

**Universidade do Minho**

Cátia Matos Salgado

**Sistema Inteligente para a simulação  
do posicionamento e movimento de pessoas  
ou equipamentos dentro de um hospital**

**Tese de Mestrado**

Mestrado Integrado em Engenharia Biomédica

Trabalho efetuado sob a orientação de:

**Professor Doutor José Machado**

**16 de Novembro de 2012**

# Agradecimentos

Embora uma dissertação seja, pela sua finalidade académica, um trabalho individual, há contributos que não podem e nem devem deixar de ser realçados. Por essa razão, desejo expressar os meus sinceros agradecimentos:

Ao professor José Machado, por ter estado sempre disponível para o esclarecimento de dúvidas, mas acima de tudo quero agradecer-lhe a motivação que me deu para trabalhar no projeto.

Ao meu amigo Bruno Gomes, por nunca dizer que não, porque me consegue ajudar sempre em tudo e, acima de tudo, pelo café maravilhoso!

Ao Luciano Bessa, pela enorme ajuda que me deu e principalmente por ter tido sempre disponibilidade e boa disposição.

À minha tia e amiga Manuela Cortinhas um especial agradecimento pelo empenho e dedicação, e porque a sua ajuda foi fundamental na reta final do trabalho.

Ao Nuno, porque tem uma paciência infinita.

À Rebecca, pela sua disponibilidade em ajudar a escrever o resumo da tese em inglês.

À minha amiga Carina Cortinhas, agradeço sinceramente o interesse e todas as dicas que me deu.

À minha querida irmã Mariana pela sua ternura, preciosa em todos os momentos.

Ao meu irmão Renato pelo entusiasmo demonstrado em relação ao trabalho.

Finalmente um especial agradecimento ao meu pai e à minha mãe, por estarem sempre presentes e porque sempre me apoiaram em tudo. Sem eles a realização deste trabalho não teria sido possível.



# Resumo

Os órgãos de gestão dos hospitais optam cada vez mais pela implementação de sistemas de monitorização baseados na tecnologia RFID, tendo em vista sobretudo a redução de custos associados à perda e roubo de equipamentos, o aumento da segurança de pacientes e profissionais e a deteção do acesso de pessoal a zonas não autorizadas. Testes e estudos de otimização, nomeadamente em relação à configuração dos sensores na arquitetura da rede RFID, devem ser realizados para aumentar o desempenho destes sistemas. A simulação surge neste contexto como uma importante ferramenta de apoio, uma vez que permite que os estudos sejam endereçados a partir de um ambiente computacional, evitando as desvantagens inerentes à realização de testes no próprio hospital. Recentemente, o estudo da questão de conservação de energia, em redes de sensores de *tracking* de objetos, tem atraído muita atenção. A previsão de trajetórias pode ser utilizada para determinar o conjunto de sensores que num determinado momento devem ser desativados para reduzir consumos energéticos e, conseqüentemente, aumentar o tempo de vida do sistema.

O objetivo deste trabalho é integrar as duas temáticas referidas – simulação e previsão – no desenvolvimento de um sistema inteligente capaz de simular e prever a trajetória de uma entidade numa área preenchida com sensores. A área escolhida para a simulação consiste no piso pediátrico de um hospital no Norte de Portugal, onde um sistema RFID de monitorização se encontra atualmente implementado para a monitorização de pacientes. A análise do sistema desenvolvido é realizada através do estudo de cenários, onde estatísticas obtidas em diferentes condições de simulação são analisadas sob diversos prismas. Os resultados provam que o sistema proposto é capaz de simular com sucesso o movimento de uma ou mais entidades num ambiente hospitalar real, e de estimar localizações com uma precisão média de 62% para um modelo de simulação que tem em consideração a existência de variabilidade e aleatoriedade no movimento. A previsão é realizada através da aplicação de um algoritmo de Data Mining, denominado SK-Means, ao histórico de percursos de cada entidade, e permite a obtenção de padrões

de movimento específicos e distintos para cada uma delas. As localizações previstas são apresentadas ao utilizador em tempo real, de forma dinâmica.

Pretende-se que no futuro o sistema possa ser implementado no hospital para permitir a visualização em tempo real dos pacientes.

# Abstract

Many hospitals choose to rely on RFID tracking systems to avoid the theft or loss of equipment, increase patient and staff safety, finding missing patients or staff, and issuing warnings about personnel access to unauthorized areas. Optimization studies, concerning the configuration of the sensors in the RFID network architecture, must be made in order to increase the performance of these systems. The simulation arises in this context as an important supporting tool, since it allows studies to be addressed from a computing environment, avoiding the disadvantages of testing in the hospital. Recently, the issue of energy saving in object tracking sensor networks has attracted much attention. Trajectory prediction is a software approach which provides, in real time, the set of sensors that can be deactivated to reduce power consumption and thereby increase the system's lifetime.

Hence, the system proposed here aims to integrate the aforementioned strategies - simulation and prediction - in the development of an intelligent tracking simulation system able to simulate and predict an entity's trajectory in an area fitted with sensors. The chosen locale for the simulation of object movements was the paediatric floor of a hospital in the North of Portugal, where an RFID monitoring system is currently employed for patient monitoring. The analysis of the system is developed through the study of scenarios where statistics obtained in different simulation conditions are analyzed from several perspectives. The results show that the proposed system is able to successfully simulate the movement of one or more entities, in an actual hospital environment, and to estimate locations with an average accuracy of 62 % for a simulation model which takes into account the existence of variability and randomness in the movement. The prediction is accomplished by applying a Data Mining algorithm, designated SK-Means, to discover object movement patterns through historical trajectory data. The predicted locations are presented to the user in real time, dynamically.

It is anticipated that this system will be incorporated in the hospital, to allow real time visualization of patients, in the future.



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Contextualização e Enquadramento . . . . .	3
1.2	Objetivos . . . . .	6
1.3	O Hospital . . . . .	8
1.4	Simulação . . . . .	13
1.5	Estrutura da Dissertação . . . . .	16
<b>2</b>	<b>Trabalho Relacionado</b>	<b>19</b>
2.1	Redes de Sensores . . . . .	19
2.1.1	Object Tracking Sensor Networks . . . . .	19
2.1.2	Previsão de trajetórias em OTSNs . . . . .	22
2.1.3	Redes RFID . . . . .	24
2.1.4	RFID nos Hospitais . . . . .	27
2.2	Extração de Conhecimento de Bases de Dados . . . . .	30
2.2.1	Data Mining . . . . .	30
2.2.2	Clustering . . . . .	34
2.2.3	Algoritmo Simple K-Means . . . . .	36
<b>3</b>	<b>Arquitetura e Implementação</b>	<b>41</b>
3.1	Visão Geral: sistema anterior <i>versus</i> sistema proposto . . . . .	41
3.2	Descrição dos Componentes . . . . .	43
3.2.1	Simulador . . . . .	43
3.2.2	Weka . . . . .	47
3.3	Tecnologias e metodologias . . . . .	49
<b>4</b>	<b>Simulador</b>	<b>51</b>
4.1	Interface Gráfica . . . . .	51
4.2	Simulação da trajetória de uma pessoa . . . . .	57
4.2.1	Corredores . . . . .	59
4.2.2	Quartos . . . . .	60
4.2.3	Algoritmo . . . . .	63

4.3	Previsão da trajetória . . . . .	67
4.4	Resultados . . . . .	69
<b>5</b>	<b>Resultados</b>	<b>73</b>
5.1	Cenário 1: sensores nas entradas e saídas . . . . .	74
5.2	Cenário 2: sensores a delimitar todos os corredores principais .	77
5.3	Cenário 3: reprodução dos sensores existentes no Hospital . .	79
5.4	Cenário 4: colocação de sensores nas portas de divisões . . . .	86
5.5	Cenário 5: simulação com vários objetos em simultâneo . . . .	90
5.6	Análise dos cenários . . . . .	93
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>101</b>
6.1	Conclusões . . . . .	101
6.2	Sugestões e Trabalho Futuro . . . . .	105

# Lista de Figuras

1.1	Ilustração do sistema proposto. . . . .	9
1.2	Planta do piso hospitalar. . . . .	12
2.1	Arquitetura de um sistema RFID. . . . .	24
2.2	Estágios que constituem o processo de Extração de Conhecimento em Bases de Dados. . . . .	32
2.3	Mudança de posição dos centróides (indicados pelo sinal +) e atribuições de dados (indicadas pela cor) durante uma execução do algoritmo K-Means. . . . .	39
3.1	Arquitetura do sistema proposto e interação entre os diferentes módulos. . . . .	43
3.2	Diagrama ilustrativo do modelo de domínio - primeira versão. . . . .	45
3.3	Diagrama ilustrativo do modelo de domínio - versão proposta. . . . .	46
3.4	Simulação. . . . .	48
4.1	Representação gráfica do piso hospitalar no simulador com legenda das 5 diferentes zonas de circulação. . . . .	52
4.2	Barra de opções da janela do caminho real durante o movimento. . . . .	54
4.3	Diferentes etapas da simulação. . . . .	55
4.4	Diferentes etapas da simulação para vários objetos em simultâneo. . . . .	56
4.5	Barra de opções da janela de simulação. . . . .	56
4.6	Simulação de movimento nos corredores do Hospital. . . . .	61
4.7	Simulação referente a 52 movimentos nos quartos do Hospital. . . . .	62
4.8	Simulação referente a 40 movimentos nos quartos do Hospital. . . . .	63
4.9	Aspeto do ambiente gráfico do simulador: janela de resultados. . . . .	70
4.10	Legenda da janela de resultados . . . . .	70
5.1	Possíveis percursos do objeto e localização dos elevadores. . . . .	75
5.2	Previsão de trajetória: cenário 1. . . . .	76
5.3	Gráfico da precisão em função do número de registos: cenário 1. . . . .	77
5.4	Simulação de trajetória: cenário 2. . . . .	79

5.5	Previsão de trajetória: cenário 2. . . . .	80
5.6	Simulação de trajetória: cenário 2. . . . .	81
5.7	Previsão de trajetória: cenário 2. . . . .	82
5.8	Previsão de trajetória final: cenário 2. . . . .	83
5.9	Resultados de previsão: cenário 2. . . . .	84
5.10	Gráfico da precisão em função do número de registros: cenário 2. . . . .	84
5.11	Simulação de trajetória: cenário 3 . . . . .	85
5.12	Previsão de trajetória: cenário 3. . . . .	85
5.13	Simulação de trajetória: cenário 3. . . . .	86
5.14	Simulação de trajetória: cenário 3. . . . .	87
5.15	Previsão final de trajetória: cenário 3. . . . .	87
5.16	Gráfico da precisão em função do número de registros: cenário 3. . . . .	88
5.17	Simulação e previsão: cenário 4. . . . .	89
5.18	Possíveis percursos estipulados para os 3 objetos. . . . .	91
5.19	Simulação de monitorização de 3 objetos em simultâneo. . . . .	92
5.20	Previsão associada ao objeto com id 1. . . . .	92
5.21	Previsão associada ao objeto com id 2. . . . .	93
5.22	Previsão associada ao objeto com id 3. . . . .	93
5.23	Objetos que se movem a velocidades diferentes. . . . .	94
5.24	Simulação de trajetória no bloco 4: cenário 2 . . . . .	96
5.25	Previsão de trajetória no bloco 4: cenário 2 . . . . .	96
5.26	Resultados de previsão no bloco 4: cenário 2. . . . .	97
5.27	Relatório de precisão média para os diferentes cenários - repetição de procedimento. . . . .	98
5.28	Gráfico da precisão em função do número de registros: cenário 2 - repetição de procedimento. . . . .	99

# Lista de Tabelas

4.1	Movimento nos corredores . . . . .	60
5.1	Relatório de precisão média: cenário 1. . . . .	76
5.2	Relatório de precisão média em função do percurso: cenário 1. . . . .	77
5.3	Relatório de precisão média: cenário 2. . . . .	79
5.4	Relatório de precisão média em função do percurso: cenário 2. . . . .	80
5.5	Relatório de precisão média: cenário 3. . . . .	82
5.6	Relatório de precisão média em função do percurso: cenário 3. . . . .	83
5.7	Relatório de precisão média para os diferentes cenários. . . . .	94
5.8	Relatório de registos da base de dados: cenário 2 . . . . .	97
5.9	Duração do modelo de previsão em diferentes condições. . . . .	100



# Lista de Algoritmos

1	SK-Means . . . . .	38
2	Próxima posição . . . . .	66
3	Cálculo de $x_{dir}$ . . . . .	67



# Siglas e Acrónimos

**RFID** Radio-Frequency Identification

**API** Application Programming Interface

**HIS** Healthcare Information System

**AIDA** Agency for Integration, Diffusion and Archive of Medical Information

**OTSN** Object Tracking Sensor Network

**FBM** Foward-Backward Matching

**CRPM** Continuous Route Pattern Mining

**PES** Prediction-based Energy Saving scheme

**EPC** Electronic Product Code

**DMT** Data Mining Techniques

**KDD** Knowledge Discovery in Databases

**DMK** Data Mining Knowledge

**AI** Artificial intelligence

**MVC** Model View Controller

**Weka** Waikato Environment for Knowledge Analysis

**GUI** Graphical User Interface

**SQL** Structured Query Language



# Capítulo 1

## Introdução

A presente dissertação serve o propósito de descrever todo o trabalho desenvolvido na criação de um Sistema Inteligente para a simulação do posicionamento e do movimento de pessoas ou equipamentos dentro de um hospital. O projeto surge no âmbito da dissertação de mestrado do curso Mestrado Integrado em Engenharia Biomédica da Universidade do Minho.

No 1º capítulo é feita uma contextualização e enquadramento do tema, são enunciadas as motivações e os objetivos propostos, assim como um enquadramento em relação ao Hospital onde se pretende implementar o sistema. Neste capítulo, também é apresentada uma contextualização referente à definição, aplicações e importância da simulação em geral e da simulação de trajetórias em ambiente hospitalar em particular. Por último, é apresentada a estrutura da dissertação.

### 1.1 Contextualização e Enquadramento

É comum encontrar-se em ambientes globalizados os denominados objetos em movimento: pessoas, produtos ou objetos que se movem continuamente, alterando constantemente a sua localização ao longo do tempo. Recentes avanços tecnológicos nas áreas da informação e comunicação, como Identificação por Rádio Frequência (Radio-Frequency Identification - RFID), redes Wireless e sensores inteligentes, fazem com que nos aproximemos cada vez mais da “Internet das Coisas” (*Internet of Things*): uma visão de uma infra-

estrutura global de objetos em rede. Uma rede deste género pode fornecer informação acerca da localização de objetos em movimento, de forma omnipresente e em tempo real, contribuindo para o apoio à decisão nos serviços que trabalhem com este tipo de metodologia [1–3].

Recentemente têm sido escritos diversos artigos sobre redes de sensores devido à ampla gama de potenciais aplicações destes sistemas. O *tracking* de objetos e pacientes continua a ser um dos maiores desafios nas operações hospitalares, tornando-se um tópico muito importante em redes de sensores [4, 5]. O termo “*tracking*” está subentendido como a capacidade de um sistema monitorizar o fluxo de objetos através de um determinado número de localizações, e o registo da sua presença nas localizações pré-definidas ao longo do caminho (sensores). A identificação e localização inequívocas de uma entidade (pessoa ou objeto) num dado momento, e ao longo do tempo, podem fornecer a base para o desenvolvimento de diversas soluções para problemas críticos dentro do sistema hospitalar [6].

Muitos pacientes, incluindo os pacientes psiquiátricos, pacientes de trauma e idosos, constituem frequentemente um perigo para si mesmos e para outras pessoas ao circularem fora da área que lhes é permitida, seja quando acedem a uma área restrita, seja quando abandonam sem autorização o hospital. A incapacidade de encontrar médicos, enfermeiros ou outros profissionais pode significar mais horas de trabalho e uso menos efetivo de pessoal. A ocorrência de furtos de equipamentos, saídas não autorizadas do hospital e a procura de equipamentos desaparecidos por parte da equipa de enfermeiros ou outro pessoal médico é uma perda de tempo e de recursos, e dispersa a concentração nos cuidados ao paciente [4]. A falta de controlo no equipamento leva a uma sobrelotação, perda e roubo de equipamentos que, por sua vez, implica despesas desnecessárias e excessivas para o hospital. A solução para estes problemas passa inexoravelmente pela capacidade de ter conhecimento, em tempo real, da localização exata ou aproximada de diferentes entidades no hospital. Nestas situações é necessário que os sistemas de *tracking* hospitalar atuem de forma a ser possível detetar com a maior celeridade a sua localização [7].

Como exemplo real deste problema refere-se o caso apresentado por Qu

et al. [4], de três hospitais nos Estados Unidos da América que implementaram um sistema de *tracking* em 12000 equipamentos. O sistema enviava um alarme para a segurança sempre que um equipamento era movido para além dos seus limites. Graças a este sistema, um dos hospitais foi capaz de poupar cerca de 200.000 dólares no primeiro ano, devido à redução de perdas e roubos e consequentemente de investimentos em equipamentos e adiamento de novas compras. Além disto, a solução adoptada permitiu aos enfermeiros poupar 30 minutos por turno em procuras.

Um sistema que permita a localização de objetos e pessoas dentro de um hospital, deve fornecer a localização em tempo real dos mesmos, tornando-se imperativo reduzir ao máximo a ocorrência de erros de forma a evitar acidentes. Em ambientes hospitalares de grande escala, porém, a identificação e monitorização espacial de pacientes, visitas, *staff*, provedores de serviço, medicamentos e equipamentos através de redes de sensores são processos extremamente complexos e muitas vezes não respondem às necessidades nem oferecem informação atempada para o apoio à decisão [6]. Estas barreiras criam a necessidade de implementar novas tecnologias e metodologias para a resolução de problemas que necessitam de ser previamente estudados e analisados, para posteriormente poderem ser implementados no sentido de melhorar a qualidade do serviço e facilitar o trabalho de profissionais.

Estudos recentes descrevem algumas das atuais aplicações de monitorização, localização e identificação através de sistemas RFID e afirmam que é possível orientar esta tecnologia no sentido das necessidades de *tracking* de medicamentos, exames de laboratório, equipamentos, pacientes e profissionais [4]. Atualmente, a maior parte dos sistemas de *tracking* hospitalar de pessoas e equipamentos funcionam com base nesta tecnologia e são diversas as publicações recentes que destacam a utilização de RFID na área da saúde [4, 8–14].

No entanto, é muito importante que as arquiteturas destes sistemas sejam estudadas e testadas previamente à instalação nos hospitais, por forma a, tendo em consideração os custos associados à implementação e orçamentos previstos, minimizar e otimizar o número de componentes RFID utilizados no processo, avaliar o impacto de mudanças no sistema e, acima de tudo,

evitar erros decorrentes de falhas nos sistemas. A área da saúde é aliás, uma área onde se exige segurança, excelência e fiabilidade, e onde a ocorrência de erros, em particular de falhas nos sistemas de software, não é tolerável. Neste contexto, sistemas eficientes, com maior fiabilidade e processamento inteligente de dados podem revelar-se bastante úteis no apoio à decisão.

## 1.2 Objetivos

Para responder às características referidas anteriormente propõe-se a criação de um sistema inteligente, representado na Figura 1.1, que faça a simulação de um sistema de *tracking* RFID e a previsão de localizações de pessoas e equipamentos dentro de um hospital.

Um dos objetivos passa por desenvolver uma interface gráfica intuitiva, em Java, usando as funcionalidades do *Swing toolkit*, a partir da que foi criada por Gonçalves et al. [15]. Pretende-se que esta interface permita visualizar a planta de um serviço hospitalar, preenchida com sensores e paredes, e o movimento de um ou mais objetos dentro dessa área em tempo real.

Relativamente aos sensores RFID, pretende-se que o simulador seja capaz de testar a sua posição, detetar anomalias decorrentes de uma má posição dos mesmos, avaliar a precisão do sistema de previsão em função destes e, dependendo do caso em estudo, detetar e corrigir erros previamente à instalação, alteração ou adição de novos componentes no sistema real. O objetivo da simulação neste contexto é evitar as claras desvantagens inerentes à realização de testes no próprio hospital, e que se podem resumir no uso desnecessário de recursos humanos e físicos. Outro parâmetro a ter em atenção será a localização ideal dos sensores dentro da área selecionada, uma vez que a localização de cada ponto de deteção de movimento influencia o custo global do sistema [1, 11].

Assume-se que o comportamento de objetos e pessoas em movimento segue uma certa tendência e é baseado em eventos, em vez de completa aleatoriedade, e por isso, pode ser previsto a partir de movimentos anteriores [16–21]. Como tal, pretende-se que o sistema seja capaz de prever localizações separadamente para cada entidade a simular, e que o faça com base

na análise de trajetórias anteriormente percorridas. Através das previsões de localização obtidas pretende-se calcular a precisão do modelo a cada nova trajetória simulada. Por outras palavras, o simulador deverá fornecer, no final de cada movimento, informação relacionada com a precisão do modelo utilizado, ou seja, dados quantitativos que reflitam o desempenho da previsão face à simulação através da comparação do caminho previsto com o caminho simulado.

A simulação consiste em criar condições ou hipóteses para estudar ou experimentar situações reais e os simuladores são usados para avaliar o impacto de novos processos ou *design* de equipamentos no desempenho geral de um sistema [22–24]. O modelo de simulação do posicionamento e movimento de pessoas ou equipamentos deverá ter em consideração o comportamento descrito por estas entidades na realidade, e deverá ser apresentado ao utilizador em tempo real de forma clara e intuitiva. Além disso, pretende-se que seja capaz de capturar constantemente as mudanças nas condições do ambiente e ser atualizado de forma a que essas mudanças sejam imediatamente comunicadas ao modelo de previsão e avaliadas da forma correta em tempo real. Para tal deverão ser criados dois agentes que comuniquem entre si durante todo processo de simulação e previsão. Como resultado final, espera-se que com base no algoritmo de Data Mining e na interação conjunta de dois modelos, representados pelos dois agentes com diferentes tarefas associadas - previsão e simulação - seja possível fornecer um modelo de previsão eficaz, realista e dinâmico na retribuição das localizações mais prováveis para uma determinada entidade ao longo do tempo.

No contexto do modelo de previsão em particular, e em modelos bioinformáticos no geral, a questão que se impõe é como avaliar o desempenho do sistema ou, mais concretamente, como comparar as localizações previstas com o caminho simulado. Muitos métodos de comparação centram-se na determinação de um valor único que represente uma medida de distância entre ambos. Aqui a solução mais óbvia consiste em calcular a percentagem de previsões corretas. A média desta percentagem tem sido usada na literatura [25, 26] como um indicador útil para avaliar o desempenho de modelos de previsão, e será usada neste trabalho nesse sentido.

Para garantir fiabilidade, segurança e fácil acesso aos dados, todos os resultados obtidos em cada simulação/previsão deverão ser armazenados numa base de dados MySQL. Para além das tecnologias já descritas será também utilizada a Interface de Programação de Aplicativos (Application Programming Interface - API) da ferramenta de Extração de Conhecimento Weka, e a plataforma Pentaho Plataform Designer para a criação de relatórios e estatísticas em diferentes cenários de análise. Os cenários deverão contemplar as diferentes trajetórias simuladas e o impacto das condições do ambiente na precisão do modelo de previsão.

O sistema integrado de simulação e previsão pretende servir como uma ferramenta de apoio em ambiente hospitalar em diferentes contextos, e vários benefícios podem ser retirados do uso da simulação de trajetórias, seja isoladamente seja integrada com um sistema de previsão. O ambiente de simulação estudado neste trabalho é um piso pediátrico hospitalar com um número variável de sensores RFID e um ou mais objetos observáveis. O simulador destina-se a hospitais como ferramenta de apoio à decisão em diferentes questões relacionadas com o fluxo de pessoas e previsão de trajetórias.

### 1.3 O Hospital

Uma vez que o simulador desenvolvido tem como referência um piso de um Hospital no Norte do país, convém fazer um ponto de situação sobre o sistema de *tracking* que atualmente se encontra em funcionamento nesse piso e as motivações que estão na origem da sua adoção.

O rapto de recém-nascidos é uma realidade e um fenómeno que acontece mais frequentemente do que geralmente se pensa [10]. No últimos 25 anos, 123 crianças, incluindo recém-nascidos, foram roubados de hospitais americanos, de acordo com dados estatísticos obtidos do Centro Nacional de Crianças Desaparecidas e Exploradas (NCMEC). Em Portugal, dois casos de rapto em menos de dois anos colocaram o Hospital considerado neste trabalho numa situação muito delicada [27]. Em 2006, uma bebé recém-nascida foi raptada da maternidade enquanto a mãe se encontrava a jantar na cantina do hospital. Em 2008 ocorreu uma situação semelhante quando uma mulher que se

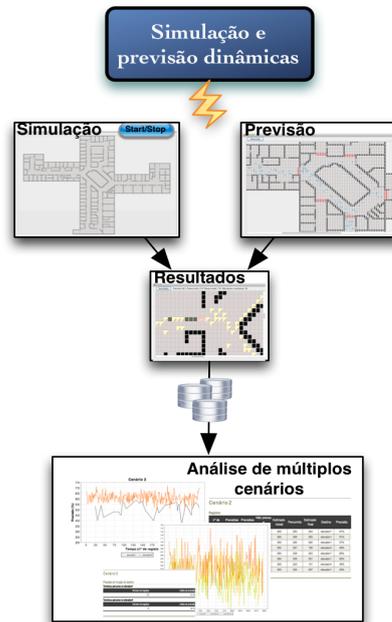


Figura 1.1: Ilustração do sistema proposto.

fez passar por enfermeira retirou o bebé recém-nascido dos braços da mãe e utilizou o elevador para fugir daquele hospital. A adoção dum sistema de monitorização constante e contínuo de recém-nascidos em ambiente hospitalar é um método de prevenção cada vez mais recorrente e que garante melhorar a sua segurança dentro do hospital [10]. Ciente disto, a administração do Centro Hospitalar em questão decidiu realizar procedimentos no sentido de garantir a segurança dos seus recém-nascidos através de uma tecnologia de monitorização baseada em RFID.

Uma plataforma para o *tracking* de pessoas e objectos em ambientes fechados através de um sistema de sensores e *tags* RFID encontra-se atualmente instalada no Hospital, e foi implementada em parceria com a Universidade do Minho em integração com a plataforma AIDA [28]. A plataforma encontra-se também integrada com outra denominada *BabyMatch*, que permite a localização e *tracking* de recém-nascidos e crianças através do uso de pulseiras com proteção anti abertura [28]. As pulseiras estão equipadas com baterias recarregáveis e são automaticamente activadas quando colocadas num bebé.

A cada recém-nascido é associada uma *tag* RFID que permite a sua identificação e localização e o acesso a outra informação que lhe esteja associada. Similarmente, cada médico, enfermeiro ou visita contém um cartão de identificação com *tag* RFID para que seja possível determinar quem entra ou sai da enfermaria a qualquer momento. Todos os eventos são registados numa base de dados que armazena um completo registo de todas as actividades que estão a decorrer. Através dos alertas gerados, o sistema deteta se o recém-nascido foi retirado do quarto por um enfermeiro ou por uma visita, ou ainda, por uma pessoa estranha.

O sistema que foi implementado no Hospital, propõe assegurar uma contínua monitorização de pacientes e *staff* especializado através da determinação da sua posição no espaço em qualquer momento. É baseado numa arquitetura multi-agentes e inclui mecanismos de lógica e tecnologias sensíveis a contexto para monitorizar pacientes, profissionais, recursos físicos e o ambiente que os circunda, e pode ser integrado com outros sistemas do hospital, nomeadamente para receber alarmes ou notificações. O sistema assenta na interação entre três módulos:

- Módulo do utilizador e do ambiente: refere-se aos pacientes, profissionais, recursos físicos e ao ambiente que os circunda. A informação obtida constitui os dados de entrada do sistema.
- Módulo de monitorização: garante uma monitorização eficiente através do uso de sensores e tecnologias sensíveis a contexto. A localização de pacientes, profissionais e recursos é garantida através do uso de sensores RFID.
- Módulo de lógica: a sua principal função é realizar ações de acordo com as condições verificadas sendo responsável por gerir a informação reunida nos módulos anteriores. Inclui os perfis de possíveis utilizadores: pacientes, médicos e enfermeiros são claramente identificados para ser possível que o sistema os localize e identifique inequivocamente com o objetivo de melhorar a segurança e a qualidade do serviço de saúde.

O objectivo máximo de um sistema deste género é que todos os pacientes,

profissionais e recursos sejam monitorizados e a sua localização seja sempre conhecida através dos sensores RFID que detetam proximidade das *tags*. Supondo que um paciente se move de um quarto para o outro, a vantagem do sistema é que permite detetar esse movimento e o envio de uma notificação ao médico responsável pelo paciente, com as coordenadas da sua localização. Quando o paciente regressa ao seu quarto uma nova notificação é enviada a avisar da corrente situação. O mesmo raciocínio pode ser aplicado aos equipamentos médicos. A monitorização do ambiente também é realizada através de sensores de deteção de movimento que possibilitam detetar a presença de elementos estranhos ao sistema que podem ser prejudiciais aos pacientes ou profissionais. O sistema consegue distinguir se a entidade detetada é um paciente, médico ou enfermeiro ou ainda uma visita. Diferentes níveis de permissão estão associados a diferentes espaços.

A figura 1.2 mostra a distribuição de sensores na planta do piso hospitalar. A vermelho representam-se os sensores de deteção de passagem e a azul os que detetam a proximidade das *tags* RFID. As áreas do hospital estão divididas por zonas, ligadas ao perfil do paciente, *staff* e visitas. As deteções de movimento entre zonas de *tags* associadas a perfis geram alarmes de diferentes tipos. O acesso de um paciente a uma ou mais zonas está limitado, e por isso um dos fatores implicado no alarme tem que ver com o estatuto de permissão em entrar numa zona particular que a pessoa detém. Por exemplo, se um paciente for hospitalizado devido a uma condição crítica que o impede de estar em contacto com outras pessoas, apenas médicos e enfermeiros podem entrar no seu quarto, pelo que se for detetada uma presença estranha é imediatamente enviado um alerta pelo sistema para que as ações devidas sejam realizadas para parar a ameaça. Os sensores RFID foram colocados em sítios estratégicos que pacientes, *staff* e visitas têm necessariamente de cruzar para mudar de zonas, assim como em algumas zonas centrais e pontos de cruzamento para controlar a sua presença. A arquitetura desenvolvida assenta numa plataforma OSGi que é uma aplicação dinâmica baseada em Java, que permite a adição de novas funcionalidades e dispositivos ao sistema em qualquer momento.

Dentro de um Sistema de Informações Hospitalares (Healthcare Informa-

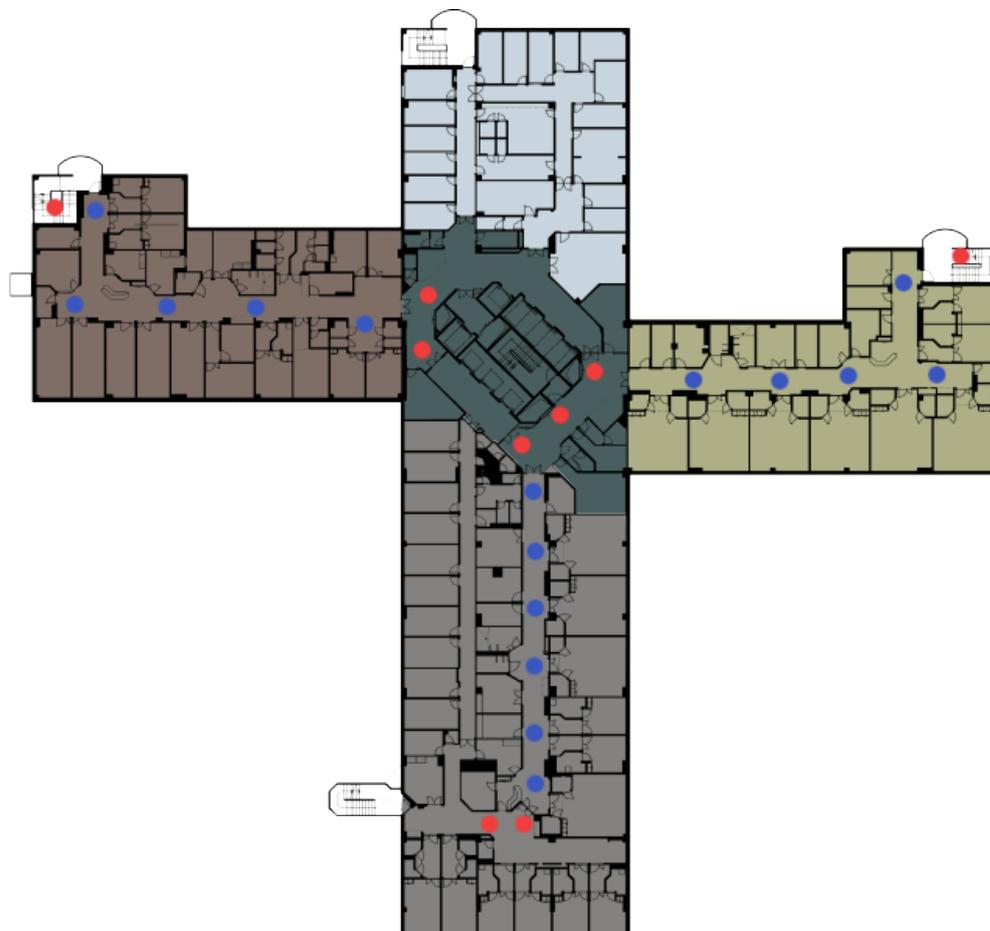


Figura 1.2: Planta do piso hospitalar.

tion System - HIS) existem diversos serviços que devem estar preparados para interoperarem com o sistema RFID, uma preocupação inerente aos hospitais que optam pela utilização desta tecnologia. O fluxo da informação deve ser consistente com os serviços já implementados, que vão desde os recursos humanos, ao sistema de monitorização de pacientes, registo clínico electrónico e sistemas de informação radiológica. A plataforma Agência para a Integração, Difusão e Arquivo de Informação Médica (Agency for Integration, Diffusion and Archive of Medical Information - AIDA), desenvolvida por Abelha et al. [30], é responsável pela integração e interoperabilidade entre diferentes serviços dentro de um hospital. A sua função consiste em disseminar informação entre os serviços referidos, e foi implementada com sucesso

em dois dos maiores hospitais no Norte de Portugal. Na entidade hospitalar considerada neste trabalho todos os serviços reportam a esta plataforma. Assim, a principal preocupação a ter em consideração na integração de um sistema neste HIS é a comunicação do mesmo com a AIDA, uma vez que esta já garante a sua interoperabilidade com todos os outros sistemas.

## 1.4 Simulação

A simulação consiste num ambiente dinâmico artificial, designadamente uma realidade alternativa, onde condições ou hipóteses são criadas para estudar ou experimentar situações reais através da representação de um ou mais objetos [22–24, 31, 32]. A simulação pretende fornecer os recursos necessários para prever como um determinado objeto se comporta no mundo real, permitindo retirar conclusões sobre o comportamento e viabilidade do sistema em estudo sem ser necessário a criação do mesmo. Na prática, os simuladores são usados para avaliar o impacto de novos processos ou *design* de equipamentos no desempenho geral do sistema.

Para a obtenção de resultados viáveis e exatos é muito importante que as características e comportamentos mais importantes do sistema sejam fielmente transpostos para o ambiente de simulação, e que ambos, ambiente de simulação e ambiente real, sejam o mais similares possíveis [31]. Os parâmetros do ambiente selecionados para a simulação, devem incluir não apenas o objeto que se pretende simular mas também o ambiente que o circunda. Além disso, um modelo de simulação deve ser capaz de capturar constantemente as mudanças nas condições do ambiente e ser atualizado de forma a que essas mudanças sejam avaliadas da forma correta em tempo real.

A simulação computacional deve o seu crescimento aos avanços no desempenho de sistemas computacionais sendo atualmente utilizada para modelar, avaliar e otimizar sistemas complexos, caros, perigosos ou simplesmente impossíveis de implementar num ambiente real, servindo também para simular cenários específicos que normalmente não ocorrem mas são possíveis [24, 31].

Algumas das aplicações mais comuns onde a simulação é atualmente empregue são a modelação de sistemas naturais (previsões meteorológicas,

evolução de tempestades, danos de terremotos), construção de edifícios (por exemplo arranha-céus e barragens) e teste e otimização de novas tecnologias [33]. A simulação é utilizada há cerca de 40 anos na área da saúde e é cada vez mais empregue na Educação e Formação, mais particularmente na área da intervenção cirúrgica e no apoio a práticas pedagógicas através da utilização de protótipos humanos. É aplicada neste contexto com o intuito de facilitar a aprendizagem, permitir que os alunos cometam erros num ambiente seguro e facilitar aceitação de novas técnicas, reduzindo ou eliminando o risco para o paciente [23]. Estudos comuns de simulação em ambiente hospitalar, e tal como enunciados por Benneyan [24], incluem também a realocação de camas no hospital, serviços ambulatoriais, cuidados intensivos, configurações de salas de emergência, ambulâncias e outros sistemas de emergência médica, fluxo de pacientes e tempos de espera. Para melhorar o desempenho de sistemas dinâmicos, a simulação tem também sido integrada noutras tecnologias para uma análise em tempo real. [33]

Antes de implementar uma arquitetura complexa, ou de impor modificações a sistemas já existentes, como é o caso de um sistema de *tracking* hospitalar, torna-se conveniente criar um ambiente de simulação que permita que o sistema seja testado e avaliado. Através da emulação da lógica e aleatoriedade subjacentes a um processo, tais como o fluxo de pacientes num determinado departamento e a duração aleatória do seu movimento, a simulação é útil para avaliar e comparar mudanças hipoteticamente formuladas ao sistema. A partir de um ambiente de simulação podem ser endereçadas questões do género - E se fosse colocado um sensor naquela posição? E se fosse inativado aquele sensor durante um certo período de tempo? - questões essas que podem conduzir a soluções ótimas sem as evidentes desvantagens associadas a conclusões obtidas à posteriori. Problemas que não foram antecipados podem ser expostos durante a simulação, evitando investimentos dispendiosos e demorados, já que os resultados obtidos por simulação muitas vezes contrariam a intuição inicial [24]. Uma outra vantagem da simulação reside no facto de permitir estudar dados que seriam difíceis de obter de outra forma, tal como a obtenção de padrões de movimento humanos. Além disso, permite estudar dados recolhidos artificialmente (a partir do programa), o

que poderia demorar muito tempo caso fossem recolhidos do processo real. A simulação pode ser utilizada para emular eficazmente sistemas grandes e complexos que geralmente não podem ser eficientemente avaliados usando outros métodos e pretende fornecer uma estimativa de uma medida do impacto de mudanças no sistema. [24]

Num simulador, a atividade do processo pode ser exibida à medida que a simulação se desenrola, oferecendo uma representação visual do sistema a quem não esteja familiarizado com o tema. Neste caso, uma vez que o simulador se destina a ser utilizado por profissionais em ambiente hospitalar, e o objetivo é fornecer em tempo real a previsão e simulação de trajetória, a interface gráfica é um parâmetro de grande importância.

Neste trabalho é proposto um sistema para a simulação em ambiente hospitalar diferente da generalidade de aplicações atualmente existentes nesta área. Apesar de o sistema de *tracking* de pacientes já se encontrar implementado no Hospital considerado, o simulador proposto pode ajudar a estudar o comportamento do mesmo sob diversos casos específicos, como por exemplo testar a variação do seu desempenho face à alteração da posição e número de sensores; detetar padrões anómalos na trajetória de determinado paciente através do mining das suas trajetórias anteriores; selecionar e ativar sensores com base no modelo de previsão, para diminuir consumos energéticos e aumentar o tempo de vida dos recursos; servir de apoio em cenários de alarme, por exemplo quando um sensor falha e perde o paciente de vista, o modelo de previsão pode permitir descobrir a sua localização.

Outra possível aplicação, facilitada pelo facto de o Hospital em questão promover o recurso a simulacros para avaliar a eficácia dos seus sistemas de segurança [27], consiste na simulação de percursos efetuados por diferentes grupos de pessoas em casos de emergência, através da análise e reprodução no sistema de comportamentos observados a partir de simulações reais, em que se regista o comportamento de um número relativamente reduzido de pessoas numa situação crítica, como por exemplo um incêndio ou uma ordem de evacuação. Nesta situação pretende-se reproduzir em ambiente simulatório a resposta das pessoas a tais cenários, nomeadamente o fluxo de pessoas nas saídas de emergência, extrapolando a situação concretizada na prática para

uma de dimensões maiores, em que um elevado número de pessoas circula nesse momento dentro do hospital. Uma vez que tal situação é difícil ou mesmo impossível de concretizar/simular na prática, o objetivo passa por, através da adição de várias entidades ao sistema que imitem os diversos percursos efetuados, preparar o Hospital para uma situação real.

Através da simulação de percursos efetuados por diferentes pessoas pertencentes a uma mesma categoria dentro de um hospital o sistema permitirá prever trajetórias não apenas de uma pessoa em particular, mas relativamente a um grupo, permitindo aprender comportamentos comuns e inerentes a profissionais de uma determinada área (enfermeiros de clínica geral, pediatras, radiologistas), a frequentadores do mesmo espaço (profissionais, pacientes e visitas do serviço de pediatria), a pacientes de uma ala específica (pacientes psiquiátricos), a deslocações típicas de equipamentos importantes, permitindo por um lado determinar as zonas que necessitam de um maior número de sensores RFID, seja de deteção ou/e de proximidade das *tags* e, por outro lado, associar a movimentação de equipamentos com situações que ocorrem frequentemente, como acidentes de viação em que várias pessoas dão entrada nas urgências de uma só vez, para prever os locais onde a utilização dos equipamentos mais requisitados é exigida, permitindo otimizar os locais onde estes são guardados.

## 1.5 Estrutura da Dissertação

A presente dissertação divide-se em seis capítulos: introdução, trabalho relacionado, arquitetura e implementação, simulador, resultados e por fim conclusões e trabalho futuro.

No primeiro capítulo é feita uma contextualização e enquadramento do problema e apresentam-se as motivações e os objetivos que estão na origem do trabalho desenvolvido. Além disso, é feita uma contextualização referente ao Hospital que está na base do projeto e à temática da simulação, referindo as vantagens e possíveis aplicações do simulador desenvolvido em ambiente hospitalar. Neste capítulo também é apresentada a estrutura da dissertação.

No segundo capítulo, referente ao enquadramento teórico, são abordados

os temas relacionados com o sistema proposto, necessários à compreensão do mesmo, e que resumem o trabalho de investigação realizado. Os principais temas abordados incluem Redes de Sensores de *Tracking* de Objetos, onde se realça a temática da previsão de trajetórias devido ao enorme potencial que demonstra na conservação de energia em sistemas que operam com este tipo de redes, como é o caso de sistemas RFID de *tracking* hospitalar. A tecnologia RFID é abordada sob este prisma sendo dado especial ênfase às suas aplicações em ambiente hospitalar. Por último é abordada a temática da Extração de Conhecimento de Bases de Dados, nomeadamente do processo de Data Mining, essencial ao modelo de previsão desenvolvido. Neste contexto, é explicada a técnica adotada para a previsão de trajetórias: *clustering* através do algoritmo SK-Means.

O terceiro capítulo visa apresentar a arquitetura do sistema desenvolvido bem como, as tecnologias utilizadas com recurso a esquemas que permitem uma melhor compreensão do mesmo.

No quarto capítulo apresenta-se o trabalho realizado no desenvolvimento do simulador. São explicados os resultados obtidos em termos de modelos de simulação e de previsão, o algoritmo de simulação de trajetórias de pessoas e a interface gráfica.

O quinto capítulo, referente aos resultados, apoia-se na realização de cinco diferentes cenários de simulação para o estudo do sistema sob diferentes prismas. São apresentados relatórios de precisão do modelo de previsão em diferentes circunstâncias de simulação.

Por último, no sexto capítulo são apresentadas as conclusões retiradas durante a realização do projeto e é feita uma consolidação daquilo que ainda pode ser realizado no futuro no sentido de melhorar o sistema proposto.



# Capítulo 2

## Trabalho Relacionado

Neste capítulo será introduzida a temática das Redes de Sensores, nomeadamente de Redes de Sensores de *Tracking* de Objectos, com destaque para o trabalho que tem sido realizado neste contexto, no âmbito da previsão de trajetórias. Redes de Sensores por RFID e a sua aplicabilidade e utilidade nos hospitais são assuntos que também serão abordados neste capítulo. Será ainda abordado o tema da Extração de Conhecimento de Bases de Dados, mais concretamente o processo de Data Mining. Pretende-se também, explicar em que consiste a técnica de Data Mining utilizada no simulador proposto, denominada *clustering*, e o respetivo algoritmo aplicado - SK Means.

### 2.1 Redes de Sensores

#### 2.1.1 Object Tracking Sensor Networks

O avanço da tecnologia em micro sensores, atuadores, processadores incorporados e RFID facilitaram o desenvolvimento de redes de sensores de grande escala, com aplicações em diversos campos incluindo a biologia, área militar e vigilância [2, 20, 33–36]. Os sensores são atualmente cada vez mais pequenos e com um maior poder de comunicação, sendo capazes de detetar o estado de um determinado objeto e ao mesmo tempo transferir toda a informação que lhes chega para plataformas ou bases de dados através de tecnologias sem fio [5, 20, 37, 38].

Para a implementação de uma rede de sensores sem fios, vários nós de sensores estacionários são distribuídos ao longo da área de interesse, designada por região de monitorização. Um nó de sensor tem a responsabilidade de monitorizar o objeto que penetra na sua área de deteção e reportar o estado de objetos em movimento com uma certa frequência de comunicação, que é ajustável à rede e aos requisitos da aplicação. Cada um destes nós é dotado de funções de computação, deteção e comunicação e possui conhecimento prévio sobre a sua localização dentro da rede, o que lhes permite determinar a posição do objeto [1, 19, 34]. Em algumas redes, determinar o estado de um objeto em movimento pode implicar a operação conjunta de vários nós de sensores. Apesar de ser uma questão pertinente, está fora do âmbito deste estudo, pelo que será considerado que cada nó de sensor consiste numa representação lógica de um conjunto de nós de sensores que conjuntamente determinam o estado/posição do objeto móvel. Além dos nós de sensores existe também uma estação base que atua como uma interface entre a rede de sensores e as aplicações, dando instruções e recolhendo os dados de interesse.

Diversos autores têm estudado nos últimos anos os assuntos relacionados com os sistemas de *tracking* de objetos em redes de sensores, uma temática bastante recorrente na literatura [1, 5, 19, 20, 34, 37, 38]. As redes de sensores em que os sensores são capazes de detetar a posição de um objeto em tempo útil são denominadas Redes de Sensores de Tracking de Objetos (Object Tracking Sensor Networks - OTSNs).

As OTSNs são atualmente aplicadas em diversas áreas, podendo-se destacar a gestão de cadeias de fornecimento, monitorização em serviços de saúde, deteção de animais, *tracking* de alimentos e prevenção de acidentes, por exemplo: deteção de pacientes e equipamento médico; deteção de explosivos num edifício; deteção de alimentos fora do prazo de validade. Espera-se que tais aplicações sejam cada vez mais utilizadas na vida real, verificando-se que aparecem frequentemente em contextos até então inacessíveis [1].

Uma vez que os atuais sensores são suportados por baterias, que podem ser difíceis de substituir, um dos problemas mais críticos relacionados com as OTSNs e que tem atraído muita atenção nos últimos anos devido

às suas vastas aplicações é o consumo de energia que afeta o seu tempo de vida [16, 19, 20, 26, 34, 37, 38]. Um consumo mais baixo de energia, por parte destes sensores, significa um prolongamento do tempo de vida do sistema [16, 20, 34]. Diversos estudos têm sido realizados relativamente à eficiente utilização de energia em *tracking* de objetos em redes de sensores. Estes estudos podem ser classificados em duas categorias. Um deles foca-se na redução dos custos de comunicação, isto é, no número de mensagens que são transmitidas dentro da rede para atualização da posição do objeto em movimento [19, 38]. No entanto, enquanto que os componentes de computação e de sensores consomem menos energia do que os componentes de comunicação, não deixam de ser importantes fontes de dissipação de energia, especialmente depois dos custos de comunicação serem otimizados, o que constitui um problema [37]. A segunda categoria centra-se em localizar os objetos de uma forma energeticamente eficiente [19, 34, 37]. Adicionalmente, na maioria dos estudos realizados os autores apostam numa melhoria do *design* do hardware de sistema, e poucos consideram uma abordagem por software [16, 37].

Um recorrente método utilizado para evitar maiores gastos energéticos consiste em organizar a rede de sensores segundo uma estrutura hierárquica, uma vez que certas áreas podem estar sujeitas a uma maior frequência de entrada e saída de objetos do que outras, pelo que deverão existir mais sensores à volta dessas áreas [16]. Uma abordagem mais intuitiva consiste na inativação, ou redução dos tempos de operação dos componentes de alta potência, dos nós que não estão a ser usados [34, 37]. No entanto, apesar desta estratégia permitir uma redução dos consumos energéticos pode fazer com que os OTSNs percam o objeto de vista, pelo que é necessário um estudo preciso do sistema antes da adoção desta estratégia. Para cumprir tal objetivo diversos autores têm apostado em esquemas baseados em previsão desenvolvidos com o intuito de reduzir o número de nós ativos durante o *tracking* [16, 19–21, 34, 37]. Um sistema de sensores em rede que seja capaz de prever com precisão a trajetória dos objetos, que pretende monitorizar, possibilita a ativação dos nós mais próximos da área da previsão, reduzindo o número de nós ativos durante o *tracking* e consequentemente o consumo

energético da rede. Uma vez que o mecanismo de previsão tem um grande impacto no desempenho do *tracking*, a questão de como prever a localização de um objeto em movimento é da maior importância [34].

Neste trabalho, o modelo de previsão de trajetórias proposto, que funciona através de padrões de movimento obtidos por simulação, pretende ser utilizado sob o ponto de vista do aumento da eficiência energética para a área selecionada.

### 2.1.2 Previsão de trajetórias em OTSNs

Diversos estudos têm sido realizados no contexto da previsão de trajetórias com base em padrões de movimento, através da aplicação de algoritmos de Data Mining. De seguida, referem-se alguns deles e a sua aplicabilidade e utilidade nas OTSN's.

Chen et al. [21] propõem um modelo para a previsão do destino e futura trajetória de uma pessoa de forma integrada, através do *mining* de dados de movimento recolhidos por GPS. Para a determinação dos pontos de origem e destino foi usado o algoritmo de *clustering Forward-Backward Matching* (FBM) e para a extração de padrões de movimento o *Continuous Route Pattern Mining* (CRPM). O modelo demonstrou uma precisão de 60% na previsão de trajetórias.

Lin et al. [16] e Tseng and Lin [20] propõem um modelo de previsão de padrões de movimentos temporais em OTSNs assente na aplicação de um algoritmo de Data Mining chamado STMP-Mine. O modelo tem como principal foco o problema da conservação de energia nestas redes e pretende utilizar os padrões obtidos para a redução de erros de previsão. Os resultados comprovam que o modelo fornece um bom desempenho em termos de escalabilidade e eficiência energética do sistema.

O estudo realizado por Xu et al. [37] insere sobre a temática da poupança de energia em OTSNs e fornece um esquema que inclui modelos de previsão, mecanismos de recuperação e mecanismos de acordar nós de sensores que se encontram num modo de baixo consumo. Os autores propõem um modelo de previsão denominado *Prediction-based Energy Saving scheme*

(PES), que pretende minimizar o número de nós que participam nas atividades de *tracking* através da desativação de nós que não sejam necessários. Esses nós deverão ser colocados num estado adormecido com o objetivo de, tal como o nome indica, fornecer uma solução para a poupança de energia. Os mesmos autores, em [19], propõem um novo modelo baseado em previsões para realização de relatórios energeticamente eficientes em redes de sensores, através da minimização do número de transmissões de longa distância entre os nós de sensores e a estação. Os autores adoptam um mecanismo denominado *Dual Prediction-based Reporting* (DPR) em que os nós dos sensores e a estação base fazem a (mesma) previsão de futuros movimentos dos objetos móveis, baseados no histórico de movimentos desse objeto e através de modelos matemáticos. As transmissões de leituras dos sensores são evitadas desde que as previsões sejam consistentes com o movimento real do objeto. Através da monitorização dos objetos, os nós dos sensores são capazes de determinar se as previsões são consistentes com o estado real do movimento do objeto e agir em consonância: se forem consistentes nenhuma atualização deve ser comunicada à estação, caso contrário os nós dos sensores deverão corrigir as previsões erradas. O esquema de monitorização adoptado - *Continuous Monitoring Scheme* (CM) - ativa um nó de sensor apenas quando um objeto entra na sua área de deteção e mantém-no ativo até que o objeto a abandone. Todos os nós que não tenham nenhum objeto na sua zona de deteção mantêm-se inativos. O modelo garante uma considerável poupança de energia sob várias condições.

Dois esquemas de classificação, avaliados e comparados através de 3 algoritmos de DM, são introduzidos por Anagnostopoulos et al. [26]. Dos algoritmos testados - J48, Naive Bayes e JRip - o que obteve melhores resultados em termos de precisão da previsão foi o J48, com uma precisão média a oscilar consoante o grau de aleatoriedade associado ao movimento: 35% de precisão para movimentos definidos apenas por aleatoriedade; 45% de precisão para movimentos com 50% de aleatoriedade e 95% para movimentos determinísticos (trajetórias regulares).

### 2.1.3 Redes RFID

A Identificação por Rádio Frequência, ou RFID, consiste num método que através da emissão de ondas rádio possibilita a identificação e localização automática de objetos à distância, e o armazenamento e retribuição de dados remotamente [4, 8, 11, 39].

Apesar das diversas aplicações RFID existentes, a arquitetura de um sistema RFID é sempre composta por três componentes básicos: etiquetas ou *tags* RFID colocadas no objeto a identificar, leitores, ou sensores, para a extração da identificação das *tags* e sistemas de *backend* capazes de gerir a informação gerada durante o processo [10, 14, 40–42]. A rede de Código Eletrónico de Produto (Electronic Product Code - EPC) é atualmente o exemplo mais conhecido de sistema *backend*, responsável por filtrar as mensagens recebidas através do seu *middleware*<sup>1</sup> e tratar a informação necessária para fornecer serviços [40]. A Figura 2.1 mostra a arquitetura básica de um sistema RFID.

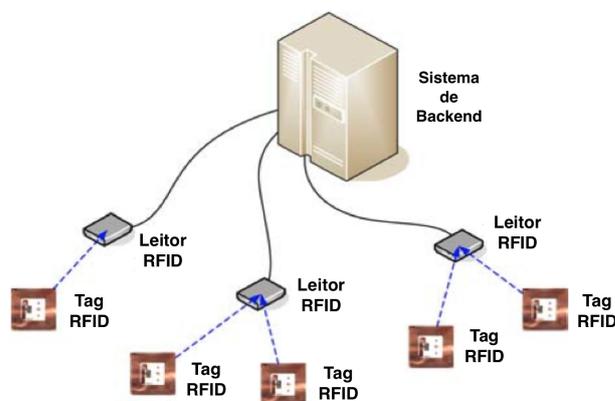


Figura 2.1: Arquitetura de um sistema RFID. Adaptado de [40]

As etiquetas são constituídas por um circuito integrado de chips e antenas que lhes conferem uma dupla funcionalidade: receber o sinal dos leitores e transmitir a sua informação de identificação de volta para o leitor [40, 42]. Ao

---

<sup>1</sup>Middleware ou mediador, é um programa responsável por fazer a mediação entre o software e as aplicações.

anexar estas *tags* aos equipamentos, e dando ao pessoal autorizado emblemas etiquetados, os sistemas de computador conseguem mostrar a sua localização dentro do edifício, tornando mais fácil localizar o objeto ou o membro do *staff* em qualquer ala, laboratório ou escritório onde se encontrem. A vantagem das *tags* RFID é que estas usam um dispositivo de memória que lhes permite armazenar uma quantidade significativa de dados relativos ao objeto que lhes está associado, que podem ser usados para fornecer listas de equipamentos, localização, estado e histórico. Os dados podem também ser utilizados para realçar se um item abandonou uma área onde era suposto permanecer, prevenindo que equipamentos ou documentos críticos deixem um local [7].

Os leitores RFID são implantados em sítios estratégicos para detetarem o aparecimento, desaparecimento ou outros comportamentos das *tags* RFID. A informação passada pelas *tags* aos leitores é convertida numa mensagem que o leitor por sua vez transmite ao sistema de *back-end* [40, 41].

Uma vez que as *tags* RFID são constantemente interrogadas pelo leitor, o sistema RFID gera uma enorme quantidade de dados que precisa de ser tratada. O *middleware* existente entre os leitores e as aplicações, é responsável por gerir estes dados. Todos os pedidos feitos pelas aplicações aos leitores passam pelo *middleware*, enquanto que os identificadores únicos de *tags* e outras possíveis informações recebidas pelos leitores são passadas a outros serviços externos ou a aplicações. As funções fundamentais do *middleware* incluem filtragem, contagem e agregação de dados das *tags*, permitindo o tratamento da larga quantidade de dados gerada pelo sistema RFID [40].

Contrariamente aos códigos de barras, as *tags* RFID podem ser lidas à distância, não sendo necessário estarem na linha de visão do leitor. Além disso, um sistema RFID é capaz de detetar a presença e distinguir diferentes *tags* RFID na mesma área, de forma autónoma (sem a ajuda de pessoas) [8, 11, 40]. Devido a estes factores, é hoje em dia uma tecnologia dominante em diferentes áreas. De facto, as aplicações RFID estendem-se às áreas de controlo de tráfego, indústria farmacêutica, manufatura, transportes, logística, construção, ambientes clínicos, controlo de gado, requisição de livros em bibliotecas, entre outras [11, 41, 43]. A sua popularidade foi no entanto tardia, devido em grande parte ao seu custo comparativamente com

os códigos de barras [10, 11, 40].

Os sistemas RFID representam uma mudança significativa nas aplicações de *tracking* podendo ser utilizados para o *tracking* de objetos e pessoas em todo o mundo. As mais variadas indústrias podem beneficiar grandemente com a implementação de sistemas RFID no *tracking* dos seus ativos, seja com a redução de investimentos em cadeias de gestão seja com o aumento da qualidade dos seus serviços [10, 11, 40, 41].

Na construção de um sistema RFID de *tracking* de objetos é fundamental testar a arquitetura da rede previamente à implementação, já que o modo como esta é desenvolvida afeta o desempenho operacional e a escalabilidade de todo o sistema. Uma estrutura hierárquica demonstra resultados positivos, nomeadamente uma boa escalabilidade para expandir o sistema e uma elevada eficiência operacional, como demonstram alguns estudos [40].

Existem dois tipos principais de sistemas de *tracking* RFID:

***tracking* ativo:** inclui *tags* e leitores RFID ativos. A *tags* RFID ativas obtêm energia a partir de uma fonte de energia interna (baterias), para transmitirem constantemente sinais rádio aos leitores ativos que se encontram a algumas dezenas de metros de distância, permitindo que a monitorização seja contínua. São muito maiores, mais caras e têm maior alcance do que as *tags* passivas. O seu custo varia entre os 10 e os 150 € e o alcance vai desde os 30 aos 90 metros. Para realizar o *tracking* de pacientes ou de material caro e de grandes dimensões o *tracking* ativo RFID é uma das soluções mais apropriadas [4, 8, 9, 11, 44, 45].

***tracking* passivo:** inclui *tags* e leitores RFID passivos. Neste caso, as *tags* não têm qualquer fonte de energia interna. Em vez disso, a energia utilizada para transmitir a informação para o leitor é aquela que é emitida pelo leitor passivo quando procede à leitura da *tag*. Contrariamente às *tags* ativas, têm um alcance de deteção reduzido (menos de 1 metro) e custam menos de 1 €. São indicadas para controlo de acesso, prevenção de roubos e inventário em objetos pequenos de baixo custo [4, 8, 9, 11, 44, 45].

### 2.1.4 RFID nos Hospitais

Até agora, o uso de RFID nos hospitais tem estado limitado às aplicações onde o uso de códigos de barras não é adequado, como é o caso do *tracking* de pacientes, já que, ao contrário dos códigos de barras, o RFID permite localizar e identificar objetos à medida que estes se movem. No entanto, apesar de alguns hospitais se mostrarem relutantes em prescindir dos investimentos realizados em códigos de barras para a adoção de novas tecnologias - RFID, cartões inteligentes e biometria - prevê-se que no futuro, soluções baseadas nas duas tecnologias se complementem em termos de funcionalidade, custos e usabilidade. Prevê-se que enquanto a tecnologia RFID evoluir o seu uso na medicina vai continuar a crescer, principalmente nas áreas de manutenção de inventários, identificação de pessoal médico e pacientes, e *tracking* de equipamentos e pessoas [9, 44].

A inexistência de um sistema de tracking de equipamentos em ambiente hospitalar, pode conduzir a situações em que equipamentos essenciais ao tratamento do paciente não sejam encontrados de forma atempada, o que por sua vez pode resultar numa qualidade de serviço fraca, baixa satisfação do paciente, perda de clientes, perda de receita e, mais tragicamente, na perda de uma vida humana [11]. A monitorização em tempo real surge como uma questão vital neste contexto, e os provedores de serviço de topo na área da saúde demonstram os benefícios do RFID na resolução destes problemas, lançando esta tecnologia para um plano de crescente destaque em hospitais [4, 8, 9, 13, 14].

Diversas aplicações RFID foram desenvolvidas para melhorar a prestação de cuidados de saúde e o bem-estar dos pacientes nos hospitais, e para fazer face aos problemas referidos [4, 8–11, 14, 44]. Entre elas destacam-se:

- Melhoria das aplicações que usam códigos de barras, por exemplo: transfusão de sangue, controlo de produtos farmacêuticos, identificação de amostras;
- Aplicações para fazer a monitorização de pacientes idosos ou desorientados;

- Aplicações para fazer a monitorização de recém-nascidos, prevenindo que estes sejam raptados ou trocados à nascença;
- Aplicações para pacientes cirúrgicos que garantam que o procedimento cirúrgico está a ser realizado no paciente certo à hora certa;
- Identificação de pacientes através de pulseiras, que quando lidas pelo leitor RFID indiquem o número de identificação, nome, data de nascimento, pedidos de admissão, informação relativa ao seguro de saúde, reações alérgicas, medicação prescrita, tipo de sangue, entre outros parâmetros;
- *Tracking* e manutenção de equipamentos. Os equipamentos incluem dispositivos médicos, como bombas de infusão, máquinas de raio-x portáteis e reguladores de oxigénio e outros bens móveis, como cadeiras de rodas e macas;
- Aplicações que usam dispositivos RFID implantáveis, que funcionam como registos médicos portáteis para os pacientes;
- *Tracking* de médicos, enfermeiros e pacientes em qualquer parte do hospital, através de crachás e cartões de identificação;
- gestão de stocks;
- gestão de arquivos médicos;

O *tracking* eficiente de equipamentos, através de sistemas baseados em RFID, permite aumentar a segurança, reduzir custos e melhorar os cuidados de saúde [9, 10]. Por um lado, ocorre uma redução no tempo de procura de equipamentos por parte do *staff* e conseqüentemente um aumento da sua produtividade. Por outro lado, ocorre uma redução significativa dos custos associados à perda e roubo de equipamentos o que leva a uma maior utilização dos mesmos [4, 8, 11]. É de salientar, que os hospitais gastam milhares de euros em equipamentos que são perdidos, roubados ou colocados em sítios errados, todos os anos [9].

Qu et al. [4] propõem um modelo para quantificar os benefícios do RFID em meio hospitalar no que diz respeito ao aumento da utilização e à redução do tempo de procura de equipamentos por parte do *staff*. O objetivo dos autores passou essencialmente por investigar e avaliar as melhorias no desempenho de um sistema de *tracking* de equipamentos após o uso desta tecnologia. Os resultados obtidos comprovaram a eficácia desta tecnologia, ao mostrar que ocorreu um aumento significativo da utilização de equipamento após a implementação do sistema.

Em 2004, num contrato de 5 anos, a Agility Healthcare Solutions – companhia que fornece soluções RFID para a indústria da saúde – ficou encarregue de desenvolver, operar e monitorizar uma rede RFID de *tracking* de 10000 peças de equipamento, em três hospitais da Virgínia. Antes de implementar a rede, a companhia realizou um estudo de vários meses da situação corrente dos hospitais em termos da sua capacidade de *tracking* de ativos. O estudo mostrou que a empresa que gere os hospitais teria o retorno do investimento realizado no primeiro ano de operações. A rede permitiu que cada hospital fosse capaz de determinar a posição exata de todos os equipamentos médicos etiquetados, e de outros bens móveis. O sistema também permitiu mostrar o estado de cada equipamento – em uso, disponível ou a precisar de manutenção – em tempo real. Centenas de leitores, com recursos embutidos para fazer a conexão a uma rede sem fios e encaminhar os dados para uma plataforma de gestão, foram colocados ao longo das três unidades de saúde. Ficou comprovado que o tempo despendido pelos enfermeiros na procura desses mesmos equipamentos sofreu uma queda de meia hora por turno por enfermeiro [45, 46].

No entanto, apesar das claras vantagens associadas à implementação de sistemas baseados neste tipo de tecnologia em ambiente hospitalar, o investimento por parte dos hospitais em soluções baseadas em RFID tem sido relativamente modesto até à atualidade, devido sobretudo aos ainda elevados custos de implementação [11, 44].

## 2.2 Extração de Conhecimento de Bases de Dados

### 2.2.1 Data Mining

Com a generalização da Internet para o público em geral, por volta do ano 2000, surgiu um enorme progresso na disponibilidade de informação e nas tecnologias de comunicação, que desempenharam um importantíssimo papel no desenvolvimento de metodologias de armazenamento de dados. A grande quantidade de dados disponíveis e um fraco entendimento humano dos processos que os geraram, levaram cientistas e pesquisadores a debruçarem-se sobre a temática da Descoberta de Conhecimento em Bases de Dados (Knowledge Discovery in Databases - KDD), nomeadamente no desenvolvimento de técnicas de Data Mining que permitissem extrair padrões importantes de grandes conjuntos de informação [35, 47, 48]. A explosão no crescimento das bases de dados e o acesso generalizado a grandes repositórios de dados estão na origem da necessidade de desenvolvimento de técnicas que consolidassem informação e conhecimento de forma inteligente. A área de pesquisa de Técnicas de Data Mining (Data Mining Techniques DMT), que surgiu por volta de 1960 como um ramo aplicado da Inteligência Artificial (AI) [47] é alias uma área cada vez mais importante no mundo científico. Para melhorar tanto a produtividade como a qualidade de software, engenheiros informáticos aplicam algoritmos de Data Mining em várias tarefas de engenharia de software [35].

KDD é um campo interdisciplinar que tem crescido fortemente nos últimos anos a partir da interseção de técnicas de gestão de bases de dados, visualização de dados, reconhecimento de padrões, aprendizagem de máquina, algoritmos matemáticos, sistemas inteligentes e estatística para a extração e identificação de informação que permanece escondida em grandes volumes de dados. [36] É uma tecnologia motivada pela necessidade de novos métodos que auxiliem a análise, entendimento e visualização de grandes quantidades de dados obtidos a partir de aplicações, de forma autónoma, por forma a revelar padrões valiosos nos mais diversos contextos. O objetivo passa

por fornecer um conjunto de metodologias para a resolução de problemas, análise, planeamento, diagnóstico, deteção, integração, prevenção, aprendizagem, inovação e apoio na tomada de decisão.

Data Mining e Extração de Conhecimento são conceitos muitas vezes confundidos e utilizados com o mesmo significado. No entanto, convém salientar que Data Mining, apesar de ser o cerne de todo o processo de KDD, constitui apenas um dos estágios envolvidos no mesmo [18, 49]. De facto, KDD é um processo que inclui diversas etapas, que deverão ser seguidas se se pretender obter um modelo de extração de conhecimento eficiente. Os estágios que constituem o processo de KDD estão presentes na figura 2.2 e são enunciados e explicados de seguida:

**Seleção de dados:** determinação do conjunto de amostras apropriadas para a consulta;

**Pré processamento de dados:** inclui o estágio de limpeza, em que ocorre uma eliminação de dados com anomalias, existindo a possibilidade de completar dados em falta e o estágio de redução de dados para casos em que se considere que uma redução no número de dados e/ou variáveis não vai influenciar os resultados de análise, o que permite uma redução do tempo de análise;

**Transformação:** diferentes tipos de dados obtidos a partir de diferentes bases de dados são convertidos num único tipo;

**Data Mining:** aplicação das DMT aos dados finais;

**Avaliação:** o conhecimento extraído é analisado de acordo com critérios de novidade, validade e utilidade. A avaliação é feita sobre um conjunto de teste que não foi utilizado no treino do algoritmo de Data Mining. O objetivo é aplicar os padrões que foram obtidos no estágio anterior ao conjunto de teste, e comparar os resultados pretendidos com os resultados reais.

O processo de Data Mining consiste basicamente na procura, de forma autónoma, de padrões que se mantêm escondidos em bases de dados. Pode

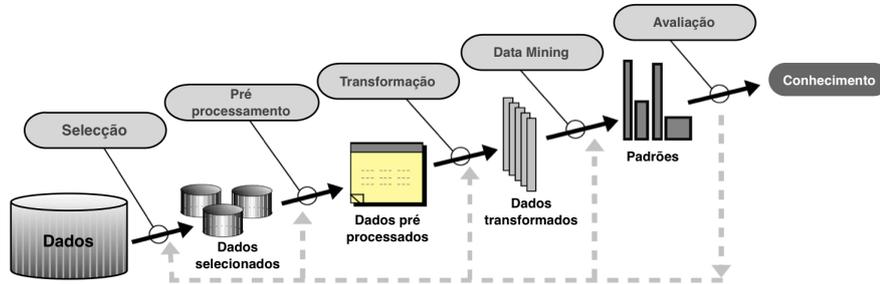


Figura 2.2: Estágios que constituem o processo de Extração de Conhecimento em Bases de Dados. Adaptado de [49]

também ser definido como a extração automática de conhecimento potencialmente útil e interessante de grandes coleções de dados armazenados em bases de dados, *warehouses* ou outros repositórios de informação, através da aplicação de algoritmos que produzam padrões ou modelos sobre os dados. [49] O Conhecimento descoberto por Data Mining (Data Mining Knowledge - DMK) está implícito nos dados e pode tomar a forma de padrões, associações entre diferentes características, mudanças, anomalias, regras, estruturas e tendências em dados temporais [36, 50].

Dependendo daquilo que é pretendido para o contexto da aplicação, os utilizadores estão geralmente habilitados para determinar a técnica de Data Mining que mais se adequa ao problema e ao conjunto de dados, podendo desta forma obter os padrões mais úteis e significativos para si [18, 36, 50].

Recentemente, Data Mining tem-se tornado numa técnica muito popular de extração de informação em diferentes áreas, devido sobretudo à flexibilidade que apresenta em operar com qualquer tipo de base de dados. É utilizado como ferramenta de apoio à decisão em variadas aplicações de negócios na medida em que fornece diferentes estratégias e modelos de previsão e descrição, inferidos a partir da observação de comportamentos reais, que podem ajudar empresas a optarem por melhores decisões e a manterem-se competitivas no mercado de trabalho [51]. Um grande número de aplicações baseadas em Data Mining têm sido desenvolvidos para as áreas de marketing, finanças,

atividade bancária, manufatura, educação e saúde [49, 52].

As DMT são aplicadas em diferentes tipos de dados, como séries temporais e espaciais, telecomunicações, Web e dados multimédia. Logicamente, todo o processo de Data Mining, incluindo a técnica a ser aplicada, depende muito do contexto da aplicação em que se insere e da natureza dos dados disponíveis [36].

Um modelo de Data Mining é obtido através da aplicação de um algoritmo de Data Mining num conjunto de dados de treino. Muitas abordagens, métodos e algoritmos têm sido desenvolvidos neste âmbito. As maiores funções de Data Mining desenvolvidas nas comunidades comerciais e de pesquisa incluem a classificação, caracterização, regras de associação, regressão e *clustering* [18, 47–49, 53, 54]. Estes métodos são sucintamente descritos de seguida:

**Classificação:** consiste no processo de descobrir um modelo ou função que descreva e distinga classes de dados, com o objetivo de ser capaz de usar o modelo para prever a classe de objetos cuja classe é desconhecida. O modelo funciona através de observações obtidas num conjunto de dados de treino (isto é, dados cuja classe é conhecida), e pressupõe o conhecimento prévio das possíveis classes. As regras de classificação são consideradas um caso particular de regras de previsão *if-then* (se-então), em que o antecedente *se* contém uma conjunção de condições para prever atributos e o conseqüente *então* o valor da previsão para o atributo. Dentro das técnicas de classificação encontram-se: árvores de decisão, análise de regressão, redes neuronais artificiais, máquina de vetores de suporte, algoritmo Naive Bayes, algoritmo K-nearest neighbor e algoritmos genéticos. Existem diversas aplicações para este tipo de abordagem. Um simples exemplo onde a classificação pode ser aplicada é na classificação de um potencial cliente entre as categorias de excelente, bom ou mau.

**Associação:** análise por associação centra-se em descobrir relações entre observações e determinar quais delas podem ser realizadas em conjunto. Por outras palavras, associação procura a identificação de factos que

possam estar direta ou indiretamente associados. Pode ser útil, por exemplo, na avaliação da existência de relações temporais entre dados.

**Regressão:** consiste na procura de uma função que represente, de forma aproximada, o comportamento apresentado pelo fenómeno em estudo. O objetivo é descobrir uma função que mapeie um item numa variável de previsão de valor real. A função de regressão mais conhecida é a regressão linear.

**Clustering:** É o nome dado ao processo de partição de um conjunto de dados em grupos (*clusters*) mais pequenos, consoante alguma noção de similaridade que apresentem entre si. Será aprofundado no próximo capítulo.

As funções de Data Mining baseiam-se em dois tipos de aprendizagem: supervisionada, ou direta, e não supervisionada ou indireta [26]. A diferença reside na forma como a avaliação é realizada: na primeira a avaliação é feita recorrendo a medidas externas aos dados, contrariamente ao que acontece na segunda em que a qualidade é avaliada sem o recurso a medidas externas. Funções de aprendizagem supervisionada são tipicamente usadas para prever um valor e são às vezes referidas como modelos de previsão, incluindo a classificação e a regressão. Técnicas não supervisionadas são tipicamente utilizadas para detetar a estrutura intrínseca, relações ou afinidades entre os dados. Estas técnicas são às vezes referidas como modelos descritivos e incluem *clustering* e modelos de associação.

O objetivo final consiste em permitir que modelos de Data Mining sejam construídos para serem usados numa variedade de tarefas de previsão e análise e para partilhar esses modelos com outras aplicações para além das bases de dados.

### 2.2.2 Clustering

*Clustering*, ou análise por *clusters*, pertence ao domínio de análise de dados e aprendizagem de máquina [51]. É o nome dado ao processo de partição

de um conjunto de dados em grupos (*clusters*) mais pequenos, consoante alguma noção de similaridade que apresentem entre si. Esta divisão é feita de modo a que objetos ou padrões similares sejam atribuídos ao mesmo *cluster* e padrões pertencentes a *clusters* diferentes sejam dissimilares entre si [51, 54–56]. O objetivo da análise por *clusters* é a classificação de objetos em subgrupos que desempenham um papel no contexto do problema em particular.

O *clustering* pode ser usado para detetar grupos naturais e identificar estruturas abstratas que possam residir no conjunto de dados, sem qualquer conhecimento básico prévio das características dos dados. Diversos métodos de *clustering* têm sido desenvolvidos. Existem duas principais categorias de processos de *clustering* de acordo com a forma como organizam os dados: hierárquicos e não hierárquicos [54]. Modelos hierárquicos são caracterizados por obter a distância entre objetos através de uma medida de distância, que é geralmente a distância euclidiana. O processo começa com todos os objetos representados como *clusters* separados e, consoante a distância calculada entre eles, dois objetos próximos juntam-se para formar um novo *cluster*, que representa um novo objeto. O processo termina quando todos os *clusters* estiverem formados, ou seja quando o número de *clusters* requisitado seja alcançado. No *clustering* não hierárquico a partição das observações ocorre em  $k$  *clusters*. Cada observação é atribuída ao *cluster* mais próximo e a média de cada *cluster* é recalculada. O SK-Means insere-se nesta categoria.

A maioria dos métodos de *clustering* têm sido desenvolvidos neste enquadramento nas últimas décadas para diversas aplicações, incluindo Data Mining, bioinformática, redes neuronais, Inteligência Artificial, estatística e recuperação de informação [56–58].

Existem diferentes algoritmos de *clustering*, que diferem entre si nas propriedades dos *clusters* calculados. A chave para perceber o funcionamento dos diferentes algoritmos é entender os *clusters* por eles obtidos. Assim, consoante o tipo de *clusters*, os algoritmos tradicionais podem ser definidos em:[54]

**Modelos de centróides:** Consiste na representação de um cluster por um

único vetor. Por exemplo, o Simple K means.

**Modelos de distribuição:** *clusters* são modelados usando distribuições estatísticas.

**Modelos de densidade:** *clusters* são vistos como regiões de densidade no espaço, ligadas entre si.

**Modelos de subespaço:** *clusters* são modelados tanto com os membros do cluster como com os atributos relevantes.

**Modelos de grupos:** não fornecem um modelo refinado, apenas informação agrupada

**Modelos baseados em gráficos:** um clique (subconjunto de nós num gráfico, tal que, cada dois nós no subconjunto são interligados) pode ser considerado como uma forma protótipal de *clustering*.

De acordo com Pandey [54], 4 das mais utilizadas técnicas de *clustering* inseridas nestes modelos são: K-means, Fuzzy C-means, *Clustering* Hierárquico e Mistura de Gaussianos.

Tipicamente, os dados necessitam de ser transformados antes de sofrerem *clustering*, geralmente porque diferentes atributos podem estar medidos em escalas diferentes, por exemplo em diferentes unidades de distância (milímetros e metros). Nestes casos, escalas diferentes podem contaminar os resultados de *clustering* e os dados devem ser padronizados para que todos os atributos estejam na mesma escala. Outra prática recorrente consiste em descartar variáveis que apresentem pouca variação, ou que têm uma grande correlação com outras variáveis, por forma a reduzir as dimensões do *clustering*.

Em alguns problemas de *clustering* o número de *clusters* a calcular tem de ser fornecido antecipadamente, como é o caso do Simple K-Means.

### 2.2.3 Algoritmo Simple K-Means

O Simple K-Means, ou abreviadamente SK-Means, é um algoritmo clássico de aprendizagem não supervisionada que resolve o problema de *clustering*

[54, 56, 58, 59].

Os potenciais benefícios deste algoritmo, relatados por Mohd et al. [58], podem ser enunciados da seguinte forma: cobre diferentes tipos de atributos; descobre *clusters* de formato arbitrário; necessita de requisitos mínimos para encontrar os parâmetros de entrada; pode ser utilizado com ruído; minimiza as diferenças entre os dados. É considerado um dos dez algoritmos de Data Mining mais influentes na comunidade de pesquisa científica [60]. A sua utilidade estende-se a diferentes áreas, sendo mais frequentemente aplicado em marketing, biologia e reconhecimento de imagem [58].

Conhecido sobretudo pela sua simplicidade, funciona pela determinação de possíveis  $k$  grupos de objetos de dados, cujos membros são de alguma forma similares. O problema fulcral consiste em, dado um conjunto de  $n$  pontos no espaço real  $d$ -dimensional  $R^d$  e um inteiro  $k$ , determinar um conjunto de  $k$  pontos em  $R^d$ , chamados centróides, que minimizem a distância média de cada ponto ao seu centro mais próximo [54, 56, 58, 60, 61].

O pseudocódigo do SK-Means é mostrado no Algoritmo 1. Este algoritmo baseia-se num processo iterativo que segue um conjunto de regras simples para classificar um determinado conjunto de dados num conjunto de  $k$  *clusters* definidos previamente [51, 54, 56, 59–63]. O primeiro passo, e a ideia principal, é definir  $k$  centróides, um para cada *cluster*, o mais distantes possível uns dos outros. O passo seguinte consiste em atribuir cada um dos pontos constituintes do conjunto de dados ao seu centróide mais próximo. Quando todos os pontos fornecidos estiverem associados a um *cluster* o algoritmo recalcula  $k$  novos centróides que reflitam a atual distribuição dos pontos em cada *cluster*. Após a obtenção destes novos  $k$  centróides uma nova ligação tem de ser feita entre os mesmos pontos e o centróide mais próximo. O algoritmo converge quando não existem mais alterações na atribuição de instâncias aos *clusters*, ou seja quando novas iterações não provocam mudanças nos centróides [51, 59, 61].

O algoritmo procura a minimização de uma função ou critério de otimização, usualmente a função erro quadrático [51, 57, 61–63] dada por:

$$\sum_{j=1}^k \sum_{i=1}^k \left\| x_i^{(j)} - c_j \right\|^2$$

Onde

$$\left\| x_i^{(j)} - c_j \right\|^2$$

é uma medida de distância entre o ponto  $x_i$  no *cluster*  $j$  e o centróide  $c_j$ . A medida de distância geralmente utilizada é a denominada distância euclidiana [63].

A função de otimização, ou função objetivo, é um indicador da distância dos vários pontos constituintes dos *clusters* aos seus respectivos centróides.

<b>Entrada:</b>	$T = \{d_1, d_2, d_3, \dots, D_n\}$	// Conjunto de elementos
	$k$	// Número de clusters pretendidos
<b>Saída</b>	: $C$	// Conjunto de clusters
<b>Início</b>		
	colocar $k$ pontos no espaço representado pelos elementos que vão sofrer <i>clustering</i> . Estes pontos representam o grupo de centróides inicial $C$ ;	
	<b>repita</b>	
	Atribuir cada elemento $d_i$ ao centróide que lhe é mais próximo;	
	Recalcular as posições dos $k$ centróides de forma a que cada centróide $C_j$ seja atualizado para ser a média dos elementos que o compõem;	
	<b>até</b> que os centróides não se movam mais no espaço ;	
<b>Fim</b>		

**Algoritmo 1:** SK-Means

A execução deste algoritmo é visualmente representada na figura 2.3.

O SK-Means é um algoritmo simples que mostra ser eficaz na produção de resultados de *clustering* em muitas aplicações práticas e é um bom candidato para operar com pontos gerados aleatoriamente [51, 55, 57, 62]. Além disso, apresenta um tempo de convergência rápido, especialmente em problemas de grandes dimensões [62].

Existem no entanto algumas dificuldades na utilização deste algoritmo. O problema mais recorrente tem que ver com a necessidade de definir an-

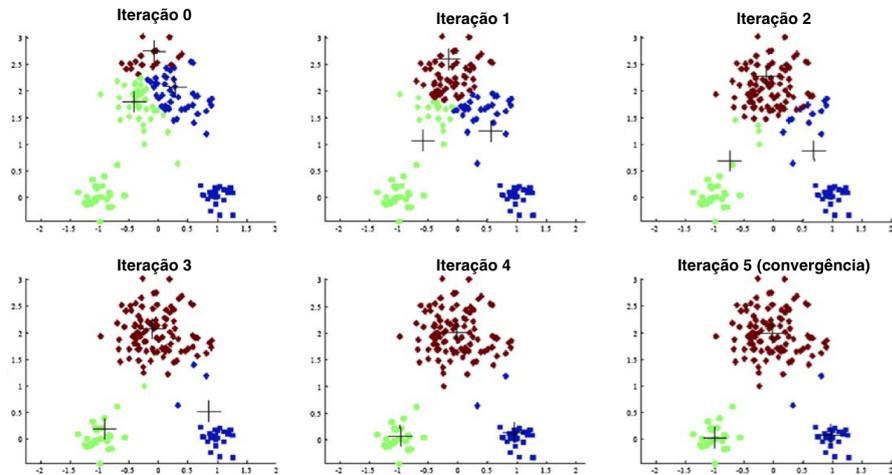


Figura 2.3: Mudança de posição dos centróides (indicados pelo sinal +) e atribuições de dados (indicadas pela cor) durante uma execução do algoritmo K-Means. Adaptado de [60]

tecipadamente o número de *clusters* para a inicialização do processo. A escolha apropriada deste valor depende do tipo de problema em questão e geralmente o utilizador necessita de diversas tentativas até obter o valor mais indicado [54, 57, 62]. Um problema referido por alguns autores está relacionado com o facto do algoritmo ser significativamente sensível aos centróides iniciais seleccionados aleatoriamente, um efeito que pode ser reduzido correndo o algoritmo várias vezes [51, 54]. O facto de poder ficar facilmente preso em soluções ótimas locais também constitui um problema em algumas aplicações [62].



# Capítulo 3

## Arquitetura e Implementação

Tal como mencionado anteriormente, o objetivo deste trabalho é o desenvolvimento de um sistema integrado de simulação e previsão, a partir do trabalho realizado por Gonçalves et al. [15]. Pretende-se transformar o sistema existente num sistema especificamente habilitado para a reprodução fiel de movimentos tipicamente demonstrados por pessoas (pacientes ou profissionais) em ambiente hospitalar.

Neste contexto, para explicar as mudanças a que o sistema foi sujeito, é importante começar por clarificar a arquitetura base em que o mesmo foi inicialmente desenvolvido, para posteriormente estabelecer a ponte entre ambos.

Desta forma, neste capítulo pretende-se apresentar uma visão geral de ambos os sistemas, assim como, uma descrição dos componentes que fazem parte da arquitetura e as tecnologias adotadas na implementação do sistema proposto.

### 3.1 Visão Geral: sistema anterior *versus* sistema proposto

O sistema originalmente proposto é constituído por um modelo de simulação de movimento assente na simulação de trajetórias aleatórias, entre dois pontos fixos, através de um algoritmo baseado no conceito de anti-

gravidade. A área percorrida pelo objeto contém 2500 posições, o objeto desloca-se apenas na diagonal e não existem delimitações no espaço. Neste estudo, o desempenho do modelo de previsão não foi avaliado em termos quantitativos.

De forma a adaptar o processo de simulação/previsão aos objetivos delineados o sistema deve ser sujeito a diversas alterações e melhorias. A adição de novas instâncias – paredes, sensores quartos e corredores – e a posterior manipulação de dados deve assentar num processo eficiente que permita que novo conteúdo seja adicionado de forma simples a qualquer momento.

Para a criação de um sistema inteligente para a simulação e previsão do movimento de pessoas em ambiente hospitalar é imperativo integrar a componente de simulação propriamente dita com outras plataformas. Nomeadamente, para o registo dos resultados obtidos em cada movimento torna-se necessário a adoção de uma estratégia de armazenamento de dados que registre os diferentes valores necessários ao cálculo do desempenho do sistema. Desta forma, existe um repositório central remoto – uma base de dados – que mantém organizados os dados fornecidos pelo sistema no final de cada movimento para posterior análise.

É ainda conveniente que exista uma interface para que os dados presentes no repositório possam ser visualizados de forma intuitiva pelo utilizador. Pretende-se uma plataforma que permita uma integração com repositórios alojados em servidores externos, para a criação de estatísticas e comparação de resultados para diferentes cenários. Tal plataforma deverá permitir a criação de relatórios dinâmicos, por forma a serem facilmente atualizados sempre que a quantidade de informação da base de dados aumenta. Como tal, optou-se pela adoção do Pentaho Report Designer para a visualização de estatísticas e relatórios de desempenho do sistema proposto.

A figura 3.1 pretende elucidar as características gerais do sistema global proposto neste trabalho e a interação entre os diversos componentes que o constituem.

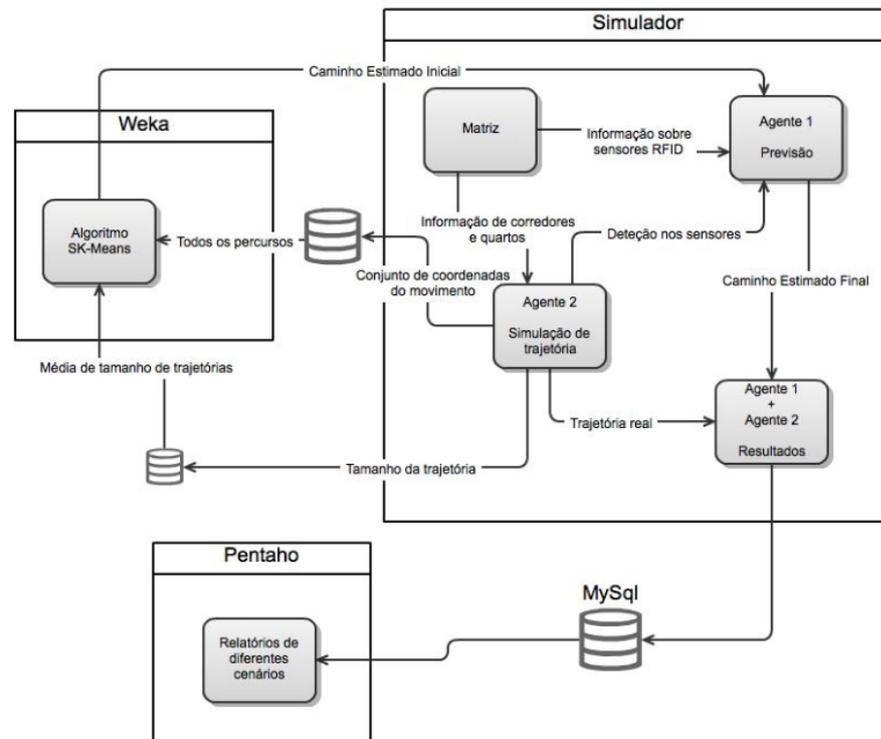


Figura 3.1: Arquitetura do sistema proposto e interação entre os diferentes módulos.

## 3.2 Descrição dos Componentes

Como se pode ver na figura 3.1 o sistema global é constituído por quatro componentes: Weka, Simulador, Base de Dados e Pentaho. Destes, apenas o Simulador e o Weka são considerados fulcrais ao correto funcionamento do modelo de simulação e previsão. A base de dados MySQL e o módulo de criação de relatórios não fazem parte do cerne da arquitetura do sistema uma vez que não são necessários para o seu correto funcionamento, servindo apenas os propósitos de avaliação do desempenho do sistema e de facilitar a manutenção, gestão e visualização dos dados.

### 3.2.1 Simulador

Para prosseguir com a descrição do sistema proposto convém começar por definir o conceito de agente em sistemas computacionais, dada a sua re-

levância no processo de simulação e previsão. A definição de agente é, no entanto, uma tarefa de difícil concretização uma vez, que um agente pode ser usado para caracterizar diferentes entidades, em diversos contextos de utilização. Abelha [64] apresenta várias definições para agente, utilizadas por diversos autores no desenvolvimento de diferentes aplicações e/ou em diferentes contextos. Independentemente do contexto da aplicação, pode-se dizer que os agentes são geralmente dotados de atributos que lhes permitem o planejamento das suas ações e a definição de estratégias de previsão e de coordenação, isto é, assumem-se como uma peça central numa metodologia de computação. São componentes persistentes e ativos que percebem, raciocinam, atuam e comunicam. Os agentes são entidades que habitam em ambientes complexos, sentem esse ambiente e atuam de modo autónomo, procurando executar um conjunto de tarefas para as quais receberam indicação. Neste trabalho, considera-se um agente como uma entidade baseada num sistema de software que permite a automatização de processos de raciocínio e tomada de decisão.

O sistema proposto, implementado em linguagem de programação Java, e representado pelo módulo Simulador, assenta na interação entre um agente responsável pelo processo de aprendizagem e um agente responsável por simular uma trajetória realista. Através da junção dos dados fornecidos por ambos os agentes é possível criar uma previsão de trajetória coincidente com a realidade. Além disso, a sua interação permite o cálculo de estatísticas e precisão do sistema.

O modelo de domínio do simulador é aqui descrito através da arquitetura *Model View Controller* (MVC), que é atualmente uma arquitetura padrão na engenharia de software. A ideia é permitir a independência funcional entre as camadas, nomeadamente o desenvolvimento, atualização e testes a cada parte isoladamente. O principal objetivo do modelo passa por separar a aplicação nas seguintes três camadas distintas:

- Camada de dados: responsável pela criação, armazenamento e manipulação de dados. Notifica a camada de visualização de mudanças de estado ocorridas no sistema. Representada pela Matriz, Objeto, Con-

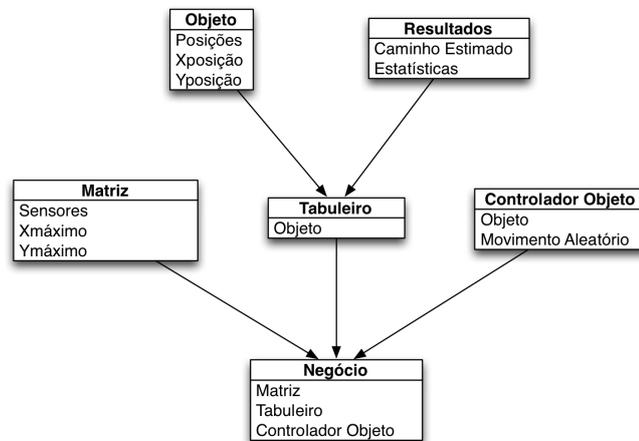


Figura 3.2: Diagrama ilustrativo do modelo de domínio - primeira versão.

trolador Objeto, Previsão e Resultados.

- Camada de visualização: representação da interface gráfica. É usada para receber a entrada de dados e mostrar o resultado ao utilizador. Representada pelo Tabuleiro.
- Camada de controlo: estabelece a ligação entre os dados e a interface gráfica. Responsável pelo controlo e mapeamento das ações. Seleciona a visualização para cada funcionalidade. Representada pelo Negócio.

Novas classes métodos e instâncias essenciais para o cumprimento dos objetivos estipulados foram adicionadas ao programa original, nomeadamente: *Previsão*, *Corredor*, *Parede*, *Quarto*, novos objetos observáveis *Objeto* e novos observadores para os mesmos designados por *ControladorObjeto*, preparando o sistema para a adição de mais objetos sistematicamente. O diagrama do modelo de domínio presente na figura 3.2 ilustra a arquitetura MVC adotada para a implementação do Simulador original, e a figura 3.3 a arquitetura atual.

Contrariamente ao que acontece no sistema inicialmente proposto, em que a previsão é calculada no final da simulação e não é visualmente apresentada ao utilizador, neste trabalho optou-se por tornar o módulo de previsão um ponto central do simulador com o objetivo de o transformar num sistema que

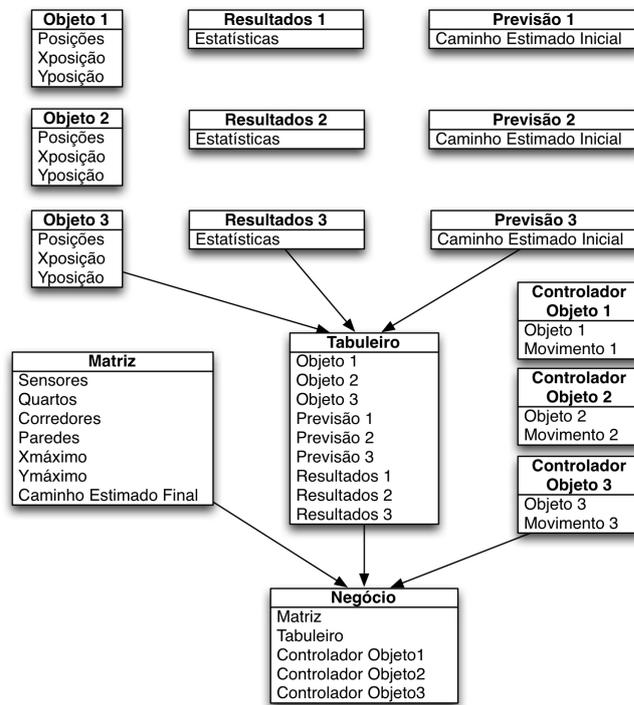


Figura 3.3: Diagrama ilustrativo do modelo de domínio - versão proposta.

integra simulação e previsão em tempo real. Deste modo, a previsão foi modificada para se tornar num processo dinâmico que oferece ao utilizador uma visão em tempo real da estimativa de localização do objeto numa interface gráfica criada para o efeito.

Em cada simulação o simulador recebe uma matriz de dados, contendo informação relativa à existência de elementos na planta hospitalar, incluindo o objeto; a localização e alcance dos sensores; a localização e área dos quartos e corredores; as delimitações físicas do espaço sob a forma de paredes e pontos aleatórios necessários ao cálculo do movimento. Com base nessa matriz, o simulador fica possibilitado de calcular e devolver ao utilizador uma estimativa de percurso (juntamente com as ferramentas do Weka); de gerar eficazmente posições para o objeto, ponto a ponto; de conjugar ambos os agentes por forma a que o que acontece na parte de simulação seja comunicado à parte de previsão em tempo real, resultando na apresentação de duas interfaces dinâmicas e sincronizadas entre si. Como resultado o simulador devolve estatísticas referentes ao desempenho do modelo de previsão face à simulação, utilizadas na análise de diferentes cenários, e todas as posições percorridas pelo objeto, necessárias ao Data Mining. Uma representação intuitiva deste processo pode ser visualizada na figura 3.4.

### 3.2.2 Weka

Cada *cluster*, obtido através da aplicação do SK-Means ao conjunto de dados, corresponde a um ponto no espaço. Tal como já foi mencionado, este algoritmo requer o conhecimento prévio do número de *clusters* a calcular. Por este motivo, juntamente com o registo de localizações, também um registo de tamanhos de movimento é mantido. Uma vez que se pretende um caminho estimado o mais similar possível do percorrido, considerou-se que o número de *clusters*, e consequentemente o tamanho do caminho estimado, corresponde em cada iteração à média dos tamanhos registados.

Desta forma, o módulo de Data Mining recebe, a cada nova iteração, o número de *clusters* a calcular e a totalidade de localizações registadas até ao momento, fornecidas pelo agente de simulação e que constituem o conjunto

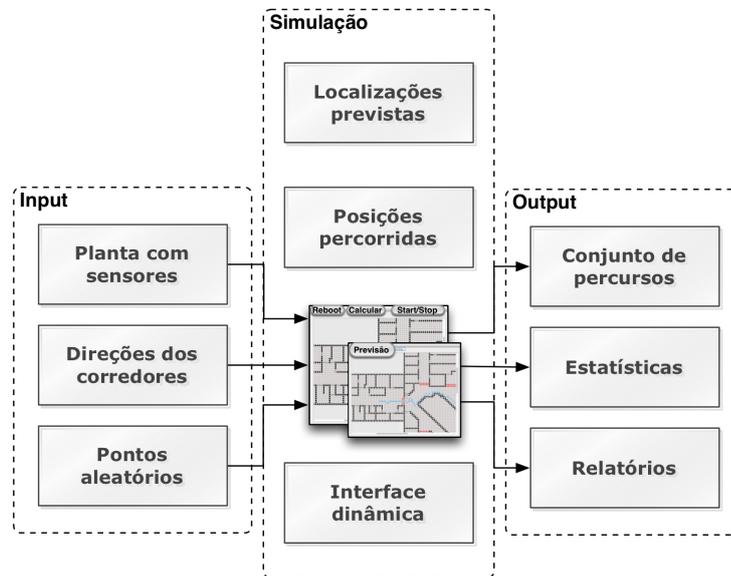


Figura 3.4: Simulação.

de treino. O conjunto de teste consiste no caminho que é simulado em cada novo movimento. As localizações percorridas pelo objeto são registadas num ficheiro do tipo *arff* sob a forma de conjuntos de coordenadas bidimensionais.

Relativamente aos estágios que constituem o processo de KDD, referidos anteriormente, e que incluem o pré-processamento, transformação, Data Mining e avaliação, apenas alguns foram implementados. O pré-processamento de dados não se revelou necessário no contexto da aplicação. O estágio de limpeza não se aplica, pois não existem dados com anomalias nem dados em falta. Aliás, a informação que é utilizada é constituída por todos os pontos que o objeto percorre em diversos movimentos, pelo que todos são importantes e nenhum se pode considerar anómalo. O estágio de redução de dados aplica-se sobretudo em grandes bases de dados, onde a quantidade de informação é enorme e onde certos parâmetros são dispensáveis à análise. Uma vez que este não é o cenário da aplicação em questão, o conjunto de dados é de dimensão pequena (na ordem das centenas de KB para um ficheiro com cerca de cem movimentos distintos) e o tempo de análise medido nos testes realizados não passou dos três segundos num ficheiro desta dimensão, este estágio também não é aplicado. Os dados utilizados no Data Mining

são todos armazenados no mesmo formato (ficheiro do tipo *arff*), pelo que também não existe necessidade de proceder à sua transformação. É de salientar que o desenvolvimento e implementação foram realizados numa máquina Core 2 Duo de 2.4GHz e 4GB de RAM.

No entanto, e dado que a área escolhida para a previsão consiste numa área estritamente delimitada, revelou-se imperativo proceder a um pós processamento de dados para evitar que pontos estimados para fora dessa mesma área sejam incluídos no caminho previsto, influenciando negativamente os resultados da previsão quando na realidade são inviáveis. Desta forma, *clusters* que não estejam incluídos dentro da área delimitada por paredes, e que em certos casos podem ser calculados pelo algoritmo por se encontrarem muito próximos de zonas onde se registou uma elevada densidade de pontos percorridos, são eliminados do caminho previsto inicial.

Depois do modelo ter sido processado através do conjunto de treino é avaliado fazendo previsões contra o conjunto de teste. Na prática este processo corresponde a uma comparação entre o caminho previsto e o caminho simulado através do cálculo da distância entre os dois.

### 3.3 Ferramentas e linguagens

O sistema de simulação proposto foi desenvolvido na linguagem de programação JAVA. Esta linguagem fornece robustez e escalabilidade ao sistema e ainda as APIs necessárias ao seu *design* – Java Swing - e à integração de Data Mining – Weka.

O Java Swing é um widget toolkit <sup>1</sup> do Java que fornece as APIs necessárias para tornar o processo de desenhar a interface gráfica mais simplificado para o programador. Desta forma, utilizando os recursos do Swing para desenhar os componentes da Interface Gráfica do Utilizador (Graphical User Interface - GUI), é possível criar etiquetas, botões, caixas de texto, *checkboxes*, painéis, entre outros, de uma forma mais intuitiva. O Java Swing facilita a criação da interface de utilizador, permitindo um elevado nível de

---

<sup>1</sup>Em computação widget toolkit refere-se a um conjunto de elementos básicos utilizados no desenvolvimento de aplicações com interface gráfica.

interação, uma visualização em tempo real da trajetória e compatibilidade com as bibliotecas subjacentes. A API Swing é uma API de alto nível que apesar de consumir mais memória RAM, é mais completa e mais flexível do que a generalidade das APIs gráficas.

A plataforma de aprendizagem *Waikato Environment for Knowledge Analysis (Weka)* fornece um conjunto completo de algoritmos de aprendizagem de máquina e ferramentas de pré-processamento de dados, para permitir a aplicação automática de algoritmos de Data Mining comuns na área da bioinformática, tais como classificação, regressão, clustering e seleção [54, 65–67]. É reconhecida como um ponto de referência em Data Mining e aprendizagem de máquina, e detém atualmente uma ampla aceitação nos meios académicos e comerciais. Qualquer técnica de aprendizagem do Weka pode ser acedida a partir de programas Java através da API Weka.

MySQL é um sistema de gestão de bases relacional, em que dados são armazenados em tabelas que estão relacionadas entre si através de regras lógicas. Permite a remoção de redundâncias, operações rápidas e consultas de alto nível. Quantidades consideráveis de dados podem ser armazenadas e acedidas através de uma consulta de Linguagem de Consulta Estruturada (Structured Query Language SQL). A base de dados MySQL tornou-se uma das bases de dados mais populares a nível internacional, devido ao seu alto desempenho, fiabilidade e facilidade de uso, e por suportar a integração com um grande número de aplicações [35]. A base de dados foi utilizada com a finalidade de manter um registo completo dos resultados da previsão obtidos para os diferentes casos de estudo.

O Pentaho Report Designer é uma aplicação de código aberto que fornece ao utilizador um ambiente de *design* visual para a criação de relatórios. Permite uma fácil integração com os dados externos guardados na base de dados MySQL, através de consultas sql, e a criação de relatórios dinâmicos, facilmente atualizados sempre que na base de dados é inserido um novo registo.

# Capítulo 4

## Simulador

### 4.1 Interface Gráfica

A estrutura da interface gráfica original foi mantida (Java Swing) e serviu de base para o desenvolvimento de uma nova configuração de apresentação ao utilizador, através da inserção de novos botões e da alteração da planta de circulação. A interface foi desenvolvida de modo a que seja possível a qualquer utilizador interagir de forma intuitiva com o simulador.

Quando o simulador é iniciado é aberta uma janela com a representação gráfica da planta do piso hospitalar, como mostra a Figura 4.1. Tentou-se que esta representação fosse o mais fiel possível à planta fornecida pelo hospital, para tornar possível o estudo de trajetórias e diferentes posições de sensores num ambiente de simulação de *tracking* real. Não foi, no entanto, seguida nenhuma escala para o *design* da planta uma vez que não se justificava uma representação rigorosa da mesma. O que se pretendia era apenas simular o movimento do objeto num ambiente real, provido de paredes, quartos e corredores.

Pela observação da Figura 1.2, na página 12, é fácil inferir que existem quatro escadas/elevadores, no canto de cada bloco. Por conveniência, e porque são os únicos locais de entrada e saída de pessoas, definiu-se que estes serão os pontos de início e fim de movimento.

Os componentes que fazem parte da interface do simulador são:

- O tabuleiro 2D, com o conjunto total de coordenadas  $xx$  e  $yy$  e a respetiva localização das paredes, quartos, portas e sensores;
- Os robots que irão percorrer o tabuleiro. Também designados por objetos.

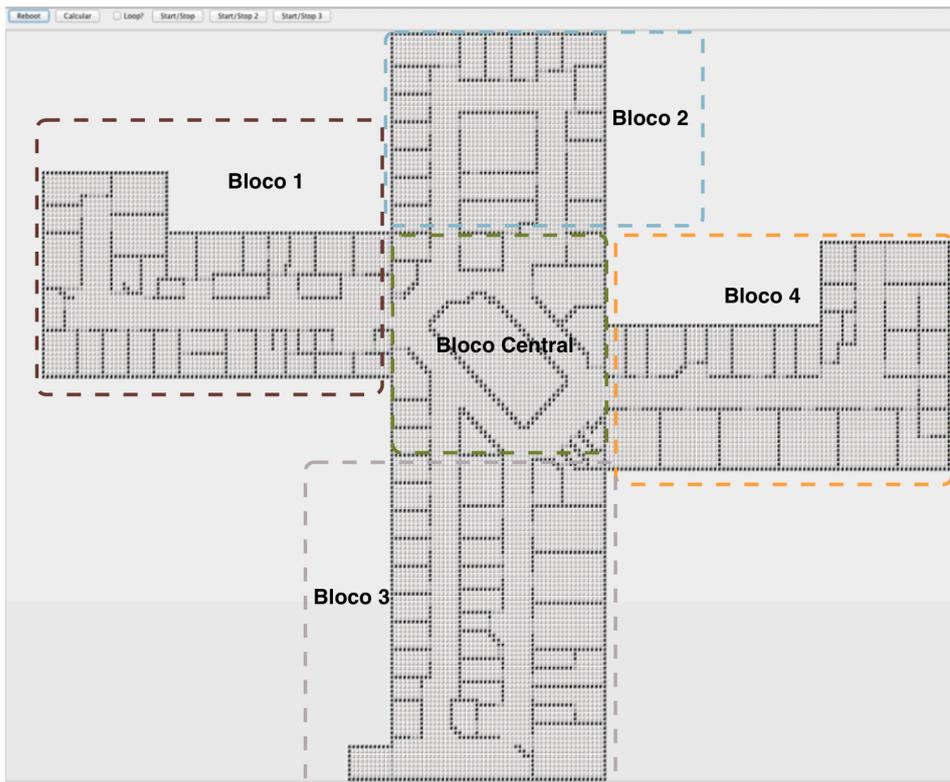


Figura 4.1: Representação gráfica do piso hospitalar no simulador com legenda das 5 diferentes zonas de circulação.

É nesta primeira janela que o utilizador pode dar a ordem de início de simulação de movimento no piso hospitalar, clicando simplesmente no botão Start/Stop associado ao objeto. Os botões encontram-se no canto superior esquerdo da janela (em pormenor na Figura 4.5). Ao clicar no botão Start/Stop ocorre uma sucessão de eventos pela seguinte ordem:

1. O agente responsável pela previsão vai analisar os registos de movimentos anteriores e retribuir um caminho estimado, para cada objeto, com

base nos *clusters* calculados.

2. Novas janelas, com uma representação gráfica similar à janela de simulação, são lançadas para a visualização do caminho estimado. O número de janelas é equivalente ao número de objetos em movimento.
3. O agente responsável pelo movimento simulado vai criar uma trajetória, coordenada a coordenada, através da geração de forças aleatórias. Este movimento é visualizado em tempo real.
4. As localizações previstas e o movimento do(s) robot(s) pode(m) ser seguidos pelo utilizador simultaneamente. Por um lado, cada janela do caminho estimado permite ao utilizador ver todos os pontos onde é possível que o objeto se encontre a qualquer instante. Por outro, a janela do caminho simulado permite ver a qualquer instante a posição real do(s) mesmo(s).
5. A janela do caminho simulado é constantemente atualizada com o nome do último sensor onde cada objeto foi detetado pela última vez.
6. Um objeto termina o seu movimento num dos três elevadores do piso.
7. A janela do caminho estimado apresenta neste momento todas as localizações previstas para o objeto em questão.
8. É iniciado o módulo de cálculo dos resultados pela comparação entre o caminho previsto e o caminho simulado.
9. A janela de resultados relativa ao objeto que terminou o movimento é lançada e o utilizador pode ver as localizações que foram ou não previstas corretamente, ponto a ponto.
10. A simulação procede de igual forma para os objetos que ainda não terminaram o movimento.

Relativamente ao ponto 5 apresenta-se um exemplo na Figura 4.2. Neste caso o objeto foi detetado pela última vez no sensor com identificação h2.

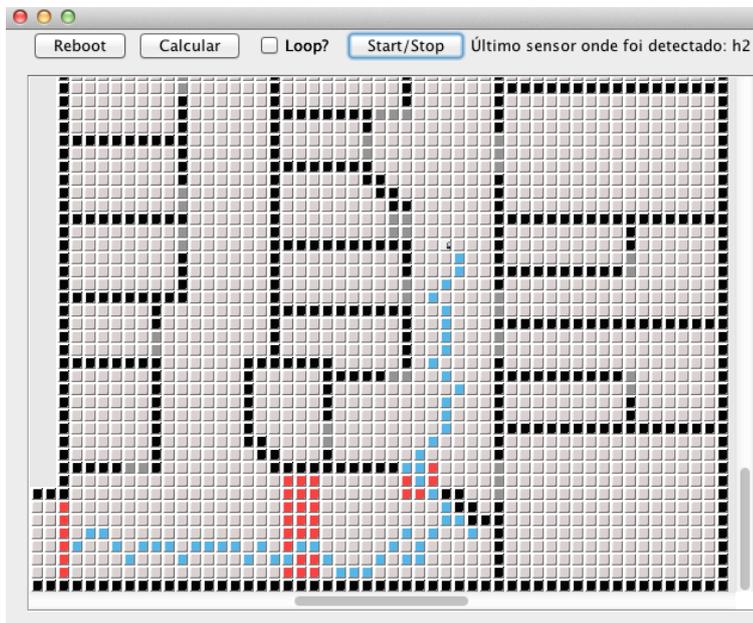


Figura 4.2: Barra de opções da janela do caminho real durante o movimento.

As Figuras 4.3 e 4.4 pretendem evidenciar a cadeia de eventos referidos sob a forma de janelas apresentadas ao utilizador em cada momento da simulação. Na primeira, apresenta-se o caso em que existe apenas um robot a percorrer o tabuleiro e na segunda, o caso em que existem três em simultâneo, podendo-se aplicar o mesmo raciocínio para qualquer número de objetos.

De seguida são explicadas as funções de cada um dos botões e da *checkbox* presentes na janela de simulação de movimento (ver Figura 4.5).

**Reboot:** serve o propósito de dar início a um novo movimento sem ser necessário reiniciar o simulador. Selecionando esta opção ocorre um *Rewind* do objeto (volta à posição inicial), os pontos que tinha percorrido são eliminados do mapa e o simulador volta ao ponto 1 das etapas referidas.

**Calcular:** este botão foi adicionado com o duplo objetivo de adicionar o registo de movimento ao ficheiro de dados em qualquer instante desejado, evitando que o registo dependa do fim da trajetória, permitindo o cálculo do desempenho da previsão em qualquer fase do movimento. A

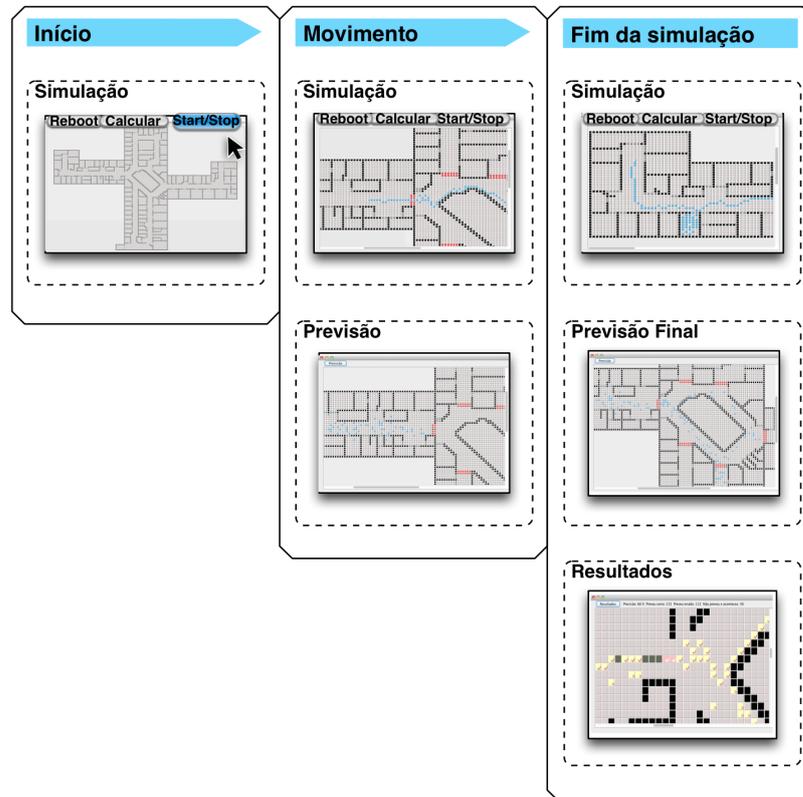


Figura 4.3: Diferentes etapas da simulação.

necessidade da introdução deste botão surgiu durante a fase de testes, altura em que não estava ainda corretamente estabelecido o algoritmo de simulação de trajetória. Ao clicar neste botão o robot pára o movimento (equivalente ao Stop), a trajetória até esse momento é adicionada ao ficheiro e são calculados os resultados da previsão.

**Loop:** caso o utilizador pretenda que uma nova simulação se inicie automaticamente no fim do movimento, deverá seleccionar a *checkbox* Loop antes de clicar no botão Start/Stop. Esta opção é particularmente útil para a realização da Extração de Conhecimento uma vez que permite a realização de várias simulações consecutivas, e consequentemente o registo de várias trajetórias que permitem a aplicação do algoritmo de Data Mining a um grande conjunto de dados. Assim, caso a opção Loop

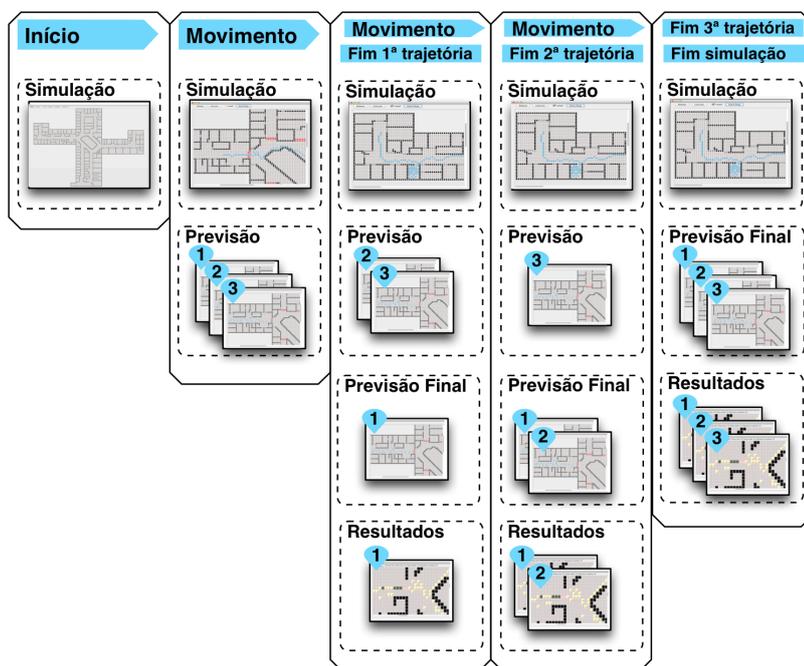


Figura 4.4: Diferentes etapas da simulação para vários objetos em simultâneo.



Figura 4.5: Barra de opções da janela de simulação.

esteja selecionada o simulador faz Reboot do tabuleiro de forma independente sempre que o objeto chegue à posição final. Caso o utilizador não dê ordem para terminar, esta operação acaba geralmente quando não existe mais espaço na memória do computador. Nesta situação é aconselhável reiniciar o simulador e repetir o processo.

**Start/Stop:** botão que permite dar início à trajetória do robot. Caso se pretenda proceder à monitorização de 2 ou mais objetos, o mesmo número de botões Start/Stop estará presente, podendo o utilizador iniciar a simulação de cada um em instantes de tempo diferentes. Clicando em Start/Stop durante a fase de movimento, o robot pára na posição em que se encontrava nesse instante, até que o utilizador clique novamente

para dar continuidade à simulação.

## 4.2 Simulação da trajetória de uma pessoa

O algoritmo de movimento selecionado por Gonçalves et al. [15] para a criação de uma trajetória aleatória fundamentou-se exclusivamente no conceito de anti-gravidade. O movimento anti-gravidade consiste numa técnica utilizada para criar movimentos aleatórios através da definição de pontos que se pretende que sejam evitados pelo objeto durante o seu deslocamento, designados por pontos gravíticos. Muito resumidamente, o algoritmo funciona através da distribuição ao longo do ambiente de simulação de pontos gravíticos, que o objeto evitará ou dos quais se aproximará, dependendo da força que lhes esteja associada. Este método é útil na simulação de trajetórias que são difíceis de prever através de técnicas de análise de padrões.

Durante o processo de adaptação do algoritmo aos objetivos delineados, tentou-se que esta abordagem fosse mantida para a simulação de trajetórias de pessoas, fazendo com que as portas representassem os pontos de atração e as paredes os de repulsão. No entanto, tal tentativa revelou-se infrutífera essencialmente devido ao facto deste método só funcionar para movimentos aleatórios, o que não se adequa ao contexto das trajetórias de pessoas. Além disso, o algoritmo apresentou-se incompatível com a presença de obstáculos (paredes) no caminho do objeto, pelo que teve de ser completamente repensado e redefinido.

Inicialmente o robot era atraído para a posição final e repellido da posição inicial do tabuleiro, sofrendo forças aleatórias pelo caminho que asseguravam que não seguia uma linha reta e constante ao longo das várias iterações. Com a introdução da nova planta, foi necessário criar uma alternativa que lidasse com a questão dos quartos e corredores. Nomeadamente, foi necessário assegurar que o robot simulasse a trajetória típica de uma pessoa, movimentando-se dentro de quartos e escritórios, seguindo através de corredores para chegar a um local desejado. Como tal, uma abordagem em que apenas componentes aleatórias são consideradas teve de ser melhorada e adaptada às condições impostas. Desta forma, revelou-se necessário adi-

cionar uma nova característica ao sistema que lidasse com este problema permitindo que o algoritmo de movimento considerasse uma componente de direção preferencial.

Através da definição de duas áreas distintas - corredores e quartos (incluindo os elevadores/escadas nesta entidade) - foi possível associar diferentes formas de movimento ao objeto dependendo da zona de circulação onde se encontra. Assim, durante um movimento o objeto está sujeito simultaneamente a componentes de carácter aleatório e componentes de carácter direcionado, que permitem que se desloque de uma forma direcionada – seguindo um caminho preferencial, escolhido – e de uma forma aleatória – a cada novo passo, cada direção possível tem a mesma probabilidade de ser escolhida. O objetivo passa por garantir que a trajetória não seja igual a outras anteriormente percorridas e que siga uma certa tendência associada à rotina do objeto.

O tamanho  $D_x$  de um movimento  $x$  é determinado pelo número de posições ou coordenadas bidimensionais  $(x, y)$  que o objeto percorre durante esse movimento. Ou seja,  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ . Na prática, o tamanho corresponde ao número de pontos percorridos pelo robot. Como consequência do algoritmo definido, o tempo que o objeto permanece nos quartos é muito variável; o número de quartos que visita no mesmo movimento não é constante e o caminho que o robot segue para atingir a posição final não é sempre o mesmo. Por estes motivos o tamanho pode variar muito entre iterações. No entanto, e ignorando alguns casos específicos, pode-se afirmar que este valor se encontra entre 300 e 500 posições, ou seja, numa iteração o objeto percorre uma média de 400 posições entre dois elevadores.

Onde antes existia um tabuleiro sem obstáculos, em que se propunha uma simples trajetória para um único robot, numa matriz com 2500 posições, agora existe a representação de um piso hospitalar, preenchido com corredores, quartos e elevadores, num total de 14506 posições onde um ou mais objetos se podem movimentar. O movimento é realizado através de passos, sendo que cada passo corresponde a um incremento ou decremento de uma unidade na posição em X e/ou Y. É de salientar que o objeto se move diagonal, vertical e horizontalmente em qualquer direção, o que não era possível com o algoritmo inicialmente proposto.

De seguida será explicado o movimento descrito pelo objeto nas duas diferentes zonas definidas: corredores e quartos.

### 4.2.1 Corredores

Uma das primeiras preocupações durante a criação do algoritmo de trajetória foi evitar que o objeto passasse várias vezes consecutivas pelo mesmo ponto e que ficasse preso na mesma zona durante tempo indefinido. Tanto em termos de resultados obtidos como em termos de aproximação a um movimento real, ficou claro que seria imperativo definir uma estratégia que evitasse estas ocorrências. A primeira abordagem a este problema passou por guardar as posições que o objeto ia percorrendo ao longo do tempo. De cada vez que o algoritmo calculava a posição seguinte, verificava se essa posição já tinha sido percorrida. Caso o objeto já tivesse passado nesse ponto o algoritmo continuaria a calcular uma nova posição até que encontrasse uma livre. Esta estratégia revelou-se no entanto inadequada em termos computacionais uma vez que não respondia com a desejada celeridade ao cálculo de um ponto alternativo. Além disso tornava-se difícil impor a este sistemas uma solução que visasse a situação em que todos os pontos em torno do ponto atual já tinham sido percorridos.

Uma melhor abordagem, de atribuição de valores para definir uma direção, foi desenvolvida e aperfeiçoada para fazer face a este problema. Desta forma, a cada corredor foi associado um conjunto de dois valores,  $x_{comp}$  e  $y_{comp}$ , que influenciam o cálculo da direção do objeto em  $xx$  e  $yy$ , respetivamente  $x_{dir}$  e  $y_{dir}$ . O valor associado a estas componentes pode ser 1, 0 ou -1, dependendo do sentido que se pretender impor ao objeto. Por exemplo, se a um corredor se associar o par (0,1), a componente  $x_{dir}$  terá um efeito muito menor no movimento do que a componente  $y_{dir}$  e o objeto mover-se-á preferencialmente segundo a direção  $yy$ . O termo preferencial é aqui empregue para denotar o facto de que associar estes valores à entidade não implica obrigatoriamente que o objeto se mova nesse sentido. O objetivo passa por fazer com que sempre que seja possível, ou seja quando não existe nenhum obstáculo, a deslocação seja nesse sentido, e que continue a ter alguma movimentação

no outro eixo (neste caso  $xx$ ). A tabela 4.1 resume sucintamente todas as hipóteses utilizadas.

$x_{comp}$	$y_{comp}$	Direção preferencial
1	1	Direita e baixo
1	-1	Direita e cima
1	0	Direita
-1	1	Esquerda e baixo
-1	-1	Esquerda e cima
-1	0	Esquerda
0	1	Baixo
0	-1	Cima
0	0	Não aplicável

Tabela 4.1: Movimento nos corredores

A Figura 4.6 mostra a janela do caminho percorrido pelo robot, com zoom no bloco 1, para um certo percurso em que o corredor horizontal admitia o par  $(-1, 0)$  para a direção preferencial, e o corredor vertical o par  $(0, -1)$ . Como se pode ver, uma vez no corredor horizontal o robot movimentava-se não apenas para a esquerda, como também de forma aleatória, para cima e para baixo.

Utilizando este método elimina-se o problema de o objeto ficar preso em alguns sítios uma vez que este se move continuamente na direção escolhida. Outra vantagem prende-se com o facto de ser mais fácil uma posterior manipulação do código para ir de encontro aos requisitos da aplicação e a alteração do curso do estudo definindo novos caminhos pela simples alteração dos valores estipulados.

### 4.2.2 Quartos

O movimento associado aos quartos foi pensado e implementado de forma diferente dos corredores pois no caso do *tracking* hospitalar é irrelevante fazer a previsão da trajetória da pessoa dentro do quarto, sendo suficiente saber se entrou ou não, e o tempo que lá permaneceu.

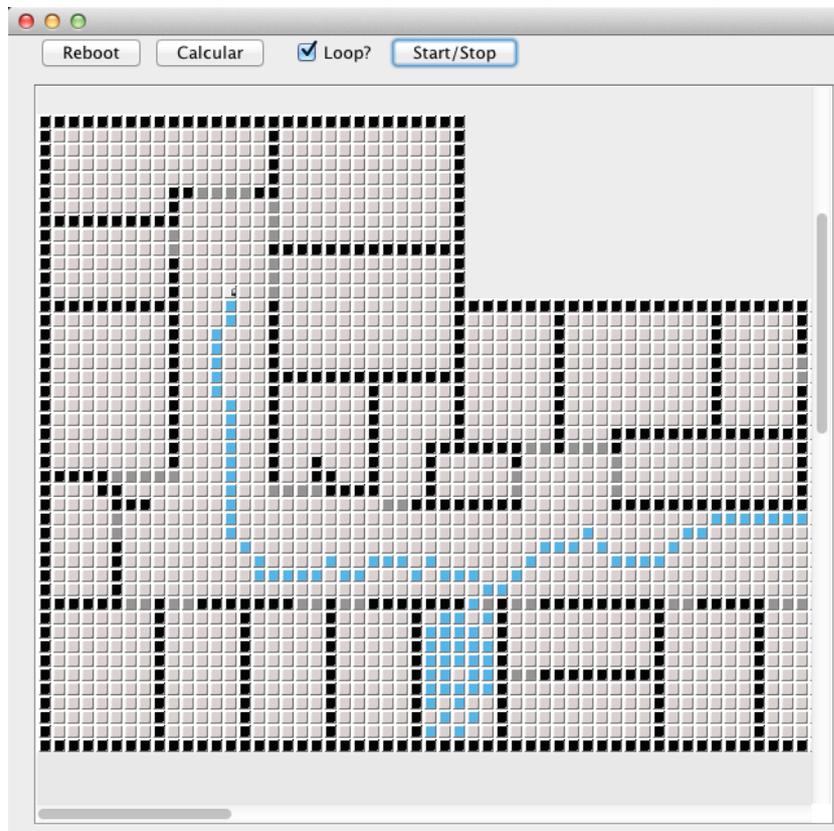


Figura 4.6: Simulação de movimento nos corredores do Hospital.

Aplicando este raciocínio, definiu-se que sempre que o objeto entra num quarto a sua próxima posição é determinada aleatoriamente. Ou seja, são gerados pontos aleatórios de forma a criar um movimento imprevisível sempre que o objeto se encontra no interior de uma área designada como quarto. Assim, a circulação do robot neste caso serve apenas o propósito de sinalizar a presença da pessoa ou equipamento no interior de uma divisão. A Figura 4.6 representa um bom exemplo desta situação.

Este método é conseguido através do algoritmo que será definido no próximo capítulo. Dado um determinado ponto aleatório, definido por uma coordenada em  $x$  ( $x_{rand}$ ) e outra em  $y$  ( $y_{rand}$ ), é calculada a força existente entre esse ponto e a localização atual do objeto através da fórmula 4.1. A direção do objeto depende unicamente do sinal da força e este por sua vez é determinado aleatoriamente. Isto significa que uma vez dentro de um quarto

o movimento do robot é completamente aleatório.

De forma a manter o objeto em circulação ao longo de todo o piso foi necessário definir um limite máximo de tempo de permanência nos quartos. Assim, sempre que um objeto entra no quarto é acionado um relógio (no algoritmo designado por *clock*) que garante que ao fim do tempo definido o objeto é forçado a sair e a continuar a sua trajetória. O objetivo é simular essencialmente uma passagem pela divisão e não uma permanência ou estadia prolongada.

No caso da simulação de trajetória presente na Figura 4.7, foi dada indicação para que a simulação se realizasse num ciclo (seleção da *checkbox* Loop) e que os percursos anteriormente descritos não fossem apagados sempre que um novo se iniciasse, com o objetivo de permitir uma melhor percepção dos quartos mais frequentemente visitados. A imagem representa um conjunto de 52 percursos, e permite ter uma visão global dos quartos/escritórios mais visitados pelo robot em questão num dos blocos do piso (bloco 1). Posteriormente, num conjunto de 40 registos, a distribuição de quartos por onde o robot passou, presente na Figura 4.8, manteve-se consistente com a que foi anteriormente verificada, ou seja, dentro dos padrões obtidos noutras simulações.

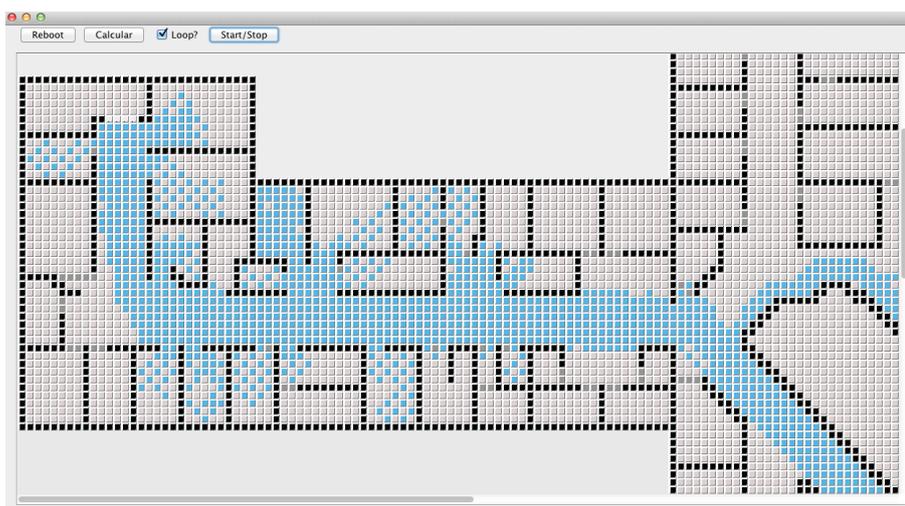


Figura 4.7: Simulação referente a 52 movimentos nos quartos do Hospital.

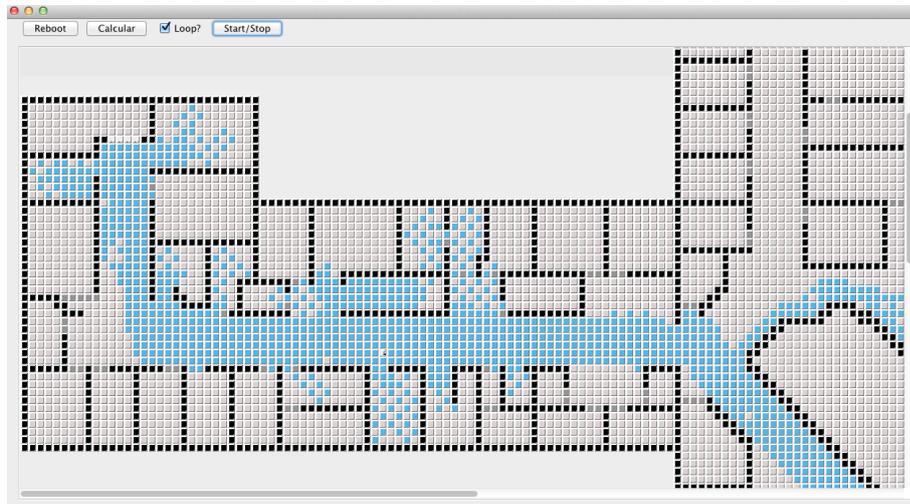


Figura 4.8: Simulação referente a 40 movimentos nos quartos do Hospital.

### 4.2.3 Algoritmo

O algoritmo de movimento baseia-se no cálculo iterativo de dois valores que conjuntamente determinam a direção do objeto segundo  $xx$  e  $yy$ , respectivamente  $x_{dir}$  e  $y_{dir}$ . O cálculo destes dois parâmetros depende unicamente de duas variáveis: duma constante, que pode ser aleatória ou previamente definida, e da distância do objeto a um ponto aleatório, pela relação 4.1.

$$Peso_{componente} = \frac{constante}{distância^2} \quad (4.1)$$

Similarmente à lei de Newton sobre gravitação universal, dada por 4.2, a fórmula utilizada assenta numa proporcionalidade direta entre o peso da componente  $x$  ou  $y$  e uma constante (na física, o produto das massas de dois objetos que se atraem) e indireta entre o peso da componente e a distância que separa o objeto de um ponto aleatório (na física correspondente à distância entre as duas massas).

$$F = G \frac{m_1 m_2}{r^2} \quad (4.2)$$

A distância a um ponto aleatório foi considerada para garantir uma certa aleatoriedade no movimento. Assim, quanto maior for este valor menor é o

valor do peso ou força da componente (similar ao que acontece entre dois corpos com massa). Quando a força em  $y$  é elevada e a força em  $x$  é muito pequena (por padrões definidos através da experiência), o objeto desloca-se apenas sobre a componente de valor elevado. Uma vez que os pontos são gerados aleatoriamente é impossível determinar com total certeza a próxima posição do objeto.

Convencionou-se que o sentido do movimento é determinado pelo sinal do peso da componente para aquele instante. Assim, de uma forma geral:

$x_{dir} > 0$  corresponde a uma deslocação de uma unidade para a esquerda;

$x_{dir} < 0$  corresponde a uma deslocação de uma unidade para a direita;

$y_{dir} > 0$  corresponde a uma deslocação de uma unidade para cima;

$y_{dir} < 0$  corresponde a uma deslocação de uma unidade para baixo.

Mais concretamente, para uma determinada posição  $P(x_{pos}, x_{pos})$ , dois pontos são gerados aleatoriamente para o cálculo de  $x_{dir}$  e  $y_{dir}$  respetivamente  $P_1(x_{rand}, y_{rand})$  e  $P_2(x_{rand}, y_{rand})$ . Se a distância de  $P_1$  a  $P(x_{pos}, x_{pos})$  for muito inferior à distância de  $P_2$ , e sendo a constante  $F$  igual em  $x_{dir}$  e  $y_{dir}$ , a componente  $x_{dir}$  é superior à  $y_{dir}$ , pelo que o objeto irá deslocar-se apenas para a direita ou para a esquerda, dependendo do sinal da constante. O pseudocódigo do algoritmo de movimento desenvolvido e implementado está presente no Algoritmo 2. Por sua vez, o pseudocódigo do algoritmo de cálculo da componente  $x_{dir}$  está presente no Algoritmo 3. O cálculo de  $y_{dir}$  segue um raciocínio análogo e por isso não é aqui descrito.

A constante *clock* garante que o robot não permanece mais do que o tempo estipulado (por exemplo 30 deslocações diferentes) num quarto ou escritório, forçando a sua saída quando este limite é atingido. Da mesma forma, foi fixado um valor (contagem) para evitar a imobilidade do robot devido à presença de parede, uma situação que ocorre quando o objeto se encontra num corredor e é forçado na direção de uma parede.

---

**Início**

```

xdir=0; ydir=0; lim=1000;
Enquanto on e não está na posição final faça
xdir←Calcxdir;
ydir←Calcydir;
xant=Objeto.xpos;; // guardar posição atual
yant=Objeto.ypos; // guardar posição atual

if objeto.(xpos+1,ypos+1) não tem parede then
| if  $xdir \in ]-lim, 0[$  e  $ydir \in ]-lim, 0[$  then
| | objeto.direitabaixo
| if  $xdir \in ]-\infty, -lim[$  e  $ydir \in ]-\infty, -lim[$  then
| | objeto.direitabaixo

else if objeto.(xpos+1,ypos-1) não tem parede then
| if  $xdir \in ]-lim, 0[$  and  $ydir \in ]0, lim[$  then
| | objeto.direitacima
| if  $xdir \in ]-\infty, -lim[$  and  $ydir \in ]lim, +\infty[$  then
| | objeto.direitacima

else if objeto.(xpos-1,ypos+1) não tem parede then
| if  $xdir \in ]0, lim[$  and  $ydir \in ]-lim, 0[$  then
| | objeto.esquerdabaixo
| if  $xdir \in ]lim, +\infty[$  and  $ydir \in ]-\infty, -lim[$  then
| | objeto.esquerdabaixo

else if objeto.(xpos-1,ypos-1) não tem parede then
| if  $xdir \in ]0, lim[$  and  $ydir \in ]0, lim[$  then
| | objeto.esquerdacima
| if  $xdir \in ]lim, +\infty[$  and  $ydir \in ]lim, +\infty[$  then
| | objeto.esquerdacima

else if objeto.(xpos-1,ypos) não tem parede then
| if  $xdir \in ]lim, +\infty[$  and  $ydir \in ]-lim, lim[$  e  $ydir \neq 0$  then
| | objeto.esquerda
| if  $xdir \in ]0, +\infty[$  e  $ydir = 0$  then
| | objeto.esquerda

else if objeto.(xpos+1,ypos) não tem parede then
| if  $xdir \in ]-\infty, -lim[$  and  $ydir \in ]-lim, lim[$  e  $ydir \neq 0$  then
| | objeto.direita
| if  $xdir \in ]-\infty, 0[$  e  $ydir = 0$  then
| | objeto.direita

```

```

else if objeto.(xpos,ypos-1)) não tem parede then
  | if xdir ∈ ]-lim,lim[ and ydir ∈ ]lim,+∞[ e xdir≠0 then
  |   | objeto.cima
  | if xdir=0 e ydir ∈ ]0,+∞[ then
  |   | objeto.cima

else if objeto.(xpos,ypos+1)) não tem parede then
  | if xdir ∈ ]-lim,lim[ and ydir ∈ ]-∞,-lim[ e xdir≠0 then
  |   | objeto.cima
  | if xdir=0 e ydir ∈ ]-∞,0[ then
  |   | objeto.baixo

else
  | // encontrou parede
  | objeto.setPos(xant,yant); // permanece na mesma posição
  | contagem+=1

if (objeto.xpos,objeto.ypos).temQuarto then
  | clock+=1
else
  | clock=0

```

**Fim** \_\_\_\_\_

**Algoritmo 2:** Próxima posição

```

if TemCorredor and contagem < 3 then
  if xcomp = 1 then k = 10000000;
  if xcomp = -1 then k = -10000000;
  if xcomp = 0 then newRandomNumber ∈ ]-1000,1000[;
  k = RandomNumber;
if TemQuarto or não TemCorredor or contagem ≥ 3 then
  if TemQuarto and clock > 30 then setPos(xporta, yporta);
  else newRandomNumber ∈ ]-1000000,1000000[;
  k = RandomNumber
New(xrand, yrand);
xdir = 
$$\frac{k}{\sqrt{(xrand-xpos)^2+(yrand-ypos)^2}}$$


```

Algoritmo 3: Cálculo de  $x_{dir}$ 

### 4.3 Previsão da trajetória

De cada vez que o objeto percorre o tabuleiro, o seu movimento é registado num ficheiro de dados estruturado, onde cada coordenada corresponde a um atributo. Estes registos são analisados através de um algoritmo de Data Mining para obter padrões de movimento que são posteriormente usados para prever o próximo caminho percorrido pelo objeto. A previsão do caminho permite estimar os pontos por onde objeto vai passar quando não existem sensores para capturar a sua posição. Desta forma, quando o simulador é iniciado, é imediatamente estimada uma trajetória para o objeto, que reflete os padrões de movimento de trajetórias anteriores.

É importante salientar que quando o ficheiro que contém os registos de localizações se encontra vazio o sistema naturalmente não consegue prever o próximo caminho. Neste caso, prossegue-se com a simulação da trajetória e registo de pontos com a componente de previsão temporariamente inativa.

O algoritmo de Data Mining utilizado, tal como já foi referido, assenta na determinação de um conjunto de *clusters* que na prática correspondem ao conjunto de pontos onde é estimado que o objeto se encontre em algum momento da simulação. A métrica utilizada para o cálculo dos  $k$  centróides,

referidos no Algoritmo 1, foi a distância euclidiana, predefinida na plataforma Weka.

Durante a implementação do algoritmo de movimento, trajetórias de tamanhos muito distintos foram testadas, pelo que se tornou mais prático optar por não fixar o número de *clusters*, e defini-lo automaticamente consoante os registos. Desta forma, o número de *clusters* retribuído pelo Weka é diferente a cada nova simulação e corresponde à média aritmética dos tamanhos de caminhos percorridos pelo robot que foram registados.

O caminho estimado deveria no entanto, por questões de eficiência, ter o mesmo tamanho do caminho percorrido a cada nova simulação mas uma vez que se pretende mostrar a previsão e a simulação em simultâneo, o caminho estimado tem forçosamente de ser calculado previamente ao início da simulação, pelo que o número de *clusters* tem de ser fornecido sem conhecimento prévio do tamanho que a trajetória real corresponde irá apresentar. Como tal, a alternativa adotada consiste em fazer uma estimativa do tamanho do futuro caminho, com base em observações anteriores.

O caminho estimado inicialmente calculado, devolvido pelo agente de previsão, não corresponde no entanto ao caminho estimado final utilizado no conjunto de teste. De facto, o caminho estimado apresentado ao utilizador varia consoante a posição dos sensores e consoante a posição do robot relativamente a estes. Isto significa que o caminho estimado final é geralmente inferior ao caminho estimado inicial. Para clarificar esta questão pode-se tomar como exemplo o cenário em que existem sensores implementados em todas as portas. Neste caso, o caminho estimado inicial contém pontos espalhados por todo o tabuleiro, incluindo pontos dentro de quartos onde o objeto não entra durante todo o movimento. Como existem sensores a detetar a entrada e saída de quartos, prever que o objeto em algum momento do movimento se poderia encontrar num quarto onde nunca foi detetado não faria sentido. No entanto, uma vez que estes pontos existem de facto no caminho estimado inicial (calculado previamente à obtenção da localização por parte dos sensores) tornou-se necessário aplicar uma filtragem a esse caminho. Desta forma, o caminho estimado inicial pode ser modificado ao longo do percurso do robot se certos conjuntos de pontos lhe forem retirados consoante cada caso

particular. Aplicando este raciocínio definiu-se que à medida que o robot se move vão sendo guardados em memória os pontos pertencentes ao caminho estimado inicial que até à posição atual do robot são possíveis localizações. Ou seja, supondo que existem sensores em todas as portas, e que objeto se encontra no primeiro bloco sem ter entrado em nenhum quarto, o caminho estimado final nesse momento é constituído por todos os pontos do caminho estimado inicial que não fazem parte de quartos. Caso o objeto não entre em nenhum quarto até ao fim do movimento, esse será o caminho estimado final usado no cálculo dos resultados. Por outro lado, não havendo sensores implementados, todas as localizações são possíveis; o caminho estimado final e inicial são coincidentes. Na secção 5 serão explicados em detalhe alguns cenários para melhor entendimento desta questão.

## 4.4 Resultados

Uma vez terminado o percurso do objeto, e com base nas localizações previstas e no caminho realmente percorrido, é possível calcular a precisão do modelo e avaliar a ocorrência de erros.

Os resultados lançados no final de cada trajetória permitem ao utilizador ter uma fácil perceção da eficiência do modelo de previsão, ou mais concretamente, da similaridade entre os caminhos - estimado e percorrido - ponto a ponto. A Figura 4.9 apresenta um exemplo de janela de resultados apresentada ao utilizador no final do movimento. A sua respetiva legenda encontra-se na Figura 4.10.

As previsões corretas e incorretas são calculadas iterando o caminho percorrido e o caminho estimado. No contexto desta aplicação em particular não é forçoso, ou relevante, considerar como pontos corretamente previstos apenas aqueles que o robot especificamente cruzou, e uma margem pode ser considerada para a definição de pontos corretamente previstos. De facto, fazendo um paralelismo entre o Hospital e o simulador pode-se dizer que o comprimento de cada unidade de posição (cada ponto ou quadrado) do simulador corresponde a 0.5 metros. Ou seja, traduzindo na prática, prever que o objeto passaria 0.5 metros ao lado do local por onde passou não é neces-

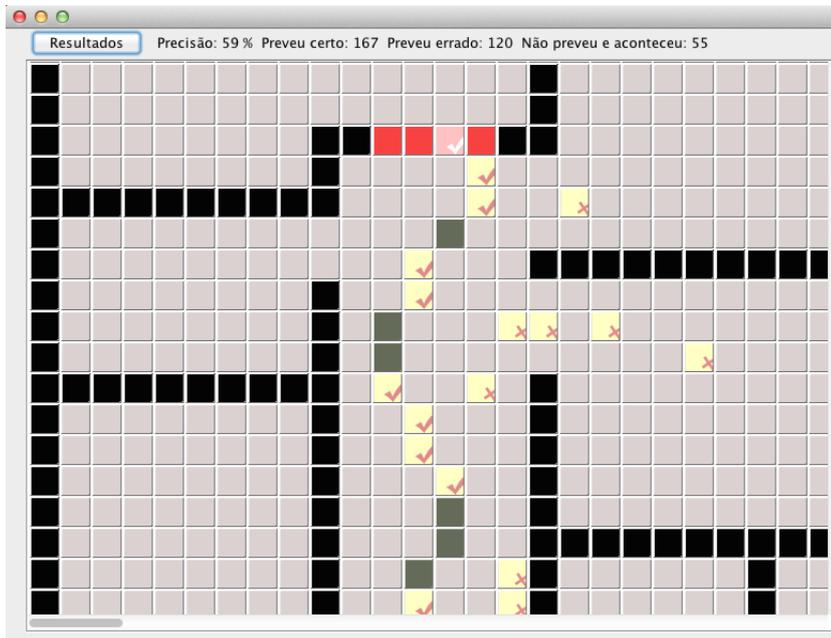


Figura 4.9: Aspeto do ambiente gráfico do simulador: janela de resultados.

Detectado por sensor: 

Preveu certo: 

Preveu errado: 

Não preveu e aconteceu: 

Figura 4.10: Legenda da janela de resultados

sariamente uma previsão errada se uma certa margem de erro for tolerada. Desta forma, definiu-se um raio para o qual a previsão de caminho continua correta, que consiste em considerar todos os pontos circundantes daquele que foi realmente previsto.

Os resultados calculados e apresentados no final de cada trajetória são os seguintes:

**Preveu certo:** número total de pontos corretamente previstos por Data Mining;

**Preveu errado:** número de pontos onde foi previsto que o objeto passasse em algum momento da sua trajetória e não passou;

**Não preveu e aconteceu:** número total de pontos onde não foi previsto que o objeto passasse e por onde efetivamente passou;

**Precisão:** razão entre o número total de pontos previstos corretamente e o número total previsões, dada pela seguinte fórmula:

$$Precisão = \frac{\textit{preveu certo}}{\textit{preveu certo} + \textit{preveu errado}} \times 100$$

Tal como já foi referido, todos estes valores são guardados, a cada iteração, numa base de dados para uma posterior análise.



# Capítulo 5

## Resultados

No presente capítulo pretende-se apresentar e ilustrar os diversos cenários de simulação estudados. Será feita a análise de 4 diferentes cenários com o objetivo de avaliar quantitativamente a influência da posição e alcance dos número de sensores na precisão do modelo, através da variação deste parâmetros.

Para cada um destes cenários, um novo ficheiro de registo de movimentos foi criado com o intuito de avaliar a progressão da precisão do sistema para cada ambiente separadamente. Assim, o primeiro registo de resultados de cada um dos cenários corresponde à primeira previsão feita para um conjunto de 4 simulações prévias, realizadas com o intuito de servir de base para o início do sistema simulação/previsão. Desta forma, cada um dos cenários é avaliado com conjuntos de dados diferentes, obtidos em percursos anteriormente registados, já que cada um é simulado independentemente dos outros.

Serão apresentadas janelas da interface gráfica obtidas sob diferentes circunstâncias e será discutida a questão do desempenho do modelo de previsão em cada cenário em termos de precisão média. Para tal, apresentar-se-ão gráficos de evolução da precisão em função do número de registos, será realizada uma comparação entre os resultados obtidos em cada cenário e uma avaliação global do desempenho do modelo de previsão.

Uma vez que as trajetórias simuladas pretendem aproximar-se ao máximo de possíveis rotinas de pessoas, para um determinado local de origem do

percurso existem pelo menos dois diferentes locais de destino, com probabilidades diferentes de serem escolhidos. Sob este ponto de vista pretende-se também avaliar a precisão do sistema quando a pessoa se desvia do seu percurso mais habitual.

Além da posição dos sensores também a variação de outros parâmetros é analisada: as trajetórias definidas e o número de objetos que o sistema observa. Tal como já foi mencionado, diferentes trajetórias podem ser definidas pela simples alteração da direção associada aos corredores. Estas serão apresentadas juntamente com a simulação simultânea de mais do que um objeto em movimento.

Em qualquer um dos cenários, sempre que o objeto é captado numa zona em que existe um sensor de deteção de movimento, o modelo de previsão é atualizado com essa informação. Desta forma, quando o objeto está de facto a ser localizado num ponto específico, a janela de previsão mostra a sua localização presente e não uma previsão. As Figuras mostram respetivamente as duas interfaces, simulação e previsão, no mesmo instante em que a presença física do objeto é detetada na zona de um sensor colocado na entrada de uma divisão.

As trajetórias estipuladas nos cenários em que apenas se monitoriza o movimento de um objeto, definidas através da conjugação de diferentes direções nos corredores estão presentes na Figura 5.1. Partindo sempre do mesmo ponto (elevador 3), o objeto termina o seu movimento no elevador 1 ou 4, sendo a probabilidade de terminar no primeiro muito superior devido à configuração dos corredores.

## 5.1 Cenário 1: sensores nas entradas e saídas

No primeiro caso de estudo a simulação de trajetória e o Data Mining foram realizados para a situação em que não existem sensores distribuídos ao longo da planta, exceto aqueles nos elevadores essenciais para sinalizar o início e o fim da trajetória. A Figura 5.2 mostra a janela do caminho previsto neste caso.

As localizações previstas (pontos marcados a azul) estão distribuídas ao

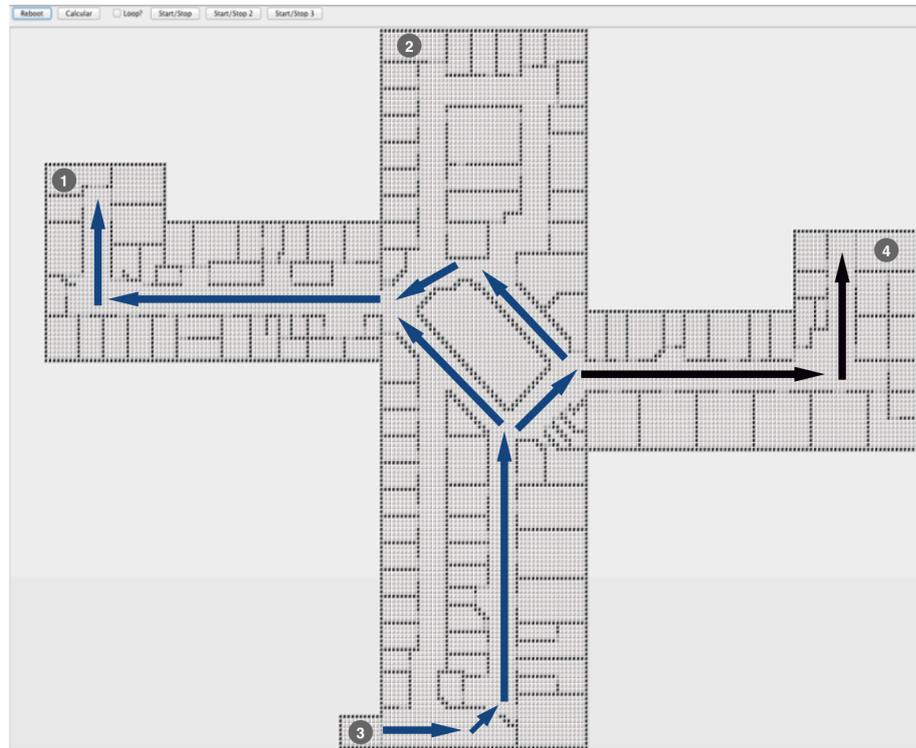


Figura 5.1: Possíveis percursos do objeto e localização dos elevadores.

longo de toda a área disponível, em qualquer momento da simulação, uma vez que não existem sensores a delimitar a possível localização do objeto. Desta forma, quando os únicos sensores presentes são os que marcam a entrada e saída de elevadores o caminho estimado engloba os 5 blocos de circulação (4 alas e bloco central) e qualquer um dos pontos a azul é uma possível localização para o robot em qualquer fase do seu movimento.

Os resultados obtidos no final de 300 percursos, presentes nas Tabelas 5.1 e 5.2, mostram que o sistema é capaz de prever as localizações do objeto com uma precisão média de 56%, num ambiente em que este apenas é detetado em 2 sensores RFID de deteção de movimento. Este valor aumenta para 59% quando o objeto descreve o seu trajeto típico e decresce para 34% quando foge dessa mesma trajetória. É importante salientar que por percurso típico entende-se o caminho assinalado a azul na Figura 5.1 onde a direção do objeto é a mesma indicada pelas setas. Este percurso engloba dois possíveis

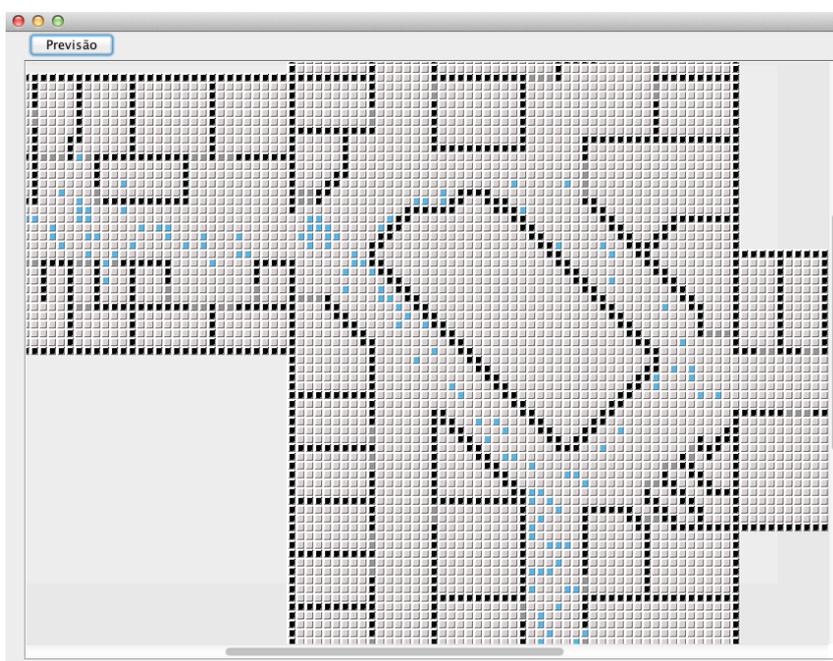


Figura 5.2: Previsão de trajetória: cenário 1.

caminhos, dependendo da direção seguida no bloco central, e conduz o objeto a abandonar o piso pelo elevador 1. O caminho atípico, assinalado a preto, serve para exemplificar que o sistema é capaz de, aleatoriamente, provocar uma alteração na rotina da pessoa. Em 300 percursos o objeto seguiu 267 vezes pelas mesmas alas do hospital, ou seja, pelo caminho assinalado a azul, e 33 vezes por um bloco diferente do habitual (bloco 4).

O gráfico da Figura 5.3 mostra a evolução da precisão ao longo de diferentes iterações, em função da zona por onde a pessoa abandonou o piso.

Cenário 1	
<b>Sensores nas entradas e saídas</b>	
Número de registos:	300
<b>Média da precisão:</b>	<b>56.3%</b>
<b>Número de sensores:</b>	<b>4</b>

Tabela 5.1: Relatório de precisão média: cenário 1.

## 5.2. CENÁRIO 2: SENSORES A DELIMITAR TODOS OS CORREDORES PRINCIPAIS

### Cenário 1

Precisão em função do destino

Terminou percurso no elevador1

Número de registos	média da precisão
267	59.1%

Terminou percurso no elevador4

Número de registos	média da precisão
33	34.3%

Tabela 5.2: Relatório de precisão média em função do percurso: cenário 1.

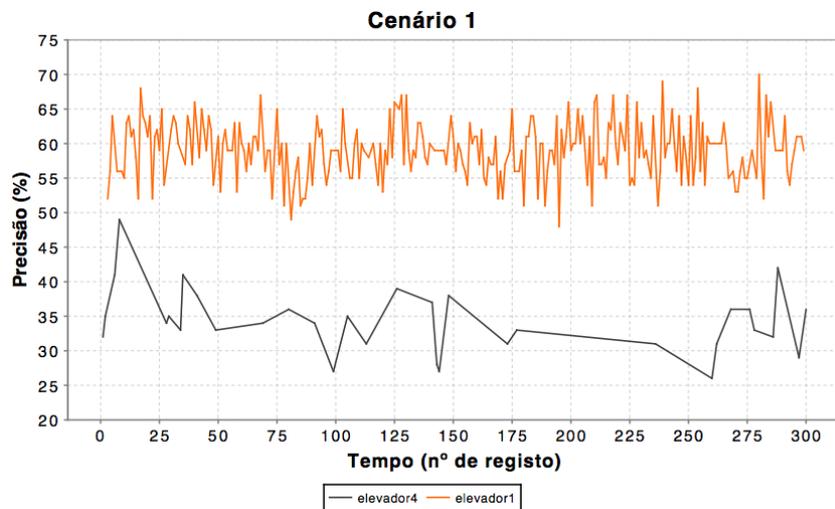


Figura 5.3: Gráfico da precisão em função do número de registos: cenário 1.

## 5.2 Cenário 2: sensores a delimitar todos os corredores principais

Uma forma de reduzir a possível localização atual do objeto é colocar sensores a limitar as entradas e saídas dos corredores, limitando os 5 blocos em 5 áreas distintas. Neste caso o simulador não apresenta ao utilizador estimativas de localizações para blocos onde o objeto não entra, e as localizações previstas são restritas a uma certa área, dependendo do último sensor onde o objeto foi detetado.

Em termos práticos, esta abordagem permite melhorar a precisão do sistema através da redução do número de pontos incorretamente previstos, exis-

tentes em áreas não frequentadas pelo objeto, que influenciariam negativamente a precisão da previsão caso fossem considerados.

Na próxima série de figuras pretende-se exemplificar a dinâmica da apresentação das localizações previstas ao utilizador. As Figuras 5.4 e 5.5 mostram respetivamente a posição atual do objeto, com zoom no bloco central, e o correspondente caminho estimado para aquele instante, constituído unicamente por pontos incluídos na zona delimitada por sensores. Quando o objeto muda de zona (Figura 5.6) a estimativa da sua localização é imediatamente alterada, passando a representar outro conjunto de possíveis localizações (Figura 5.7).

Pela observação da Figura 5.6 depreende-se que o objeto terminará o seu percurso no bloco lateral esquerdo (bloco 1), uma vez que se sabe que o algoritmo definido não permite que volte no sentido inverso.

Uma vez terminado o percurso, a janela do caminho previsto é atualizada, passando a apresentar todo o caminho previsto para aquela trajetória, como mostra a Figura 5.8. Os resultados obtidos, presentes na Figura 5.9 mostram que neste caso o sistema previu a trajetória do robot com uma precisão de 66%.

Fazendo uma média aritmética da precisão obtida em vários movimentos, onde a previsão a cada iteração é feita com base em todas as trajetórias/iterações anteriores, obtém-se o valor de 60.5%, tal como mostra a Tabela 5.3. No entanto, tal como no cenário anterior, se a precisão for analisada separadamente para as duas situações distintas, referentes a trajetórias típicas e atípicas, este valor cresce para 61.3% para o caso em que o objeto se comporta dentro dos seus parâmetros normais e decresce para 50.6% nos cenários em que se verifica um comportamento diferente do habitual (Tabela 5.4).

O gráfico da evolução da precisão em função do número de registos encontra-se na Figura 5.10.

### 5.3. CENÁRIO 3: REPRODUÇÃO DOS SENSORES EXISTENTES NO HOSPITAL79

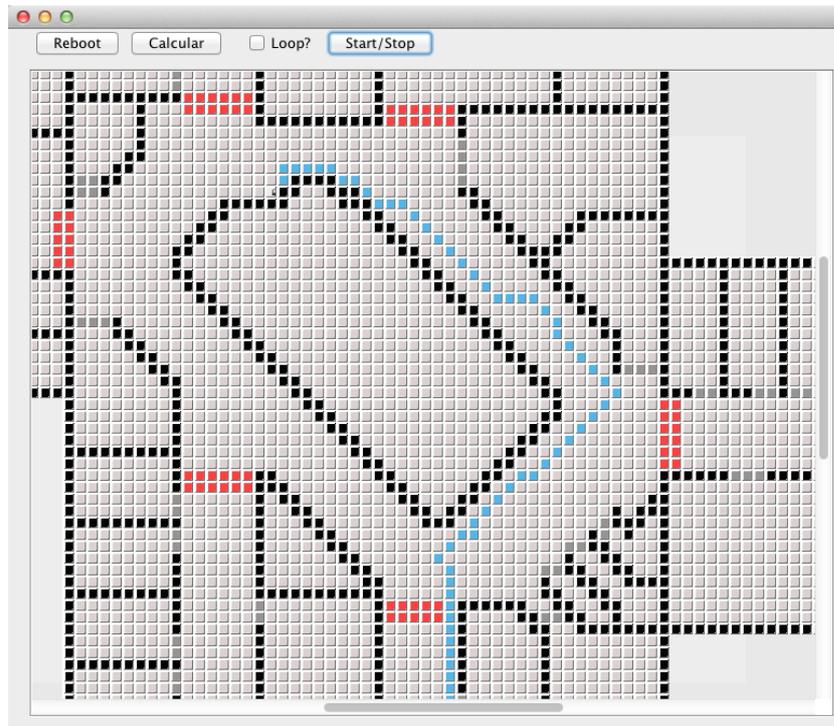


Figura 5.4: Simulação de trajetória: cenário 2.

---

#### Cenário 2

Sensores a delimitar os corredores principais

Número de registos: 300

Média da precisão: 60.1%

Número de sensores: 10

---

Tabela 5.3: Relatório de precisão média: cenário 2.

## 5.3 Cenário 3: reprodução dos sensores existentes no Hospital

Neste cenário a disposição dos sensores foi feita com base na informação referente à distribuição de sensores no piso hospitalar (Figura 1.2 na página 12). Apesar de não existir sensor de deteção de movimento do elevador 3 do piso hospitalar, uma vez que na série de estudos realizados se estipulou que o objeto dava entrada no piso por esta zona decidiu-se manter o mesmo no

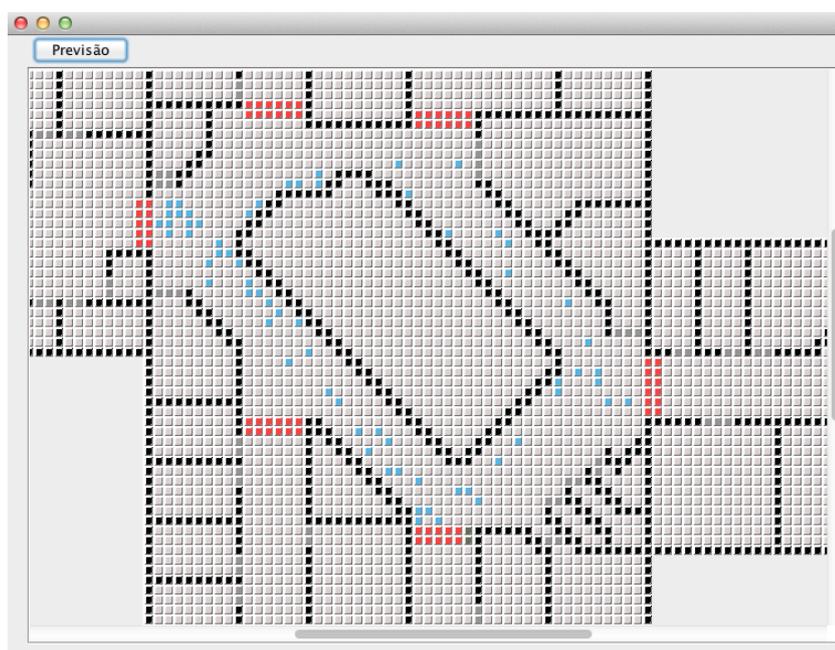


Figura 5.5: Previsão de trajetória: cenário 2.

## Cenário 2

### Precisão em função do destino

#### Terminou percurso no elevador1

Número de registos	média da precisão
267	61.3%

#### Terminou percurso no elevador4

Número de registos	média da precisão
33	50.6%

Tabela 5.4: Relatório de precisão média em função do percurso: cenário 2.

ambiente de simulação para efeitos práticos. A sua existência não influencia os resultados uma vez que serve apenas mais uma vez para simular que o aparecimento do objeto foi detetado nessa zona. Assim, os sensores introduzidos na planta do simulador correspondem aos sensores de deteção de movimento atualmente em funcionamento no hospital.

O robot iniciou o seu percurso na zona circundante às escadas/elevadores, seguiu no sentido ascendente e pelo caminho visitou um dos quartos presentes nessa área (Figura 5.13). Chegando à zona central do piso, três alternativas

### 5.3. CENÁRIO 3: REPRODUÇÃO DOS SENSORES EXISTENTES NO HOSPITAL81

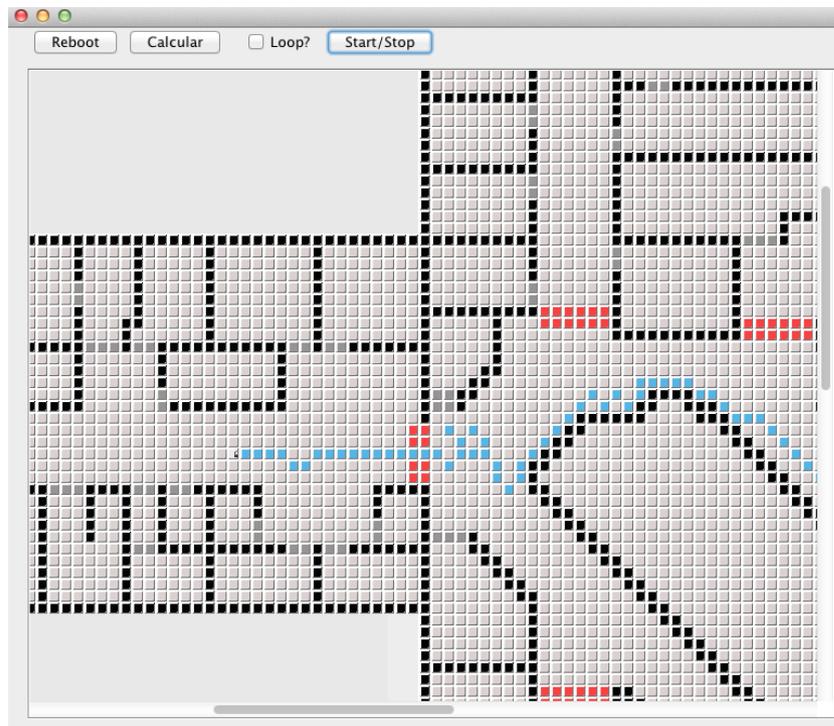


Figura 5.6: Simulação de trajetória: cenário 2.

são possíveis: seguir pela direita e de seguida pelo bloco 4; seguir pela direita e continuar para cima em direção ao bloco 1; seguir pela esquerda e dirigir-se ao bloco 1. Neste caso, a aleatoriedade associada ao facto de ser mais provável que o robot siga pela esquerda, determinaram que o robot prosseguisse nesse sentido (Figura 5.11) terminando o seu tracking no momento em que entra na zona associada ao elevador (Figura 5.14). As Figuras 5.11 e 5.12 mostram respetivamente a simulação de movimento e as correspondentes localizações previstas, num determinado momento em que a presença do objeto foi detetada no bloco central, no cenário em que se reproduziu a atual disposição de sensores RFID do Hospital.

No momento em que o robot chega ao seu destino, a janela do caminho estimado (Figura 5.15) representa todos os pontos que farão parte do conjunto que será usado na avaliação do desempenho do sistema e onde não estão incluídos pontos pertencentes a zonas delimitadas por sensores que o objeto não percorreu.

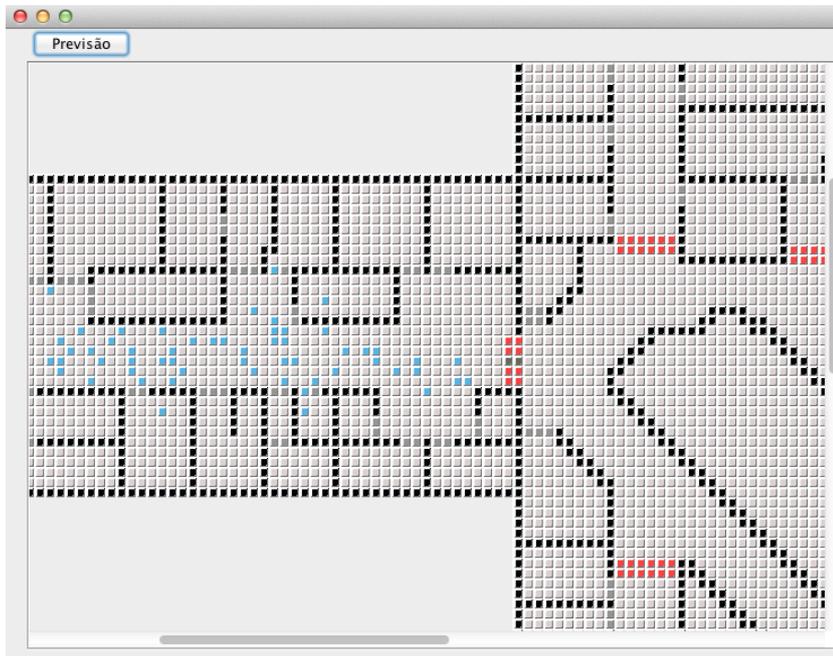


Figura 5.7: Previsão de trajetória: cenário 2.

Os resultados obtidos podem ser visualizados no Pentaho Report Designer, que a qualquer momento pode ser atualizado para fornecer uma atualização do desempenho do sistema para os diferentes cenários. Neste caso, quando o total de diferentes movimentos descritos pelo robot ao longo do tempo atingiu o número 300, a precisão média do modelo de previsão registada corresponde a cerca de 62.5% (Figura 5.5). Este valor varia ligeiramente (cerca de 6 pontos percentuais) quando se analisa apenas as situações em o movimento termina no elevador 1, uma vez que o valor obtido para trajetórias incomuns, mais concretamente 8% das trajetórias, se distancia apenas 7.5 pontos percentuais (Tabela 5.6).

<b>Cenário 3</b>	
<b>Reprodução da atual distribuição de sensores no Hospital</b>	
Número de registos:	300
<b>Média da precisão:</b>	<b>62.5%</b>
<b>Número de sensores:</b>	<b>10</b>

Tabela 5.5: Relatório de precisão média: cenário 3.

### 5.3. CENÁRIO 3: REPRODUÇÃO DOS SENSORES EXISTENTES NO HOSPITAL83

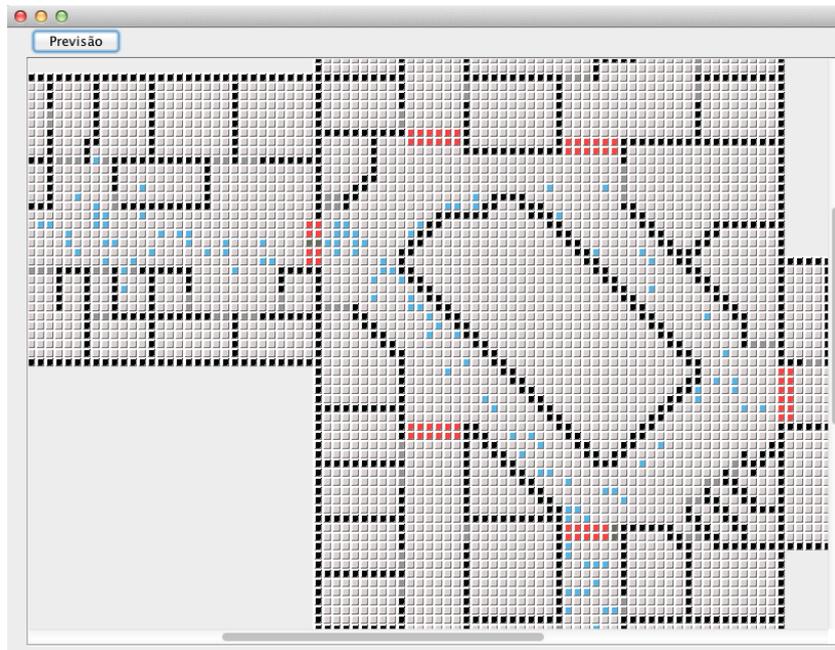


Figura 5.8: Previsão de trajetória final: cenário 2.

#### Cenário 3

##### Precisão em função do destino

##### Terminou percurso no elevador1

Número de registos	média da precisão
277	63.1%

##### Terminou percurso no elevador4

Número de registos	média da precisão
23	55.6%

Tabela 5.6: Relatório de precisão média em função do percurso: cenário 3.

O gráfico da evolução da precisão em função do número de registos para este cenário está presente na Figura 5.16.

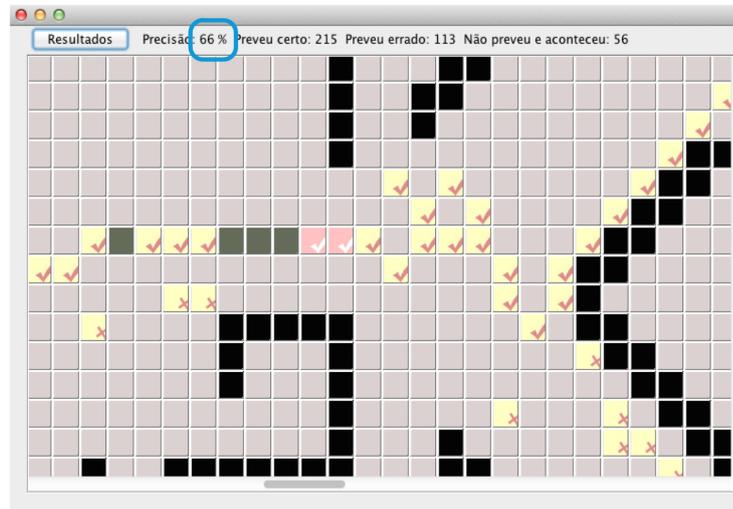


Figura 5.9: Resultados de previsão: cenário 2.

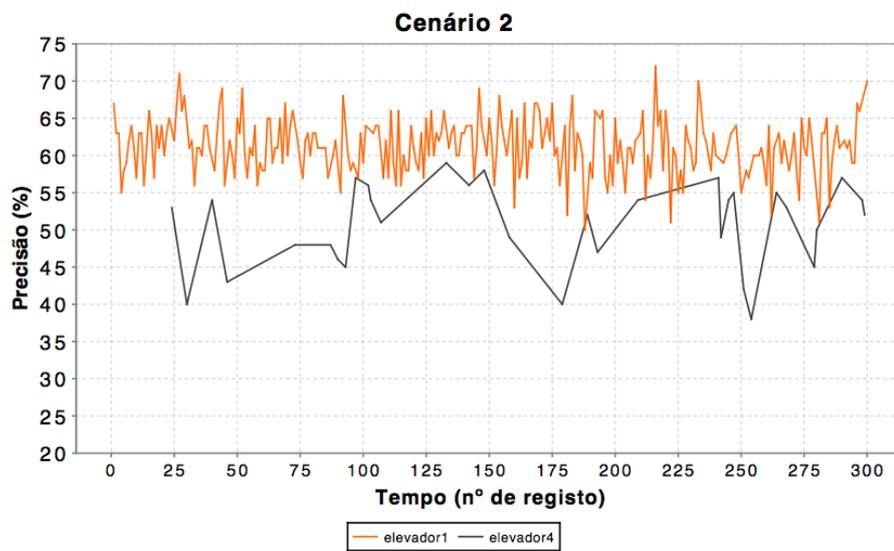


Figura 5.10: Gráfico da precisão em função do número de registos: cenário 2.

5.3. CENÁRIO 3: REPRODUÇÃO DOS SENSORES EXISTENTES NO HOSPITAL85

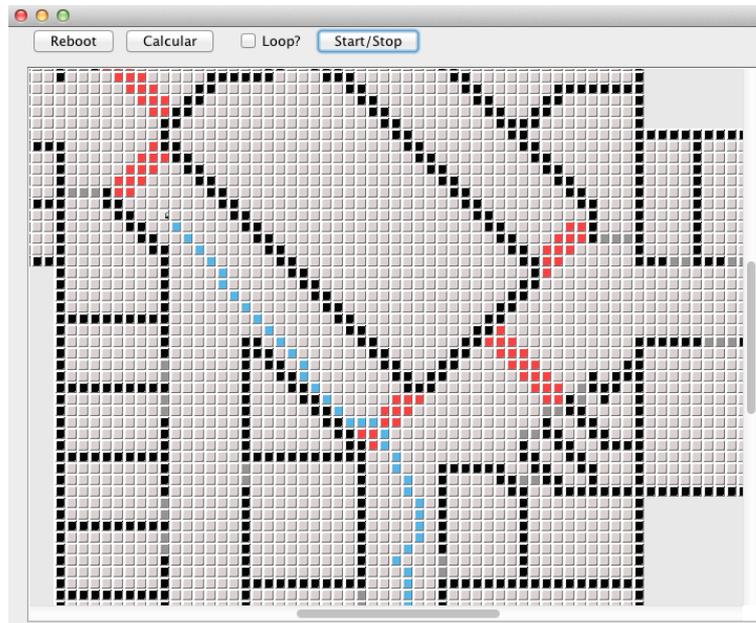


Figura 5.11: Simulação de trajetória: cenário 3

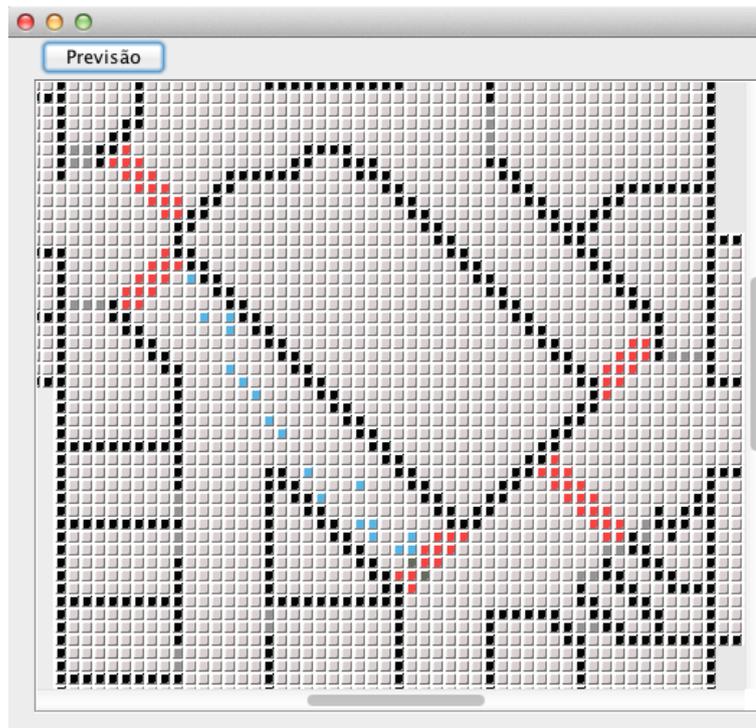


Figura 5.12: Previsão de trajetória: cenário 3.

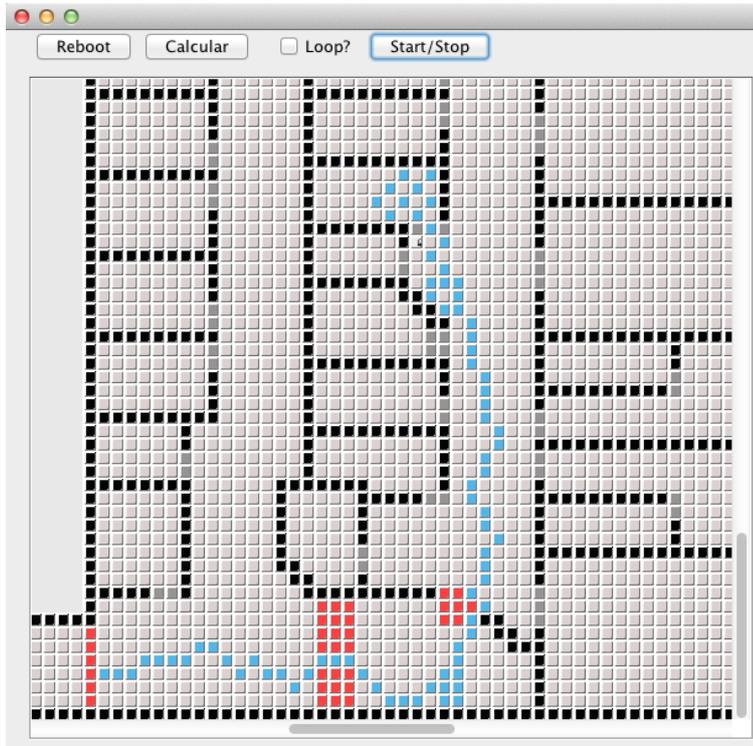


Figura 5.13: Simulação de trajetória: cenário 3.

## 5.4 Cenário 4: colocação de sensores nas portas de divisões

Nos cenários anteriores analisaram-se os casos em que a previsão é limitada a zonas de circulação grandes, como corredores ou blocos inteiros. Neste cenário, pretende-se salientar o facto de que o sistema de simulação funciona de igual forma se a zona limitada por sensor estiver restrita a 4 portas. Ou seja, tal como é capaz de fornecer um modelo de previsão com base nas grandes áreas onde o objeto possivelmente se encontra, o sistema é também capaz de limitar os pontos previstos a um quarto, se se der o caso de este conter um sensor a detetar a entrada ou saída de objetos ou pessoas com *tags* RFID.

Apesar de não ser uma opção viável para o hospital em termos de custos associados ao investimento e implementação de sensores de deteção de movi-

#### 5.4. CENÁRIO 4: COLOCAÇÃO DE SENSORES NAS PORTAS DE DIVISÕES87

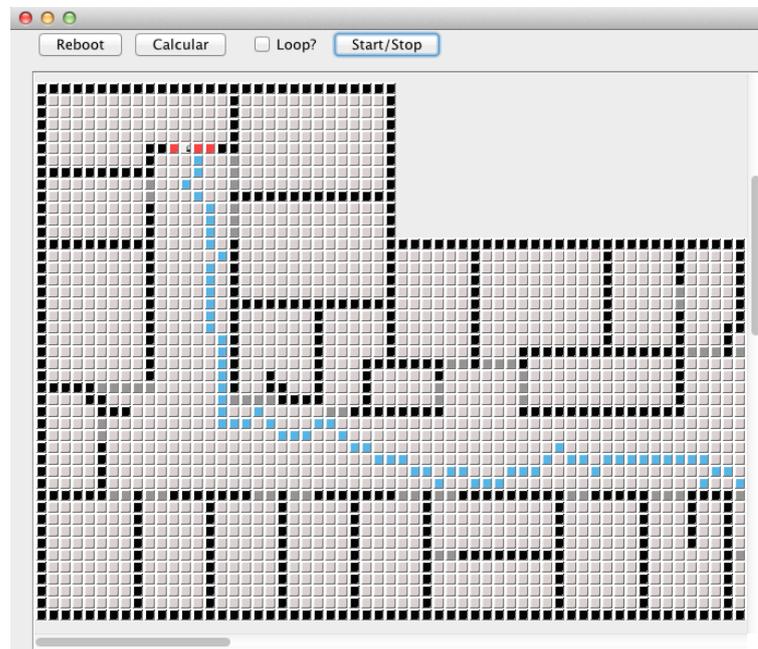


Figura 5.14: Simulação de trajetória: cenário 3.

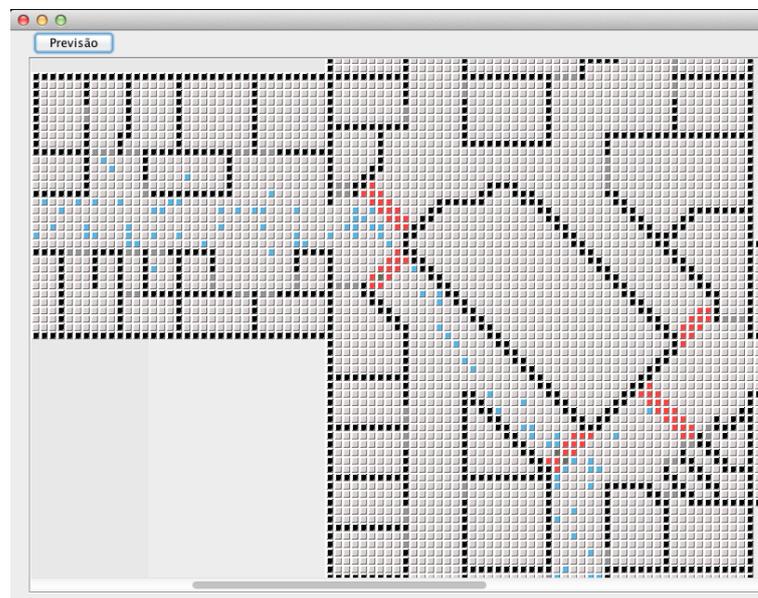


Figura 5.15: Previsão final de trajetória: cenário 3.

mento nas portas, neste cenário é estudado o caso em que existem sensores a cobrir todas as portas do piso hospitalar.

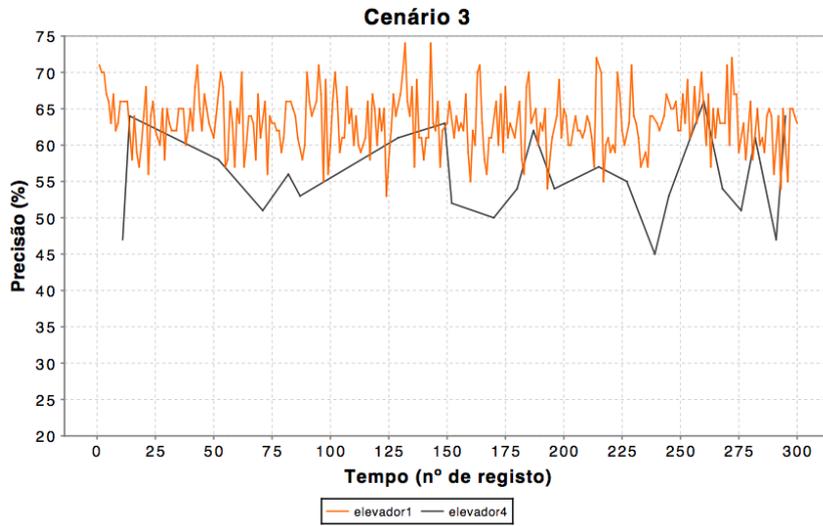


Figura 5.16: Gráfico da precisão em função do número de registos: cenário 3.

Inicialmente ponderou-se seguir uma abordagem em que apenas alguns sensores seriam colocados a controlar as entradas nos quartos. No entanto, os quartos que são frequentemente visitados por um objeto não são forçosamente os que são visitados por um outro objeto, pelo que a abordagem apresentada serve de caso geral para qualquer objeto ou qualquer quarto onde seja importante detetar, por exemplo entradas não autorizadas. Desta forma, a simulação com sensores em todas as portas serve apenas para exemplificar o modo de funcionamento do sistema quando é detetado que o objeto se encontra num quarto.

Os sensores cobrem todo o comprimento de entrada e saída e dos quartos, assegurando que a passagem do objeto é sempre detetada. Assim, se o objeto foi detetado pela última vez no interior de um quarto, o caminho estimado nesse momento vai ser constituído apenas pelos pontos previstos por Data Mining que se encontrem nesse mesmo quarto.

Neste cenário o sistema é capaz de limitar eficazmente o caminho previsto a uma área reduzida e coincidente com a realidade. No entanto, no decorrer das experimentações verificou-se que este método estava dotado de um pequena falha: se no conjunto de pontos inicialmente previstos não existissem

#### 5.4. CENÁRIO 4: COLOCAÇÃO DE SENSORES NAS PORTAS DE DIVISÕES 89

pontos referentes ao quarto onde o objeto entrou (porque não é frequente visitar esse quarto), o modelo de previsão não apresentava quaisquer pontos, o que da ótica do utilizador resultaria numa completa ignorância face à possível localização do objeto. Para resolver este problema, optou-se por impor ao agente de previsão a inclusão de um ponto no centro do quarto, que apesar de não ter sido determinado por Data Mining é acrescentado à previsão e permite ao utilizador perceber a presença do objeto na zona correta. No momento demonstrado na Figura 5.17, em que a posição do objeto é referente ao interior de um quarto, a previsão é dada pelos únicos pontos calculados nesse espaço (neste caso apenas um ponto).

No final do percurso a janela do caminho estimado é atualizada de forma a ser possível visualizar a previsão final, que inclui pontos no interior de quartos visitados pelo robot, determinados por Data Mining ou por inferências obtidas a partir da realidade, e de todos os pontos previstos para os corredores.

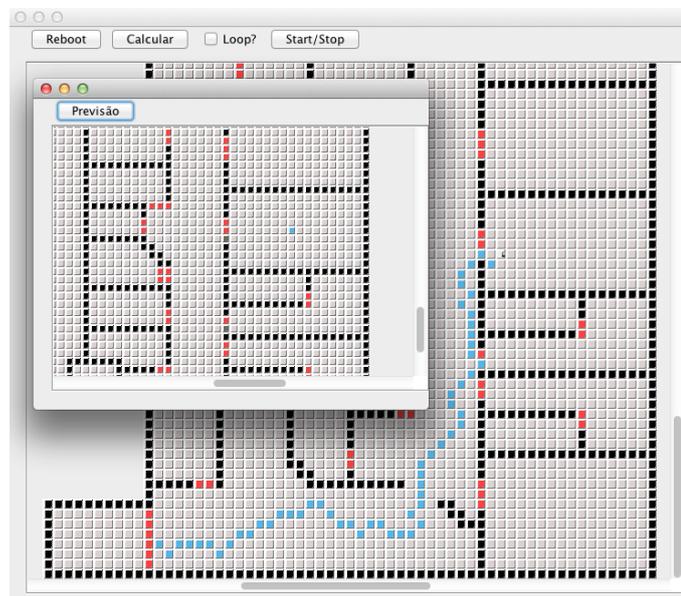


Figura 5.17: Simulação e previsão: cenário 4.

## 5.5 Cenário 5: simulação com vários objetos em simultâneo

Até aqui os cenários centraram-se na demonstração de diferentes percursos, na exemplificação da dinâmica da interface gráfica e na análise do modelo de previsão e dos resultados em função da localização, número de sensores e percurso seguido. Até aqui o sistema operou com um objeto apenas. No entanto, os sistemas de monitorização de pessoas e equipamentos com recurso a redes de sensores são desenvolvidos e aplicados no sentido de realizar a monitorização simultânea de várias entidades na mesma região. O presente cenário pretende demonstrar que o sistema desenvolvido funciona de igual forma quando a simulação é realizada com mais do que um objeto a percorrer o tabuleiro ao mesmo tempo.

O teste foi realizado para 3 objetos, identificados através do seu id. O objeto 1, representado pela cor azul claro, inicia o seu percurso no elevador 3 e termina no elevador 1 ou 4. Por sua vez, os objetos 2 e 3, representados respetivamente a azul escuro e verde, são inicialmente detetados no elevador 1 sendo o seu destino o elevador 3 ou 4. Estas duas trajetórias, que podem ser visualizadas na Figura 5.18, foram estipuladas a título exemplificativo.

Os objetos pretendem representar entidades distintas (pacientes, *staff* ou equipamentos) que ao longo do tempo vão alterando a sua localização dentro do hospital. Por este motivo, o registo de percursos efetuados por cada entidade em representação, é armazenado em diferentes ficheiros para que a aplicação do algoritmo de Data Mining possa ser realizado sobre cada um independentemente. Desta forma, os diferentes objetos seguem percursos inteiramente independentes entre si e a previsão de localização de cada um é apresentada em diferentes janelas. Além disso, o instante em que os objetos iniciam o movimento difere de objeto para objeto uma vez que o movimento de cada um é controlado pelo utilizador através dos três botões que existem de Start/Stop, criados para o efeito.

Na Figura 5.19 captou-se o momento em que 3 objetos se encontram relativamente próximos. O objeto 2, por exemplo, inicialmente detetado no sensor existente na zona de elevador do bloco 1, permaneceu durante algum

## 5.5. CENÁRIO 5: SIMULAÇÃO COM VÁRIOS OBJETOS EM SIMULTÂNEO91



Figura 5.18: Possíveis percursos estipulados para os 3 objetos.

tempo no interior de um quarto e de seguida dirigiu-se para o bloco central. As Figuras 5.20, 5.21 e 5.22 mostram as janelas de previsão obtidas para cada entidade no instante captado na Figura 5.19. Por não ser um percurso habitualmente descrito pelo objeto 2, o número de pontos previstos para o corredor onde este se encontra é consideravelmente reduzido.

À medida que os objetos abandonam o piso através das saídas existentes, os resultados da previsão vão também sendo apresentados ao utilizador em diferentes janelas numa sucessão de eventos similar à exemplificada na Figura 4.4 da página 56.

Os objetos podem ser programados para se moverem a velocidades diferentes (Figura 5.23).

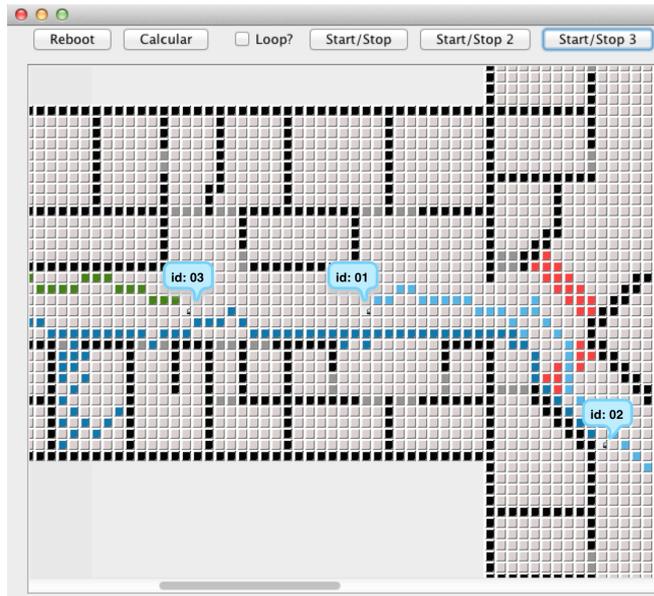


Figura 5.19: Simulação de monitorização de 3 objetos em simultâneo.

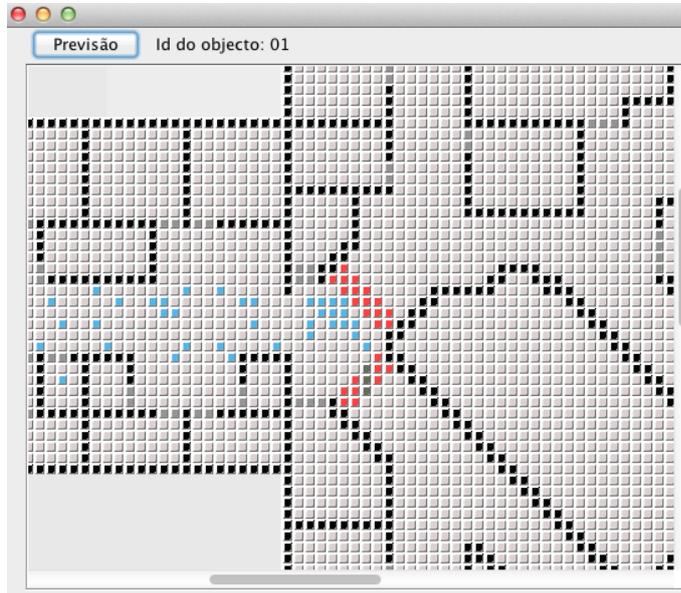


Figura 5.20: Previsão associada ao objeto com id 1.

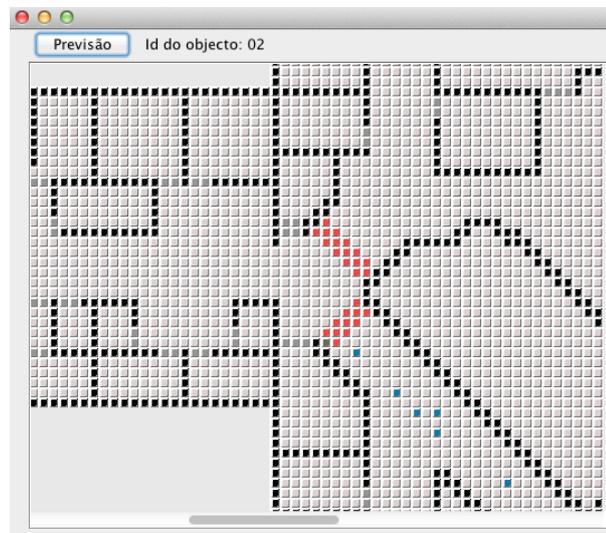


Figura 5.21: Previsão associada ao objeto com id 2.

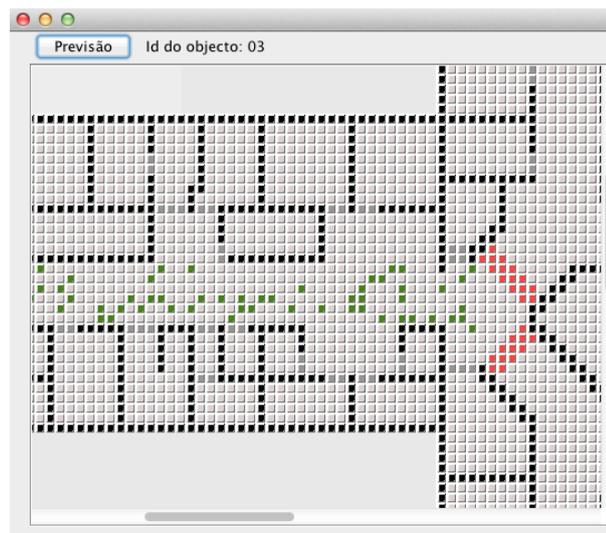


Figura 5.22: Previsão associada ao objeto com id 3.

## 5.6 Análise dos cenários

O primeiro cenário, referente a simulações com 4 sensores nas entradas e saídas, foi estudado para mostrar a influência da localização e número de sensores no desempenho do modelo de previsão, através da comparação com os cenários em que se simulou o movimento numa área preenchida com senso-

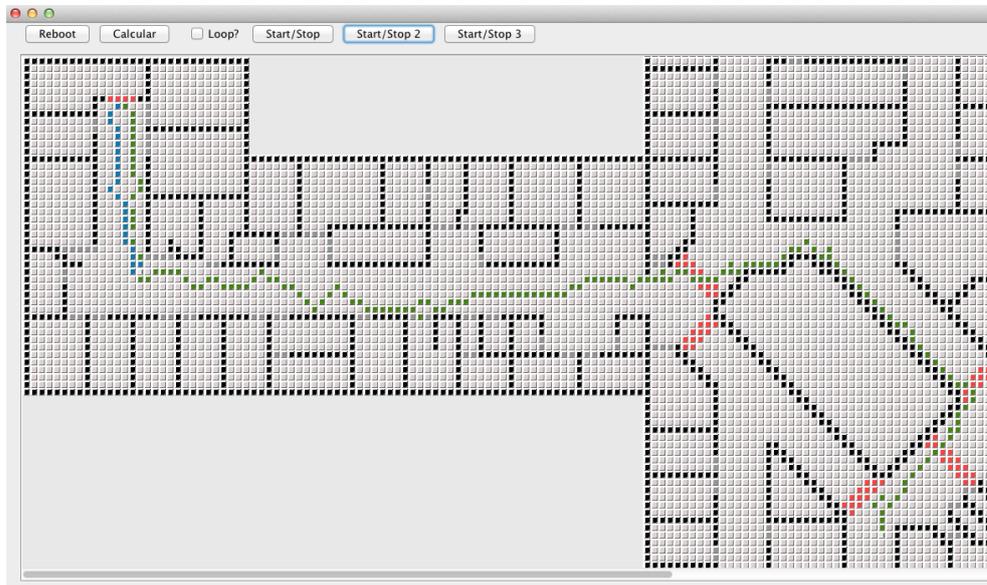


Figura 5.23: Objetos que se movem a velocidades diferentes.

res. Os resultados obtidos, resumidos na tabela 5.7, mostram que o sistema consegue prever a localização de um objeto com uma precisão média superior a 60% nos dois cenários em que existem sensores a detetar movimento nos corredores do hospital, e com uma precisão média de 56% quando não existem sensores. No cenário 2 a precisão média foi de 60.1%, e no cenário 3 de 62.5%, o que se traduz numa diferença percentual, comparativamente ao cenário 1, de 3.7 pontos no primeiro caso e 6.1 no segundo.

Análise de cenários				
Destino	Cenário 1	Cenário 2	Cenário 3	Cenário 4
elevador1	59.1%	61.3%	63.1%	62.2%
elevador4	34.3%	50.6%	55.6%	32.9%
Precisão	56.4%	60.1%	62.5%	60.8%

Tabela 5.7: Relatório de precisão média para os diferentes cenários.

Esperava-se contudo que esta diferença fosse significativamente acentuada, o que na prática não se verificou. Através da análise e comparação entre as previsões finais, obtidas em diversos movimentos pertencentes a diferentes cenários, podem-se enunciar os motivos pelos quais a melhoria do desempenho não é mais significativa. Em primeiro lugar, convém destacar a

disparidade existente entre os valores médios de precisão obtidos para diferentes pontos de destino do objeto. A Tabela 5.7 mostra esses valores para cada um dos cenários em estudo. A partir da análise da tabela, da visualização de movimentos idênticos em cenários distintos e dos caminhos percorridos em diferentes registos, podem apontar-se algumas razões para explicar a reduzida diferença verificada entre o desempenho do modelo nos casos em que existem sensores e no caso em que não existem.

Pela observação das Figuras 5.24 e 5.25, referentes a um percurso efetuado pelo objeto no cenário 2, percebe-se que a distribuição de localizações previstas para o bloco onde o objeto se encontra é muito reduzida, contendo apenas 3 pontos. A simulação pertence ao registo número 87 do cenário 2 e, tal como se pode ver na Figura 5.10, nesse momento o objeto tinha passado apenas 5 vezes nesse mesmo bloco (bloco 4), o que justifica a fraca aglomeração de previsões nesta zona. Uma vez que existem sensores a limitar a zona central do piso, os pontos inicialmente previstos para o bloco 1, onde seria mais comum encontrar o objeto devido aos comportamentos anteriormente descritos (como se pode ver, por exemplo, na Figura 5.7), não são aceites como previsões incorretas. Contrariamente, se o mesmo percurso se realizasse num ambiente de simulação sem sensores, todos os pontos do bloco 1 seriam considerados previsões incorretas, o que influenciaria negativamente o valor da previsão.

Analisando a Tabela 5.8, que contém registos de alguns dos movimentos simulados no segundo cenário, obtêm-se os valores associados ao registo 87 referentes a: número de localizações inicialmente previstas – 264; tamanho do caminho percorrido – 211 e tamanho do caminho previsto final - 156. A diferença de 108 pontos entre a previsão inicial e final representa o número de pontos que num cenário sem sensores seriam adicionados ao número de previsões incorretas. Ou seja, a precisão para a mesma trajetória, caso esta decorresse no ambiente referido no primeiro cenário, seria 30%, 18% abaixo do valor obtido (Figura 5.26). Esta constatação é válida se se ignorar o facto de que o Data Mining é realizado sobre um conjunto de registos diferentes, evidência que não tem implicações nesta questão uma vez que o objeto é o mesmo em todos os cenários supracitados.

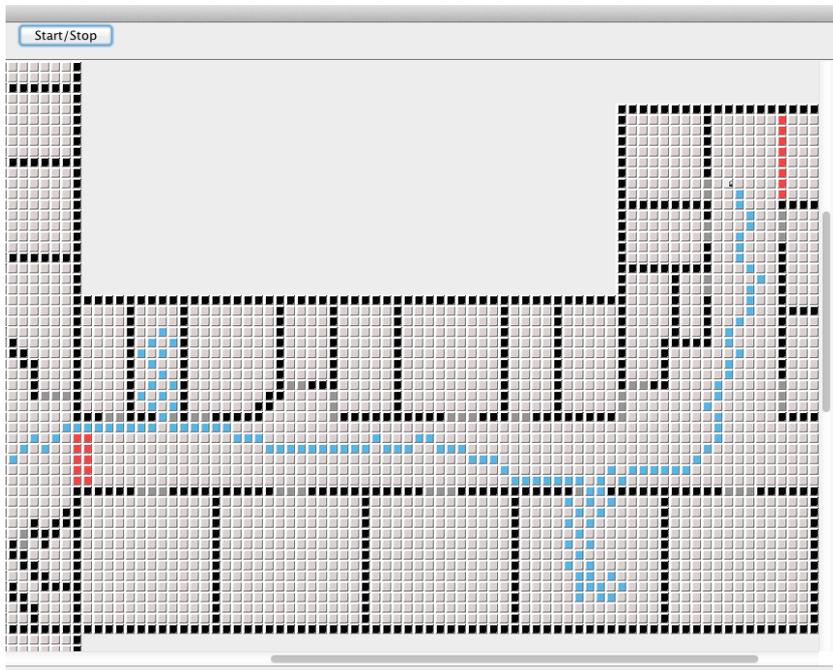


Figura 5.24: Simulação de trajetória no bloco 4: cenário 2

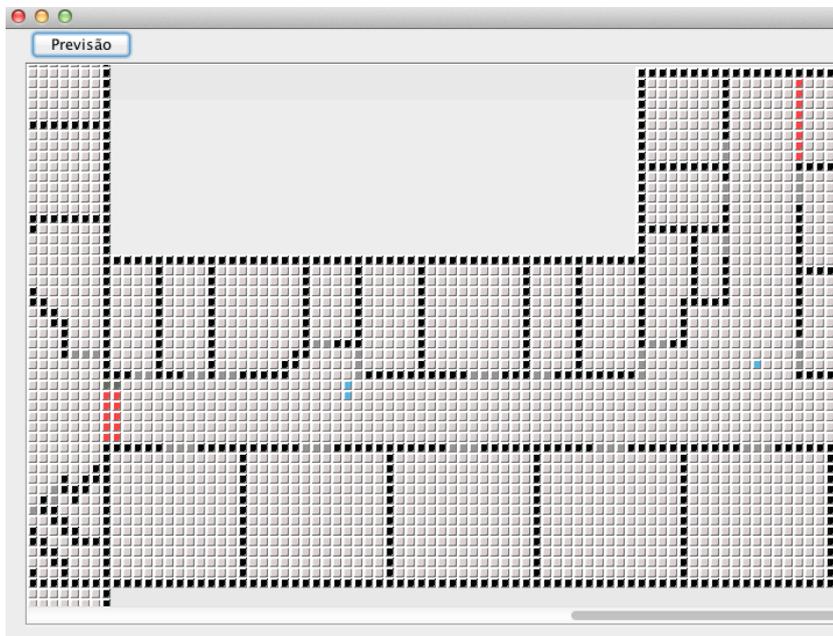


Figura 5.25: Previsão de trajetória no bloco 4: cenário 2

Relativamente ao valor da precisão obtido na Figura 5.26 - 48% - podem-se fazer as seguintes observações: apesar de os pontos do bloco 1 serem eliminados do número de previsões erradas, o número de pontos percorridos e não previstos sofre uma subida muito acentuada em comparação com o normal, e o número de previsões corretas decresce devido à quase completa inexistência de pontos no bloco 4. Uma vez que a primeira condição não influencia a previsão, a queda no desempenho do modelo, relativamente a registos em que o objeto termina o percurso no elevador 1, deve-se unicamente do decréscimo de previsões corretas.

### Cenário 2

#### Registos

nº de registo	Previsões correctas	Previsões incorrectas	Não preveu e aconteceu	Estimado inicial	Percorrido	Estimado final	Destino	Precisão
84	181	117	77	263	263	254	elevador1	61%
85	193	124	58	263	256	255	elevador1	61%
86	170	129	109	263	284	250	elevador1	57%
87	81	90	211	264	297	156	elevador4	48%
88	173	116	50	264	228	253	elevador1	60%
89	201	124	49	264	255	251	elevador1	62%
90	84	104	134	263	223	151	elevador4	46%
91	166	137	64	263	235	257	elevador1	55%

Tabela 5.8: Relatório de registos da base de dados: cenário 2

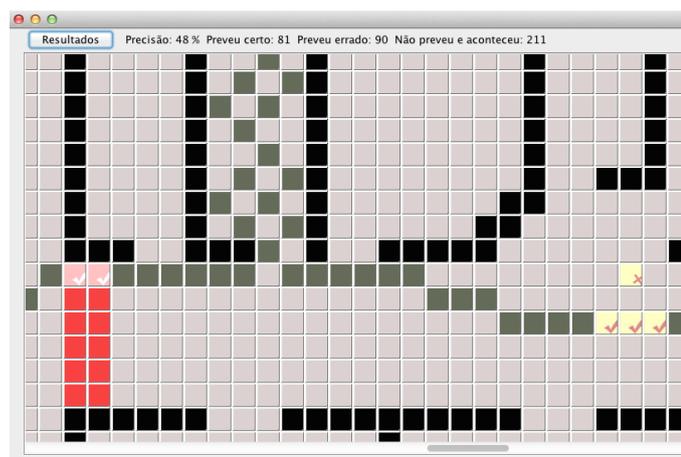


Figura 5.26: Resultados de previsão no bloco 4: cenário 2.

Similarmente, e analisando o problema por outro prisma, quando o objeto descreve a sua trajetória típica, pelo bloco 1, todos os pontos correspondentes ao bloco 4 são eliminados do número de previsões erradas (no segundo e terceiro cenários), o que implica uma variação pouco significativa na precisão, devido essencialmente à fraca distribuição de pontos nessa zona.

É importante referir que uma vez que a comparação é realizada ponto a ponto, o caminho estimado final deveria conter o mesmo número de pontos que o caminho percorrido. No entanto, não foi possível aplicar essa condição devido às restrições associadas ao SK-Means e ao facto da previsão não ser calculada à medida que o objeto acede a novas zonas.

Com o intuito de verificar a validade das estatísticas obtidas nos diferentes cenários procedeu-se à repetição da avaliação da precisão média para novos conjuntos de registos, obtidos em condições idênticas às referidas em cada cenário. Os resultados obtidos para um conjunto de 300 trajetórias estão sumariados na Tabela 5.27. Tal como seria de esperar, os resultados comprovam que o desempenho da precisão do modelo é consistente com as medições anteriores.

Análise de cenários			
Destino	Cenário 1	Cenário 2	Cenário 3
elevador1	58.1%	59.9%	63.3%
elevador4	30.8%	50.7%	56.5%
Precisão	56.5%	58.8%	62.6%

Figura 5.27: Relatório de precisão média para os diferentes cenários - repetição de procedimento.

Sabendo que a previsão é feita com base em todas as trajetórias anteriores, seria de esperar que a precisão do modelo aumentasse com o número de registos, o que pela observação dos gráficos de evolução não se pode dizer que aconteça com clareza em nenhum dos cenários estudados. No entanto, analisando o gráfico referente à evolução da precisão obtido para o segundo conjunto de registos, referentes à Tabela 5.27, e que consta na Figura 5.28, já se denota a existência de uma tendência de crescimento nos valores da precisão à medida que o histórico de trajetórias aumenta. Independentemente do cenário, pode-se afirmar que o algoritmo de previsão converge rapida-

mente para os valores de precisão apresentados. Adicionalmente, os gráficos comprovam que a percentagem de situações em que o objeto abandona o piso hospitalar pelo elevador 4 é similar às obtidas anteriormente.

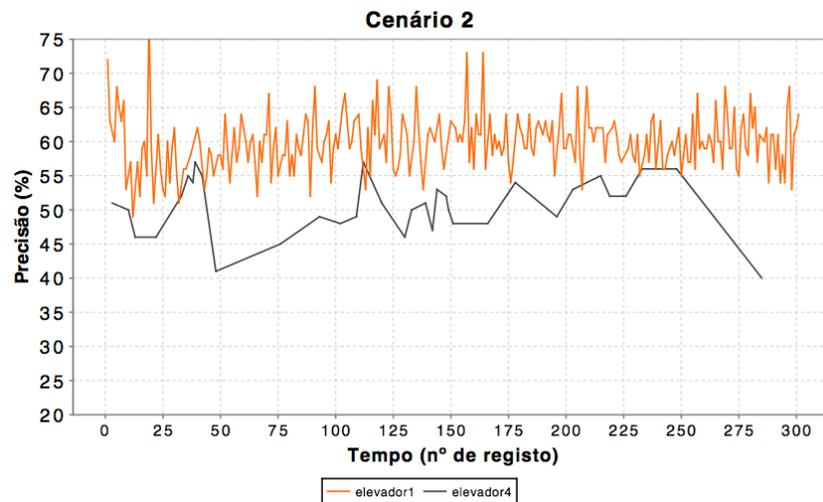


Figura 5.28: Gráfico da precisão em função do número de registos: cenário 2 - repetição de procedimento.

Relativamente ao cenário 5 convém realçar a influência do número de objetos simuláveis no desempenho do modelo de previsão, nomeadamente em relação ao tempo de análise do histórico de trajetórias de cada objeto. Quando o Data Mining é aplicado a um conjunto de 20 registos para cada um dos 3 objetos o simulador demora cerca de 3 segundos a iniciar. Este valor é proporcional ao tamanho do ficheiro e ao número de objetos. Para um total de aproximadamente 300 registos, o tempo de análise para um único objeto é de cerca de 8 segundos, e para 3 objetos de 23 segundos. A Tabela 5.9 apresenta o tempo de duração do módulo de previsão para diferentes registos, para 1 ou 3 objetos. Os valores permitem inferir o tempo de duração do modelo para qualquer número de objetos que se pretenda observar, dependendo do tamanho dos ficheiros aos quais se pretenda aplicar o algoritmo de Data Mining. De facto, pode-se concluir que a duração do modelo de previsão segue uma progressão aritmética: para um ficheiro de aproximadamente 50kB (correspondente a cerca de 20 registos), a adição de um robot implica um custo de 1 segundo no tempo de inicialização do

simulador. Os tempos de duração dependem obviamente da máquina onde a simulação é realizada.

Tamanho (kB)	Duração 1 objeto (segundos)	Duração 3 objetos (segundos)
50	< 1	< 3
300	< 2	7
900	8	23

Tabela 5.9: Duração do modelo de previsão em diferentes condições.

# Capítulo 6

## Conclusões e Trabalho Futuro

### 6.1 Conclusões

O sistema inicialmente proposto por Gonçalves et al. [15] visava a simulação de trajetória num contexto global, sem nenhum foco de contextualização específico. A modificação do sistema tendo em vista os objetivos delineados requereu em primeiro lugar um grande empenho no entendimento daquilo que já tinha sido feito. Em particular, muito tempo teve de ser dedicado à compreensão da arquitetura e implementação do simulador, nomeadamente do código Java que inicialmente o fundamentava. Esta tarefa não foi de toda uma tarefa simples, ou fácil, dada a complexidade inerente à compreensão de toda a linguagem de programação que constitui um programa escrito por outros.

O primeiro passo no sentido de cumprir os objetivos passou por adaptar a área de simulação a um ambiente hospitalar real, onde seria possível simular a trajetória de uma pessoa dentro do contexto. Como tal, começou-se por implementar a planta do piso do Hospital na interface gráfica do simulador, através da introdução de novas instâncias de dados no programa. Onde anteriormente existia um ambiente composto por 2500 posições distintas, onde todas as localizações constituíam possíveis lugares para a presença física do objeto passou a existir uma área delimitada por paredes, com 14506 posições que podem constituir espaços livres de circulação, incluindo zonas de quartos,

corredores ou sensores, e espaços definidos por portas e paredes.

Uma vez que num hospital é elevado o número de pessoas que circulam ao mesmo tempo numa mesma zona, e com o propósito de aproximar o mais possível o ambiente simulatório da realidade, tornou-se imperativo alterar o sistema integrado de simulação e previsão de modo a permitir a inserção e manipulação de novos robots, para permitir a circulação em simultâneo, e em tempo real, de vários objetos no ambiente de simulação. Esta capacidade do sistema é particularmente importante para a sua utilização como ferramenta de apoio à decisão em ambiente hospitalar. De facto, além de atender às condições que circundam o objeto, a simulação deve focar-se também no objeto em si. Este objetivo foi conseguido com sucesso no simulador, devido à arquitetura do simulador. A título demonstrativo, apresentou-se o caso de simulação de monitorização simultânea de três objetos apenas, uma vez que se prova que o sistema apresenta escalabilidade para a inserção de um maior número de objetos.

Foi necessário adaptar a trajetória puramente aleatória do objeto a um movimento típico de uma pessoa dentro de um edifício, que tal como já foi referido é definido por comportamentos de rotina na generalidade dos casos. Assim sendo, o algoritmo foi estudado e modificado com o intuito de conter tanto uma componente aleatória, que o tornasse imprevisível e não repetitivo, como uma componente direcionada que impusesse lógica ao movimento e evitasse que o objeto se movimentasse sem um sentido ou rumo aparente. A solução mais viável passou por atribuir direções a cada um dos corredores existentes e definir um movimento aleatório dentro dos quartos e escritórios, abordagem que provou ser capaz de simular o movimento caracteristicamente rotineiro de uma pessoa. Mais concretamente, analisando os movimentos virtualmente simulados verifica-se que:

- O robot pode percorrer corredores diferentes, que o levam a destinos distintos, em simulações consecutivas, dependendo da probabilidade de atingir determinado corredor que é definida pelo programador ao desenhar e posicionar os corredores no ambiente.
- A escolha dos quartos que visita a cada novo percurso é aleatória, re-

presentando no entanto um padrão quando várias trajetórias são analisadas. Ou seja, o robot frequenta geralmente o mesmo grupo de quartos entre diferentes simulações mas diferentes quartos entre simulações consecutivas, de forma autónoma.

- O tempo que permanece em cada quarto é variável.
- Quando se encontra num corredor a sua trajetória é contínua e num sentido.
- Quando entra numa zona de acesso aos pisos inferiores ou superiores a sua monitorização termina.

Seguindo esta estratégia é possível definir diferentes locais de origem e novas trajetórias alterando apenas as direções associadas aos corredores. Desta forma, conclui-se que o algoritmo desenvolvido é capaz de fornecer uma trajetória que vai de encontro aos comportamentos que são observados nos humanos.

Outra importante alteração aplicada ao sistema inicial foi a criação de uma interface de visualização do caminho previsto por Data Mining em tempo real, possibilitando ao utilizador uma visualização simultânea da previsão e do movimento simulado. Tal como se pretendia, os percursos seguidos por cada objeto são visualizados em simultâneo no tabuleiro, enquanto que a previsão e os resultados finais associados a cada um é apresentada em diferentes janelas. Além de ser aplicado previamente à simulação, o modelo de previsão sofreu alterações que visavam a melhoria da dinâmica do sistema através da interação entre as duas partes – simulação e previsão. Não só a simulação se apresenta como um processo dinâmico que em tempo real fornece ao utilizador a representação da trajetória gerada, como o modelo de previsão assenta num processo visualmente intuitivo, que consoante o que observa da realidade (simulação referida como realidade do ponto de vista da previsão) modifica a sua saída, apresentando uma distribuição de pontos concordante com a realidade recebida da camada de dados e de simulação, e que se define como o conjunto de localizações previstas e aceites para o objeto a um dado momento.

O SK-Means é um algoritmo de Data Mining simples que mostrou ser eficaz na produção de *clusters* na aplicação em questão, e um bom candidato para operar com os pontos gerados na simulação de trajetória. O modelo de previsão provou ser capaz de estimar localizações com uma precisão média de 62% para um modelo de simulação que tem em consideração a existência de variabilidade no movimento, definida por imposição de componentes aleatórias que asseguram que o percurso descrito por uma mesma entidade não é sempre igual. Verificaram-se no entanto algumas dificuldades na utilização do algoritmo de Data Mining. Uma delas baseia-se no facto de ser necessário fornecer o número de *clusters* para a inicialização do algoritmo. Neste contexto, a solução adotada consistiu em definir o número de *clusters* como uma média dos tamanhos anteriormente simulados para o objeto em estudo. Originalmente este problema não persistiu porque o caminho estimado só era calculado no final da simulação, e aí não existiam dúvidas de que ambos os caminhos deveriam ter o mesmo tamanho, e logo o número de *clusters* era direto. Uma outra limitação do algoritmo prende-se com a questão da continuidade dos pontos no espaço. Em vez de devolver um conjunto de pontos adjacentes, que teoricamente representassem uma possível trajetória, o algoritmo devolve pontos isolados sem ligação com outros pontos. Esta condição invalida a expressão previsão de trajetória, e deve ser substituída por previsão de localizações. Uma vantagem do algoritmo, relacionada com a sua simplicidade, prende-se com o reduzido tempo de computação necessário para operar um conjunto de dados relativamente extenso em termos de número de diferentes trajetórias, um fator importante e a ter em consideração em qualquer sistema computacional. Além disso, o algoritmo demonstra convergir rapidamente para os valores ótimos. Por este motivo, é razoável afirmar que a sua aplicação a um grande conjunto de dados é desnecessária na aplicação desenvolvida. Pelo contrário, o desempenho máximo do modelo é obtido logo nas primeiras previsões retribuídas, pelo que não se justifica o tempo de análise despendido em grandes ficheiros. Por este motivo, a adição de dezenas de entidades ao ambiente de simulação torna-se possível em termos de eficiência computacional.

Resumindo, o sistema proposto é capaz de simular com sucesso a tra-

jetória de entidades com particularidades distintas num ambiente hospitalar real, registar essas trajetórias e aplicá-las num algoritmo de Data Mining para a obtenção de padrões de movimento específicos para cada uma das entidades, e funciona com uma ou mais entidades em circulação em simultâneo, indo de encontro às necessidades do hospital, onde várias pessoas e equipamentos são etiquetados para serem monitorizados. A plataforma de simulação foi desenvolvida para melhorar e fazer ajustamentos antes do objetivo final: a implementação num dos maiores hospitais portugueses como ferramenta de apoio à decisão.

## 6.2 Sugestões e Trabalho Futuro

Desenhar e introduzir as paredes, quartos, corredores e portas no código inicialmente fornecido foi uma tarefa que, na ótica do programador, consumiu demasiado tempo. A fim de agilizar o simulador, propõe-se como trabalho futuro uma aplicação que, integrada neste, seja capaz de automaticamente reconhecer, numa imagem de formato típico (jpeg, png), as instâncias referidas e de as transpor para o ambiente de simulação. Tal solução seria obviamente indispensável para generalizar o simulador a qualquer aplicação.

A previsão de percurso deverá ser feita em tempo real, de acordo com o percurso realizado e a informação daí proveniente relativa aos sensores onde se deteta movimentação do objeto. De facto, seria mais eficiente e lógico adicionar pontos ao caminho previsto à medida que o objeto passasse pelos sensores, em vez de eliminar aqueles que não são possíveis. Tal trabalho requereria a implementação de um novo algoritmo de Data Mining, já que o atual, SK-Means, carece de eficiência quanto à aplicação de restrições ou condições iniciais. Adicionalmente, a eficiência do modelo de previsão deverá ser melhorada através da adoção de um algoritmo mais adequado ao modelo em questão. Nomeadamente, a escolha de um método que retribua uma previsão de percurso (em detrimento da previsão de possíveis localizações) é aconselhada para o problema em questão, assim como uma visualização sincronizada com a velocidade a que o objeto provavelmente se desloca, para efeitos de uma perceção visual imediata e mais realista da sua localização.

O objetivo de introduzir o simulador nos sistemas de informação do hospital passa por permitir que este receba alertas do sistema de monitorização a informar sobre a deteção de um determinado paciente num sensor e a partir daí agir em consonância: identificando o objeto, prevendo a sua trajetória a partir do ponto onde foi encontrado com base no conhecimento dos seus comportamentos e simulando a sua trajetória, ponto a ponto, para que hipoteticamente possa ser facilmente encontrado. Desta forma, poderá ser útil para simular e prever a localização de um paciente de risco em tempo real, agindo em concordância com as deteções reais, numa interface gráfica intuitiva que pode ser utilizada por qualquer profissional médico em tal situação. É de salientar que o atual sistema implementado no Hospital funciona à base de alertas, e portanto não permite a visualização em tempo real da localização das entidades com *tags* RFID. Para que o sistema possa ser utilizado no hospital este deverá ser integrado no sistema AIDA para garantir interoperabilidade com os outros sistemas, em particular com o sistema de monitorização que atualmente se encontra implementado para o *tracking* de pacientes por RFID.

Relativamente à temática das OTSNs em ambiente hospitalar, sugere-se para o hospital em questão a adoção de uma estratégia de conservação de energia que funcione com base no modelo de previsão fornecido. Através do conhecimento de padrões de movimento e da identificação das entidades que são monitorizadas no departamento de pediatria, supõe-se que será possível adaptar facilmente o sistema para que devolva no decorrer do tempo os nós de sensores que podem ser inativados e o período de tempo que devem permanecer adormecidos até que sejam novamente necessários.

# Bibliografia

- [1] Cleopatra Bardaki, Panos Kourouthanassis, and Katerina Pramadari. Modeling the information completeness of object tracking systems. *Journal of Strategic Information Systems*, 20(3):268–282, September 2011. ISSN 0963-8687. doi: 10.1016/j.jsis.2011.03.004.
- [2] Shin-Chih Tu, Guey-Yun Chang, Jang-Ping Sheu, Wei Li, and Kun-Ying Hsieh. Scalable continuous object detection and tracking in sensor networks. *Journal of Parallel and Distributed Computing*, 70(3):212–224, March 2010. ISSN 0743-7315. doi: 10.1016/j.jpdc.2009.12.001.
- [3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, Oct 2010.
- [4] Xiuli Qu, LaKausha T. Simpson, and Paul Stanfield. A model for quantifying the value of rfid-enabled equipment tracking in hospitals. *Advanced Engineering Informatics*, 25(1):23–31, Jan 2011.
- [5] Vincent S. Tseng and Eric Hsueh-Chan Lu. Energy-efficient real-time object tracking in multi-level sensor networks by mining and predicting movement patterns. *The Journal Of Systems and Software*, 82(4):697–706, April 2009.
- [6] José Machado, Miguel Miranda, Pedro Gonçalves, António Abelha, José Neves, and Alberto Marques. Aidatrace - interoperation platform for active monitoring in healthcare environments.
- [7] *Pro-Found. Who? What? Where? For Hospital Equipment and Staff.* CoreRFID. URL [www.corerfid.com](http://www.corerfid.com).

- [8] Haleh Hakim, Raymond Renouf, and John Enderle. Passive rfid asset monitoring system in hospital environments. In *Bioengineering Conference, 2006. Proceedings of the IEEE 32nd Annual Northeast*, pages 217–218. IEEE, 2006.
- [9] Jill A. Fisher and Torin Monahan. Tracking the social dimensions of rfid systems in hospitals. *International Journal of Medical Informatics*, 77:176–183, 2008.
- [10] David C. Wyld. Preventing the worst case scenario: An analysis of rfid technology and infant protection in hospitals. *The Internet Journal of Healthcare Administration*, 10(1), 2010.
- [11] Asil Oztekin, Foad M. Pajouh, Dursun Delen, and Leva K. Swim. An rfid network design methodology for asset tracking in healthcare. *Decision Support Systems*, 49(1):100–109, April 2010. ISSN 0167-9236. doi: 10.1016/j.dss.2010.01.007.
- [12] Remko van der Togt, Piet. J.M. Bakker, and Monique W.M. Jaspers. A framework for performance and data quality assessment of radio frequency identification (rfid) systems in health care settings. *Journal of Biomedical Informatics*, 44(2):372–383, 2011. doi: 10.1016/j.jbi.2010.12.004.
- [13] Yu-Ju Tu, Wei Zhou, and Selwyn Piramuthu. Identifying rfid-embedded objects in pervasive healthcare applications. *Decision Support Systems*, 46(2):586–593, 2009. doi: 10.1016/j.dss.2008.10.001.
- [14] M. Laskowski, BCP Demianyk, G. Naigeboren, BW Podaima, MR Friesen, and RD McLeod. Rfid modeling in healthcare. *Sustainable Radio Frequency Identification Solutions*, pages 217–240, 2010.
- [15] Pedro Gonçalves, Luciano Alves, Tiago Sá, Cesar Quintas, Miguel Miranda, António Abelha, and José Machado. Object trajectory simulation - an evolutionary approach. 2009.

- [16] Kawuu W. Lin, Ming-Hua Hsieh, and Vincent S. Tseng. A novel prediction-based strategy for object tracking in sensor networks by mining seamless temporal movement patterns. *Expert Systems with Applications*, 37(4):2799–2807, April 2010.
- [17] Ling Chen, Mingqi Lv, Qian Ye, Gencai Chen, and John Woodward. A personal route prediction system based on trajectory data mining. *Information Sciences*, 181(7):1264–1284, April 2011.
- [18] Ilhami Colak, Seref Sagiroglu, and Mehmet Yesilbudak. Data mining and wind power prediction: A literature review. *Renewable Energy*, 46: 241–247, 2012.
- [19] Yingqi Xu, Julian Winter, and Wang-Chien Lee. Dual prediction-based reporting for object tracking sensor networks. *Mobile and Ubiquitous Systems, Annual International Conference on*, 0:154–163, 2004. doi: <http://doi.ieeecomputersociety.org/10.1109/MOBIQ.2004.1331722>.
- [20] Vincent S. Tseng and Kawuu W. Lin. Energy efficient strategies for object tracking in sensor networks: A data mining approach. *The Journal of Systems and Software*, 80(10):1678–1698, October 2007. ISSN 0164-1212. doi: 10.1016/j.jss.2006.12.561.
- [21] Ling Chen, Mingqi Lv, and Gencai Chen. A system for destination and future route prediction based on trajectory mining. *Pervasive and Mobile Computing*, 6(6):657–676, December 2010.
- [22] António Abelha, Cesar Analide, José Machado, José Neves, Manuel Santos, and Paulo Novais. Ambient intelligence and simulation in health care virtual scenarios. In Luis Camarinha-Matos, Hamideh Afsharmanesh, Paulo Novais, and Cesar Analide, editors, *Establishing The Foundation Of Collaborative Networks*, volume 243 of *IFIP Advances in Information and Communication Technology*, pages 461–468. Springer Boston, 2007. ISBN 978-0-387-73797-3.

- [23] José Machado, Antonio Abelha, Paulo Novais, João Neves, and José Neves. Improving patient assistance and medical practices through intelligent agents. *Proc. of the Workshop on Agents applied in health care*, pages 41–50, May 2008.
- [24] James C. Benneyan. An introduction to using computer simulation in healthcare: patient wait case study. *Journal of the Society for Health Systems*, 5:1–15, 1997.
- [25] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus A. F. Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics Review*, 16(5):412–424, February 2000.
- [26] Theodoros Anagnostopoulos, Christos B. Anagnostopoulos, Stathes Hadjiefthymiades, Alexandros Kalousis, and Miltos Kyriakakos. Path prediction through data mining. *International Conference on Pervasive Services*, 0:128–135, 2007. doi: <http://doi.ieeecomputersociety.org/10.1109/PERSER.2007.4283902>.
- [27] A segurança do recém-nascido em meio hospitalar. Technical report, Entidade Reguladora da Saúde, Março 2011.
- [28] *TraceMe - A Tool for Safety and Security in Clinical Governance using RFID and Integration of Location Services in a Hospital Environment*, 2009. Proceedings of the IECON09 – The35th Annual Conference of the IEEE Industrial Electronics Society.
- [29] Pedro Eduardo Gomes, Alberto Marques, Ângelo Costa, Paulo Novais, and José Neves. Patient monitoring under an ambient intelligence setting. *Future Trends - International Symposium on Ambient Intelligence*, 72:185–188, June 2010.
- [30] António Abelha, José Machado, Victor Alves, and José Neves. Data warehousing through multi-agent systems in the medical arena. 2004.

- [31] Davide Carneiro, Ricardo Costa, Paulo Novais, João Neves, José Machado, and José Neves. Simulating and monitoring ambient assisted living.
- [32] Mindi Anderson, Mary Lou Bond, Tiffany L. Holmes, and Carolyn L. Cason. Acquisition of simulation skills: Survey of users. *Clinical Simulation in Nursing*, 8:e59–e65, 2012.
- [33] Lingguang Song and Neil N. Eldin. Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling. *Automation in Construction*, 27:32–39, 2012.
- [34] Jenq-Muh Hsu, Chao-Chun Chen, and Chia-Chi Li. Poot: An efficient object tracking strategy based on short-term optimistic predictions for face-structured sensor networks. *Computers and Mathematics with Applications*, 63(2):391–406, Jan 2012.
- [35] Niyati Aggarwal, Amit Kumar, Harsh Khatter, and Vaishali Aggarwal. Analysis the effect of data mining techniques on database. *Advances in Engineering Software*, 47:164–169, 2012.
- [36] Hsu-Hao Tsai. Global data mining: An empirical study of current trends, future forecasts and technology diffusions. *Expert Systems with Applications*, 39(9):8172–8181, July 2012. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.01.150.
- [37] Yingqi Xu, Julian Winter, and Wang-Chien Lee. Prediction-based strategies for energy saving in object tracking sensor networks. *Mobile Data Management, IEEE International Conference on*, 0:346, 2004. doi: <http://doi.ieeecomputersociety.org/10.1109/MDM.2004.1263084>.
- [38] Chih-Yu Lin, Wen-Chih Peng, and Yu-Chee Tseng. Efficient in-network moving object tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5:1044–1056, 2006. ISSN 1536-1233.
- [39] Yung-Ting Changa, Shabbir Syed-Abdula, Chung-You Tsaia, and Yu-Chuan Lid. A novel method for inferring rfid tag reader recordings

- into clinical events. *International Journal of Medical Informatics*, 80: 872–880, 2011.
- [40] Jiann-Liang Chen, Ming-Chiao Chen, Chien-Wu Chen, and Yao-Chung Chang. Architecture design and performance evaluation of rfid object tracking systems. *Computer Communications*, 30(9):2070–2086, Jun 2007.
- [41] Mohammed F.M. Ali, Mohammed I. Younis, Kamal Z. Zamli, and Widad Ismail. Development of java based rfid application programmable interface for heterogeneous rfid system. *Journal of Systems and Software*, 83(11):2322–2331, 2010. doi: 10.1016/j.jss.2010.07.030.
- [42] Wei-Pang Liao, Tom M. Y. Lin, and Shu-Hsien Liao. Contributions to radio frequency identification (rfid) research: An assessment of sci-, ssci-indexed papers from 2004 to 2008. *Decision Support Systems*, 50(2):548–556, Jan 2011.
- [43] Jongchul Song, Carl T. Haas, and Carlos H. Caldas. A proximity-based method for locating rfid tagged objects. *Advanced Engineering Informatics*, 21(4):1474–1486, 2007. doi: 10.1016/j.aei.2006.09.002.
- [44] António Aguilar. A patient identification system using rfid and ieee 802.11b wireless networks. Master’s thesis, School of Information and Communication Technology at Royal Institute of Technology, Stockholm, Sweden, March 2007.
- [45] Eduardo C. Macalanda. Radio frequency identification (rfid) for naval medical treatment facilities (mtf). Master’s thesis, Naval Postgraduate School, September 2006.
- [46] Jonathan Collins. Hospitals get healthy dose of rfid. *RFID Journal*, April 2004.
- [47] Shu-Hsien Liao, Pei-Hui Chu, and Pei-Yuan Hsiao. Review: Data mining techniques and applications - a decade review from 2000 to 2011. *Expert*

- Syst. Appl.*, 39(12):11303–11311, September 2012. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.02.063.
- [48] Alex A. Freitas. Advances in evolutionary computing. chapter A survey of evolutionary algorithms for data mining and knowledge discovery, pages 819–845. Springer-Verlag New York, Inc., New York, NY, USA, 2003. ISBN 3-540-43330-9.
- [49] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3): 37–54, 1996.
- [50] Fernando Alonso, Loïc Martínez, Aurora Pérez, and Juan P. Valente. Cooperation between expert knowledge and data mining discovered knowledge: Lessons learned. *Expert Syst. Appl.*, 39(8):7524–7535, June 2012. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.01.133.
- [51] T. Velmurugan and T. Santhanam. Performance evaluation of k-means and fuzzy c-means clustering algorithms for statistical distributions of input data points. *European Journal of Scientific Research*, 46(3):320–330, 2010.
- [52] Wu He. Examining students online interaction in a live video streaming environment using data mining and text mining. *Computers in Human Behavior*, 29(1):90–102, 2013. doi: 10.1016/j.chb.2012.07.020.
- [53] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier, 2006. ISBN 9781558609013.
- [54] Ankit Pandey. Data mining techniques using weka. *IT for Business Intelligence*.
- [55] Hui Xiong, Junjie Wu, and Jian Chen. K-means clustering versus validation measures: a data distribution perspective. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery*

- and data mining*, KDD '06, pages 779–784, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. doi: 10.1145/1150402.1150503.
- [56] Osama Abu Abbas. Comparisons between data clustering algorithms. *The International Arab Journal of Information Technology*, 5(3):320–325, 2008.
- [57] Sanjay; Alsabti, Khaled; Ranka and Vineet Singh. An efficient k-means clustering algorithm. *Electrical Engineering and Computer Science*, 1997.
- [58] Wan Maseri Binti Wan Mohd, A.H.Beg, Tutut Herawan, and K.F.Rabbi. An improved parameter less data clustering technique based on maximum distance of data and lioyd k-means algorithm. *Procedia Technology*, 1:367–371, 2012.
- [59] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 577–584, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.
- [60] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, December 2007. ISSN 0219-1377. doi: 10.1007/s10115-007-0114-2.
- [61] S.S. Mehta, D.A. Shete, N.S. Lingayat, and V.S. Chouhan. K-means algorithm for the detection and delineation of qrs-complexes in electrocardiogram. *IRBM*, 31(1):48–54, 2010. doi: 10.1016/j.irbm.2009.10.001.
- [62] Yau King Lam and Peter W. M. Tsang. exploratory k-means: A new simple and efficient algorithm for gene clustering. *Applied Soft Computing*, 12(3):1149–1157, March 2012.

- [63] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, July 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.1017616.
- [64] António Abelha. *Sistemas Multiagente como Suporte a Trabalho Cooperativo em Unidades Hospitalares*. PhD thesis, Universidade do Minho, Braga, Portugal, 2004.
- [65] Eibe Frank, Mark Hall, Len Trigg, Geoffrey Holmes, and Ian H. Witten. Data mining in bioinformatics using weka. *Bioinformatics*, 20(15):2479–2481, October 2004. ISSN 1367-4803. doi: 10.1093/bioinformatics/bth261.
- [66] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278.
- [67] Ian Witten Eibe, Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations. In *Proc ICOP/NIP/ ANZIIS/ANNES99 Future Directions for Intelligent Systems and Information Sciences*, pages 192–196. Morgan Kaufmann, 1999.