários parâmetros das luminárias inteligentes.

O sistema proposto é uma solução distribuída. Isto significa que não existe um ponto de falha único, como acontece com os sistemas centralizados. Para além disso, também facilita a instalação do sistema, porque cada luminária inteligente é completamente independente, e facilita a escalabilidade da solução.

O funcionamento do sistema depende do algoritmo que irá correr no microprocessador de cada luminária inteligente. O sistema opera com base nos sensores de movimento e na informação recebida das luminárias vizinhas. Quando é detetado movimento, todas as luminárias num determinado raio acendem-se. Esse raio denomina-se raio de iluminação, cuja dimensão é configurável. O sistema procura também determinar a direção de deslocação do utilizador e acende as luminárias nessa direção, em antecipação, até uma certa distância. A essa distância chamamos distância de antecipação, que também é configurável. Ao fim de alguns segundos, se não for detetado movimento, as luminárias vão escurecendo até chegarem a uma intensidade mínima predefinida ou se apagarem. Um objetivo fundamental deste trabalho será explorar a capacidade do sistema em iluminar a área à volta dos utilizadores e identificar corretamente a sua direção de deslocação, iluminando esse espaço antecipadamente. Outro objetivo será estimar a poupança de energia obtida da solução proposta e comparar com outras soluções já existentes.

Como o sistema é destinado a espaços de grandes dimensões, seria necessário adquirir um número elevado de componentes, que teria custos significativos e seria complexo de realizar. Em alternativa, a solução será desenvolvida e testada com recurso a um simulador.

1.3 Estrutura do Documento

O documento está dividido em seis capítulos. No próximo capítulo é introduzido o trabalho relacionado, onde são apresentados diferentes sistemas de iluminação inteligente. No terceiro capítulo é apresentada e descrita a arquitetura da solução desenvolvida. O quarto capítulo será onde se fará uma descrição mais detalhada do funcionamento do algoritmo. No quinto capítulo são apresentados os testes efetuados e os respetivos resultados. Por fim, no sexto capítulo apresentam-se as conclusões do trabalho realizado e funcionalidades que poderão ser adicionadas no futuro.

2.Trabalho Relacionado

Nesta secção é apresentado o trabalho relacionado. Em cada subsecção é apresentado um sistema de iluminação inteligente e são descritas as suas funcionalidades e o seu funcionamento.

No final é feita uma pequena análise dos sistemas apresentados, onde se irá discutir as conclusões a que se chegou.

2.1 Sistema de iluminação baseado numa rede de sensores

Solução proposta

No artigo [4] é proposto o conceito de um sistema de iluminação de rua onde as lâmpadas ligam-se antes de os peões ou veículos passarem e desligam-se ou diminuem a intensidade quando não passa ninguém.

Descrição da solução

O sistema proposto tem as seguintes características:

- Fácil instalação e extensão: as lâmpadas inteligentes podem ser instaladas individualmente, tendo apenas de configurar os parâmetros. Por ser um sistema distribuído não é necessário um computador central que controle o sistema;
- Baixo custo: as lâmpadas inteligentes devem ter um baixo custo, para que haja retorno do investimento o mais rapidamente possível;
- Fácil atualização: o *firmware* de cada lâmpada inteligente tem que ser atualizada de forma rápida e fácil;
- Autodiagnóstico: as lâmpadas inteligentes têm de registar as falhas, em que se detetou movimento, mas não houve uma notificação antecipada das outras unidades;
- Sem diferenças das lâmpadas normais: as lâmpadas estão ligadas ainda antes de os peões ou veículos passarem. Os peões e os condutores não devem conseguir distinguir se as lâmpadas são inteligentes ou não;
- Segurança: o objetivo deste sistema é poupar energia e ao mesmo tempo manter a segurança. Para tal, é necessário que as lâmpadas que estão dentro do campo de visão dos peões ou condutores estejam ligadas;
- Privacidade: é necessário ter em atenção com a maneira como os dados recolhidos pelos sensores são tratados, pois podem fornecer informação dos comportamentos de alguém. Um dos objetivos deve ser manter a privacidade de todos aqueles que entram ou saem de uma zona com iluminação inteligente.

O sistema é constituído pelos seguintes componentes:

- Unidade da lâmpada
Consiste numa matriz de LED's de potência ajustável, sensor de luminosidade, sensor de movimento, dispositivo de comunicação e controlador. A lâmpada LED é ligada caso os sensores detetem movimento e envia uma mensagem às outras unidades. Desliga-se ou reduz a intensidade se não detetar nenhum movimento durante um certo intervalo de tempo.
- Unidade de sensor
Consiste no sensor de movimento, dispositivo de comunicação e controlador. Se detetar movimento envia uma mensagem às outras unidades. Esta unidade é instalada em vários sítios para garantir que as lâmpadas na rua são ligadas antes do peão ou veículo reparem nelas.
- Ponto de acesso
Consiste no dispositivo de comunicação e controlador. É usado no caso da distância entre as unidades das lâmpadas e as unidades de sensores ser demasiado grande para comunicarem entre si.

A Figura 1 mostra um exemplo do sistema de iluminação inteligente. As lâmpadas ligam-se antes do peão chegar e desligam-se ou reduzem a intensidade quando não está ninguém.

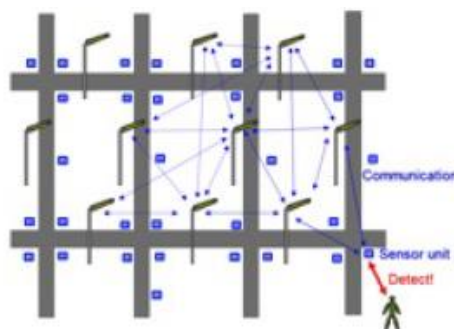


Figura 1 – Exemplo do sistema de iluminação inteligente

O sistema descrito não foi implementado, pelo que apenas se apresentam vários pontos associados à sua conceção.

A. Tipo de controlo

Existem duas maneiras de implementar o sistema:

- Controlo centralizado
Com controlo centralizado, temos um computador central que recebe dados de todos os sensores e enviam comandos às lâmpadas inteligentes. A unidade de controlo controla a iluminação através dos comandos recebidos do computador central, envia os dados obtidos

pelos sensores e transmite os dados enviados pelas outras lâmpadas, sensores e computador central.

Cada lâmpada não tem muita informação e capacidade de processamento, enquanto o computador central dispõe de toda a informação para controlar todas as luzes.

Esta solução tem alguns problemas. Um dos problemas é a robustez; se um computador controlar todo o sistema, a falha desse computador implica a falha do sistema. Uma solução seria ter vários computadores com o estado do sistema, no entanto a sincronização entre os computadores é um problema.

- **Controlo distribuído**

Com controlo distribuído, cada lâmpada decide ligar-se ou desligar-se com base na informação recebida. A unidade de controlo controla a intensidade da iluminação autonomamente, com base na informação recebida dos sensores e das outras unidades e tem de decidir o destino da informação obtida pelos sensores.

B. Rede

O desenho da rede depende do tipo de controlo da iluminação inteligente. Com o controlo centralizado, o tráfego da rede passa todo pelo computador central, sendo necessário adotar métodos para reduzir o tráfego entre os sensores e o computador central.

Com controlo distribuído, cada lâmpada ou sensor tem de decidir o destino da informação a enviar, sendo também importante adotar métodos para reduzir o tráfego nas redes de sensores.

C. Tempo para ligar e desligar

O objetivo da iluminação de rua é manter a segurança e o objetivo da iluminação inteligente é poupar energia. Portanto, o momento de ligar ou desligar as luzes é importante. Se as luzes se desligarem passado pouco tempo está-se a pôr em causa a segurança; existe um compromisso entre manter a segurança e poupar energia. Existem diferentes métodos para decidir quando se deve apagar a luz:

- **Temporizador:** cada lâmpada tem um temporizador, que indica quando é que a luz se deve desligar. O tempo de cada temporizador é diferente, pois as luzes têm de estar ligadas enquanto estiverem no campo de visão do peão ou do condutor. Para tal, é necessário fazer um levantamento para saber durante quanto tempo as luzes estão nos seus campos de visão.
- **Não há deteção de veículos ou peões:** a lâmpada desliga-se quando não há movimento detetado por nenhum dos sensores. No entanto pode acontecer o caso dos sensores não detetarem um peão ou veículo, pelo que é necessário ter um algoritmo que consiga prevenir esta situação.

Embora este sistema não tenha sido implementado e seja um sistema de iluminação pública, alguns dos princípios utilizados neste sistema podem ser utilizados noutros sistemas de iluminação baseados em redes de sensores.

2.2 Twilight

A Twilight [5] é uma empresa de controlo de iluminação inteligente e especializam-se em sensores para o exterior, controlo de iluminação sem fios e *software* de gestão de iluminação para aplicações de iluminação de rua. Dos vários produtos disponíveis, o CityManager e o CitySense são os que mais se relacionam com este trabalho.

CityManager

O CityManager [6] é um *software* de gestão de iluminação que permite monitorizar, gerir e controlar toda a infraestrutura de iluminação da cidade. O CityManager tem as seguintes características:

- Permite configurar, monitorizar, gerir e controlar a iluminação remotamente, individualmente ou em grupo, com informação em tempo real;
- Permite visualizar informações relativamente a uma lâmpada ou a um grupo de lâmpadas. As informações disponíveis incluem o tipo de lâmpada, as definições da lâmpada, gastos de energia, as deteções do sensor e a monitorização do tráfego;
- Fornece uma análise automática e uma avaliação dos dados recolhidos;
- Consegue identificar falhas e reportá-las, através de relatórios, de forma automática;
- As lâmpadas estão representadas numa interface gráfica no GoogleMaps e que permite localizar, monitorizar e controlar as lâmpadas;
- Todas as comunicações são cifradas para que o funcionamento do sistema e o envio de dados sejam seguros;
- É compatível com diferentes tipos de lâmpadas e diferentes controlos de lâmpadas;
- Suporta níveis de iluminação que variam consoante o contexto e condições, tais como, horas de mais ou menos movimento, nascer e pôr-do-sol, alterações no clima e situações de emergência;
- É compatível com *software* e hardware de terceiros, o que permite a implementação de serviços adicionais;

CitySense

O CitySense [7] é um sensor que ajusta a intensidade da luz consoante detete a presença de peões ou de veículos. As lâmpadas escurecem quando ninguém é detetado e acendem-se, iluminando a área em torno da zona onde se detetou a presença. O CitySense tem as seguintes características:

- Os sensores de presença, a comunicação sem fios e o controlo da iluminação estão integrados num único produto. Isto faz com que a instalação seja bastante simples;

- Quando é detetada uma presença podem ativar-se até 20 lâmpadas vizinhas;
- Consegue controlar diferentes tipos de lâmpadas (vapor de sódio, vapor de mercúrio, fluorescentes e LED);
- São usados um conjunto de sensores que combinados com um algoritmo inteligente conseguem detetar peões e veículos, enquanto filtra objetos indesejados (pequenos animais, chuva e neve);
- Pode ser controlado e monitorizado através do CityManager;
- Notifica falhas e outros aspetos relativos à manutenção através do CityManager.

Caso de estudo – Parque de estacionamento exterior

Neste caso de estudo [8] utilizou-se o CityManager e o CitySense num parque de estacionamento exterior. O parque de estacionamento está localizado nos arredores de Groningen, na Holanda e é um Park+Ride (parque de estacionamento com acessos à rede de transportes públicos), é usado por milhares de pessoas diariamente. Os carros são deixados no parque e as pessoas apanham os transportes públicos para irem para a cidade. O sistema de transporte público funciona até à meia-noite e a partir dessa hora a ocupação do parque é bastante reduzida. No entanto, por uma questão de segurança, todas as lâmpadas do parque estão acesas na intensidade máxima, a noite inteira. Isto causa um grande desperdício de energia, para além de aumentar a conta da energia da câmara municipal.

Para resolver este problema, a câmara municipal escolheu instalar unidades CitySense em todo o parque de estacionamento. As luzes funcionam com 20% de intensidade quando não é detetada nenhuma presença. Assim que é detetada uma presença, as luzes à volta do ocupante brilha com intensidade máxima. Para além disto, a câmara municipal também instalou o software CityManager para monitorizar as lâmpadas. Tudo isto permitiu ter poupanças de energia de 54%, sem comprometer a segurança dos peões e veículos.

2.3 SSL (Smart Street Lighting)

Solução proposta

Os autores do artigo [9] apresentam-nos o SSL (Smart Street Lighting), um sistema que liga e desliga a iluminação de rua dinamicamente, baseado na localização dos peões e na sua “zona de segurança”. A localização de cada peão é obtida através do seu *smartphone*, que envia periodicamente informação de localização e configuração, para o servidor SSL. Cada poste de iluminação está equipado com um dispositivo de rádio, que recebe informações do servidor SSL.

Descrição da solução

O sistema SSL tem as seguintes características:

- Eficiência energética: como as lâmpadas só são ligadas quando são “necessárias”, obtêm-se grandes poupanças de energia;
- Zonas de segurança: o sistema permite que cada utilizador configure a sua zona de segurança. A zona de segurança refere-se à área, por utilizador, que é iluminada;
- Conhecimento do contexto: não é necessário que um utilizador do sistema explicitamente execute uma aplicação, no seu *smartphone*, para ativar o controlo da iluminação. Isso é tratado em segundo plano, no entanto é possível interagir com o sistema e alterar, por exemplo, o tamanho da área de segurança;
- Privacidade e segurança: todos os dados enviados automática e periodicamente do dispositivo do utilizador são anónimos. Informação trocada com o utilizador é reescrita no ciclo seguinte, impedindo a criação de um histórico do utilizador;
- Expansibilidade do sistema: todos os componentes de *software* foram desenhados para serem extensíveis e escaláveis.

O sistema é constituído pelos seguintes componentes:

- Postes de iluminação com *motes* ZigBee instalados;
- Um servidor SSL;
- Estação base de rádio, que é responsável pela comunicação entre o servidor e os postes de iluminação;
- Dispositivos dos utilizadores (*smartphones*), com GPS, ligação à Internet e aplicações para *smartphone* para os administradores e utilizadores;

Para determinar que lâmpadas se devem ligar ou desligar em torno do utilizador, utilizaram-se zonas para o utilizador (zona radial) e para as lâmpadas ativáveis (zona poligonal de infraestrutura). As lâmpadas são ligadas apenas se houver sobreposição das zonas do utilizador e da infraestrutura (Figura 2). Para além disto, a última posição do utilizador tem que respeitar uma restrição de tempo,

caso contrário a posição do utilizador é dada como inválida. A Equação 1 apresenta uma descrição mais formal:

$$z \in Z, l \in L, u \in U \quad on(l) \equiv \exists z, \exists u : inside(l, z) \wedge inside(u, z) \\ \wedge active(l) \wedge active(z) \wedge distance(u, l) < r \wedge age(u) < t$$

Equação 1

Explicação da equação:

- z é uma zona, pertencente ao conjunto de zonas Z ;
- l é uma lâmpada, pertencente ao conjunto de lâmpadas L ;
- u é um utilizador, pertencente ao conjunto de utilizadores U ;
- $on(l)$ indica o estado da lâmpada l ;
- $inside(l, z)$ é verdade se a lâmpada l está na zona z ;
- $inside(u, z)$ é verdade se o utilizador u está na zona z ;
- $active(l)$ e $active(z)$ são verdade se a lâmpada l e a zona z estão ativas, pois podem ser desativadas pelo administrador;
- $distance(u, l)$ descreve a distância entre o utilizador u e a lâmpada l ;
- r descreve a distância da zona de segurança;
- $age(u)$ é a idade da última posição conhecida do utilizador u ;
- t é a restrição de tempo.

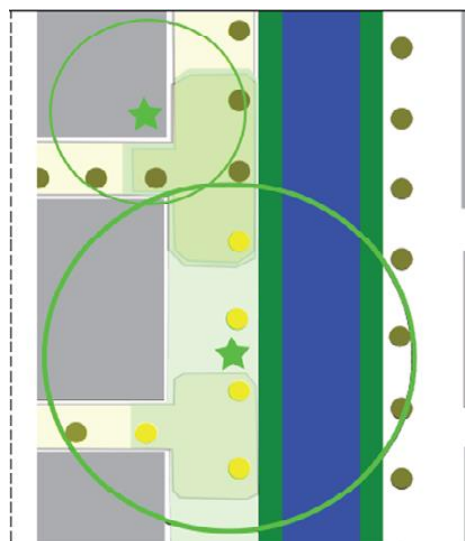


Figura 2 – Exemplo do SSL. As zonas cinzentas representam edifícios; a zona azul representa um obstáculo (estrada, rio); os pontos amarelos são lâmpadas acesas; as estrelas representam a localização dos utilizadores; os círculos representam as zonas de segurança dos utilizadores.

Para o servidor SSL, qualquer computador com o sistema operativo Linux pode ser usado ou, em alternativa, um computador com o sistema operativo Windows com o Cygwin instalado. O desempenho do computador irá depender do tamanho do sistema SSL, i.e. o número de utilizadores, frequência de atualização da localização dos utilizadores, número de postes de iluminação e número de polígonos de infraestruturas.

Os nós ZigBee dos postes de iluminação têm duas funções. A primeira é encaminhar qualquer mensagem que recebam, exceto se a mensagem for destinada ao nó que a recebeu. A segunda função é controlar a lâmpada, ligando-a ou desligando-a conforme necessário. A comunicação entre os nós ZigBee dos postes de iluminação e o servidor SSL é feita através de um nó ZigBee chamado “nó base”, que funciona como uma *gateway* entre o servidor SSL e os restantes nós ZigBee.

A aplicação de administração tem como função a atribuição da posição aos postes de iluminação. Antes de aceder à aplicação, é necessário inserir a chave de administração. Após o registo, os administradores podem adicionar, atualizar ou remover lâmpadas da base de dados, através do seu identificador.

A aplicação do utilizador normalmente corre em segundo plano, transmitindo a localização do utilizador periodicamente, juntamente com o tamanho da zona de segurança para o servidor SSL. Os utilizadores apenas interagem com o sistema SSL, por exemplo, quando alteram o tamanho da zona de segurança. Por uma questão de segurança, a identificação dos utilizadores é feita através de um identificador temporário, gerado aleatoriamente. Como medida de segurança adicional, é usado um mecanismo que verifica a integridade dos dados do utilizador, para prevenir o acesso não autorizado ao sistema. Para prevenir que pessoas não autorizadas falsifiquem a posição de um utilizador, juntamente com a informação, também é enviada uma chave de autenticação gerada a partir dos dados a serem enviados e dos dados enviados anteriormente. Com isto o servidor SSL, consegue verificar a integridade dos dados do utilizador e impedir que a mesma informação seja usada ilicitamente uma segunda vez.

O acesso à interface da aplicação de administração é protegido através de uma chave de administração temporária. Antes do primeiro uso do SSL, todos os postes de iluminação têm que estar registados usando a aplicação no dispositivo do administrador. Para fazer o registo dum poste de iluminação é necessário o endereço físico do nó ZigBee e a sua localização GPS. Os dados da localização são enviados para o servidor SSL e guardados numa base de dados espacial. Assim que as lâmpadas estão registadas na base de dados, é usada uma aplicação *web* para manter as zonas de infraestruturas (adicionar ou remover zonas poligonais, postes de iluminação, etc.).

No dispositivo do utilizador (*smartphone*), a aplicação SSL envia, periodicamente, a posição atual do utilizador e o raio da zona de segurança ao servidor SSL. Assim que o servidor recebe nova informação, pesquisa na base de dados pelas lâmpadas que serão ligadas ou desligadas, cria pacotes com os comandos a serem executados e envia-os para os nós ZigBee.

Esta solução, apesar de ter funcionalidades interessantes, como a zona de segurança, apresenta uma infraestrutura bastante complexa, exigindo ao utilizador que disponha de um *smartphone*, com GPS e ligação à Internet. Para além disso, a utilização de um computador central que controla todo o sistema apresenta-se como um ponto de falha do sistema.

2.4 Sistema de controlo de iluminação baseado numa rede sensores

Solução proposta

Os autores do artigo [10] apresentam-nos um sistema de controlo da iluminação das estradas. Este sistema é baseado nas redes de sensores sem fios e permite monitorizar e controlar, em tempo real, a iluminação de vias públicas.

Descrição da solução

O sistema pode ser dividido em duas partes: o centro de controlo e as zonas. O centro de controlo é constituído pelo computador anfitrião e por um modem GPRS. Cada zona é constituída por um módulo de comunicação e por lâmpadas inteligentes. São utilizadas duas redes diferentes para a comunicação sem fios: o GPRS é utilizado para comunicar a longas distâncias, entre o centro de controlo e as diferentes zonas e a rede ZigBee é utilizada para comunicações a curta distância, ou seja, comunicações dentro de cada zona. O sistema consegue controlar até 4096 zonas e cada zona consegue ter até 65535 lâmpadas, o que consegue satisfazer as necessidades de iluminação das vias públicas de uma cidade.

Cada lâmpada é constituída por três partes: um módulo de comunicação ZigBee, um módulo de controlo e um microprocessador. O módulo de comunicação sem fios utiliza a tecnologia ZigBee por ser muito flexível e por construir a rede autonomamente, sem intervenção externa. O módulo de controlo tem a função de controlar a intensidade da luz, ligar ou desligar a lâmpada e verificar o estado do equipamento. O sistema permite três intensidades: intensidade alta, intensidade média e intensidade baixa.

A segurança também é algo que foi considerado neste sistema. O algoritmo de cifra adotado foi o AES-128. No entanto, com este algoritmo, qualquer texto cifrado que provenha do mesmo texto original é sempre igual. Isto faz com que reutilizando o texto cifrado se consiga entrar no sistema. A solução encontrada foi criar um contador síncrono, onde o valor do contador é adicionado ao texto antes de ser cifrado. Esta solução faz com que o mesmo texto, depois de cifrado, seja diferente devido à adição do valor do contador.

O módulo de comunicação de cada zona tem como função converter a informação recebida da rede GPRS e convertê-la para a rede ZigBee e vice-versa. Este módulo é constituído por um módulo GPRS, um módulo ZigBee e um microprocessador.

O *software* de gestão do sistema oferece uma maneira simples e prática para os utilizadores controlarem o sistema. Este *software* corre no computador anfitrião e permite monitorizar, controlar, fazer pesquisas e controlar automaticamente o sistema. O *software* tem as seguintes funcionalidades:

- Consegue monitorizar e mostrar as condições de funcionamento das lâmpadas;
- Quando existe algum evento anormal, um alarme automático é despoletado;

- O *software* permite ligar ou desligar as lâmpadas e controlar a sua intensidade individualmente ou em grupo;
- Permite fazer pesquisas de acordo com as condições do utilizador e mostrar os seus resultados;
- Consegue ser programado para ligar ou desligar e controlar a intensidade das luzes a determinada hora;
- O *software* automaticamente cifra e decifra toda a informação enviada e recebida.

Avaliação experimental

O sistema apresentado foi testado em pequena escala, durante pouco mais de dois meses. Os resultados obtidos mostram que as poupanças de energia rondam os 40%. Também se mostrou que o sistema é estável e fiável, sendo que a taxa de perda de dados na comunicação é inferior a 3% e o *software* de gestão tem uma interface simples e é fácil de utilizar.

2.5 Controlo de iluminação baseado em sistema de vídeo

Solução proposta

No artigo [11], os autores apresentam-nos um sistema embebido de processamento de vídeo, que controla a intensidade da iluminação num parque de estacionamento, através da deteção de movimento. A intensidade da iluminação é aumentada quando são detetadas pessoas ou veículos em movimento e diminuída em caso contrário, conseguindo assim reduzir o consumo de energia.

Na solução proposta são utilizadas lâmpadas LED de intensidade regulável e um sensor de vídeo inteligente que permite controlar a intensidade da iluminação. Este sistema utiliza análise de vídeo em tempo real para detetar movimento.

Descrição da solução

Este sistema foi desenhado para ser utilizado em parques de estacionamento ao ar livre, com postes de iluminação com uma altura superior a 10 metros e que iluminem uma área superior a 1000 metros quadrados. A câmara está montada juntamente com a lâmpada e está apontada diretamente para o chão.

Existem três configurações diferentes de iluminação: *desligado* durante o dia, *baixo* quando não é detetada atividade e *alto* quando é detetada atividade. A mudança entre as configurações é feita pela análise de vídeo em tempo real.

A análise de vídeo tem como objetivo detetar a existência de veículos ou pessoas. Estes alvos têm de ser detetados e monitorizados enquanto se movem. Se os alvos pararem completamente ainda têm de ser monitorizados pelo sistema durante algum tempo, apesar dos carros estacionados não serem do interesse do sistema.

Um dos principais desafios da análise de vídeo consiste em detetar os alvos com sucesso numa grande área, em tempo real. Para que tal seja possível, a resolução do vídeo é limitada a 640x480 pixels.

A calibração da câmara é bastante simples, sendo apenas necessário conhecer a altura a que a câmara vai ser instalada. Inicialmente é gerada uma *lookup table*, que contém o tamanho e a orientação de uma pessoa, para cada pixel da imagem (Figura 3). A partir desta tabela consegue-se obter o tamanho relativo de qualquer alvo que seja detetado na imagem. Também é criada uma máscara de visibilidade que exclui as áreas onde os alvos são demasiados pequenos para serem detetados.

O módulo de deteção de movimento identifica que pixels se estão a mover numa imagem utilizando a diferenciação temporal de três *frames*. Considera-se que um pixel está em movimento se o valor da sua intensidade é diferente em ambos os *frames* de referência do futuro e do passado.

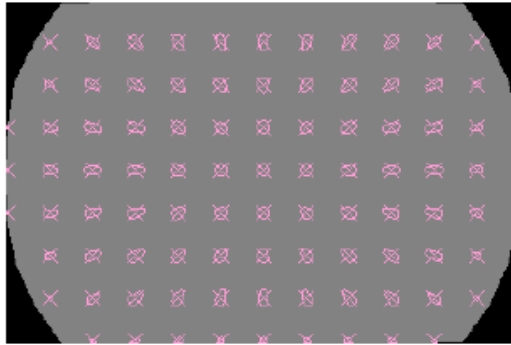


Figura 3 – Máscara de visibilidade e exemplos de alvos humanos. As regiões a negro não são processadas.

A saída de módulo de detecção de movimento é uma imagem booleana que representa os pixels em movimento. O próximo passo é combinar os pixels em *blobs* usando a análise de componentes ligados. Como este passo é o mais intensivo computacionalmente, foi adotada uma estratégia do tipo dividir e conquistar. É feito um processo de varrimento vertical e horizontal que divide a máscara do movimento numa lista de regiões mais pequenas. A análise de componentes ligados é depois aplicada a cada uma destas regiões de movimentos.

A saída da análise de componentes ligados é uma lista de *blobs* isolados. Esses *blobs* podem corresponder a diferentes objetos, tais como:

- Alvos válidos (pessoas e veículos);
- Alvos válidos parcialmente detetados;
- Objetos que causam ruído de fundo, como pó, insetos ou gotas de chuva;
- *Blobs* de ruído devido ao *jitter* ou *aliasing* da câmara.

Os alvos humanos são os mais importantes nesta aplicação, pelo que não se pretende comprometer a segurança para poupar energia. Portanto é necessário que o sensor de vídeo tenha uma taxa muito baixa de falsos negativos. Para atingir este objetivo, são examinados todos os *blobs* cujo tamanho de imagem é menor que o duma pessoa. Tenta-se aumentar o tamanho do *blob* verificando, para cada pixel da fronteira do *blob*, os pixels na sua vizinhança. O *blob* é expandido, passando a incluir o pixel que está a ser testado, se as duas condições seguintes forem cumpridas:

- O valor da intensidade do pixel é muito semelhante à do pixel do *blob* a que está ligado;
- A alteração menor entre os dois *frames* de referência é maior do que metade da mudança média do *blob*.

O aumento do tamanho do *blob* está limitado pelo tamanho da imagem duma pessoa, obtido a partir da calibração da câmara. Após o processo de crescimento do *blob*, *blobs* separados são juntos se a sua junção não for superior ao tamanho de um *blob* de um humano. De seguida, são removidos todos os *blobs* cuja área é inferior à área de um *blob* de um humano.

Os insetos e as gotas de chuva são dois dos alvos inválidos mais vistos e que não podem ser filtrados com base no tamanho por estarem perto da câmara. Devido ao sensor utilizado, estes alvos têm a característica de terem uma cauda longa e brilhante. Esta característica é usada para filtrar estes

alvos. Na última fase da filtragem, são detetados e removidos os falsos *blobs* causados pelo *jitter* e *aliasing* da câmara.

Após o processo de filtragem, ainda podem existir *blobs* que não correspondem a alvos válidos. Para serem considerados alvos válidos os *blobs* têm de ter uma velocidade consistente numa janela de tempo e têm de ser consistentes na aparência.

Avaliação experimental

O desempenho do sistema foi avaliado utilizando cerca de 10 horas de vídeo. Os dados foram obtidos ao longo de vários dias, sob diferentes condições ambientais (dia, noite, sol e chuva). As métricas utilizadas para medir o desempenho são as deteções falhadas e as falsas deteções.

Na instalação do sistema, adicionou-se um atraso de 2 minutos após o alvo deixar o campo de visão antes de se diminuir a intensidade das luzes. O desempenho foi avaliado tendo em conta este atraso de 2 minutos, estando ilustrado na Tabela 1.

Como podemos observar na Tabela 1, o maior problema foi a chuva, que foi responsável por grande parte das deteções falhadas e das falsas deteções. Tirando isso, as deteções falhadas eram normalmente pessoas que não se moviam o suficiente e as falsas deteções eram geralmente causadas por insetos.

| Dados | Duração | Deteções falhadas | Falsas deteções |
|------------------|----------------|--------------------------|------------------------|
| Tudo | 9:53 h | 1,68% | 15,90% |
| Noite | 5:17 h | 1,13% | 13,22% |
| Noite, chuva | 1:00 h | 8,00% | 44,49% |
| Noite, sem chuva | 4:17 h | 0,67% | 1,07% |

Tabela 1 – Desempenho do sistema

2.6 Discussão

Ao analisarmos o estado da arte, podemos verificar que os sistemas apresentados apesar de terem todos como objetivo reduzir o consumo de energia, possuem arquiteturas bastantes diferentes. Na Tabela 2 apresentam-se as principais características de cada sistema de uma forma resumida. As soluções apresentadas, embora não sejam destinadas a espaços fechados, mas a ambientes exteriores, são as que mais se assemelham ao nosso problema pois reduzem o consumo de energia, iluminando as áreas apenas na presença de pessoas e utilizam estratégias de controlo de iluminação.

Com os sistemas [6] e [10] verifica-se a importância de este tipo de sistemas possuir uma aplicação que permita controlar o sistema de iluminação. As aplicações utilizadas nesses dois sistemas permitem controlar o sistema remotamente e ao mesmo tempo facilitar tarefas de manutenção do sistema e oferecer outras funcionalidades.

Analisando os sistemas dos artigos [7] e [9], ambos iluminam a área em torno do utilizador, mas a abordagem seguida por cada um é bastante diferente. Enquanto o primeiro é um sistema distribuído, gerido por uma aplicação de controlo, o segundo é um sistema centralizado, que funciona com base na localização dos utilizadores. Tendo em conta que atualmente o *hardware* é relativamente barato e poderoso e que estes sistemas iluminam a via pública, a utilização de uma arquitetura centralizada pode não ser a melhor opção. O risco de um sistema centralizado falhar e desligar a iluminação pública de uma cidade pode justificar a aposta numa abordagem distribuída, ainda que esta se mostre mais complexa de implementar.

| | Tipo de Iluminação | Arquitetura | Recolha de dados | Software de controlo |
|---|---------------------------------------|--------------------|--|-----------------------------|
| Sistema de iluminação baseado numa rede de sensores | Iluminação pública | - | Sensores instalados em vários pontos da rua | Não |
| Tvilight | Iluminação pública | Distribuído | Sensores instalados nas luminárias (CitySense) | Sim (CityManager) |
| SSL (Smart Street lighting) | Iluminação pública | Centralizado | Smartphone do utilizador | Não |
| Sistema de controlo de iluminação baseado numa rede de sensores | Iluminação pública | Centralizado | Módulo de controlo lâmpada inteligente | Sim |
| Controlo de iluminação baseado em sistema de vídeo | Parques de estacionamento ao ar livre | Distribuído | Câmaras de vídeo | Não |

Tabela 2 – Resumo das características dos sistemas apresentados

3. Descrição da Solução Proposta

Neste capítulo é apresentada uma descrição do trabalho a desenvolver, os objetivos que o sistema tem que cumprir, as restrições relacionadas com este trabalho, a abordagem escolhida e algumas considerações em relação à implementação.

3.1 Descrição Funcional

Na presente dissertação pretende-se desenvolver um sistema distribuído de iluminação com controlo inteligente. O sistema proposto tem como alvo espaços de grande dimensão, onde exista um grande número de lâmpadas. Com este sistema pretende-se reduzir os gastos com a energia e com a manutenção, contribuindo para reduzir a pegada ecológica.

Por este ser um sistema distribuído, a inteligência do sistema reside nas luminárias. Cada luminária é uma luminária inteligente que possui capacidade de processamento, consegue detetar movimento, troca informação com as luminárias vizinhas e controla a intensidade da luz. Cada luminária representa um nó no sistema, sendo constituídas por uma lâmpada LED, regulável em intensidade, um sensor de movimento, um módulo de comunicação rádio e um microprocessador. Assume-se a existência de quatro intensidades: alta, média, baixa e mínima, cujo valor concreto pode ser configurado.

As luminárias inteligentes são exatamente iguais entre si e assume-se que a sua disposição física no espaço tem por base uma disposição matricial. Como o sistema funciona com base nas luminárias inteligentes a instalação de um sistema torna-se simples, assim como, a realização de alterações que venham a ser necessárias no futuro. Outra vantagem da abordagem distribuída é a não existência de um ponto de falha único, como acontece nos sistemas centralizados.

O funcionamento do sistema tem por base a utilização de sensores que detetam o movimento dos utilizadores e funciona como descrito em seguida:

- Quando é detetado movimento, as lâmpadas dentro de um determinado raio a partir da zona onde houve deteção de movimento são ligadas. Este raio será designado raio de iluminação (Figura 4). O nó que detetou movimento fica com intensidade alta, os nós na orla do raio de iluminação ficam com intensidade baixa e os restantes nós ficam com intensidade média;
- Quando o sistema determina que o utilizador se desloca numa direção, as lâmpadas que estão na direção prevista acendem-se ao longo duma determinada distância e com o raio de iluminação. Essa distância será designada de como a distância de antecipação (Figura 5). Os nós ao longo da distância de antecipação ficam com intensidade alta;
- Se não for detetado movimento durante algum tempo as lâmpadas que estão acesas vão escurecendo até atingirem a intensidade mínima.

O sistema deve funcionar como descrito mesmo com múltiplos utilizadores no espaço, que se podem deslocar individualmente ou em grupo e em qualquer direção.

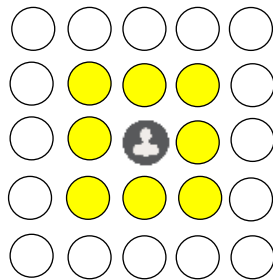


Figura 4 – Foi detetado movimento. As lâmpadas à volta do utilizador acenderam-se, com o raio de iluminação igual a 1.

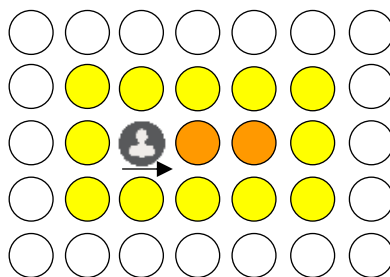


Figura 5 – O utilizador moveu-se da esquerda para a direita. Acenderam-se as lâmpadas na direção prevista (a laranja) e à sua volta, com raio de iluminação igual a 1 e distância de antecipação igual a 2.

3.2 Principais componentes do sistema

O componente básico deste sistema é a luminária inteligente, que já foi descrita acima. Presente em cada luminária está um algoritmo responsável pelo comportamento do sistema.

O algoritmo é a parte principal do sistema e define o comportamento de uma luminária, sendo responsável por interpretar a informação recebida, alterar o estado da lâmpada e enviar informação às outras luminárias inteligentes. O algoritmo deve permitir que o sistema funcione corretamente, ocupando o mínimo de recursos computacionais possível e procurando minimizar o número de mensagens transmitidas.

Durante o funcionamento do sistema existem parâmetros que podem ser alterados, como por exemplo, o raio de iluminação. Para isso será utilizada uma aplicação de configuração. A aplicação de configuração permite gerir e monitorizar o sistema. É executada num computador sendo utilizada para

configurar o sistema durante a instalação e alterar os parâmetros do sistema consoante as preferências do utilizador. Permite adicionar e remover luminárias do sistema e visualizar informações e estatísticas do sistema relativas ao consumo e poupança de energia.

A comunicação entre a aplicação de configuração e as luminárias é assegurada por um elemento *gateway*. Quando algum parâmetro do sistema é alterado através da aplicação de configuração, essa informação é enviada para o *gateway* que depois transmite essa informação para as luminárias inteligentes. O elemento *gateway* constituído por um módulo de comunicação rádio e por um módulo que converte o formato da informação recebida da aplicação no formato usado pelas luminárias e vice-versa.

3.3 Implementação

O objetivo deste trabalho é o desenvolvimento de parte dos componentes do sistema, nomeadamente, o algoritmo e a aplicação de configuração. O desenvolvimento e o teste de um protótipo real do sistema proposto implicaria adquirir um grande número de componentes, para implementar um número significativo de luminárias inteligentes. Tal seria um trabalho muito extenso e teria custos consideráveis. Em vez disso será utilizado um simulador para dar suporte ao desenvolvimento e teste dos algoritmos do sistema.

De seguida irão ser descritos os componentes do trabalho a desenvolver, os requisitos que devem cumprir e como foram implementados.

3.3.1 Simulador

O simulador é uma aplicação que irá dar suporte ao teste do funcionamento do sistema. Com o simulador é possível criar ambientes de teste com diferentes dimensões e com um qualquer número de nós e de utilizadores. Os testes serão realizados simulando múltiplos utilizadores a deslocarem-se no espaço e visualizando o estado das lâmpadas, permitindo verificar se o comportamento do sistema corresponde ao esperado.

Escolha do Simulador

Para o simulador existem duas opções: criar um simulador de raiz específico para este trabalho ou utilizar um simulador para redes de sensores sem fios, que esteja disponível gratuitamente.

Se utilizarmos um simulador que já esteja disponível, existe uma curva de aprendizagem, que varia consoante a informação disponível e a própria complexidade do simulador. Além disso,

geralmente, é necessário desenvolver o projeto numa linguagem imposta pelo simulador, que pode não ser a mais adequada ao desenvolvimento do trabalho.

Por outro lado, se criarmos um simulador de raiz específico para este projeto existe um gasto de tempo associado ao seu desenvolvimento. No entanto, tem a vantagem de poder ser utilizada qualquer linguagem de programação e a flexibilidade de adicionar ou remover funcionalidades como desejado.

Numa primeira fase, tentou-se utilizar um simulador existente. Para que o simulador pudesse ser utilizado neste trabalho, teria que cumprir os seguintes requisitos:

- Possuir uma interface gráfica que permita visualizar as luminárias e os utilizadores, enquanto a simulação corre;
- Configurar o ambiente de teste, permitindo alterar o tamanho e formato do espaço e o número de nós e de utilizadores presentes;
- Permitir definir os trajetos dos utilizadores e simular o seu movimento;
- Controlar a simulação, a qual pode ser colocada em pausa e retomada a qualquer instante;

Um levantamento dos simuladores disponíveis foi realizado, sendo que os simuladores mais promissores foram testados. Entre os simuladores testados encontram-se o OMNet++ [12], o J-Sim [13], SIDnet-SWANS [14] e Ptomely II [15].

Nos testes efetuados, nenhum dos simuladores permitia executar todas as operações acima descritas, pelo que se optou por desenvolver um simulador específico para este trabalho.

Implementação

Para o desenvolvimento do simulador e do restante projeto, a linguagem de programação escolhida foi o Java, por possuir diversas bibliotecas e *frameworks* que poderiam facilitar o desenvolvimento. Em particular, para este trabalho foi utilizada a *framework* Swing para desenvolver a interface gráfica. O ambiente de desenvolvimento utilizado foi o NetBeans, que possui a ferramenta GUI Builder, que facilita a criação de interfaces gráficas.

Funcionamento

A Figura 6 representa o simulador desenvolvido. O simulador é controlado através dos vários botões presentes e possui a área do mapa (área retangular), onde é possível observar a simulação. Para além das funcionalidades descritas acima também foram acrescentadas as seguintes funcionalidades:

- Executar a simulação passo a passo;
- Alterar a velocidade de execução da simulação;
- Mostrar as poupanças de energia obtidas;
- Simular a falha de um nó;

- Mostrar todas as mensagens trocadas entre os nós e estatísticas relacionadas com o envio de mensagens.

Para executar uma simulação, antes de mais é necessário definir o ambiente de teste. Para tal são utilizados dois ficheiros de texto: o *map script* e o *user script*. O *map script* contém um conjunto de comandos que define o ambiente de teste, ou seja, o seu tamanho e formato, o número de nós, o raio de iluminação e distância de antecipação, entre outras coisas.

O *user script* contém os comandos que controlam os utilizadores, indicando a sua posição inicial e o seu caminho. Os dois ficheiros são seleccionados através de um explorador quando carregamos nos botões *Map Script*, e *User Script* e são carregados para o simulador através do botão *Load*. De seguida aparece na área do mapa os nós e os utilizadores nas suas posições iniciais. Para desenhar o mapa considera-se que a disposição dos nós é matricial e que os eixos dos *xx* e dos *yy* têm a orientação representada na Figura 6.

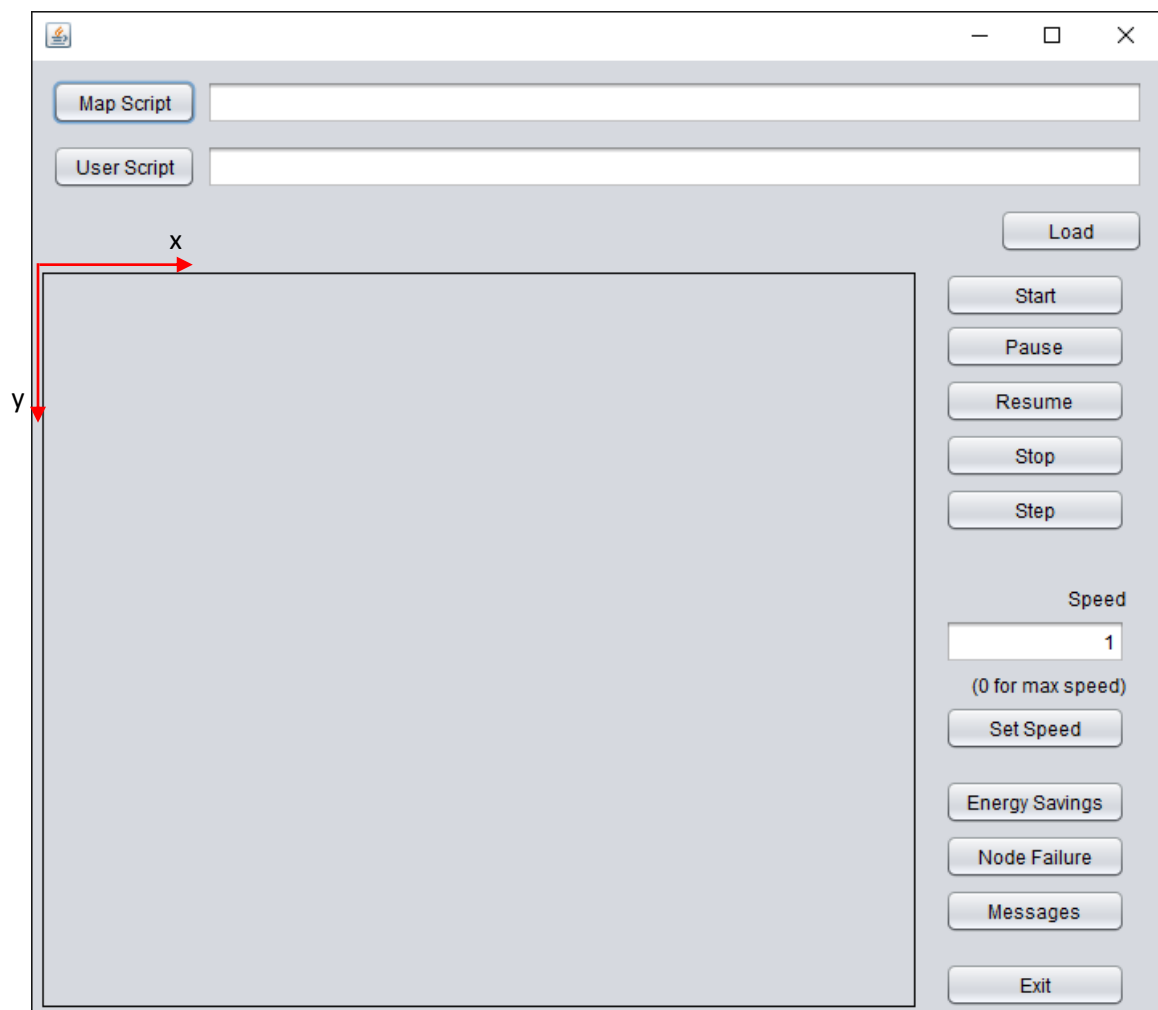


Figura 6 – Ecrã inicial do simulador. As setas a vermelho representam a orientação dos eixos dos *xx* e dos *yy*.

Antes de iniciarmos uma simulação é possível ainda alterar a velocidade da mesma. O valor da velocidade representa o tempo, em segundos, entre cada movimento dos utilizadores. O valor inicial da velocidade é 1, o que significa que os utilizadores se movem a cada segundo. Para alterar o valor da velocidade inserimos o valor desejado no campo *Speed* e clicamos em *Set Speed*. O valor da velocidade tem que ser superior ou igual a 0. O valor 0 representa a velocidade máxima do simulador, que corresponde a 0,25 segundos.

A simulação é iniciada assim que carregamos no botão *Start*. Depois de iniciada a simulação, podemos efetuar pausas, retomar e parar a simulação em qualquer altura. Isto é feito, respetivamente, através dos botões *Pause*, *Resume* e *Stop*. Também é possível correr a simulação passo a passo. Para tal, efetua-se a pausa da simulação e de seguida carrega-se no botão *Step* para que os utilizadores se movam para a próxima posição.

Quando a simulação termina aparece uma caixa de diálogo a informar o fim da simulação. Terminada a simulação, se carregarmos no botão *Energy Savings* é apresentada uma nova janela com informação relativa às unidades de energia gastas se as luminárias estivessem ligadas na intensidade máxima, a energia gasta utilizando o sistema e a poupança obtida (Figura 7).

Se carregarmos no botão *Messages* aparece uma nova janela onde podemos ver várias informações relativas ao envio de mensagens e uma tabela onde estão todas as mensagens enviadas (Figura 8). As informações mostradas são o número de mensagens enviadas, o número de nós que enviaram pelo menos uma mensagem, a média de mensagens enviadas por cada nó que enviou pelo menos uma mensagem e a média de mensagens enviadas por cada nó que enviou pelo menos uma mensagem por unidade de tempo. Se quisermos reiniciar a simulação carrega-se nos botões *Stop* e *Load*, caso a simulação ainda esteja a decorrer, ou apenas no botão *Load* se a simulação já estiver terminada. Para sairmos do simulador basta carregar no botão *Exit*.

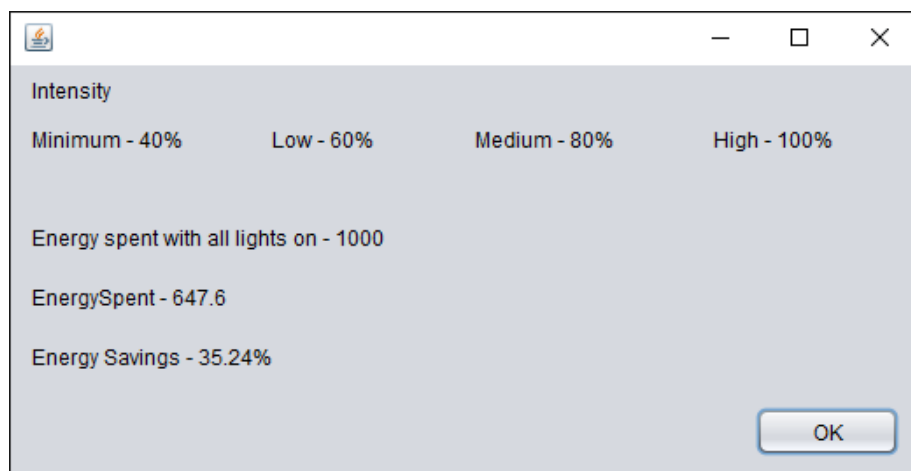


Figura 7 – Janela onde podemos ver a poupança de energia.

Number of messages sent: 2161
 Number of nodes that sent a message: 73
 Average number of messages sent per node that sent a message: 29.60274
 Average number of messages sent per node that sent a message over time: 2.960274

| Node | Message | Receivers |
|--------|---|-----------|
| [0, 3] | MessageLR - Origin: [0, 3] Sender: [0, 3] Lighting Radius: 2 Ori... | [0, 2] |
| [0, 3] | MessageLR - Origin: [0, 3] Sender: [0, 3] Lighting Radius: 2 Ori... | [0, 4] |
| [0, 3] | MessageLR - Origin: [0, 3] Sender: [0, 3] Lighting Radius: 2 Ori... | [1, 3] |
| [0, 3] | MessageLR - Origin: [0, 3] Sender: [0, 3] Lighting Radius: 2 Ori... | [1, 2] |
| [0, 3] | MessageLR - Origin: [0, 3] Sender: [0, 3] Lighting Radius: 2 Ori... | [1, 4] |
| [0, 4] | MessageLR - Origin: [0, 4] Sender: [0, 4] Lighting Radius: 2 Ori... | [0, 3] |
| [0, 4] | MessageLR - Origin: [0, 4] Sender: [0, 4] Lighting Radius: 2 Ori... | [0, 5] |
| [0, 4] | MessageLR - Origin: [0, 4] Sender: [0, 4] Lighting Radius: 2 Ori... | [1, 4] |
| [0, 4] | MessageLR - Origin: [0, 4] Sender: [0, 4] Lighting Radius: 2 Ori... | [1, 3] |
| [0, 4] | MessageLR - Origin: [0, 4] Sender: [0, 4] Lighting Radius: 2 Ori... | [1, 5] |
| [1, 4] | MessageLR - Origin: [0, 3] Sender: [1, 4] Lighting Radius: 1 Ori... | [2, 5] |
| [1, 4] | MessageLR - Origin: [0, 3] Sender: [1, 4] Lighting Radius: 1 Ori... | [1, 5] |
| [1, 4] | MessageLR - Origin: [0, 3] Sender: [1, 4] Lighting Radius: 1 Ori... | [2, 4] |

Figura 8 – Janela onde podemos ver informações do envio de mensagens. Também existe uma tabela com todas as mensagens enviadas.

Comandos do *Map Script*

O map script é o local onde o ambiente de teste é definido. Apesar do simulador utilizar uma disposição matricial dos nós, pretende-se que a criação de ambientes de teste, com os comandos disponíveis, tivesse o menor número de restrições possível.

É possível definir ambientes de teste com qualquer número de nós e qualquer formato. Para criar espaços próximos dos existentes na realidade podemos desenhar paredes, que são representadas no simulador como um segmento de reta. Permite adicionar e remover nós individualmente ou em conjunto e alterar os vários parâmetros do sistema e de cada nó, alguns dos quais serão explicados em detalhe mais à frente. É possível adicionar comentários utilizando o carácter '#' seguido de um espaço. Exceto onde indicado, todos os comandos recebem inteiros como argumentos. Em seguida são descritos os comandos que permitem definir o ambiente de teste para cada simulação:

- `matrix c r`

Comando que define o tamanho da matriz do ambiente de teste. Recebe como argumentos dois inteiros maiores que zero: o número de colunas (*c*) e o número de linhas (*r*). O *map script* começa sempre com este comando e só pode ser chamado uma vez;

- `rectangle_add x y l h`

Este comando define um retângulo e cria os nós que estão dentro da área do retângulo. Recebe quatro inteiros como argumentos: as coordenadas da origem do retângulo (*x* e *y*) e o

comprimento (l) e a altura (h). A origem do retângulo corresponde ao ponto do canto superior esquerdo;

- `rectangle_remove x y l h`

Este comando define um retângulo e remove os nós que estão dentro da área do retângulo. Recebe quatro inteiros como argumentos: as coordenadas da origem do retângulo (x e y) e o comprimento (l) e a altura (h). A origem do retângulo corresponde ao ponto do canto superior esquerdo.

- `circle_add x y r`

Comando que define um círculo e cria os nós que estão dentro da área do círculo. Recebe como argumentos as coordenadas da origem do círculo (x e y) e o seu raio (r);

- `circle_remove x y r`

Comando que define um círculo e remove os nós que estão dentro da área do círculo. Recebe como argumentos as coordenadas da origem do círculo (x e y) e o seu raio (r);

- `wall_add x1 y1 x2 y2`

Este comando permite criar uma parede. Uma parede é representada no simulador como uma linha entre dois pontos. É utilizada para representar obstáculos que possam existir num ambiente real e não afeta o resultado da simulação. Recebe como argumentos as coordenadas de dois pontos, o ponto um ($x1$ e $y1$) e o ponto dois ($x2$ e $y2$). As coordenadas podem ser números racionais;

- `node_add x y`

Comando que adiciona um nó numa determinada posição. Recebe as coordenadas do nó a remover (x e y);

- `node_remove x y`

Comando que remove um nó numa determinada posição. Recebe as coordenadas do nó a remover (x e y);

- `neighbour_remove x y nx ny`

Este comando permite remover um nó da lista de vizinhos de um determinado nó. Recebe como argumentos as coordenadas do nó (x e y) e as coordenadas do nó vizinho a remover (nx e ny). Este comando é especialmente útil quando existe uma parede entre dois nós e não se pretende que eles comuniquem;

- `lighting_radius i`
Este comando altera o raio de iluminação. Recebe como argumento o novo raio de iluminação (*i*);
- `anticipation_distance i`
Comando que altera a distância de antecipação. Recebe como argumento a nova distância de antecipação (*i*);
- `base_intensity x y i`
Este comando permite alterar a intensidade base de um nó. Recebe como argumentos as coordenadas do nó (*x* e *y*) e a intensidade (*i*). O valor da intensidade é um valor entre 0 e 3, onde o 0 corresponde à intensidade mínima, o 1 corresponde à intensidade baixa, o 2 à intensidade média e o 3 à intensidade alta.
- `intensity i p`
Este comando permite alterar a potência, em percentagem, de uma intensidade. Recebe como argumentos a intensidade (*i*) e a nova percentagem (*p*) que tem de ser um inteiro entre 0 e 100. O valor da intensidade é valor entre 0 e 3, onde o 0 corresponde à intensidade mínima, o 1 corresponde à intensidade baixa, o 2 à intensidade média e o 3 à intensidade alta.
- `special_direction x y dir newdir1 newdir2 ...`
Comando que define uma direção especial num determinado nó. Recebe como argumentos as coordenadas do nó (*x* e *y*), uma direção (*dir*) e uma lista de direções.
Cada direção é identificada por um inteiro entre 1 e 8. Na Tabela 3 podemos ver a correspondência entre cada direção e o seu identificador.
- `timeout i`
Comando que define o valor do temporizador inicial presente em cada nó. Recebe como argumento um inteiro que corresponde ao valor do temporizador (*i*).
- `node_failure x y`
Comando que permite simular a falha de um nó. Recebe como argumentos dois inteiros que correspondem às coordenadas do nó que vai ter a falha (*x* e *y*). Um nó quando falha não deteta movimento, não recebe nem envia mensagens e a lâmpada está desligada.

Comandos do *User Script*

O user script permite a criação de utilizadores e a definição do caminho de cada um. Cada utilizador tem um identificador, uma posição inicial e o caminho que tem de percorrer. Podem-se mover entre quaisquer dois pontos do ambiente de teste. Os utilizadores possuem coordenadas especiais

com o valor (-1, -1) e quando se movem para essas coordenadas, desaparecem do mapa durante um intervalo de tempo, não sendo detetado o seu movimento. Tal como no *map script*, é possível adicionar comentários utilizando o caracter '#' seguido de um espaço. Os comandos disponíveis permitem que os utilizadores se movam livremente pelo mapa, podendo até controlar a velocidade a que se movem. A seguir são apresentados os comandos do *user script*:

- `start i x y`

Comando que cria um utilizador com o identificador i nas coordenadas (x, y) . O identificador do utilizador é um inteiro;

- `move i x y seg`

Este comando move o utilizador com o identificador i para as coordenadas (x, y) ao longo de seg segundos. Se este comando for utilizado com as coordenadas (-1, -1), o utilizador desaparece e não se move durante seg segundos.








| Direção | Representação Gráfica | Identificador |
|------------------------------|---|---------------|
| Cima |  | 1 |
| Diagonal direita para cima |  | 2 |
| Direita |  | 3 |
| Diagonal direita para baixo |  | 4 |
| Baixo | | 5 |
| Diagonal esquerda para baixo |  | 6 |
| Esquerda |  | 7 |
| Diagonal esquerda para cima |  | 8 |

Tabela 3 – Correspondência entre as direções e os seus identificadores.

3.3.2 Abordagem ao Algoritmo

Os nós do sistema têm uma capacidade de processamento limitada, pelo que o algoritmo tem de necessitar do mínimo de recursos possível. Isto significa que cada nó irá ter o mínimo de informação possível. Cada nó tem as seguintes características:

- Tem um identificador único, definido pelas suas coordenadas;
- Cada nó apenas comunica com os seus vizinhos, que são os nós adjacentes;
- Os vizinhos de cada nó são conhecidos *a priori*;
- A comunicação entre nós é feita através de mensagens.

O número de direções que um nó consegue detetar são oito (cima, baixo, esquerda, direita e as diagonais), visto que cada nó apenas comunica com os seus vizinhos e a sua disposição é matricial.

As luminárias do sistema encontram-se numa de quatro intensidades: mínima, baixa, média e alta. A intensidade mínima é a menor das quatro e é utilizada quando os nós não detetam movimento, nem recebem mensagens. As intensidades baixa e média são utilizadas quando houve uma deteção e o nó está dentro do raio de iluminação. A intensidade alta é utilizada quando o nó detetou movimento ou está dentro da distância de antecipação.

O funcionamento do algoritmo pode ser dividido em duas partes: a primeira deteção de movimento de um utilizador e as deteções seguintes. Todos os nós quando inicializados começam com a intensidade mínima. Na primeira deteção de movimento, o nó que deteta o movimento muda a sua intensidade para alta e envia uma mensagem aos vizinhos a avisar a deteção. A mensagem é propagada até ao raio de iluminação. Os nós na orla do raio de iluminação ficam com intensidade baixa e os nós entre a orla e o nó que detetou o movimento ficam com intensidade média (Figura 9).

Considerando deteções em luminárias vizinhas, ocorridas num intervalo de tempo curto, será assumido que se trata de uma ou mais pessoas a deslocarem-se e o algoritmo irá estimar a direção de deslocação. Com base nessa informação, o nó que efetuou a deteção mais recente envia uma mensagem para o vizinho na direção prevista, a qual se irá propagar pela distância de antecipação e outra mensagem para os restantes vizinhos a avisar da deteção de movimento que se irá propagar pelo raio de iluminação. Os nós que estão na direção prevista ficam com intensidade alta. Cada nó que esteja na direção prevista também irá enviar uma mensagem aos seus vizinhos como se tivesse detetado movimento. Essa mensagem irá propagar-se pelo raio de iluminação. Caso o utilizador não siga a direção prevista, será calculada uma nova direção com base no movimento do utilizador.

Se não houver deteção de movimento, ao fim de alguns segundos os nós vão diminuir a sua intensidade passando de intensidade alta para média, de média para baixa e de baixa para mínima.

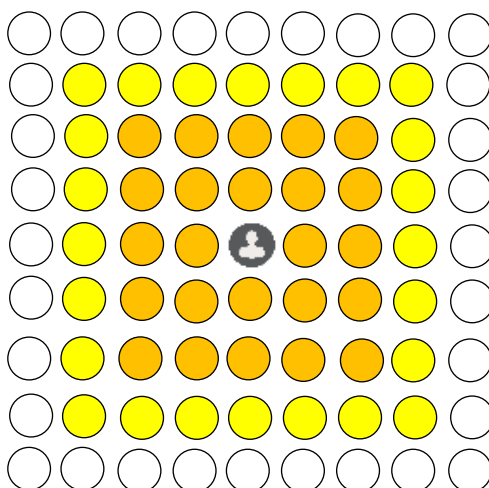


Figura 9 – Detecção de um utilizador, com raio de iluminação igual a 3. O nó onde está o utilizador tem intensidade alta, os nós a laranja têm intensidade média e os nós a amarelo têm intensidade baixa.

3.3.3 Aplicação de configuração

A aplicação de configuração permite gerir e monitorizar o sistema. No entanto, como o sistema foi desenvolvido com o auxílio do simulador, o objetivo da aplicação passou a ser a criação e edição de *map scripts*.

Funcionalidades

Na Figura 10 podemos ver o ecrã inicial da aplicação de configuração. Dos vários botões presentes, os que nos interessam neste momento são o *New Setup* e *Edit Setup*. O *New Setup* permite-nos criar um novo *map script* e o *Edit Setup* permite-nos escolher um *map script* criado anteriormente e editá-lo. Quando carregamos no botão *New Setup*, abre-se uma nova janela (Figura 11) onde podemos criar o *map script*. Se carregamos no botão *Edit Setup* aparece uma nova janela que permite seleccionar um ficheiro de texto. A edição do *map script* é feita com os restantes botões do ecrã inicial.

A aplicação de configuração tem as seguintes funcionalidades:

- Adicionar e remover nós;
- Definir os vizinhos de cada nó;
- Alterar o raio de iluminação;
- Alterar a distância de antecipação;
- Alterar a potência, em percentagem, de cada uma das intensidades;
- Adicionar e remover paredes;
- Definir a intensidade base de cada nó;
- Definir as direções especiais para cada nó;
- Alterar o *timeout*;
- Simular a falha de um nó.

Cada uma destas funcionalidades abre uma nova janela. Estas funcionalidades são comuns quer à criação, quer à edição de um *map script*. Quando criamos um *map script* temos primeiro de escolher um nome para o ficheiro e definir o tamanho da matriz, através dos campos X e Y e só quando carregamos no botão *Create* é que o ficheiro é criado e guardado. Quando editamos o *map script*, em cada uma das janelas ao carregarmos no botão *Ok*, o ficheiro é automaticamente alterado.

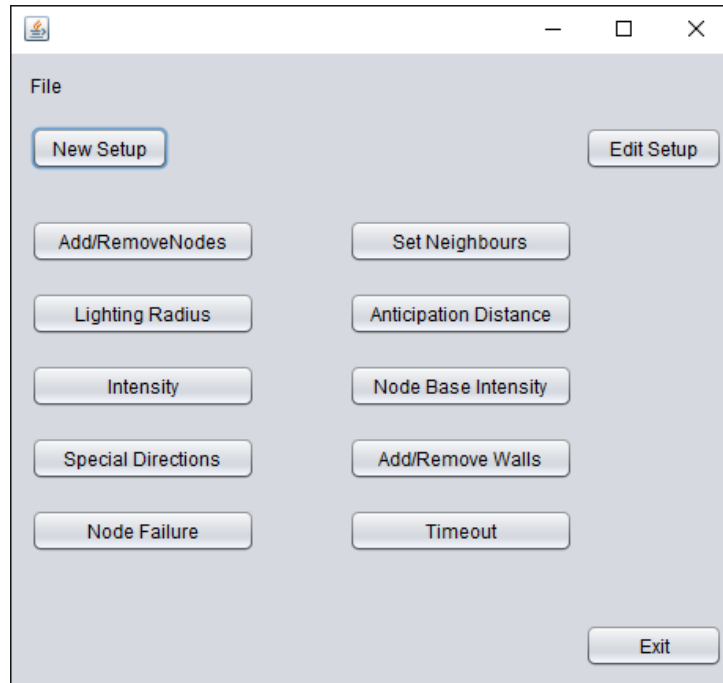


Figura 10 – Ecrã inicial da aplicação de configuração.

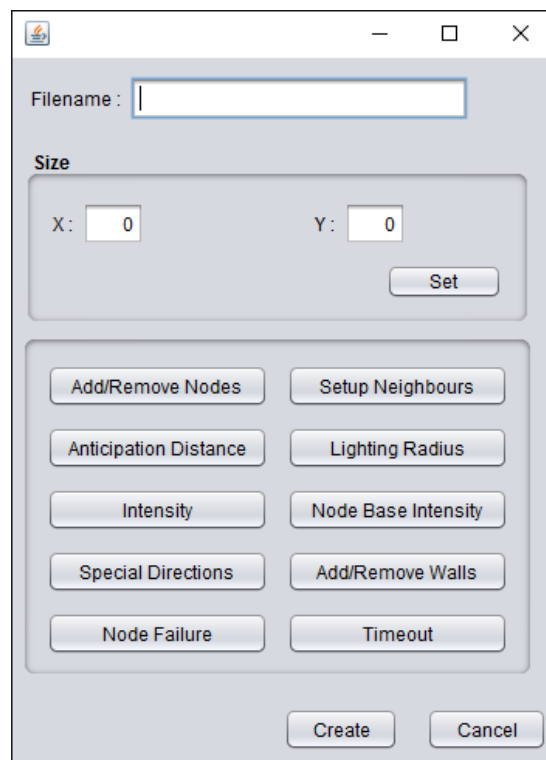


Figura 11 – Janela onde podemos criar um novo *map script*.

Em seguida vão ser descritas as várias funcionalidades em detalhe:

- Adicionar e remover nós

Esta funcionalidade permite adicionar ou remover nós conforme desejado. Ao carregarmos no botão *Add/Remove Nodes*, é aberta uma nova janela (Figura 12) onde estão todos os nós. À medida que vamos clicando nos círculos pretos, conseguimos definir o formato do ambiente de teste. Os círculos a preto são os nós que não vão ser removidos e a branco temos os nós que vão ser removidos.

- Definir os vizinhos de cada nó

Para se definir os vizinhos de um nó, tem de carregar no botão *Setup Neighbours*, que irá abrir uma janela similar à da Figura 13. Nessa janela podemos seleccionar um nó clicando sobre um círculo negro. Quando seleccionamos um nó aparece-nos uma lista com os vizinhos por baixo de *Neighbours* e uma lista com os vizinhos removidos por baixo de *Removed*. Para removermos um vizinho seleccionamos um dos vizinhos e carregamos no botão *Remove*. Para adicionarmos um vizinho seleccionamos um dos vizinhos removidos e carregamos no botão *Add*.

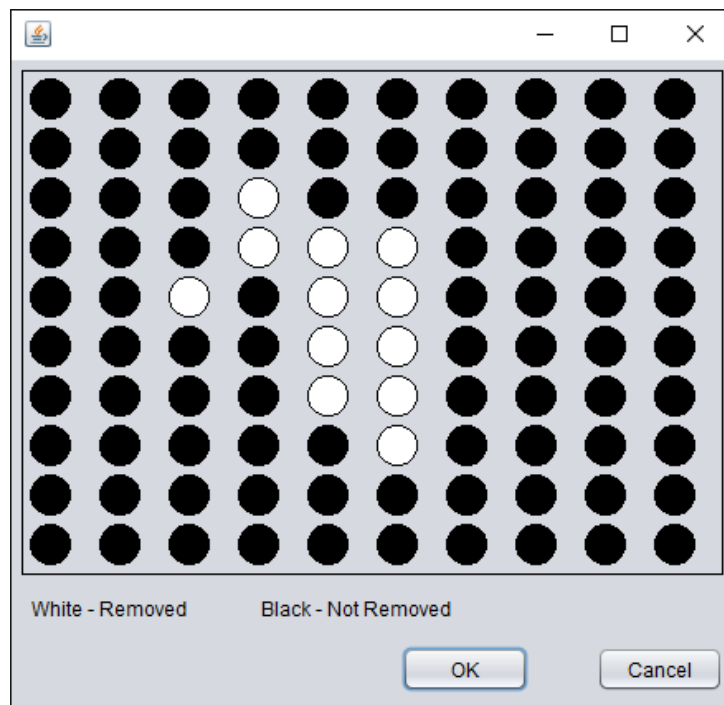


Figura 12 – Janela onde se removem os nós.

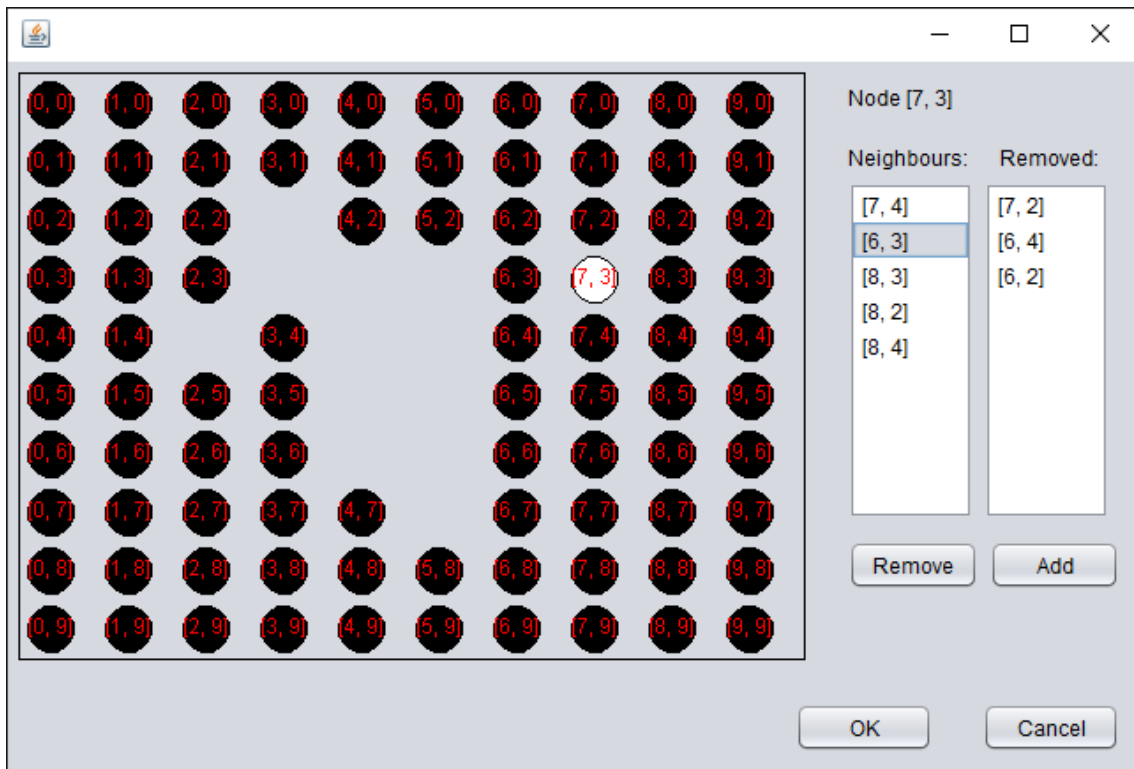


Figura 13 – Janela onde se define os vizinhos de cada nó.

- Alterar o raio de iluminação

Esta funcionalidade permite alterar o raio de iluminação. Ao carregarmos no botão *Lighting Radius* aparece uma janela com um campo onde é possível definir o raio de iluminação (Figura 14). O campo vem preenchido inicialmente com o valor atual do raio de iluminação;

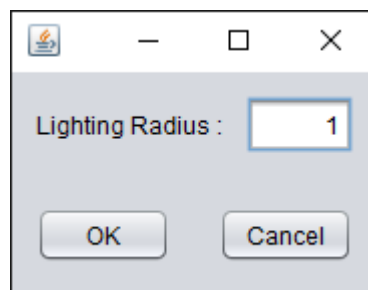


Figura 14 – Janela onde se define o raio de iluminação.

- Alterar a distância de antecipação

Para se alterar a distância de antecipação carrega-se no botão *Anticipation Distance* e aparece uma janela com um campo onde é possível definir a distância de antecipação (Figura 15). O campo vem preenchido inicialmente com o valor atual da distância de antecipação;

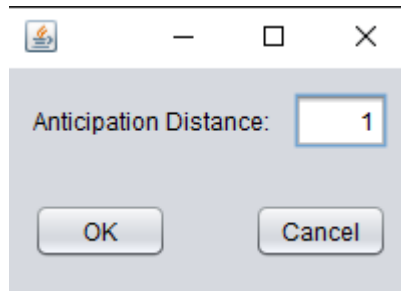


Figura 15 – Janela onde se define a distância de antecipação.

- Alterar a potência, em percentagem, de cada uma das intensidades
Com esta funcionalidade conseguimos definir a potência em percentagem para cada intensidade. Quando carregamos no botão *Intensity* aparece uma janela com quatro campos, cada um corresponde a uma das intensidades (Figura 16). Cada campo vem preenchido com o valor da potência de cada uma das intensidades.

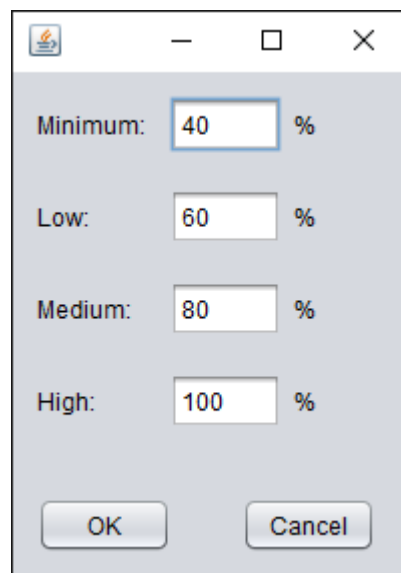


Figura 16 – Janela onde se define a intensidade em percentagem de cada uma das intensidades.

- Adicionar e remover paredes
Esta funcionalidade permite adicionar e remover paredes. Ao carregarmos no botão *Walls* aparece uma nova janela, similar à da Figura 17, onde podemos adicionar e remover paredes. Para criarmos uma parede, clicamos em qualquer ponto da área do mapa e mantendo o botão do rato premido, arrastamos o rato para onde queremos e largamos o botão. É criada uma linha entre esses dois pontos e que vai ter um identificador único. Para removermos uma parede, basta seleccionarmos o identificador da parede na combo box e carregar no botão *Remove*;

- Definir a intensidade base de cada nó

Para se alterar a intensidade base de cada nó, carrega-se no botão *Node Base Intensity* e aparece uma janela, como na Figura 18, onde cada círculo representa um nó. Seleccionamos um nó clicando nele, que fica de cor branca. Ao lado temos as coordenadas do nó seleccionado e uma combo box na qual podemos escolher a nova intensidade base e confirmar a opção carregando no botão *Set*;

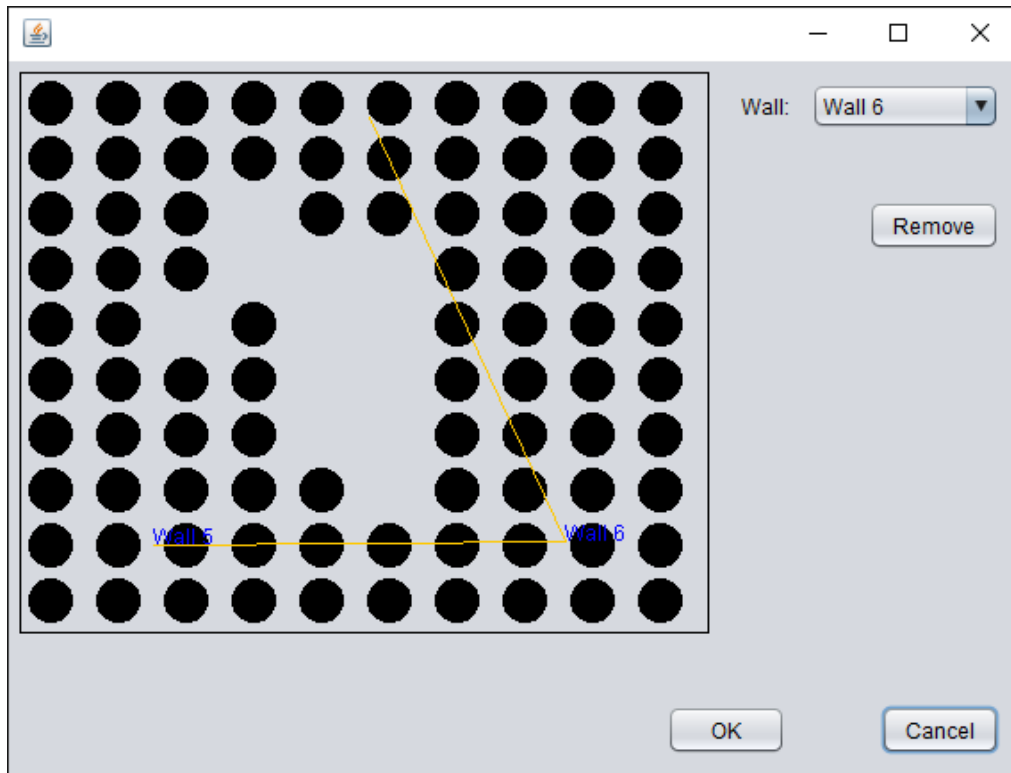


Figura 17 – Janela onde se adiciona e remove paredes.

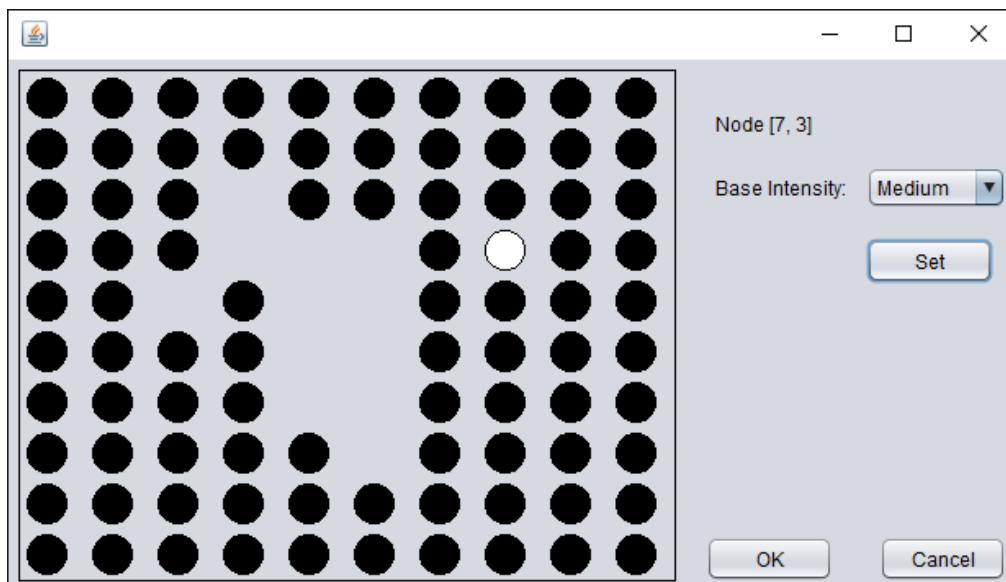


Figura 18 – Janela onde se define a intensidade base de cada nó.

- Definir as direções especiais para cada nó
 Para definir as direções especiais de um nó, carrega-se no botão *Special Directions* e aparece uma nova janela, como na Figura 19. Nessa janela podemos selecionar um nó clicando sobre um círculo negro. Quando selecionamos um nó aparece-nos uma lista com as direções especiais desse nó por baixo de *Special Directions* e uma lista com todas as direções por baixo de *Directions*. Para adicionarmos uma direção especial selecionamos uma direção da lista de direções e carregamos no botão *Add*. Se quisermos remover uma direção especial, selecionamos a direção especial que vai ser removida e carregamos no botão *Remove*.
- Alterar o *timeout*
 Esta funcionalidade permite alterar o *timeout* dos nós. Ao carregarmos no botão *Timeout* aparece uma janela com um campo onde podemos alterar o valor do *timeout* (Figura 20). O campo vem preenchido com o valor atual do *timeout*.
- Simular a falha de um nó
 Com esta funcionalidade pode-se simular uma falha de funcionamento de um nó. Para isso carrega-se no botão *Node Failure* e de seguida aparece uma janela, similar à da Figura 21. Os nós que irão falhar são selecionados clicando sobre os círculos negros.

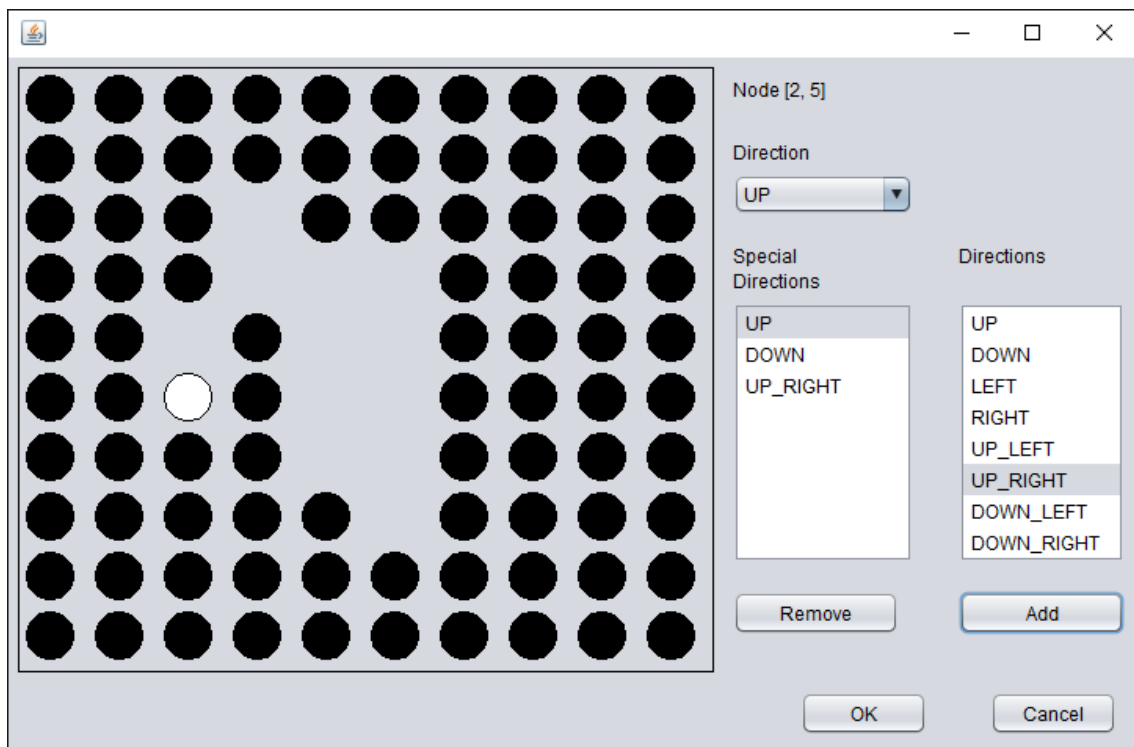


Figura 19 – Janela onde se define as direções especiais de qualquer nó.

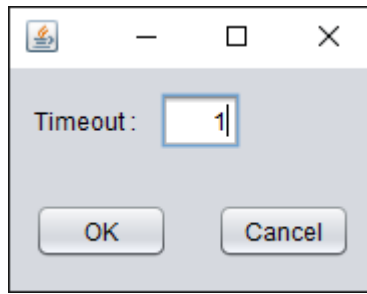


Figura 20 – Janela onde se define o valor do *timeout*.

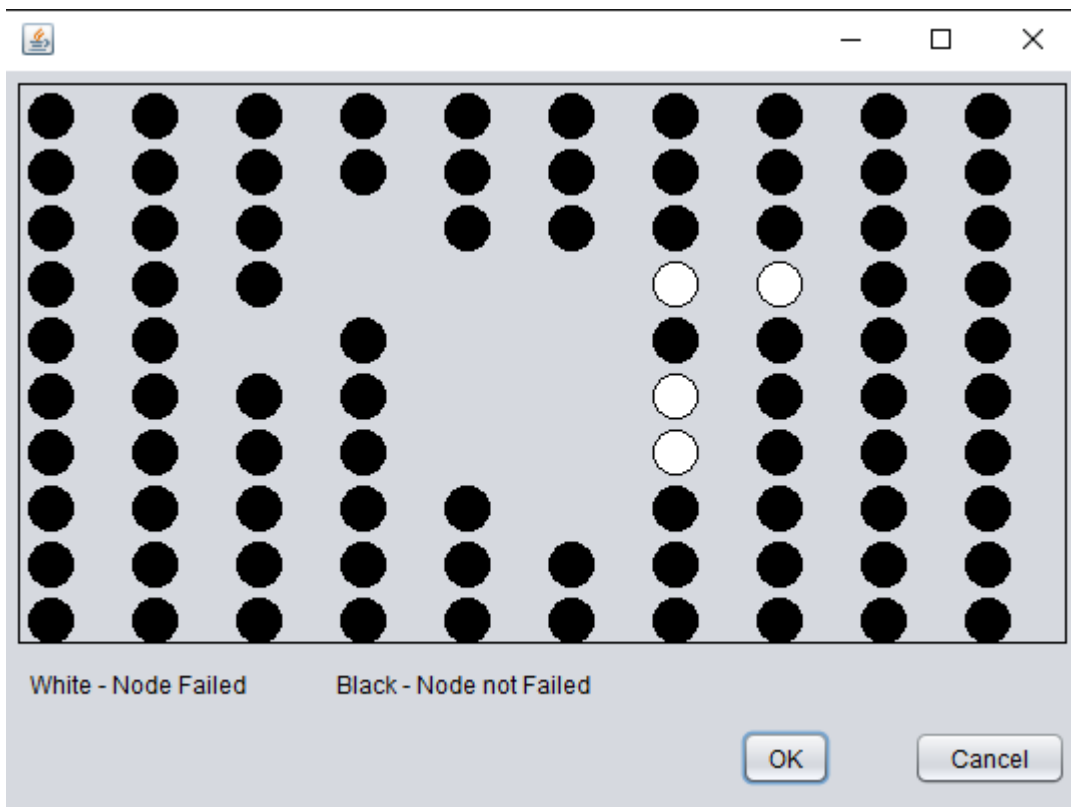


Figura 21 – Janela onde se simula a falha de um ou mais nós.

4. Algoritmo

O componente principal do sistema é o algoritmo. Neste capítulo será descrito as mensagens utilizadas, o método de envio de mensagens utilizado, a máquina de estados e uma descrição detalhada do funcionamento do algoritmo. Também será apresentar melhoramentos que foram feitos e limitações da abordagem tomada.

4.1 Tipos de Mensagens

A comunicação entre os nós é feita através de mensagens. Existem dois tipos de mensagens para os vários eventos: as mensagens RI (Raio de Iluminação) e as mensagens DA (Direção Antecipada).

A mensagem RI é utilizada para avisar os nós que estão dentro do raio de iluminação. Cada mensagem RI tem os seguintes campos:

- Origem: contém o identificador do nó que detetou o movimento, ou seja, as suas coordenadas;
- *Timestamp* da origem: corresponde ao *timestamp* em que a origem detetou o movimento;
- Remetente: contém o identificador do nó que enviou a mensagem;
- *Timestamp*: o *timestamp* de quando a mensagem foi enviada. É o identificador da mensagem;
- Raio de iluminação.

A mensagem DA é utilizada para avisar os nós que estão na distância de antecipação. Cada mensagem DA tem os seguintes campos:

- Origem: contém o identificador do nó que detetou o movimento, ou seja, as suas coordenadas;
- *Timestamp* da origem: corresponde ao *timestamp* em que a origem detetou o movimento;
- Remetente: contém o identificador do nó que envio a mensagem;
- *Timestamp*: o *timestamp* de quando a mensagem foi enviada. É o identificador da mensagem;
- Distância de antecipação.

4.2 Envio de Mensagens

O número de mensagens enviadas é algo que temos de ter em conta ao implementarmos o sistema. Este sistema destina-se a espaços fechados de grande dimensão, que têm períodos de maior movimento, onde o número de deteções e conseqüentemente de mensagens geradas será muito superior ao normal. Se os nós de cada vez que enviam uma mensagem a enviassem para todos os vizinhos, seria necessário uma capacidade de processamento maior para processar todas as

mensagens, encarecendo cada luminária. Para que isso não aconteça, foi necessário encontrar um método de envio de mensagens que reduza o número de mensagens enviadas, sem comprometer o funcionamento do sistema.

Na primeira detecção de movimento, o nó que detetou o movimento envia uma mensagem RI para todos os seus vizinhos (Figura 22). Os nós que recebem essa mensagem e a propagam não necessitam de a enviar para todos os seus vizinhos, caso contrário haveria nós que receberiam a mesma mensagem mais do que uma vez.

A solução encontrada é o envio da mensagem para o nó que está na direção em que a mensagem foi recebida e para os nós das direções adjacentes (Figura 23). Com este método reduz-se bastante o número total de mensagens enviadas e o número de mensagens repetidas que cada nó recebe.

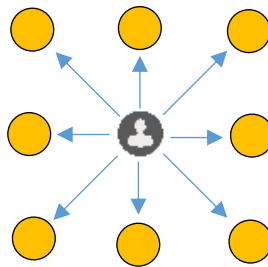


Figura 22 – Envio de uma mensagem RI a partir do nó que detetou o movimento.

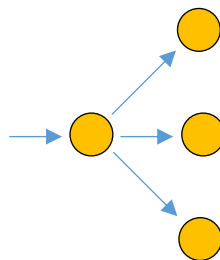


Figura 23 – Nó que recebeu uma mensagem RI. Envia uma nova mensagem RI na direção recebida e nas direções adjacentes.

A partir da segunda deteção de movimento o nó que detetou o movimento envia uma mensagem DA na direção determinada e aos restantes vizinhos envia uma mensagem RI (Figura 24). Quando um nó recebe uma mensagem DA, se a distância de antecipação ainda não foi percorrida, é enviada uma mensagem DA na direção prevista e para os restantes vizinhos, exceto o remetente é enviada uma mensagem RI (Figura 25). Se a distância de antecipação já foi percorrida, o nó que recebeu a mensagem DA envia para todos os vizinhos, exceto o remetente, uma mensagem RI (Figura 26).

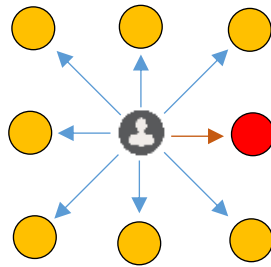


Figura 24 – Envio de uma mensagem RI (setas azuis) e de uma mensagem DA (seta castanha) a partir do nó que detetou o movimento.

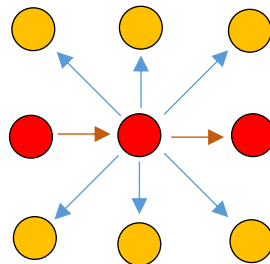


Figura 25 – Envio de uma mensagem RI a partir do nó que recebeu uma mensagem DA.

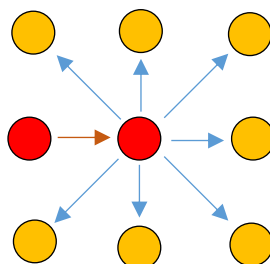


Figura 26 – Envio de uma mensagem RI a partir do nó que recebeu uma mensagem DA.

4.3 Estados

Cada nó, internamente, funciona como uma máquina de estados. Para cada estado e para cada entrada existe uma saída bem definida. Existem no total cinco estados: Mínimo, Baixo, Médio, Antecipado e Detetado. Cada nó começa no estado Mínimo. A saída de cada estado corresponde a uma das quatro intensidades: mínima, baixa, média e alta. Há cinco entradas que alteram o estado de um nó:

- Detecção de movimento (D);
- Mensagem DA (DA);
- Mensagem RI com o raio de iluminação superior a 1 ($RI > 1$);
- Mensagem RI com o raio de iluminação igual a 1 ($RI = 1$);
- O temporizador ter chegado a zero (T).

Na Tabela 4 está representada a tabela de transição de estados e na Tabela 5 podemos observar a correspondência entre o estado e a saída. Na Figura 27 temos a máquina de estados.

| Estado Atual | Entrada | Estado Seguinte | Saída |
|--------------|---------|-----------------|--------|
| Mínimo | T | Mínimo | MÍNIMA |
| Mínimo | RI=1 | Baixo | BAIXA |
| Mínimo | RI>1 | Médio | MÉDIA |
| Mínimo | AD | Antecipado | ALTA |
| Mínimo | D | Detetado | ALTA |
| Baixo | T | Mínimo | MÍNIMA |
| Baixo | RI=1 | Baixo | BAIXA |
| Baixo | RI>1 | Médio | MÉDIA |
| Baixo | AD | Antecipado | ALTA |
| Baixo | D | Detetado | ALTA |
| Médio | T | Baixo | BAIXA |
| Médio | RI=1 | Médio | MÉDIA |
| Médio | RI>1 | Médio | MÉDIA |
| Médio | AD | Antecipado | ALTA |
| Médio | D | Detetado | ALTA |
| Antecipado | T | Médio | MÉDIA |
| Antecipado | RI=1 | Antecipado | ALTA |
| Antecipado | RI>1 | Antecipado | ALTA |
| Antecipado | AD | Antecipado | ALTA |
| Antecipado | D | Detetado | ALTA |
| Detetado | T | Médio | MÉDIA |
| Detetado | RI=1 | Detetado | ALTA |
| Detetado | RI>1 | Detetado | ALTA |
| Detetado | AD | Antecipado | ALTA |
| Detetado | D | Detetado | ALTA |

Tabela 4 – Tabela de transição de estados.

| Estado | Saída |
|------------|--------|
| Detetado | ALTA |
| Antecipado | ALTA |
| Médio | MÉDIA |
| Baixo | BAIXA |
| Mínimo | MÍNIMA |

Tabela 5 – Correspondência entre estado e saída

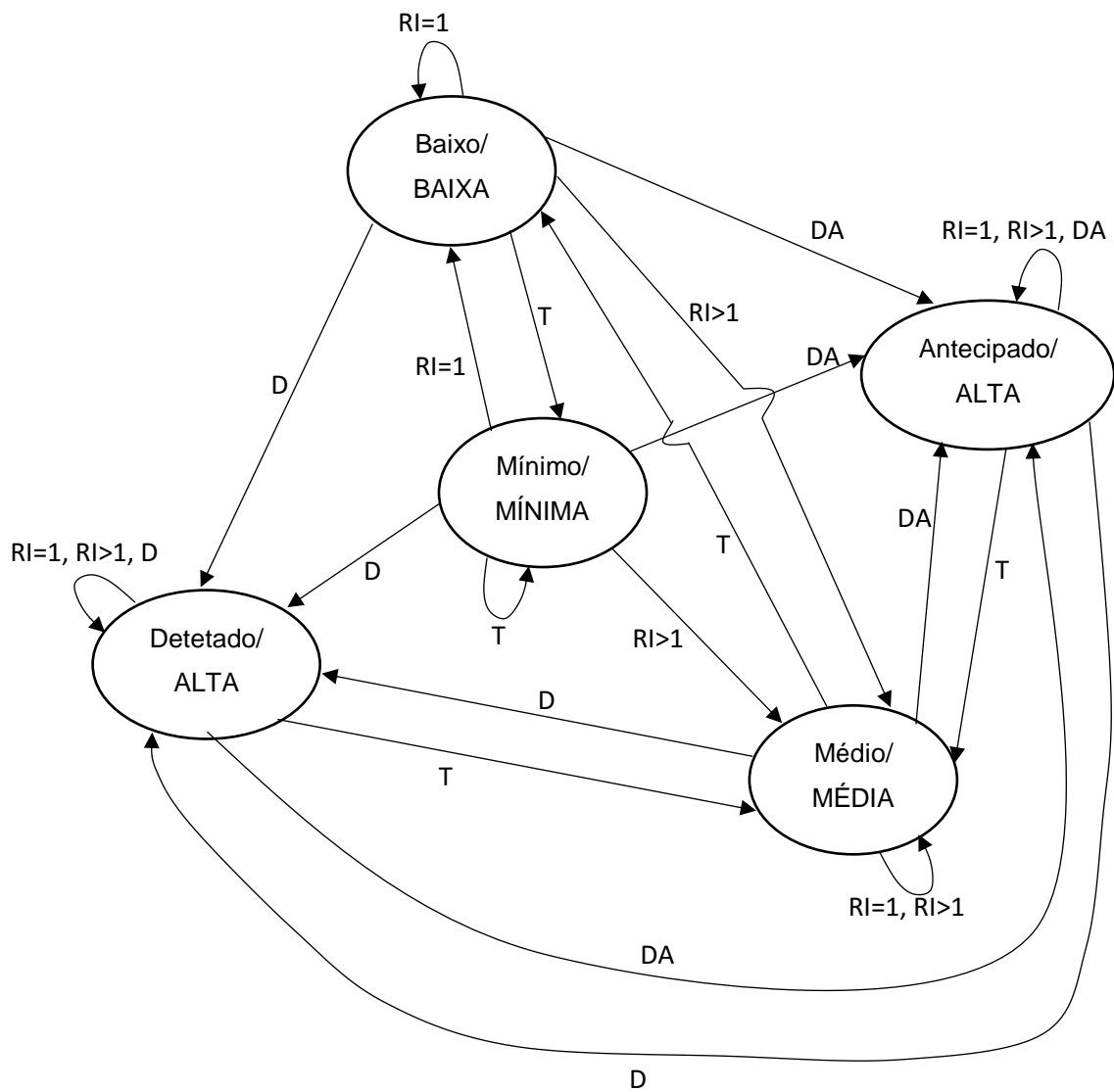


Figura 27 – Máquina de estados.

4.4 Descrição detalhada

Cada nó tem apenas a informação necessária ao funcionamento do algoritmo. Quando um nó é criado, possui a seguinte informação:

- As coordenadas, que servem como identificador;
- O estado inicial, que corresponde ao estado Mínimo;
- A intensidade, inicializada com a intensidade mínima;
- O *timeout*, que é o intervalo de tempo que o nó fica num estado antes de passar para o estado seguinte;
- O raio de iluminação;
- A distância de antecipação;
- A lista dos vizinhos.

Para além desta informação, cada nó também tem uma lista de mensagens, onde são colocadas as mensagens que estão por processar, a lista de deteções, que é uma estrutura de dados onde se guarda o *timestamp* da última deteção de cada vizinho, e um temporizador. O temporizador é iniciado com o valor do *timeout* e é reiniciado quando chega a zero ou o nó muda de estado.

Nas Figuras 28 e 29 temos dois fluxogramas que mostram o funcionamento do algoritmo. O fluxograma da Figura 28 representa o ciclo principal do algoritmo, de espera e processamento de mensagens. O fluxograma da Figura 29 mostra o que acontece quando existe uma deteção de movimento, que interrompe o ciclo principal.

Para percebermos melhor o funcionamento do algoritmo vamos supor que temos um utilizador a deslocar-se numa área com o tamanho de 10 por 10, como por exemplo na Figura 30. O raio de iluminação e distância de antecipação são iguais a 2 e o *timeout* é igual a 3.

Quando o utilizador se desloca (Figura 30), o nó que deteta o movimento altera o seu estado para Detetado e coloca o temporizador com o valor 3. Como não é possível determinar a direção do utilizador é enviada uma mensagem RI para todos os vizinhos. Nessa mensagem os campos da origem e do remetente são as coordenadas do nó, o *timestamp* e o *timestamp* da origem são iguais e o raio de iluminação é igual a 2.

Os nós vizinhos, quando fazem o tratamento da mensagem RI, alteram o seu estado para Médio, pois o raio de iluminação na mensagem é maior que 1. Como foi um vizinho que detetou o movimento, os nós guardam o *timestamp* da mensagem associado às coordenadas do nó que a enviou, na lista de deteções. Cada *timestamp* possui uma validade, após a qual é eliminado da lista de deteções. Isto significa que cada deteção tem uma validade e impede que os nós guardem informação sobre deteções que possa estar desatualizada, que faria o sistema comportar-se incorretamente. Os nós enviam uma mensagem RI, utilizando o método de envio de mensagens descrito acima, onde os campos origem e *timestamp* da origem são iguais aos campos da mensagem anterior, o campo remetente é as coordenadas do nó que envia a mensagem, o *timestamp* é a marca temporal da altura em que a mensagem é enviada e o raio de iluminação é decrementado, ficando com o valor 1.

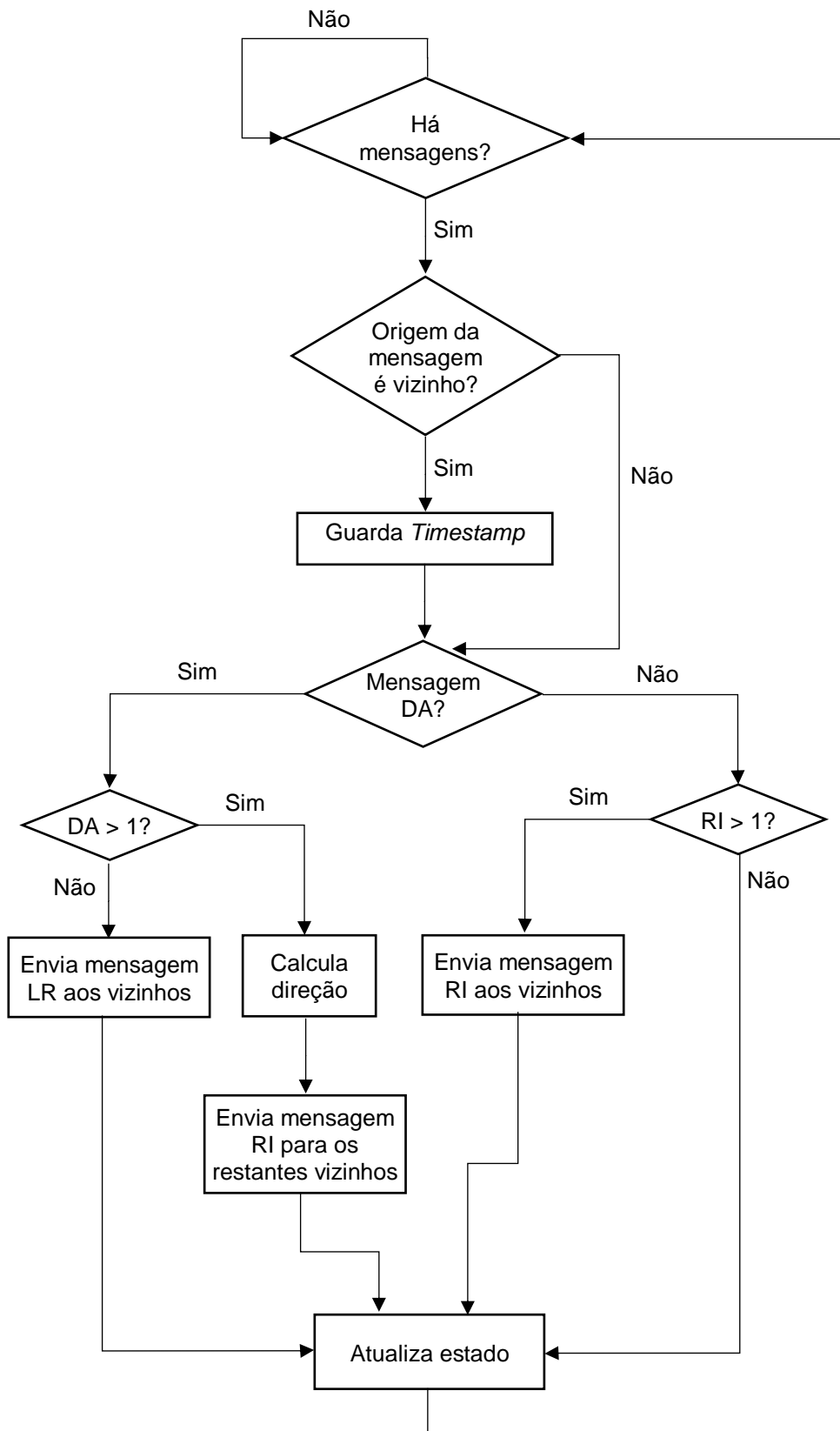


Figura 28 - Fluxograma do ciclo principal algoritmo.

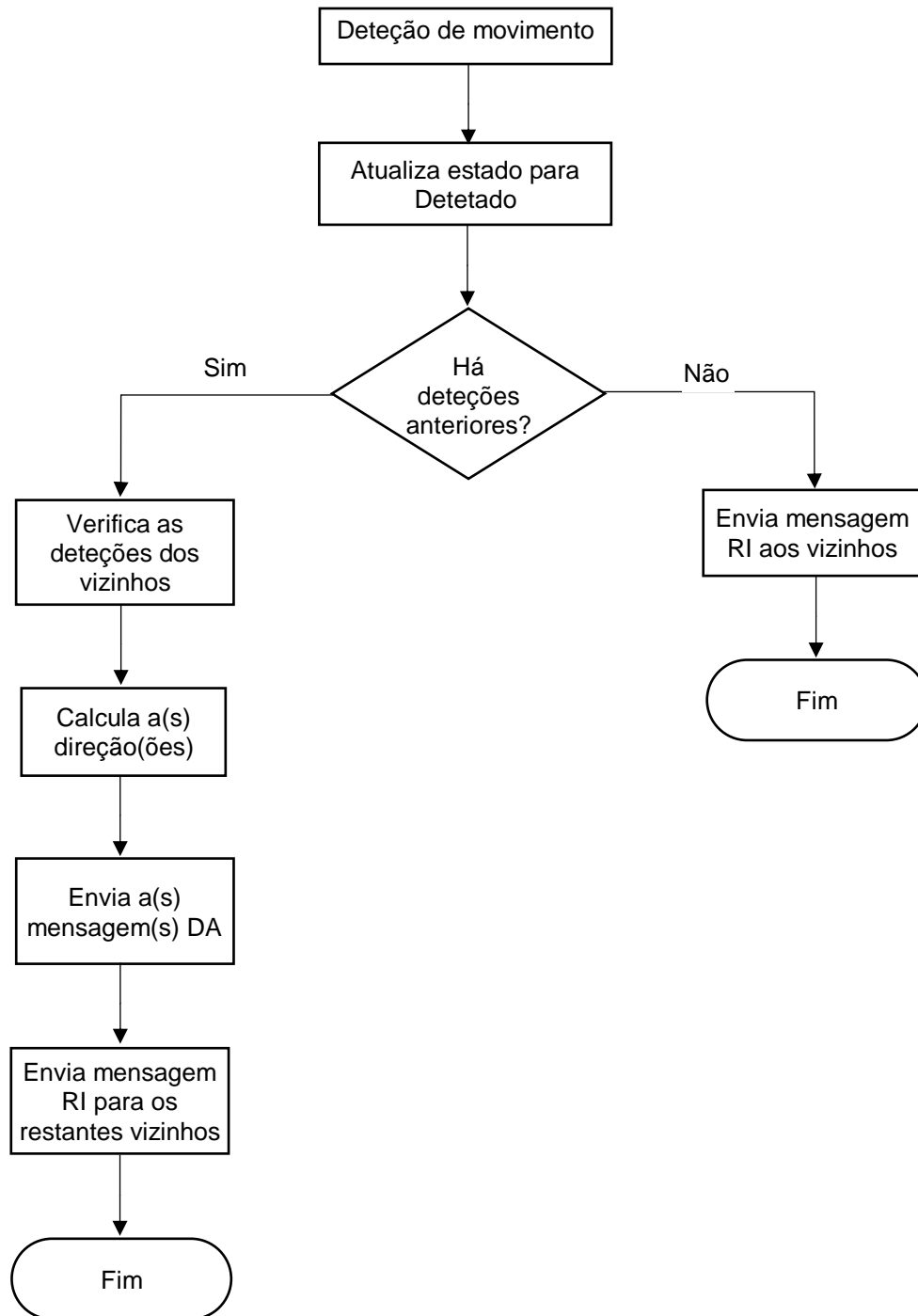


Figura 29 - Fluxograma quando existe uma deteção de movimento.

Os outros nós que receberam esta segunda mensagem RI alteram o seu estado para Mínimo, pois o raio de iluminação é igual 1. Como já se chegou ao limite do raio de iluminação, não é necessário enviar mais nenhuma mensagem.

Na segunda detecção de movimento (Figura 31), o nó que detetou o movimento já consegue calcular a direção do utilizador. Após calcular a direção, são enviadas uma mensagem RI e uma mensagem DA. A mensagem DA é enviada para o nó vizinho que está na direção calculada e a origem e o remetente da mensagem são as coordenadas do nó que detetou o movimento, o *timestamp* e o *timestamp* da origem são iguais e a distância de antecipação tem o valor 2. A mensagem RI é preenchida de maneira similar à mensagem DA e é enviada para os restantes vizinhos. A mensagem RI é depois propagada como já foi descrito acima.

O nó que recebe a mensagem DA altera o seu estado para Antecipado e muda para intensidade alta. Como a distância de antecipação é maior que 1, este nó também irá enviar uma mensagem DA e uma mensagem RI. Na mensagem DA os campos da origem e do *timestamp* da origem são iguais ao da mensagem recebida, o campo do remetente é as coordenadas do nó, o campo *timestamp* é preenchido com a marca temporal e a distância de antecipação é decrementada, ficando com o valor 1. A mensagem DA mantém a direção, sendo enviada para o nó correspondente. A mensagem RI é preenchida de forma similar à mensagem DA, com a diferença de que o campo do raio de iluminação vai ter o valor 2. É depois enviada para os restantes vizinhos, exceto o remetente da mensagem DA.

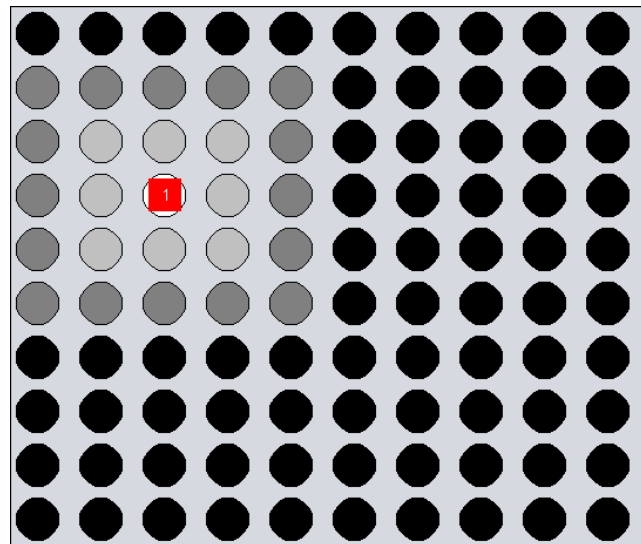


Figura 30 – Primeira detecção de movimento de um utilizador.

O nó que recebeu a mensagem DA altera o seu estado para Antecipado e muda para intensidade alta. Como o valor da distância de antecipação é igual a 1, este nó apenas envia uma mensagem RI para todos os vizinhos exceto o nó que enviou a mensagem DA. A mensagem RI tem no campo da origem a origem da mensagem DA, o *timestamp* de origem é o mesmo que na mensagem DA, o remetente é as coordenadas do nó, o campo *timestamp* é preenchido com a marca temporal e o raio de iluminação vai ter o valor 2. A partir da segunda detecção este processo repete-se até o sistema deixar de detetar o utilizador.

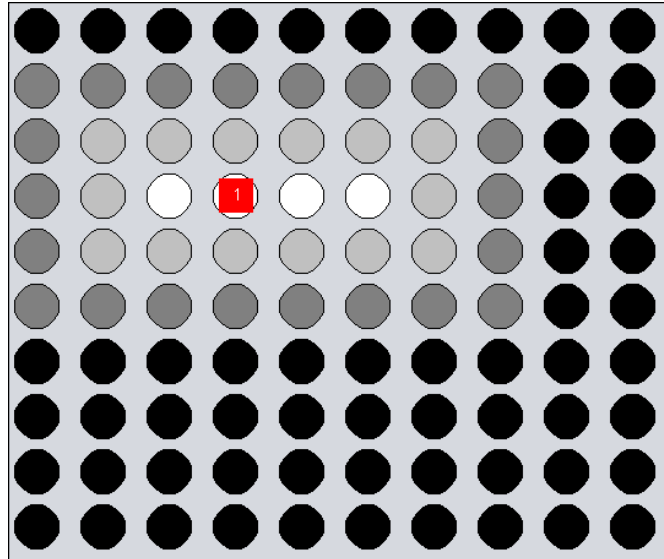


Figura 31 – Segunda deteção de movimento de um utilizador.

Quando existe mais do que um utilizador, o sistema comporta-se como descrito acima, enquanto os utilizadores não se cruzam. Na Figura 32 temos um exemplo com dois utilizadores.

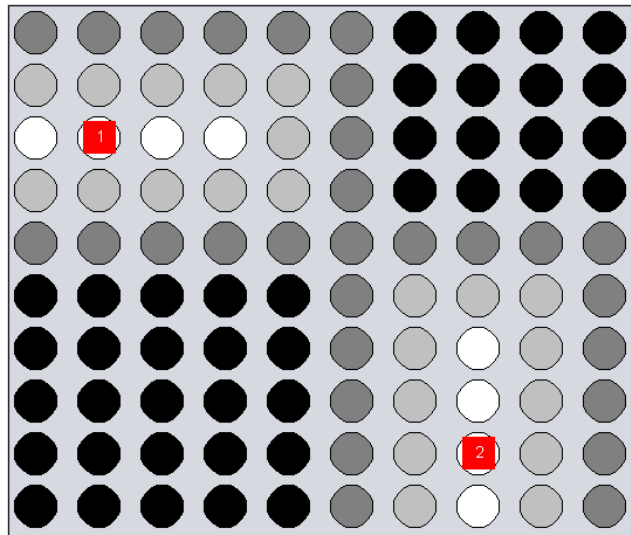


Figura 32 – Dois utilizadores a movimentarem-se.

Quando os utilizadores se cruzam, são iluminadas as direções que os utilizadores percorrem. Tal deve-se à lista de deteções. Quando um nó deteta movimento, verifica na lista de deteções quais os vizinhos que detetaram movimento recentemente e para esses vizinhos é enviada uma mensagem DA, sendo assim possível iluminar todas as direções dos utilizadores. Na Figura 33 temos um exemplo, onde dois utilizadores se cruzam e as direções em que seguiam foram corretamente iluminadas.

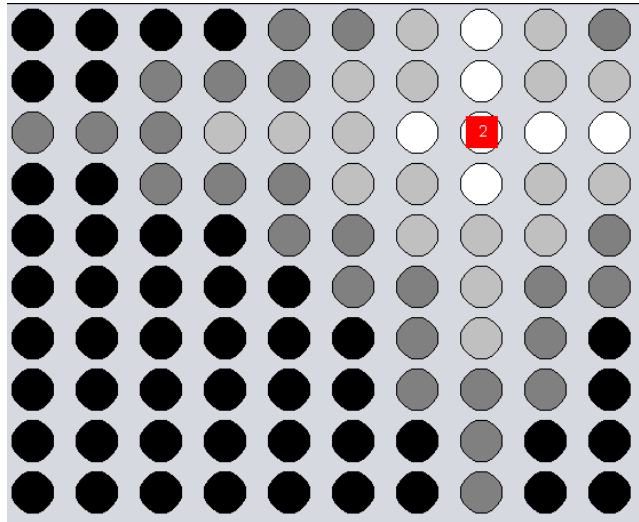


Figura 33 – Dois utilizadores cruzaram-se.

Os nós sempre que processam mensagens têm uma política de se manterem sempre na intensidade mais alta possível. Podemos considerar a seguinte ordem para a intensidade: mínima > baixa > média > alta. Isto significa que se um nó receber uma mensagem que o ponha numa intensidade inferior, a sua intensidade não se irá alterar. Se receber uma mensagem que o mantenha na mesma intensidade, o nó mantém apenas reinicia o temporizador. Caso receba uma mensagem que o coloque numa intensidade superior, o nó altera a sua intensidade.

4.5 Limitações

O algoritmo apresentado tem algumas limitações, em particular quando vários utilizadores estão próximos uns dos outros. Neste caso existem várias deteções por diferentes nós num curto de espaço de tempo, que faz com que os nós calculem direcções que nenhum dos utilizadores está a seguir. Na Figura 34 temos um exemplo, onde dois utilizadores que se estão a mover da esquerda para a direita e por haver várias deteções num curto espaço de tempo, as diagonais estão no estado Antecipado erradamente.

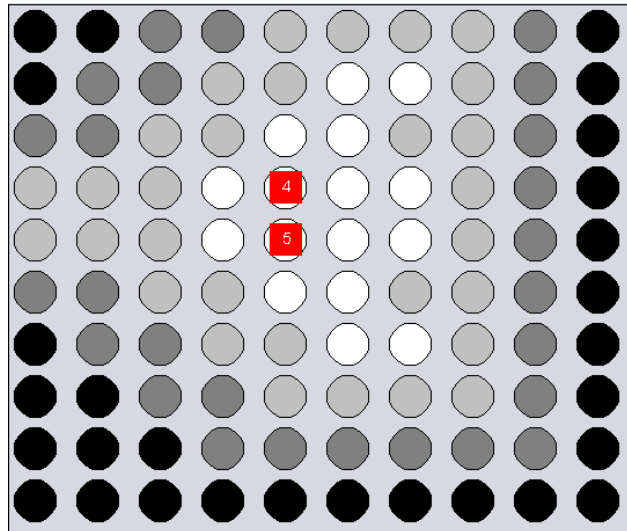


Figura 34 – Cálculo da direção errada.

4.6 Melhoramentos

Ao algoritmo base que foi descrito acima, foram feitos alguns melhoramentos que se considerou fazerem sentido acrescentar, tendo em conta o tipo de espaço a que este sistema se destina.

O primeiro melhoramento é a definição de uma intensidade base. Cada nó passou a ter uma intensidade base que é a intensidade mais baixa que esse nó pode tomar. O nó pode aumentar a sua intensidade, mas nunca irá para nenhuma intensidade abaixo da intensidade base. A intensidade base está pré-definida com a intensidade mínima, mas pode ser alterada para qualquer outra intensidade. O objetivo por detrás desta funcionalidade é permitir que hajam áreas, que por diversas razões, têm de estar sempre iluminadas, como por exemplo as saídas de emergência.

O segundo melhoramento refere-se às direções especiais. As direções especiais permitem que um nó quando deteta movimento numa determinada direção possa enviar a mensagem DA para quaisquer outras direções. Isto é feito através de uma estrutura de dados onde para cada direção está associada uma lista com as direções especiais. Esta funcionalidade serve para criar pequenos caminhos, quando se verifica que os utilizadores geralmente seguem uma direção diferente daquela prevista pelo sistema. Na Figura 35 temos um exemplo onde um utilizador que se desloca para a direita foi detetado por um nó tinha duas direções especiais.

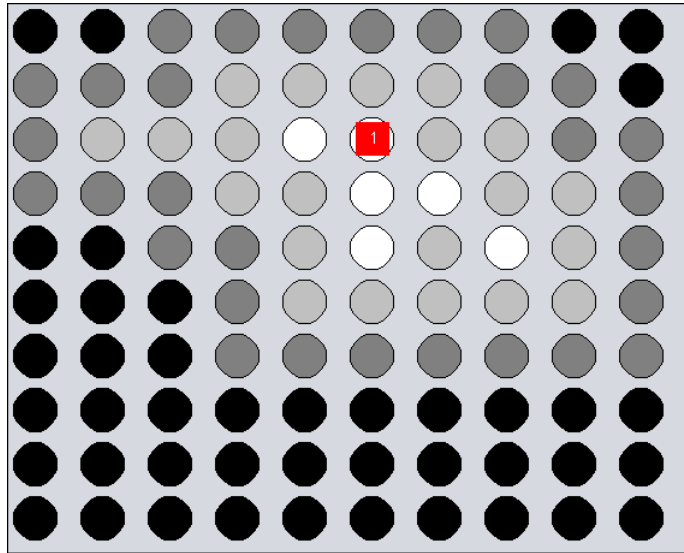


Figura 35 – Exemplo das direções especiais.

5. Avaliação

Neste capítulo irá ser feita uma avaliação do algoritmo desenvolvido. Serão feitos vários testes com o objetivo de verificar se o algoritmo cumpre os objetivos propostos e se comporta como esperado.

5.1 Método de Avaliação

O algoritmo será avaliado através de vários testes. Em cada teste é avaliado o comportamento do algoritmo utilizando um conjunto de situações que procuram ser semelhantes à utilização que o sistema teria no mundo real. O algoritmo será avaliado utilizando dois critérios:

- Poupança de energia;
- Número médio de mensagens enviadas por nó por unidade de tempo.

O principal objetivo deste sistema é permitir poupar energia, daí a poupança de energia ser um dos critérios de avaliação. Irá considerar-se que o algoritmo cumpre o objetivo se nos testes efetuados as poupanças de energia forem pelo menos de 40%. O cálculo da poupança de energia é feita através duma média ponderada de todas as intensidades. Na Equação 2 podemos ver a fórmula utilizada para o cálculo. O p é a percentagem da poupança de energia, min é o valor da intensidade mínima em percentagem, t_{min} são as unidades de tempo que todos os nós estiveram na intensidade mínima, b é o valor da intensidade baixa em percentagem, t_b são as unidades de tempo que todos os nós estiveram na intensidade baixa, med é o valor da intensidade média em percentagem, t_{med} são as unidades de tempo que todos os nós estiveram na intensidade média, a é o valor da intensidade alta em percentagem, t_a são as unidades de tempo que todos os nós estiveram na intensidade alta e o t são as unidades de tempo decorridas.

$$p = \frac{min \times t_{min} + b \times t_b + med \times t_{med} + a \times t_a}{t}$$

Equação 2

Com o critério do número médio de mensagens enviadas por nó por unidade de tempo pretende-se avaliar a carga de processamento a que cada luminária está sujeita. Este critério tem um fator determinante na escolha dos componentes de *hardware* para cada luminária inteligente. Se o número de mensagens enviadas for elevado cada luminária irá necessitar mais capacidade de processamento, logo terá *hardware* mais sofisticado e com um custo superior. Para calcular o valor deste critério só se tem em conta os nós que enviaram pelo menos uma mensagem, caso contrário os valores obtidos não iriam corresponder à realidade. O cálculo é feito utilizando uma média aritmética. Na equação 3 podemos ver a forma como é calculada a média de mensagens, onde o m é a média de mensagens

por nó, por unidade de tempo, me é o número de mensagens enviadas, n é o número de nós que enviaram pelo menos uma mensagem e t são as unidades de tempo decorridas.

$$m = \frac{me}{n \times t}$$

Equação 3

Cada teste será composto por um mapa e por vários utilizadores que se irão deslocar no mapa. O mapa tem de ser simples, para permitir o máximo de flexibilidade nos testes executados e ser minimamente realista, importando algumas das características dos espaços-alvo do sistema.

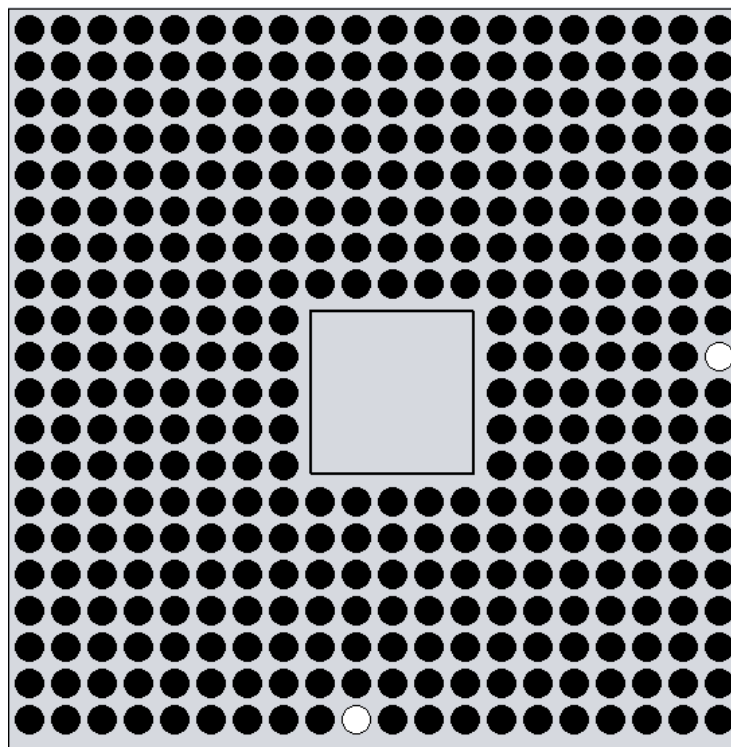


Figura 36 – Mapa utilizado nos testes.

O mapa que irá ser utilizado está representado na Figura 36 e na Figura 37 temos o seu *map script*. A principal característica dos espaços-alvo é o seu tamanho e conseqüentemente o elevado número de luminárias. Por isso o mapa tem uma área de 20 x 20 luminárias. Se considerarmos que cada luminária ilumina um raio de 2 metros à sua volta, verificamos que o mapa tem uma área de 40 por 40 metros quadrados, que é uma área considerável. No centro do mapa existe uma área sem luminárias e com paredes à sua volta, que representa a entrada/saída. Os círculos brancos são duas luminárias que têm a sua intensidade base com a intensidade alta e representam saídas de emergência. Para cada nó a intensidade mínima será 25% da potência da lâmpada, a intensidade baixa

será 50%, a intensidade média será 75% e a intensidade máxima será 100%. O valor do *timeout* antes de cada nó passar para a intensidade abaixo é de 2 unidades de tempo.

Para avaliar o algoritmo serão feitos, no total, seis testes. Em cada teste o número de utilizadores e o seu padrão de movimento vão ser diferentes. Os utilizadores irão ter dois padrões de movimentos, o primeiro onde os utilizadores irão percorrer caminhos semelhantes e o segundo onde o percurso de cada utilizador irá ser diferente.

Para cada um destes padrões de movimento serão feitos três testes, variando o número de utilizadores. O primeiro teste será feito com dois utilizadores, para simular as horas de menor uso, o segundo teste será com quatro utilizadores e representado o período de tempo em que o espaço tem alguns utilizadores e o terceiro teste, que será com sete utilizadores e representa as alturas de maior uso do espaço.

```
matrix 20 20
rectangle_add 0 0 20 20
lighting_radius 2
anticipation_distance 2
node_remove 8 8
node_remove 8 9
node_remove 8 10
node_remove 8 11
node_remove 8 12
node_remove 9 8
node_remove 9 9
node_remove 9 10
node_remove 9 11
node_remove 9 12
node_remove 10 8
node_remove 10 9
node_remove 10 10
node_remove 10 11
node_remove 10 12
node_remove 11 8
node_remove 11 9
node_remove 11 10
node_remove 11 11
node_remove 11 12
node_remove 12 8
node_remove 12 9
node_remove 12 10
node_remove 12 11
node_remove 12 12
base_intensity 19.0 9.0 3
base_intensity 9.0 19.0 3
intensity 0 25
intensity 1 50
intensity 2 75
intensity 3 100
timeout 2
wall_add 8.3 8.3 8.3 12.8
wall_add 12.8 8.3 8.3 8.3
wall_add 8.3 12.8 12.8 12.8
wall_add 12.8 12.8 12.8 8.3
```

Figura 37 – *Map script* do mapa utilizado nos testes.

5.2 Testes

5.2.1 Teste 1 – Percursos semelhantes, utilização baixa

O objetivo deste teste é testar o comportamento do sistema nas alturas em que estão poucos utilizadores a percorrer caminhos semelhantes. Para tal, foram usados dois utilizadores, que percorreram os caminhos representados na Figura 38. A linha azul representa o caminho que o utilizador 1 percorre e a linha a laranja representa o caminho do utilizador 2. Na Figura 39 temos o *user script* utilizado para este teste.

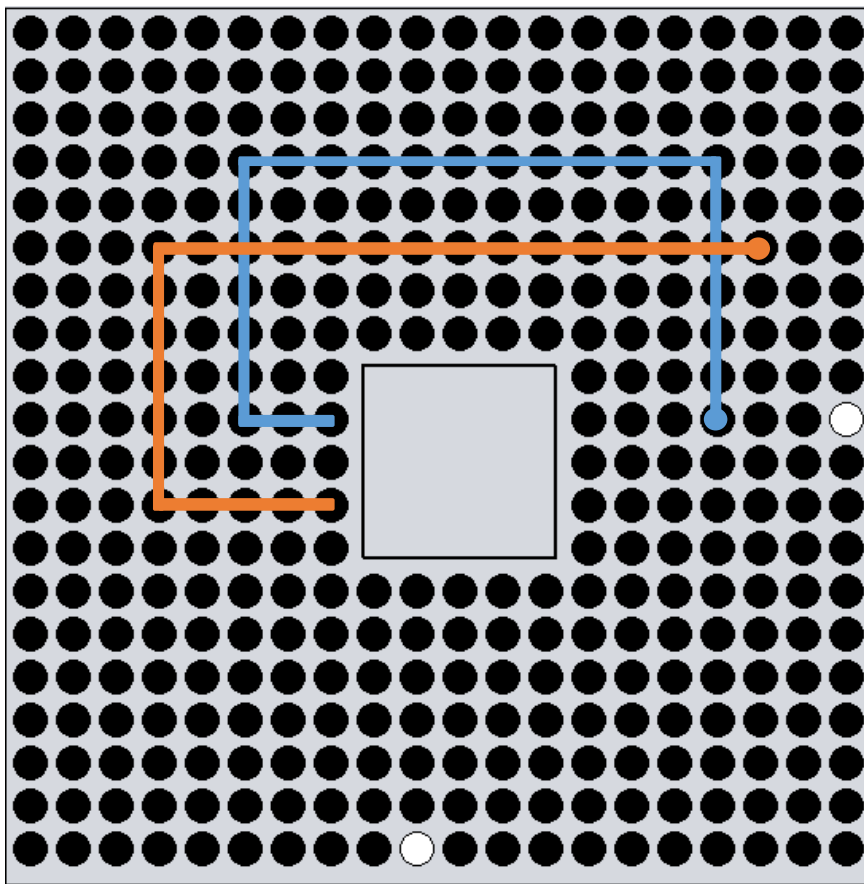


Figura 38 – Caminhos percorridos pelos utilizadores no teste 1.

```

start 1 -1 -1      start 2 -1 -1
move 1 7 9 1       move 2 -1 -1 3
move 1 5 9 2       move 2 7 11 1
move 1 5 3 6       move 2 3 11 4
move 1 16 3 11     move 2 3 5 6
move 1 16 9 6      move 2 17 5 14
  
```

Figura 39 – *User script* utilizado no teste 1.

Na Tabela 6 podemos observar algumas estatísticas da poupança de energia. O campo Energia total refere-se à energia que seria gasta se todas as luminárias estivessem na máxima intensidade, durante o teste. O campo Energia Gasta é a energia gasta pelas luminárias inteligentes durante o teste. O campo Poupança é a poupança de energia obtida pelas luminárias inteligentes.

| | |
|----------------------|--------|
| Energia Total | 10875 |
| Energia Gasta | 3580 |
| Poupança | 67,08% |

Tabela 6 – Estatísticas da poupança de energia no teste 1.

Na Tabela 7 estão presentes dados relativos ao envio de mensagens. Pode-se observar o número total de mensagens enviadas, o número de nós que receberam pelo menos uma mensagem, a média de mensagens recebidas por cada nó que recebeu pelo menos uma mensagem, a duração do teste em unidades de tempo e a média de mensagens recebidas por cada nó que recebeu pelo menos uma mensagem por unidade de tempo.

| | |
|--|-------|
| Mensagens enviadas | 4555 |
| Nós que receberam mensagens | 152 |
| Média de mensagens recebidas por nó | 29,97 |
| Duração do teste | 29 |
| Média por unidade de tempo | 1,03 |

Tabela 7 – Estatísticas do envio de mensagens no teste 1.

5.2.2 Teste 2 - Percursos semelhantes, utilização média

Neste teste, testou-se a situação em que o número de utilizadores é o normal e os seus percursos são semelhantes. Foram usados quatro utilizadores, que percorreram os caminhos representados na Figura 40. O utilizador 1 percorre a linha azul, o utilizador 2 percorre a linha laranja, o utilizador 3 percorre a linha vermelha e o utilizador 4 percorre a linha verde. Na Figura 41 temos o *user script* utilizado para este teste.

Na Tabela 8 podemos observar algumas estatísticas da poupança de energia e na Tabela 9 estão presentes dados relativos ao envio de mensagens.

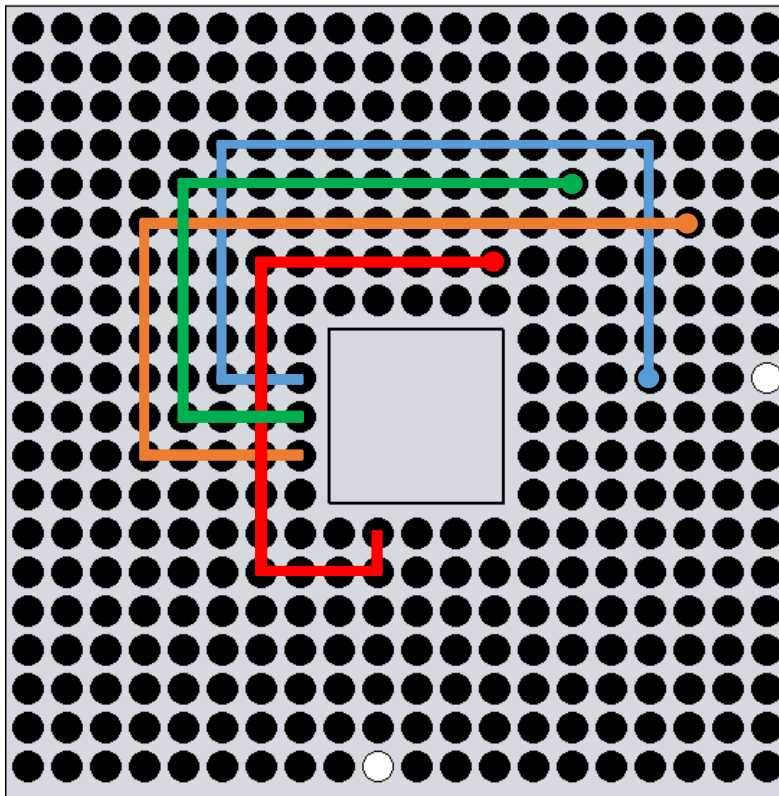


Figura 40 – Caminhos percorridos pelos utilizadores no teste 2.

```

start 1 -1 -1      start 3 -1 -1
move 1 7 9 1      move 3 -1 -1 1
move 1 5 9 2      move 3 9 13 1
move 1 5 3 6      move 3 9 14 1
move 1 16 3 11    move 3 6 14 3
move 1 16 9 6     move 3 6 6 8

start 2 -1 -1     move 3 13 6 7
move 2 -1 -1 3    start 4 -1 -1
move 2 7 11 1     move 4 -1 -1 2
move 2 3 11 4     move 4 7 10 1
move 2 3 5 6      move 4 4 10 3
move 2 17 5 14   move 4 4 4 6
                  move 4 14 4 10
                  move 4 14 1 3

```

Figura 41 – User script utilizado no teste 2.

| | |
|----------------------|--------|
| Energia Total | 10875 |
| Energia Gasta | 3880 |
| Poupança | 64,33% |

Tabela 8 – Estatísticas da poupança de energia no teste 2.

| | |
|------------------------------------|-------|
| Mensagens enviadas | 8406 |
| Nós que receberam mensagens | 198 |
| Média de mensagens por nó | 42,45 |
| Duração do teste | 29 |
| Média por unidade de tempo | 1,46 |

Tabela 9 – Estatísticas do envio de mensagens no teste 2.

5.2.3 Teste 3 – Percursos semelhantes, utilização alta

Para este teste, a situação testada foi com um número de utilizadores acima do normal e os seus percursos sendo semelhantes. Foram usados sete utilizadores, que percorreram os caminhos representados na Figura 42. Cada utilizador percorre uma das linhas representadas na Figura 42:

- Utilizador 1 – Linha azul;
- Utilizador 2 – Linha laranja;
- Utilizador 3 – Linha vermelha;
- Utilizador 4 – Linha verde;
- Utilizador 5 – Linha amarela;
- Utilizador 6 – Linha púrpura;
- Utilizador 7 – Linha branca.

Na Figura 43 temos o *user script* utilizado para este teste.

Na Tabela 10 podemos observar algumas estatísticas da poupança de energia e na Tabela 11 estão presentes dados relativos ao envio de mensagens.

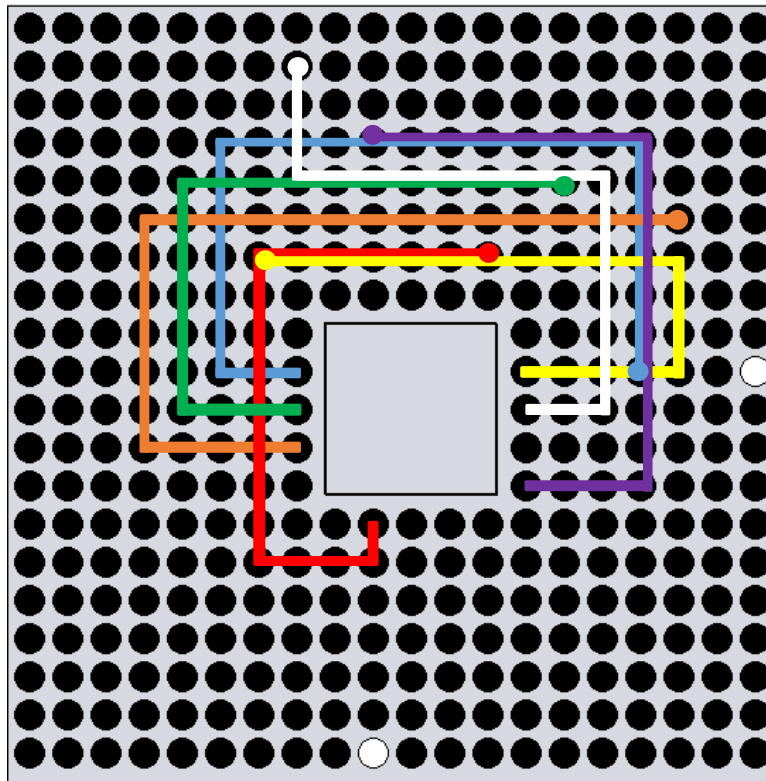


Figura 42 – Caminhos percorridos pelos utilizadores no teste 3.

```

start 1 -1 -1      start 4 -1 -1      start 6 -1 -1
move 1 7 9 1      move 4 -1 -1 2    move 6 13 12 1
move 1 5 9 2      move 4 7 10 1    move 6 16 12 3
move 1 5 3 6      move 4 4 10 3    move 6 16 3 9
move 1 16 3 11    move 4 4 4 6     move 6 9 3 7
move 1 16 9 6     move 4 14 4 10
                  move 4 14 1 3

start 2 -1 -1     start 5 -1 -1     start 7 -1 -1
move 2 -1 -1 3    move 5 -1 -1 5    move 7 13 10 1
move 2 7 11 1     move 5 13 9 1    move 7 15 10 2
move 2 3 11 4     move 5 17 9 4    move 7 15 4 6
move 2 3 5 6      move 5 17 6 3    move 7 7 4 8
move 2 17 5 14    move 5 6 6 11    move 7 7 2 2

start 3 -1 -1
move 3 -1 -1 1
move 3 9 13 1
move 3 9 14 1
move 3 6 14 3
move 3 6 6 8
move 3 13 6 7

```

Figura 43 – User script utilizado no teste 3.

| | |
|----------------------|--------|
| Energia Total | 10875 |
| Energia Gasta | 4488 |
| Poupança | 58,73% |

Tabela 10 – Estatísticas da poupança de energia no teste 3.

| | |
|------------------------------------|-------|
| Mensagens enviadas | 14269 |
| Nós que receberam mensagens | 240 |
| Média de mensagens por nó | 59,45 |
| Duração do teste | 29 |
| Média por unidade de tempo | 2,05 |

Tabela 11 – Estatísticas do envio de mensagens no teste 3.

5.2.4 Teste 4 – Percursos diferentes, utilização baixa

Com este teste pretende-se testar o comportamento do sistema quando estão poucos utilizadores no espaço e eles percorrem caminhos diferentes. Para isso, foram usados dois utilizadores, que percorreram os caminhos representados na Figura 44. A linha azul representa o caminho que o utilizador 1 percorre e a linha a laranja representa o caminho do utilizador 2. Na Figura 45 temos o *user script* utilizado para este teste.

Na Tabela 12 podemos observar algumas estatísticas da poupança de energia e na tabela 13 estão presentes dados relativos ao envio de mensagens.

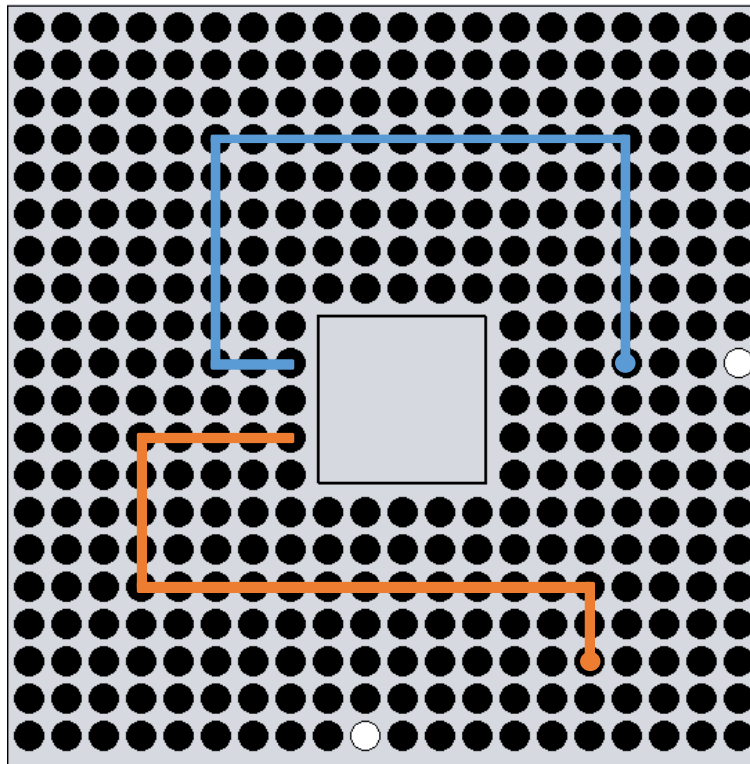


Figura 44 – Caminhos percorridos pelos utilizadores no teste 4.

```
start 1 -1 -1      start 2 -1 -1
move 1 7 9 1      move 2 7 11 1
move 1 5 9 2      move 2 3 11 4
move 1 5 3 6      move 2 3 15 4
move 1 16 3 11    move 2 15 15 12
move 1 16 9 6     move 2 15 17 2
```

Figura 45 – *User script* utilizado no teste 4.

| | |
|----------------------|--------|
| Energia Total | 10125 |
| Energia Gasta | 3390 |
| Poupança | 66,52% |

Tabela 12 – Estatísticas da poupança de energia no teste 4.

| | |
|------------------------------------|------|
| Mensagens enviadas | 4371 |
| Nós que receberam mensagens | 189 |
| Média de mensagens por nó | 23 |
| Duração do teste | 27 |
| Média por unidade de tempo | 0,86 |

Tabela 13 – Estatísticas do envio de mensagens no teste 4.

5.2.5 Teste 5 - Percursos diferentes, utilização média

O objetivo é testar comportamento do sistema estão alguns utilizadores no espaço e eles percorrem caminhos diferentes. Foram usados quatro utilizadores, que percorreram os caminhos representados na Figura 46. A linha azul representa o caminho que o utilizador 1 percorre, a linha a laranja representa o caminho do utilizador 2, a linha vermelha representa o caminho do utilizador 3 e a linha verde representa o caminho do utilizador 4. Na Figura 47 temos o *user script* utilizado para este teste.

Na Tabela 14 podemos observar algumas estatísticas da poupança de energia e na Tabela 15 estão presentes dados relativos ao envio de mensagens.

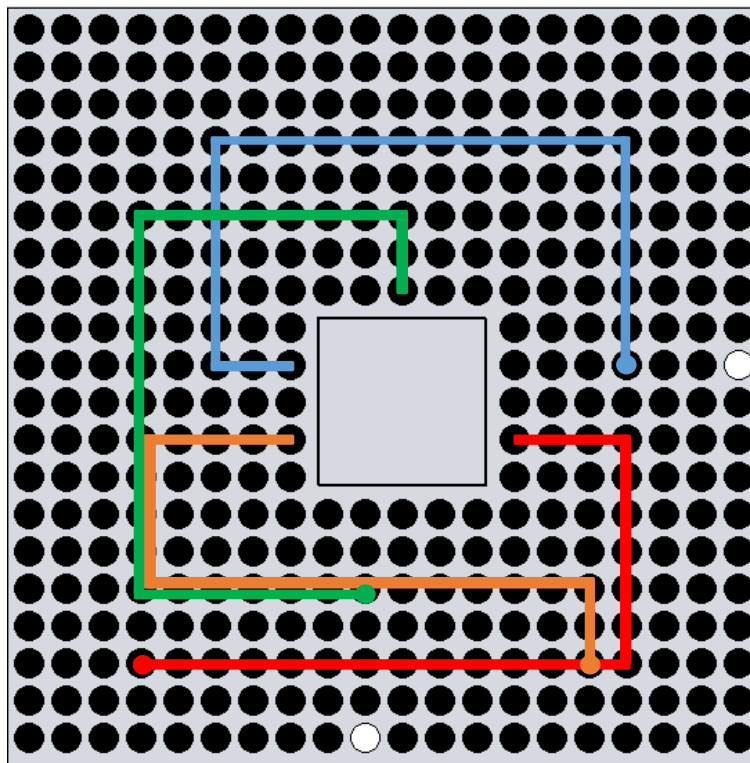


Figura 46 – Caminhos percorridos pelos utilizadores no teste 5.

```

start 1 -1 -1      start 3 -1 -1
move 1 7 9 1      move 3 13 11 1
move 1 5 9 2      move 3 16 11 3
move 1 5 3 6      move 3 16 17 6
move 1 16 3 11    move 3 3 17 13
move 1 16 9 6

start 2 -1 -1     start 4 -1 -1
move 2 7 11 1     move 4 -1 -1 1
move 2 3 11 4     move 4 10 7 1
move 2 3 15 4     move 4 10 5 2
move 2 15 15 12   move 4 3 5 7
move 2 15 17 2    move 4 3 15 10
move 2 15 17 2    move 4 9 15 6

```

Figura 47 – User script utilizado no teste 5.

| | |
|----------------------|--------|
| Energia Total | 10500 |
| Energia Gasta | 4216 |
| Poupança | 59,85% |

Tabela 14 – Estatísticas da poupança de energia no teste 5.

| | |
|------------------------------------|------|
| Mensagens enviadas | 8876 |
| Nós que receberam mensagens | 255 |
| Média de mensagens por nó | 34,8 |
| Duração do teste | 28 |
| Média por unidade de tempo | 1,24 |

Tabela 15 – Estatísticas do envio de mensagens no teste 5.

5.2.6 Teste 6 – Percursos diferentes, utilização alta

Neste teste usou-se um número de utilizadores acima do normal, em que os seus percursos têm direções diferentes. Foram usados sete utilizadores, que percorreram os caminhos representados na Figura 48. Cada utilizador percorre uma das linhas representadas na Figura 48:

- Utilizador 1 – Linha azul;
- Utilizador 2 – Linha laranja;
- Utilizador 3 – Linha vermelha;
- Utilizador 4 – Linha verde;
- Utilizador 5 – Linha amarela;
- Utilizador 6 – Linha púrpura;
- Utilizador 7 – Linha branca.

Na Figura 49 temos o *user script* utilizado para este teste.

Na Tabela 16 podemos observar algumas estatísticas da poupança de energia e na Tabela 17 estão presentes dados relativos ao envio de mensagens.

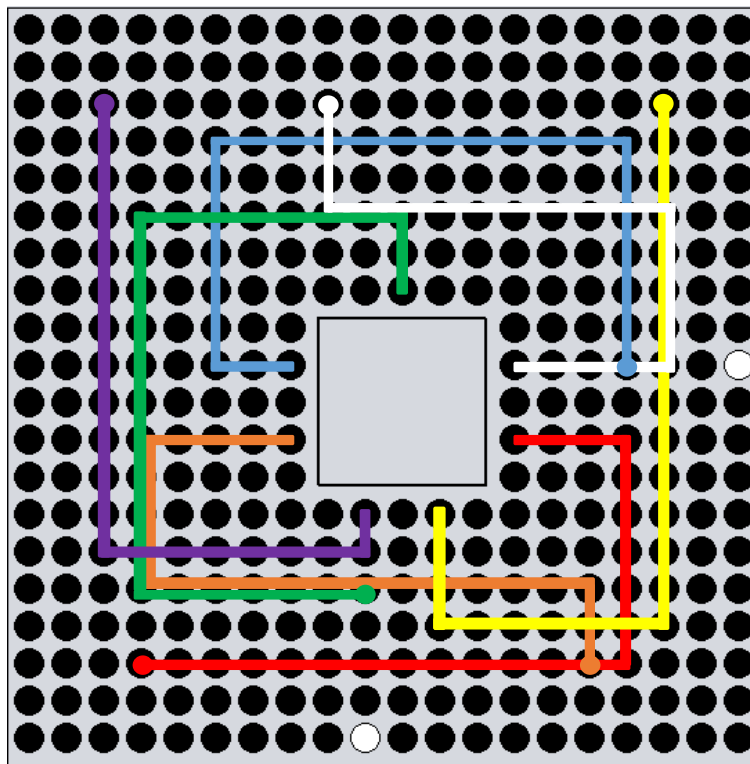


Figura 48 – Caminhos percorridos pelos utilizadores no teste 6.

```

start 1 -1 -1      start 4 -1 -1      start 6 -1 -1
move 1 7 9 1      move 4 -1 -1 1     move 6 9 13 1
move 1 5 9 2      move 4 10 7 1     move 6 9 14 1
move 1 5 3 6      move 4 10 5 2     move 6 2 14 7
move 1 16 3 11    move 4 3 5 7      move 6 2 2 12
move 1 16 9 6     move 4 3 15 10
                  move 4 9 15 6

start 2 -1 -1     start 5 -1 -1     start 7 -1 -1
move 2 7 11 1     move 5 11 13 1    move 7 13 9 1
move 2 3 11 4     move 5 11 16 3    move 7 17 9 4
move 2 3 15 4     move 5 17 16 6    move 7 17 5 4
move 2 15 15 12  move 5 17 2 14    move 7 8 5 9
move 2 15 17 2    move 7 8 2 3

start 3 -1 -1
move 3 13 11 1
move 3 16 11 3
move 3 16 17 6
move 3 3 17 13

```

Figura 49 – User script utilizado no teste 6.

| | |
|----------------------|--------|
| Energia Total | 10500 |
| Energia Gasta | 5047 |
| Poupança | 51,93% |

Tabela 16 – Estatísticas da poupança de energia no teste 6.

| | |
|------------------------------------|-------|
| Mensagens enviadas | 15070 |
| Nós que receberam mensagens | 315 |
| Média de mensagens por nó | 46,37 |
| Duração do teste | 28 |
| Média por unidade de tempo | 1,66 |

Tabela 17 – Estatísticas do envio de mensagens no teste 6.

5.3 Avaliação dos resultados

Nesta seção vão ser avaliados os resultados dos testes feitos ao sistema. Na Tabela 18 podemos observar as estatísticas da poupança de energia de todos os testes. O campo Energia total refere-se à energia que seria gasta se todas as luminárias estivessem na máxima intensidade. O campo Energia Gasta é a energia gasta pelas luminárias inteligentes durante o teste. O campo Poupança é a poupança de energia obtida pelas luminárias inteligentes. Observando a poupança de energia obtida nos testes podemos concluir:

- Quando o espaço tem pouca utilização, o padrão de movimento dos utilizadores tem pouco impacto na poupança obtida (teste 1 e 4);
- A poupança de energia é superior quando os utilizadores percorrem caminhos semelhantes. Esta diferença é mais evidente comparando os resultados dos testes 3 e 6;
- À medida que se aumenta o número de utilizadores presentes a poupança de energia vai diminuindo.
- O sistema teve um comportamento melhor do que o esperado, com uma poupança média a rondar os 60%, muito acima dos 40% esperados. Quando comparado com outros sistemas já apresentados verificamos que as poupanças apresentadas são superiores às do sistema da Tvilight (CitySense e CityManager), com 54% e bastante superiores ao sistema apresentado no artigo [10], cuja poupança de energia ronda os 40%.

| Teste | Energia Total | Energia Gasta | Poupança |
|--------------|---------------|---------------|----------|
| 1 | 10875 | 3580 | 67,08% |
| 2 | 10875 | 3880 | 64,33% |
| 3 | 10875 | 4488 | 58,73% |
| 4 | 10125 | 3390 | 66,52% |
| 5 | 10500 | 4216 | 59,85% |
| 6 | 10500 | 5047 | 51,93% |
| Média | | | 61,42% |

Tabela 18 - Estatísticas da poupança de energia nos testes efetuados.

Na Tabela 19 estão presentes dados relativos ao envio de mensagens. Pode-se observar a média de mensagens recebidas por cada nó que recebeu pelo menos uma mensagem, a duração do teste em unidades de tempo e a média de mensagens recebidas por cada nó que recebeu pelo menos uma mensagem por unidade de tempo. Com os dados recolhidos verifica-se que quando os utilizadores percorrem caminhos semelhantes a média de mensagens é superior, devido a uma maior concentração de utilizadores numa área menor e que cada nó processar cerca de 2 mensagens por unidade de tempo

é um valor bastante baixo. Isto significa que o *hardware* para as luminárias pode ser mais simples e logo mais barato.

| Teste | Média | Duração | Média por unidade de tempo |
|--------------|--------------|----------------|-----------------------------------|
| 1 | 29,97 | 29 | 1,03 |
| 2 | 42,45 | 29 | 1,46 |
| 3 | 59,45 | 29 | 2,05 |
| 4 | 23 | 27 | 0,86 |
| 5 | 34,8 | 28 | 1,24 |
| 6 | 46,37 | 28 | 1,66 |

Tabela 19 - Estatísticas relativas ao envio de mensagens nos testes efetuados.

6. Conclusão

Com este projeto era esperado desenvolver um sistema de iluminação inteligente, que funcionasse de forma distribuída. Este sistema seria constituído por luminárias inteligentes, o *gateway*, a aplicação de configuração e o algoritmo presente nas luminárias. Devido a algumas limitações, o sistema teve de ser testado recorrendo a um simulador. Como tal, esta dissertação focou-se no desenvolvimento do algoritmo e da aplicação de configuração.

O simulador utilizado foi desenvolvido especificamente para este projeto. Permitia ver o funcionamento do sistema em tempo real, controlar a simulação e configurar o ambiente de teste e conforme desejado. No final da simulação, também era possível observar estatísticas relativas ao consumo de energia e ao envio de mensagens das luminárias.

O principal componente deste sistema é o algoritmo que está presente nas luminárias. A abordagem seguida cumpre os objetivos de iluminar a área à volta do utilizador e de determinar a sua direção. Esta abordagem, nos casos em que existem vários utilizadores muito próximos, há a possibilidade de se acenderem mais luminárias do que o suposto. No entanto, mesmo com esta limitação, o sistema continua a ser muito superior, quando comparado aos sistemas mais convencionais utilizados atualmente.

Inicialmente a aplicação de configuração seria utilizada para configurar as luminárias durante a instalação e gerir o sistema. No entanto, como o sistema foi testado através de um simulador, a aplicação passou a configurar o ambiente de teste que seria utilizado no simulador.

Os testes efetuados permitiram avaliar a poupança de energia obtida e o número médio de mensagens transmitidas por unidade de tempo. Os testes foram desenhados para se assemelharem a situações reais de utilização do espaço, variando o número de utilizadores e o seu padrão de movimento. Verificou-se que o sistema permite poupanças consideráveis de energia, na ordem dos 60%. Este valor é superior a outros sistemas aqui apresentados. Para além disso também se verificou que a média de mensagens por luminária por unidade de tempo é relativamente baixo, que permite que o *hardware* para cada luminária seja mais em conta.

No futuro poder-se-ia desenvolver um protótipo da luminária inteligente. Com alguns destes protótipos seria possível fazer testes num ambiente real e obter dados mais concretos sobre sistema. Os protótipos também iriam permitir o desenvolvimento da aplicação de configuração, como esta estava inicialmente planeada.

A partir daí seria possível desenvolver funcionalidades mais avançadas, como por exemplo, o envio automático de relatórios de avarias ou programar os diferentes parâmetros para variarem consoante a hora do dia.

Seria interessante introduzir a autoconfiguração das luminárias. Isto teria como objetivo tornar as luminárias o mais próximo possível dos dispositivos *plug-and-play*. Isto significa que a luminária conseguiria descobrir a sua posição, os seus vizinhos e registar-se na aplicação de configuração automaticamente.

Outra possibilidade seria melhorar o algoritmo ou experimentar novas abordagens, que permitissem diminuir ou até eliminar completamente a limitação da abordagem atual, calculando as direções de todos os utilizadores sem erros.

Também poderia ser estudado a possibilidade de adaptar este sistema à iluminação de rua, em particular aplicá-lo na periferia das cidades, onde existe pouco movimento.

Olhando para o trabalho desenvolvido e para os objetivos que tinham de ser cumpridos, verifica-se que, apesar do trabalho desenvolvido não ser o que foi proposto inicialmente, o objetivo essencial, a criação de um algoritmo distribuído que permitisse poupar energia, foi cumprido com sucesso.

7.Referências

- [1] [EIA projects world energy consumption will increase 56% by 2040](http://www.eia.gov/todayinenergy/detail.cfm?id=12251#), U.S. Energy Information Administration, julho 2013 <http://www.eia.gov/todayinenergy/detail.cfm?id=12251#> Acedido dia 6 de julho de 2016
- [2] EU energy in figures – Statistical Pocket Book 2015, página 11, European Union, 2015 http://ec.europa.eu/energy/sites/ener/files/documents/PocketBook_ENERGY_2015%20PDF%20final.pdf Acedido dia 6 de julho de 2016
- [3] World Energy Outlook 2014, página 4, Internacional Energy Agency, 2014 http://www.iea.org/publications/freepublications/publication/WEO_2014_ES_English_WEB.pdf Acedido dia 6 de julho de 2016
- [4] N. Yoshiura, Y. Fujii, N. Ohta, Smart street light system looking like usual street lights based on sensor networks, *13th International Symposium on Communications and Information Technologies (ISCIT), 2013*
- [5] Tvilight, A Wireless Network and Sensor Technologies Company, <http://www.tvilight.com> Acedido 6 de julho de 2016
- [6] Tvilight CityManager, Smart City Sensors, Network and Light Management Software, <http://www.tvilight.com/management-software/#citymanager> Acedido dia 6 de julho de 2016
- [7] Tvilight CitySense, A Plug-and-Play Wireless Sensor, <http://www.tvilight.com/citysense/> Acedido dia 6 de julho de 2016
- [8] Tvilight Case Study – Outdoor Parking http://www.tvilight.com/wp-content/uploads/2014/03/CSE_Hoogkerk_English_Online1.pdf Acedido dia 6 de julho de 2016
- [9] R. Müllner, A. Riener, An energy efficient pedestrian aware Smart Street Lighting system, *International Journal of Pervasive Computing and Communications Vol. 7 No. 2, 2011*, pp. 147-161
- [10] LiLong, ChuXiumin, WuYong, WuQing, The Development of Road Lighting Intelligent Control System Based on Wireless Network Control, *International Conference on Electronic Computer Technology, 2009*

- [11] Z. Zhang, A. Mistry, W. Yin, P. L. Venetianer, Embedded Smart Sensor for Outdoor Parking Lot Lighting Control, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012*

- [12] OMNeT++, Discrete Event Simulator, <https://omnetpp.org/> Acedido dia 6 de julho de 2016

- [13] J-Sim, Simulation System for Building and Analyzing Quantitive Numeric Models, <http://www.physiome.org/jsim/> Acedido dia 6 de julho de 2016

- [14] SIDnet-SWANS, A Simulator and Integrated Development Plataform for Sensor Networks Application, <http://users.eecs.northwestern.edu/~ocg474/SIDnet.html> Acedido dia 6 de julho de 2016

- [15] Ptolemy II, Software Framework for Experimenting with Actor-Oriented Design, <http://ptolemy.eecs.berkeley.edu/ptolemyII/> Acedido dia 6 de julho de 2016