



INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Modelação computacional de Exercícios de Evacuação do Parque Urbano de Albarquel, Setúbal

Tiago Miguel Passos de Almeida Domingos

Mestrado em Engenharia Informática

Orientador:

Doutor Nuno Manuel Mendes Cruz David, Professor Associado,
Iscte – Instituto Universitário de Lisboa

Orientadora:

Doutora Ângela Cristina Carvalho da Silva Santos, Investigadora,
IGOT – ULisboa

Outubro, 2025

Departamento de Ciências e Tecnologias da Informação

Modelação computacional de Exercícios de Evacuação do Parque Urbano de Albarquel, Setúbal

Tiago Miguel Passos de Almeida Domingos

Mestrado em Engenharia Informática

Orientador:

Doutor Nuno Manuel Mendes Cruz David, Professor Associado,
Iscte – Instituto Universitário de Lisboa

Orientadora:

Doutora Ângela Cristina Carvalho da Silva Santos, Investigadora,
IGOT – ULisboa

Outubro, 2025

Agradecimento

Sem ordem de importância, quero primeiramente agradecer ao professor Nuno David e à professora Ângela Santos que foram essenciais na execução desta dissertação, conseguindo guiar-me sempre na direção certa.

Quero também prestar agradecimentos ao ISCTE que sempre proporcionou a estrutura necessária para que tudo sempre corresse pelo melhor.

Um agradecimento mais pessoal à minha família que sempre me apoiou de todas as formas para que eu conseguisse continuar os estudos.

Resumo

A ocorrência de tsunamis representa um risco significativo para zonas costeiras urbanas, onde a elevada densidade populacional e a falta de planeamento adequado podem resultar em cenários de grande vulnerabilidade. Neste contexto, a simulação computacional baseada em agentes aparece como uma ferramenta eficaz para avaliar estratégias de evacuação e ajudar na execução de medidas preventivas.

Este trabalho apresenta um modelo de simulação da evacuação controlada do Parque Urbano de Albarquel, em Setúbal, um espaço público com elevada afluência e situado numa área suscetível à ocorrência de tsunamis. O modelo foi desenvolvido em Python, integrando dados geoespaciais obtidos através do software QGIS. O movimento dos agentes foi modelado com base na modelação baseada em agentes, incorporando interações sociais e restrições no ambiente de forma a aproximar a simulação do comportamento real da multidão no Parque.

Foram testados diferentes cenários, variando o parâmetro de densidade populacional. Os resultados evidenciam pontos críticos de congestionamento, nomeadamente nas zonas de escadas e passagens estreitas, que comprometem a eficiência da evacuação. Embora uma evacuação completa em todos os cenários analisados não se tenha mostrado possível dentro do tempo estimado para a chegada da primeira onda, os resultados foram bastante positivos na grande maioria das simulações executadas.

Este estudo contribui para a compreensão dos riscos associados a tsunamis em Setúbal e fornece evidências que podem apoiar a Proteção Civil na melhoria de planos de evacuação e medidas de mitigação.

Palavras-chave: Evacuação, Modelação Baseada em Agentes, Tsunami, Simulação Computacional, Python, QGIS.

Abstract

The occurrence of tsunamis poses a significant risk to urban coastal areas, where high population density and lack of adequate planning can result in scenarios of great vulnerability. In this context, agent-based computer simulation appears to be an effective tool for evaluating evacuation strategies and assisting in the implementation of preventive measures.

This work presents a simulation model of the controlled evacuation of Albarquel Urban Park in Setúbal, a highly frequented public space located in an area susceptible to tsunamis. The model was developed in Python, integrating geospatial data obtained using QGIS software. The movement of agents was modelled based on the Social Force Model, incorporating social interactions and environmental constraints in order to approximate the simulation to the actual behavior of the crowd in the Park.

Different scenarios were tested, varying the population density parameter. The results highlight critical points of congestion, particularly in areas with stairs and narrow passages, which compromise the efficiency of the evacuation. Although a complete evacuation in all scenarios analyzed was not possible within the estimated time for the arrival of the first wave, the results were very positive in the vast majority of simulations performed.

This study contributes to the understanding of the risks associated with tsunamis in Setúbal and provides evidence that can support Civil Protection in improving evacuation plans and mitigation measures.

Keywords: Evacuation, Agent-Based Modeling, Tsunami, Computer Simulation, Python, QGIS.

Índice

Agradecimento	iii
Resumo.....	v
Abstract	vii
Índice de Figuras	xii
Índice de Tabelas.....	xiv
CAPÍTULO 1 Introdução.....	1
1.1 Motivação.....	1
1.2 Questões de Investigação	2
1.3 Objetivos	2
1.4 Metodologia de Investigação	3
1.5 Planeamento do documento	4
CAPÍTULO 2 Revisão da Literatura	5
2.1 Modelação Baseada em Agentes.....	5
2.2 Trabalhos Anteriores	5
2.3 Modelos e Tecnologias.....	8
2.3.1 Modelos Macroscópicos e Modelos Microscópicos	8
2.3.2 Algoritmos de Pesquisa de Caminho em Grafos (Pathfinding)	9
Algoritmo de Dijkstra.....	9
Algoritmo A*.....	10
Comparação entre A* e Dijkstra	10
2.3.3 Modelos de Movimento e Interação Local.....	11
Modelo de Força Social (SFM).....	11
2.3.4 Modelos Baseados em Regras.....	12
Modelos de Autómatos Celulares (CA)	13
CAPÍTULO 3 Modelo abstrato.....	15
3.1 Modelo do Ambiente.....	15
3.1.1 Parque Urbano de Albarquel	15
3.1.2 Representação Espacial	18
3.1.3 Representação Temporal	18
3.2 Modelo do Agente.....	18
3.2.1 Regras Físicas.....	18
3.2.2 Atributos do Agente	19

3.2.3	Comportamento e Navegação do agente	19
CAPÍTULO 4	Implementação do Modelo de Simulação	23
4.1	Software e Ferramentas	23
4.1.1	Sistema de Informação Geográfica (GIS): QGIS.....	23
4.1.2	Ambiente de Desenvolvimento: Python e Bibliotecas	24
4.1.3	Ambiente de Execução	24
4.2	Construção do Ambiente de Simulação	25
4.2.1	Representação Geoespacial e Grafo de Navegação	25
4.2.2	Aplicação de Restrições ao Grafo	27
4.3	Implementação do Agente e Respetivos Comportamentos.....	28
4.3.1	Processo de Criação e Posicionamento	28
4.3.2	Lógica de Decisão de Caminhos	29
4.4	Ciclo de Execução	32
4.4.1	Inicialização e Estrutura Temporal.....	32
4.4.2	Ciclo Interno do Agente e Cálculo de Forças.....	32
4.4.3	Validação Espacial e Movimento Tangencial.....	34
4.4.4	Sincronização e Atualização do Estado.....	34
4.5	Desafios e Soluções no Movimento de Agentes	35
4.6	Geração e Formatação dos Resultados.....	38
CAPÍTULO 5	Experiências e Resultados	40
5.1	Configuração Experimental.....	40
5.1.1	Parâmetros do Modelo	40
5.1.2	Métricas de Avaliação	45
5.2	Análise dos Cenários de Simulação	46
5.2.1	Visualização da simulação	46
5.2.2	Análise do Cenário Base	47
5.2.3	Estudo da Escalabilidade da Evacuação.....	50
5.2.4	Discussão do Realismo dos Resultados	54
5.3	Respostas às Questões de Investigação	55
CAPÍTULO 6	Conclusão e Trabalho Futuro	56
6.1	Conclusão	56
6.2	Limitações e Trabalho Futuro	57
	Referências Bibliográficas	58

Índice de Figuras

Figura 1 - Metodologia Design Science Research (Imagem criada pelo autor).....	3
Figura 2 - Exemplo de landmarks para a decisão navegacional na praia da Figueirinha. [10]..	6
Figura 3 - Demonstração do algoritmo Dijkstra [26].....	10
Figura 4 - Demonstração do algoritmo A* [29].....	10
Figura 5 - Fluxo de pedestres em diferentes sentidos através de uma zona de afunilamento [31].	12
Figura 6 - Três regras básicas do movimento de Boids [43].....	13
Figura 7 - Evacuação de pedestres em diferentes instantes (Modelo Celular) [51].	14
Figura 8 - Passeio largo e praia do Parque (foto tirada no dia 07/06/2025).....	15
Figura 9 - Escadas da praia.	16
Figura 10 - Escadas perto do café.	16
Figura 11 - Escadas mais a Oeste.....	17
Figura 12 - Escadas a cobrir um sinal de evacuação.....	17
Figura 13 – Representação exemplificativa da força de atração dos agentes ao nó.....	20
Figura 14 - Representação exemplificativa da força de repulsão entre agentes.....	21
Figura 15 - Representação exemplificativa e aproximada da soma das forças de atração e repulsão.	21
Figura 16 - Representação exemplificativa e aproximada do deslocamento dos agentes A e B segundo as suas forças finais.....	22
Figura 17 - Exemplos de utilização do QGIS [53].....	23
Figura 18 - Visualização do mapa com edifícios (cor cinza).....	25
Figura 19 - Visualização do mapa com edifícios (cor cinza) e nós do grafo.	26
Figura 20 - Exemplo de informação em um nó.....	26
Figura 21 - Diferença de altura (a vermelho) entre praia e paredão em metros.....	27
Figura 22 - Exemplo de nós sem ligação.	28
Figura 23 - Exemplo de posicionamento mais concentrado de agentes em shapes específicas.	29
Figura 24 - Aglomerado de agentes causado por não utilizarem caminhos alternativos.	30
Figura 25 - Utilização de porta para controlar o fluxo dos agentes.	31
Figura 26 - Fluxograma representativo da lógica de execução e sincronização do modelo de simulação.....	33
Figura 27 - Simulação de cenário base com 300 pessoas. Caminhos percorridos e zonas de congestão são observáveis.....	35
Figura 28 – Ligação entre nós por cima de zona inacessível pelo agente.....	36
Figura 29 - Demonstração exemplificativa de agente a ficar preso.	37
Figura 30 - Agente a desprender-se através da movimentação aleatória.	38
Figura 31 - Ficheiro CSV com o registo de uma simulação.	38
Figura 32 - Demonstração do deslocamento dos agentes com barra de tempo interativa. Hora de cada imagem no formato hh:mm:ss. A – 20:48:07. B – 20:50:26. C – 20:53:01. D – 20:56:37.....	39
Figura 33 - Relação entre os graus de inclinação das escadas e as velocidades de subida [60].	41

Figura 34 - Medição de distância horizontal das escadas [61].....	42
Figura 35 - Medição da distância vertical das escadas [62].	42
Figura 36 - Representação das escadas (lado C).	43
Figura 37 - Exemplo do mapa de calor (layer do mapa interativo).....	46
Figura 38 - Mapa interativo para visualização da simulação.	47
Figura 39 - Exemplo da distribuição dos agentes no instante inicial. Cenário base com 300 pessoas.....	48
Figura 40 - Caminhos ótimos surgem ao sair da praia.	49
Figura 41 - Zona junto ao café (imagem da esquerda) e zona junto às escadas a oeste (imagem da direita).....	50
Figura 42 - Caminhos ótimos mais acentuados ao simular com 500 agentes.....	51
Figura 43 - Mapa de calor da primeira simulação com 700 agentes.....	53

Índice de Tabelas

Tabela 1 - Comparação entre Modelos Macroscópico e Microscópico	9
Tabela 2 - Dados de simulações de cenário base com 300 agentes.	49
Tabela 3 - Dados de simulações de cenário base com 500 agentes.	51
Tabela 4 - Dados de simulações de cenário base com 700 agentes.	52

CAPÍTULO 1 INTRODUÇÃO

Nesta dissertação pretende-se realizar a simulação, por computador, de uma evacuação de emergência em caso de *tsunami*. A área de estudo localiza-se no Parque Urbano de Albarquel, em Setúbal que é muito procurado pela população e conta com uma área superior a 40 mil metros quadrados, bem como serviços de restauração e parque infantil [1].

Por se localizar à beira-rio, no caso de ocorrência de um *tsunami*, esta zona torna-se vulnerável e perigosa para a população que a frequenta, e por isso é necessário que esteja devidamente preparada e sinalizada de forma a conduzir as pessoas para um local seguro.

Para tal iremos recorrer a simulações por computador, com objetivo de tentar compreender os riscos inerentes e desenvolver soluções. Os exercícios de evacuação e simulacros despendem de muitos recursos humanos e requerem uma grande mobilização de tempo [2]. Além disso, não é um ambiente tão controlável como numa simulação por computador, que pode trazer benefícios quando se testam vários cenários diferentes.

O objetivo desta dissertação é perceber que problemas de evacuação possam surgir, simulando o ambiente e as pessoas por computador. Para tal as pessoas são representadas no mundo virtual como agentes virtuais cujo objetivo é alcançar uma das zonas seguras quando a massa de água chegar ao parque.

1.1 Motivação

Desastres naturais ocorrem constantemente e muitos são difíceis ou até mesmo impossíveis de prever, e por esse motivo tem de haver preparação para prevenir tragédias. No caso de desastres como o abordado nesta dissertação – *tsunamis* – é necessário planear rotas de fuga eficientes e que estejam ao alcance de todos.

A história de Portugal conta-nos muito sobre como estamos protegidos para este tipo de desastres. Entre sismos, incêndios e *tsunamis*, o famoso desastre de 1755 matou mais de 2000 pessoas em Setúbal, que representavam na época 16,9% da população dessa cidade [2]. Este *tsunami* chegou a Setúbal passado cerca de 30 minutos após o sismo e a profundidade da água chegou a 9 metros em algumas zonas da praia.

O *tsunami* de 1969, com uma magnitude de Ms 8.0, lembrou Portugal de que não está imune a este tipo de catástrofes. O sismo provocou danos ligeiros [3]. Embora o tsunami não tenha provocado danos ou vítimas, sabe-se que as primeiras ondas chegaram à costa de Setúbal em aproximadamente 40 minutos, deixando praias submersas em água com 2,4 metros de profundidade, calculando-se que as ondas atingiram 10,8 m/s de velocidade [2]. Estes números demonstram que mesmo em casos considerados não muito perigosos, pode haver consequências indesejáveis para a população se não houver preparação prévia.

Estudos recentes mostram que a população portuguesa não está devidamente preparada para um novo desastre. Muitos desconhecem procedimentos de evacuação bem como onde se situam as zonas seguras [4]. Além disso, existem arribas instáveis que poderão dar origem à queda de rochas, dificultando a movimentação das pessoas [2].

Neste contexto, torna-se evidente a necessidade de criar ou analisar planos de evacuação. O desenvolvimento e utilização de ferramentas de simulação computacional possibilita identificar riscos e problemas a resolver, de forma a estudar respostas mais rápidas e eficientes em situações reais, e daí a importância desta dissertação, que se propõe a abordar esta problemática.

1.2 Questões de Investigação

As questões abordadas nesta dissertação são as seguintes:

- O modelo desenvolvido simula a evacuação controlada de forma realista?
- Qual o tempo médio de evacuação total do Parque Urbano de Albarquel?
- Haverá locais do parque em que a possibilidade de fuga das pessoas que lá se encontrem seja menor?

1.3 Objetivos

O presente trabalho pretende como objetivo geral especificar, desenvolver e implementar um modelo baseado em agentes que simule a evacuação controlada do Parque Urbano de Albarquel, para apoiar a análise de cenários de risco e a identificação de padrões, de forma a gerir a segurança em emergências. Para alcançar este objetivo geral, definem-se os seguintes objetivos específicos:

1. Desenvolver o modelo da simulação: conceber um sistema baseado em agentes adaptado ao contexto do parque, definindo os agentes, as regras de interação e os parâmetros do ambiente de simulação.

2. Implementar o modelo: utilizar a linguagem de programação Python e a framework Mesa [5], integrando o sistema com um Sistema de Informação Geográfica (SIG) para representar o espaço do parque de forma rigorosa.
3. Realizar experiências de simulação: analisar a dinâmica do modelo para identificar padrões e possíveis problemas no processo de evacuação, permitindo antever soluções.

1.4 Metodologia de Investigação

A metodologia de investigação utilizada no desenvolvimento desta dissertação baseia-se no Design Science Research (DSR) [6]. O DSR, esquematizado na Figura 1, é um modelo de investigação orientado para a construção de artefactos. Nesta dissertação, o artefacto é um modelo baseado em agentes que simula a evacuação do Parque Urbano de Albarquel.

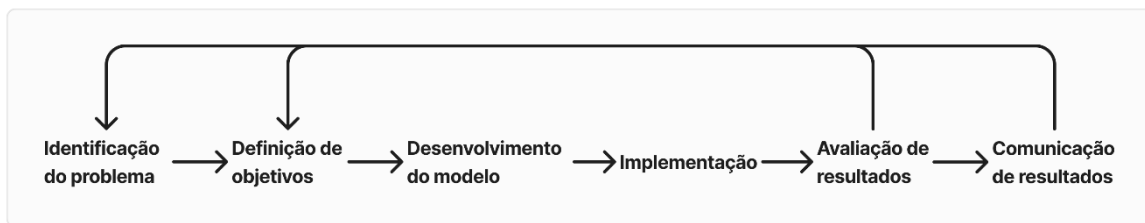


Figura 1 - Metodologia Design Science Research (Imagem criada pelo autor).

O processo de DSR é iterativo e composto por várias etapas de forma que o desenvolvimento seja recorrentemente melhorado, como mostra a Figura 1 [6].

As etapas são as seguintes:

1. Identificação do problema: A necessidade de evacuações eficientes e da identificação de riscos.
2. Definição de objetivos: Criação de um modelo de evacuação baseado em agentes de modo a identificar rotas de fuga mais eficientes e avaliar os riscos de diferentes perfis de agentes.
3. Desenvolvimento do modelo: Desenvolvimento com recurso à linguagem de programação Python e a *framework* Mesa, com integração de um Sistema de Informação Geográfica para simular o local.
4. Implementação: Implementação das funcionalidades e testes das mesmas.

5. Avaliação de resultados: Avaliação dos resultados obtidos no ponto 4, com objetivo de detetar padrões e avaliar a eficácia.
6. Comunicação de resultados: Resultados serão documentados na dissertação, que inclui a discussão sobre os mesmos e a identificação de limitações. Caso se identifique novas melhorias possíveis, o processo poderá ser iterado novamente.

1.5 Planeamento do documento

Nesta secção será exposto qual o plano e organização desta dissertação, com breves explicações sobre o mesmo.

Na revisão da literatura, **Capítulo 2**, expõe-se o estado de arte. Neste capítulo o objetivo é compreender que estudos foram realizados no mesmo domínio que esta dissertação, identificar os conceitos principais, que tecnologias foram usadas ou desenvolvidas e analisar as conclusões dos estudos.

No **Capítulo 3**, explica-se de forma abstrata o modelo. Pretende-se abordar as vantagens e desvantagens dos algoritmos existentes e descreve-se o modelo proposto. São também apresentados o *software* e as ferramentas utilizados.

O **Capítulo 4** apresenta o modelo de implementação do modelo e explica como este funciona.

O **Capítulo 5** mostra os resultados obtidos, conclusões e limitações. Uma conclusão geral da dissertação será elaborada no final.

2.1 Modelação Baseada em Agentes

Agent-Based Modeling (ABM) ou, em português, Modelação Baseada em Agentes é um domínio científico que usa um método computacional para simular as ações/decisões que agentes autónomos realizam com o ambiente ou com outros agentes autónomos. Cada agente segue um conjunto de regras que definem a forma como este reage ao ambiente e a outros agentes.

Mesmo com agentes simples é possível obter comportamentos complexos que sejam úteis para a recolha de informação. Um ABM permite uma análise macro dos efeitos causados por comportamentos dos agentes de nível micro, ou seja, ao implementar agentes simples que tenham apenas algumas regras de locomoção e de interação com o ambiente e com os seus pares, é possível ver os efeitos macro que estes agentes têm, como, por exemplo, a análise dos caminhos percorridos pelos agentes ao dirigirem-se para objetivos [7,8].

Diferentemente de modelos tradicionais, que assumem homogeneidade na população de agentes, um ABM tem as características individuais em conta, o que torna mais imprevisível quais serão os resultados das interações entre agentes com diferentes características [9]. Isto é importante para simular resultados imprevisíveis que poderiam não ser percebidos como uma possibilidade caso não fosse usado o ABM.

2.2 Trabalhos Anteriores

Um exemplo de uma simulação feita por meio de um ABM foi a evacuação da praia da Figueirinha em Setúbal [10] em que se pretendeu compreender e estudar o comportamento das pessoas ao aplicar diferentes cenários, para poderem ser identificados problemas e serem sugeridas diferentes soluções.

Para a criação deste modelo foi utilizada a linguagem de programação *Python* com recurso à biblioteca *mesa-geo*, que permite a modelação de interações dos agentes com o ambiente geográfico. Este ambiente é um *Geospace* que contém *Geoagents*. Os *Geoagents* podem ser agentes que se movem, como pessoas, ou podem ser agentes estáticos que representam áreas específicas.

Assim, para alterar o comportamento da deslocação das pessoas pode existir, por exemplo, um *Geoagent* que representa a zona de areia seca. Nesta área, a velocidade das pessoas é reduzida relativamente à velocidade normal, fazendo com que o comportamento dos agentes móveis se altere e, por conseguinte, o resultado da simulação leve em conta esse comportamento.

Para o deslocamento dos agentes, foi utilizado um sistema de *landmarks*, pontos no espaço para onde os agentes se irão mover (Figura 2). Esta implementação é simples, mas mostrou-se eficaz, pois como refere o autor, algoritmos de cálculo de caminho mais curto ou caminho mais rápido são pouco eficazes com um número elevado de agentes. Cada agente a calcular a sua rota mais rápida a cada iteração torna o modelo pesado computacionalmente.

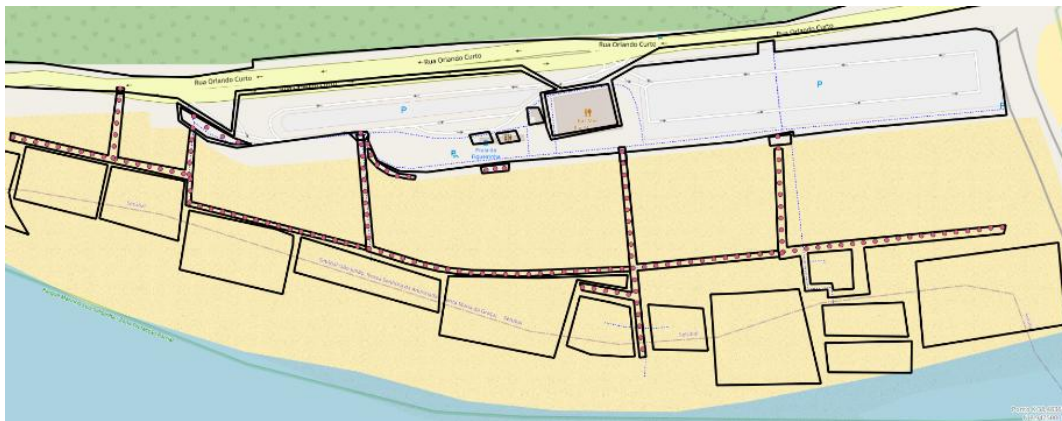


Figura 2 - Exemplo de landmarks para a decisão navegacional na praia da Figueirinha. [10]

Para situações em que as pessoas estão em grupo, como uma turma composta por um adulto e várias crianças, ou até mesmo uma simples família, o algoritmo tem em conta um líder do grupo. Este líder só pode ser um adulto e só se desloca quando todos os membros do grupo estão na sua área de influência, representada por uma distância ao redor do líder. Esta distância varia conforme o número de seguidores do líder, de modo que mesmo em fila todos os seguidores sejam abrangidos pela área de influência. Os seguidores moverem-se em direção ao líder coloca, contudo, um problema, quando os seguidores passam por obstáculos que não deveriam. Para impedir que isto aconteça foi imposta uma regra no algoritmo para que os seguidores sigam o seguidor do líder mais próximo, em vez do próprio líder. Desta forma origina-se uma fila, resolvendo o problema. Caso alguns seguidores tenham uma velocidade abaixo da do líder, estes vão acabar por sair da área de influência, fazendo com que o líder espere até que todos os seguidores voltem à área.

Como resultados, houve situações com uma alta taxa de mortalidade que se deveu, principalmente, a problemas de engarrafamento em zonas apertadas, como escadas de acesso e passadiços. Percebe-se então, que o modo mais ineficiente de evacuação da praia é em fila, como resultado de alguns agentes serem mais lentos e atrasarem todos os outros. Pôde-se concluir também que, nas condições previstas de evacuação, uma taxa de sobrevivência de 100% é irrealista, especialmente quanto maior for o número de agentes presentes na simulação.

Outro estudo relevante sobre a evacuação simulada de espaços públicos é o trabalho de Liu et al. (2020) [11]. Este estudo aborda um ABM com o propósito de estudar e analisar a zona ribeirinha de Bund, Xangai, baseando-se num desastre real que ocorreu em 2014, onde a elevada concentração de pessoas levou a uma morte.

O modelo foi desenvolvido com o *Agent Simulator of Pedestrian Flow - ASPFver4.0*. Este sistema permite simular interações dos agentes pedestres com o ambiente urbano, por meio de parâmetros como a velocidade individual de deslocamento, estímulos visuais, densidade populacional e até a topografia como fator de influência na tomada de decisão dos agentes. Neste sistema existe também uma previsão dinâmica do congestionamento das áreas, de modo a reorientar os agentes para zonas menos congestionadas. Da mesma forma que no estudo da Praia da Figueirinha, foram identificados pontos de congestionamento críticos e testadas soluções diferentes para os resolver, como, por exemplo, a alteração de infraestruturas existentes.

Uma das contribuições-chave deste estudo foi a demonstração da importância da maneira como os espaços urbanos são projetados. Por exemplo, foi observado que a simples adição de barreiras físicas em certas zonas tornou a evacuação muito mais segura ao reduzir a densidade nesses locais para evitar o efeito de empurrões em massa que pode levar a quedas perigosas e até mesmo a fatalidades.

Assim como na simulação da Praia da Figueirinha, neste estudo é também destacado como a formação de filas pode ser um problema grave nas evacuações de emergência, pois limitam a circulação dos pedestres, especialmente em locais estreitos, colocando a em causa a segurança das pessoas.

Ambos os estudos analisados demonstram haver questões a resolver com congestionamentos e movimentações em massa, mostrando a relevância da modelação baseada em agentes na simulação de evacuações para identificar e resolver diversos problemas.

2.3 Modelos e Tecnologias

Este subcapítulo aborda os fundamentos teóricos e as ferramentas práticas que sustentam o simulador desenvolvido para esta dissertação.

Serão apresentados e analisados os algoritmos responsáveis pela criação do ambiente e pela definição do comportamento dos agentes. A escolha das abordagens pretende equilibrar a aproximação da simulação ao realismo, procurando-se, ao mesmo tempo, manter uma complexidade computacional adequada, de modo a permitir a sua execução em hardware convencional.

Irão também ser apresentadas as bibliotecas de software e as ferramentas utilizadas neste trabalho.

2.3.1 Modelos Macroscópicos e Modelos Microscópicos

Os modelos de simulação de multidões podem ser categorizados em modelos macroscópicos e modelos microscópicos. As diferenças são claras e explicando de forma simples, um modelo macroscópico ignora o comportamento individual e trata a simulação de multidões como um fluxo, utilizando conceitos de dinâmica de fluidos. Já o modelo microscópico aborda a simulação com um nível de resolução mais elevado, pois cada elemento da simulação é modelado separadamente. No modelo microscópico são tidas em conta as interações e decisões individuais, algo que não acontece nos modelos macroscópicos [12, 13, 14].

Os modelos microscópicos apresentam uma desvantagem: complexidade computacional elevada. Cada agente necessita de recursos para calcular a sua rota e interações com outros agentes. Enquanto numa simulação com um reduzido número de interações isto possa ser negligenciável, numa simulação complexa com muitos agentes pode-se tornar inviável.

Apesar disso, os modelos microscópicos mostram-se muito úteis para se perceber movimentos individuais e apresentam alta-fidelidade comportamental, ou seja, é possível reproduzir interações e decisões de rotas em detalhe, algo que é essencial para analisar evacuações e testar diferentes possibilidades de rotas [15]. Na verdade, os modelos microscópicos têm sido uma ferramenta muito utilizada na área de simulação de evacuação, na medida em que os modelos macroscópicos não representam de forma adequada o comportamento individual de cada agente presente na simulação [15, 16].

Tabela 1 - Comparação entre Modelos Macroscópico e Microscópico

Característica	Modelo Macroscópico	Modelo Microscópico	Citações
Nível de detalhe	Baixo (coletivo)	Alto (individual)	[12, 13, 14, 17, 18]
Unidade de simulação	Fluxo/densidade da multidão	Cada indivíduo (agente)	[12, 13, 14, 17, 18]
Interações individuais	Ignoradas ou agregadas	Modeladas explicitamente	[12, 14, 16, 18]
Aplicações típicas	Grandes multidões, evacuação em massa, planeamento	Simulação realista, análise de comportamento, segurança	[12, 14, 16, 17]
Custo computacional	Geralmente menor	Geralmente maior	[12, 13, 14, 17]

Tendo em conta as características do problema em estudo, que envolve analisar rotas de evacuação e comportamentos de indivíduos, foi escolhida a abordagem de implementação de um modelo microscópico. Desta forma é possível representar interações pessoa-a-pessoa, algo essencial para perceber a dinâmica das diferentes estratégias de evacuação (Tabela 1). Além disso, oferece a possibilidade de analisar zonas de aglomeração e tempos de evacuação, apesar de ser mais pesado computacionalmente [15].

2.3.2 Algoritmos de Pesquisa de Caminho em Grafos (Pathfinding)

Uma vez que o ambiente desta simulação é definido por um grafo de navegação, é necessário escolher e utilizar um algoritmo que descubra o melhor caminho desde o grafo de partida ao grafo de destino/objetivo. Esta tarefa é chamada de pathfinding e é um pilar fundamental no planeamento do movimento de cada agente. No campo da teoria de grafos e inteligência artificial existem diversos algoritmos possíveis de utilizar, entre eles o algoritmo de Dijkstra e o A* (A-Estrela) [19, 20, 21, 22].

Algoritmo de Dijkstra

O algoritmo de Dijkstra, criado por Edsger W. Dijkstra em 1956, mas apenas publicado em 1959, é um dos algoritmos mais conhecidos para encontrar o caminho mais curto num grafo. Este algoritmo tem uma abordagem exaustiva ao problema, pois explora o grafo de forma radial para calcular a distância mais curta para todos os nós alcançáveis (Figura 3). Apesar desta abordagem garantir a descoberta do caminho mais curto, fá-lo de forma crescentemente ineficiente ao expandir a busca em todas as direções, sem qualquer noção da direção do nó objetivo [22, 23, 24, 25].

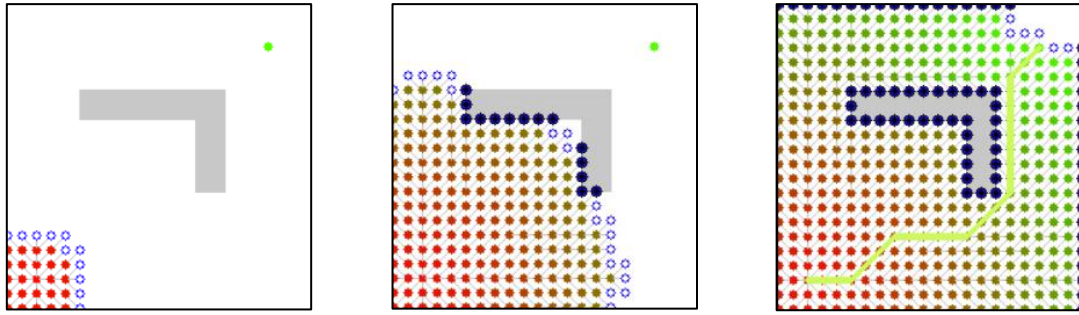


Figura 3 - Demonstração do algoritmo Dijkstra [26].

Algoritmo A*

O algoritmo A* (A-Estrela), publicado por Peter Hart, Nils Nilsson e Bertram Raphael em 1968, é uma evolução do algoritmo de Dijkstra. Este é um algoritmo de pesquisa informada, que significa que utiliza informação adicional para explorar o grafo de forma mais inteligente. Tal como o algoritmo de Dijkstra, o A* considera o custo do caminho desde o ponto de partida até ao nó atual. Porém, a grande diferença é que incorpora também uma função heurística que permite ao algoritmo estimar o custo do caminho desde o nó atual até ao nó objetivo (Figura 4). Geralmente a heurística mais usada é a distância Euclidiana em linha reta até ao destino, de forma a permitir que o A* prioriza a exploração de nós que parecem estar no caminho certo, em vez de explorar cegamente todos os nós de forma radial como faz o algoritmo de Dijkstra [19, 20, 21, 27, 28].

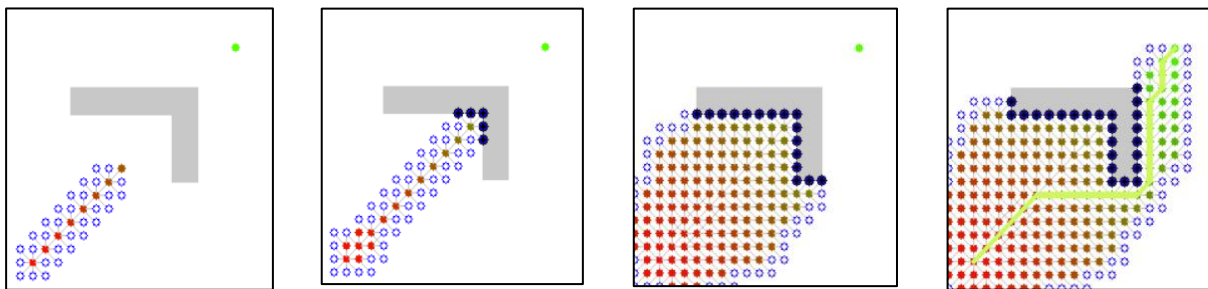


Figura 4 - Demonstração do algoritmo A* [29].

Comparação entre A* e Dijkstra

A principal característica vantajosa de ambos os algoritmos é a garantia de serem ótimos, ou seja, ambos encontrarão o melhor caminho (mais curto) de um ponto até ao objetivo. No entanto, a grande diferença é sentida na performance. A diferença do número de nós explorados entre o algoritmo A* e o Dijkstra é crucial, já que o A* explora significativamente menos nós em comparação com o Dijkstra. Quanto maior for o grafo e quanto menor for o número de obstáculos, mais diferença se sentirá na performance do algoritmo, como é o caso do Parque Urbano de Albarquel [24, 25, 30].

Numa simulação como esta, onde podem existir centenas de agentes e cada um deles necessita de calcular a sua melhor rota, é essencial que se faça um balanço entre performance e realismo, de forma a assegurar que os agentes se movem de forma inteligente e orientada para o objetivo. Assim sendo, a escolha do algoritmo A* para o problema do planeamento do caminho de cada agente tornou-se óbvia.

2.3.3 Modelos de Movimento e Interação Local

As simulações de agentes em evacuações de multidões necessitam de algoritmos realistas que simulam as interações que estes têm com o ambiente e com outros agentes. Existem diversos modelos usados neste contexto, entre eles os seguintes: Modelo de Força Social (Social Force Model - SFM), modelos baseados em regras (por exemplo, o algoritmo Boids de Craig Reynolds), modelos de Autómatos Celulares (Cellular Automata - CA) e modelos baseados em decisões cognitivas (como agentes BDI – Belief-Desire-Intention). Todos estes diferem em eficiência computacional, realismo comportamental e adaptabilidade a diferentes cenários. Veremos então quais são as características de cada um, para os podermos comparar.

Modelo de Força Social (SFM)

O Modelo de Força Social (SFM), proposto por Helbing e Molnár em 1995, assume os agentes como uma partícula com “forças sociais” que representam direções para onde o agente se deve mover. Estas forças são as seguintes: Primeiramente, um termo que representa a aceleração em direção ao objetivo. Em segundo, termos que fazem o agente manter a distância a outros agentes e a obstáculos, ou seja, forças de repulsão. Em terceiro, termos que representam efeitos atrativos que ajudam a simular grupos (por exemplo força de atração na direção de amigos ou família) [31, 32, 33].

Estas forças são combinadas em equações de movimento para resultar uma força final, que representa a direção e aceleração do agente. Quando este modelo é aplicado a muitos agentes, consegue formar interações realistas e interessantes que são observadas em multidões, razão porque o Modelo de Força Social se tornou um dos mais utilizados para simular grandes grupos de agentes [34, 35, 36]. Através da Figura 5, vemos um exemplo deste modelo a funcionar.

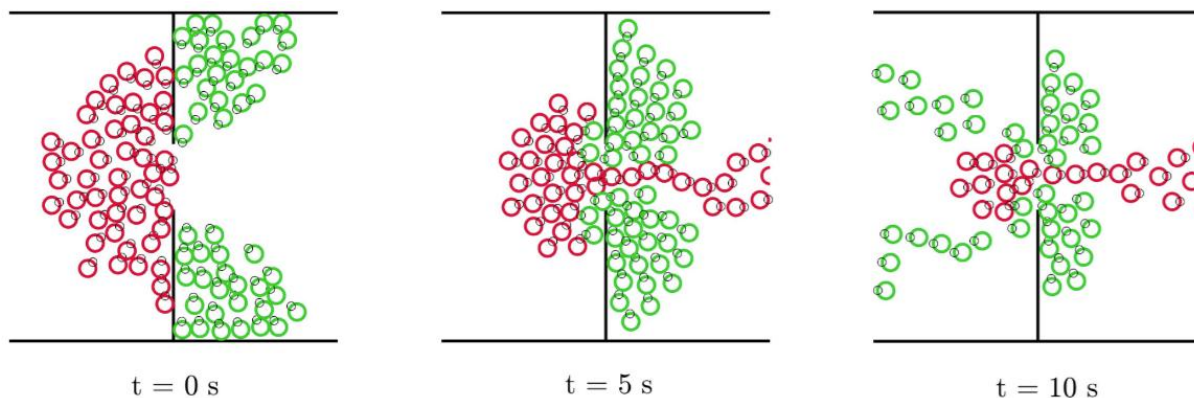


Figura 5 - Fluxo de pedestres em diferentes sentidos através de uma zona de afunilamento [31].

2.3.4 Modelos Baseados em Regras

Os modelos baseados em regras, como o modelo de Boids de Craig Reynolds, têm como objetivo simular comportamentos coletivos e interações complexas entre agentes individuais a partir de regras simples como: a regra de separação cujo objetivo é evitar a proximidade de agentes para evitar colisões, a regra do alinhamento cujo objetivo é ajustar a direção e velocidade do agente e a regra da coesão que pretende manter a coesão do grupo (Figura 6) [37, 38, 39].

Cada *boid* opera de forma autónoma em relação tanto ao ambiente quanto aos demais *boids*, sendo guiado por um conjunto de regras simples que, em conjunto, resultam em comportamentos coletivos complexos e interessantes. Por existir esta simplicidade de regras, o modelo de Boids tornou-se popular e acabou por servir de base para outros modelos, como a adição de regras novas para que seja possível existir um líder num grupo [37, 40, 41].

Estes tipos de modelos são fáceis de implementar e leves computacionalmente. Em termos de realismo, estes modelos estão, todavia, vários níveis abaixo dos modelos de força social. Contrariamente ao SFM, o modelo de Boids não implementa diretamente forças de contacto, o que prejudica o realismo da simulação, especialmente em zonas de afunilamento como portas ou escadas. Assim, embora úteis para comportamentos de grupo, modelos baseados em regras são menos comuns em estudos sobre evacuações de alta densidade, dada a necessidade de interações físicas mais precisas [37, 39, 42].

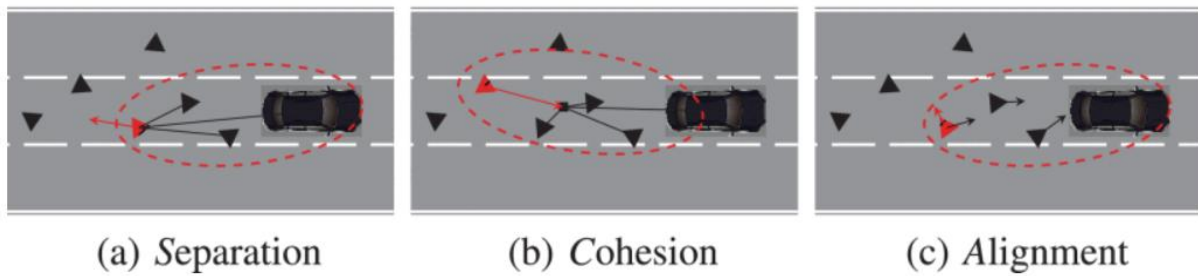


Figura 6 - Três regras básicas do movimento de Boids [43].

Modelos de Autômatos Celulares (CA)

Os Modelos de Autômatos Celulares discretizam o espaço numa grelha de células e permitem que cada indivíduo se desloque de célula em célula de acordo com regras locais (Figura 7). A cada passo de tempo discreto, cada agente pode mover-se para uma célula adjacente livre, segundo uma heurística [44].

Estes modelos destacam-se pela sua simplicidade e eficiência. Como cada agente apenas verifica as células vizinhas, as atualizações são computacionalmente leves e facilmente paralelizáveis. O modelo de autômato celular tornou-se um dos mais utilizados em simulações de evacuação por garantir elevada eficiência, grande escalabilidade e facilidade de adaptação a diferentes layouts: basta definir a grelha conforme a planta do ambiente (com portas e obstáculos representados como células) e o modelo consegue simular a evacuação nesse espaço [45, 46, 47, 48, 49].

Os modelos de autômatos celulares são amplamente utilizados para simular evacuações de grande escala, em que a rapidez da simulação e a simplicidade do modelo têm grande relevância [44, 50].

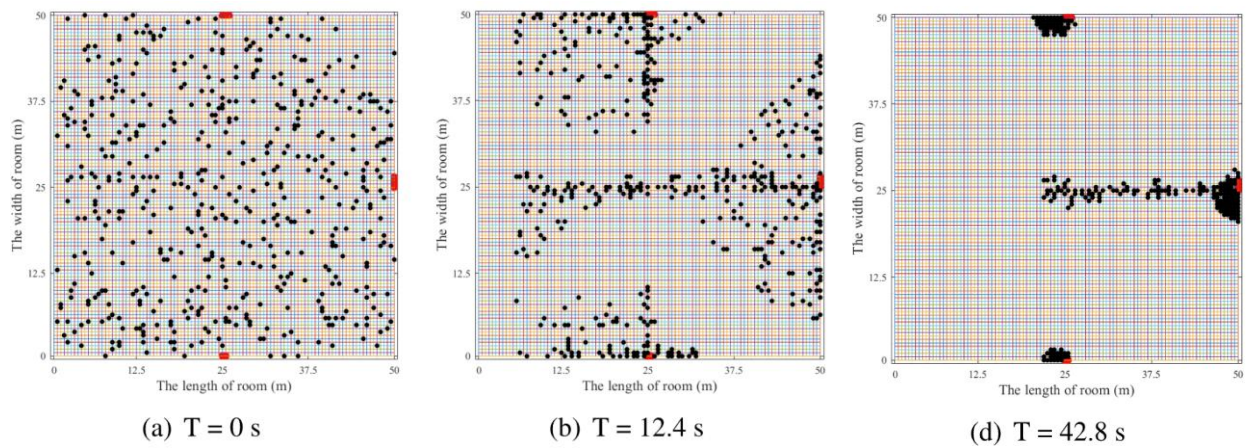


Figura 7 - Evacuação de pedestres em diferentes instantes (Modelo Celular) [51].

Cada modelo é um balanço entre realismo e viabilidade computacional, alguns simples, porém não muito realistas, outros mais complexos e mais realistas.

CAPÍTULO 3 MODELO ABSTRATO

Para estudar a dinâmica de evacuação de pedestres no parque em análise, foi adotada a metodologia de Modelação Baseada em Agentes (ABM - Agent-Based Modeling). Esta abordagem é particularmente adequada para sistemas complexos nos quais o comportamento global emerge das interações locais de múltiplos agentes autónomos.

Este capítulo dedica-se à descrição do modelo abstrato que serve de fundamento à simulação. O seu propósito é definir formalmente os componentes do sistema — o ambiente e os agentes — e as regras que definem o seu comportamento e interação.

3.1 Modelo do Ambiente

O ambiente da simulação representa o espaço físico do Parque Urbano de Albarquel. Os agentes situam-se e deslocam-se neste espaço, por isso é importante conhecer tanto o ambiente real como as características do ambiente simulado.

3.1.1 Parque Urbano de Albarquel

O Parque Urbano de Albarquel é um espaço amplo com relva, praia e passeios onde ocorrem diversos eventos que reúnem grandes grupos de pessoas no mesmo sítio. Para que as pessoas se dirijam a zonas seguras no caso de um *tsunami*, têm de passar por escadas íngremes e algumas não muito largas.

O trabalho de campo foi realizado no dia 7 de junho de 2025 com o propósito de conhecer o espaço e de apoiar as simulações realizadas no presente trabalho. Todas as fotos neste capítulo forma tiradas neste dia.

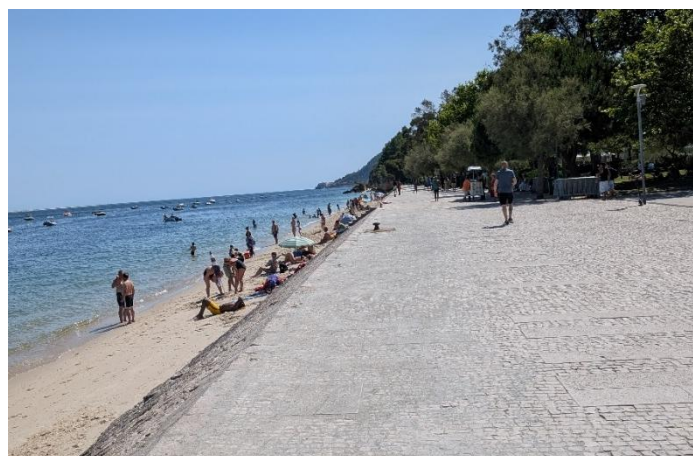


Figura 8 - Passeio largo e praia do Parque (foto tirada no dia 07/06/2025).

A Figura 8 mostra o passeio onde muitas pessoas se deslocam para observar a paisagem e passear. É possível ver também a praia que não é muito profunda, mas tem uma extensão grande ao longo de todo o parque. Na Figura 9 observa-se o acesso à praia feito através de escadas, sendo que a sua utilização é a única maneira de sair da zona de praia e chegar ao passeio.



Figura 9 - Escadas da praia.

Como existem apenas algumas escadas ao longo da praia, estas acabam por condicionar a deslocação das pessoas que tentam sair da zona da praia, tornando a evacuação mais lenta e perigosa.

Para chegar à zona segura, as pessoas têm de passar por uma de duas escadas. As escadas perto do café são estreitas e apenas duas pessoas conseguem subir lado a lado (Figura 10). Já as escadas mais a Oeste é possível três a quatro pessoas subirem simultaneamente lado a lado (Figura 11).

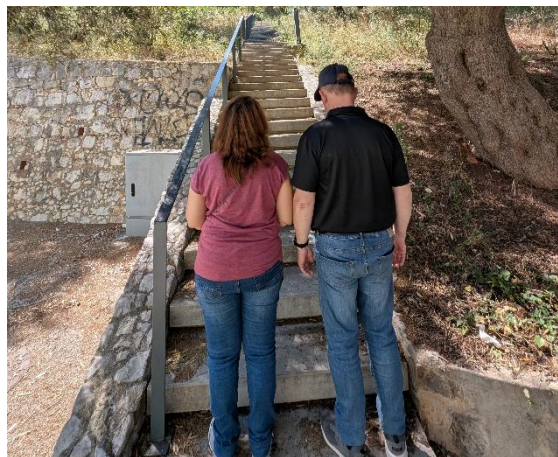


Figura 10 - Escadas perto do café.

Uma nota importante sobre as escadas perto do café é que por haverem mesas e cadeiras perto, o acesso e deslocação nesta zona fica mais difícil e pode agregar muita gente no mesmo sítio, tornando a evacuação mais lenta e perigosa. Nas escadas mais a Oeste existe uma zona de afunilamento que junta muita gente no mesmo sítio; no entanto, como estas escadas são mais largas, a evacuação é mais rápida e célere.



Figura 11 - Escadas mais a Oeste.

Um detalhe importante observado foi a presença de sinaléticas de evacuação, no entanto muitos dos sinais são praticamente impercetíveis e alguns até se encontram cobertos por estruturas, como se observa na Figura 12, enfatizando a importância da alteração da localização ou até mesmo dos sinais em si.



Figura 12 - Escadas a cobrir um sinal de evacuação.

3.1.2 Representação Espacial

O espaço físico do parque é conceptualizado como um grafo de navegação, $G = (V, E)$, onde:

- V é um conjunto de vértices (ou nós), que representam localizações discretas e alcançáveis no ambiente. Cada vértice $v \in V$ possui uma coordenada geográfica (x, y) .
- E é um conjunto de arestas, que representam as ligações ou caminhos diretos entre pares de vértices. Uma aresta $(u, v) \in E$ significa que um agente se pode deslocar diretamente entre as localizações u e v . O peso de cada aresta corresponde à distância euclidiana entre os vértices que liga.

Esta representação em grafo transforma o problema de navegação em espaço contínuo num problema de procura de caminho num espaço discreto, que é computacionalmente mais simples e eficiente.

3.1.3 Representação Temporal

O tempo no modelo é discreto e representa o estado da simulação num dado instante. A simulação avança em passos de tempo (time steps) de duração fixa, Δt . Em cada passo t , o estado de todos os agentes é recalculado com base no estado do sistema no passo $t-1$. Esta abordagem permite observar a evolução da simulação de forma iterativa.

3.2 Modelo do Agente

O agente é a entidade ativa e autónoma do modelo. Representa um indivíduo cujo objetivo é evacuar do parque. Cada agente possui um estado interno e um conjunto de regras comportamentais que definem as suas ações a cada passo de tempo.

3.2.1 Regras Físicas

Os agentes seguem regras para que a simulação seja similar à realidade. Para isso é necessário fazer com que os agentes não atravessem por paredes ou desníveis de altura impossíveis de atravessar. Tal como na vida real, os agentes não podem passar por dentro de outros agentes e podem-se empurrar mutuamente em situações de grande afluência. A velocidade dos agentes é medida em metros por segundo e alterada conforme a área em que estes se deslocam, para uma maior aproximação à realidade.

3.2.2 Atributos do Agente

O estado de um agente a qualquer momento é definido por um conjunto de atributos:

- **Identificador (ID):** Um identificador único para cada agente.
- **Posição (x, y):** As coordenadas do agente no espaço do ambiente.
- **Estado (S):** O estado atual do agente, que pode ser ATIVO (a tentar evacuar) ou EVACUADO (alcançou uma zona segura).
- **Velocidade Preferida (v_pref):** A velocidade à qual o agente se move na ausência de obstáculos ou outros agentes.
- **Destino (D):** A zona segura específica para a qual o agente se dirige.
- **Plano de Rota (P):** Uma sequência ordenada de vértices do grafo, $P = (v_1, v_2, \dots, v_n)$, que representa o caminho que o agente pretende seguir desde a sua posição atual até ao seu destino.

3.2.3 Comportamento e Navegação do agente

Considerando a necessidade de realismo na evacuação, mas ao mesmo tempo também a necessidade de escalabilidade, optou-se por uma abordagem inspirada no modelo de força social (SFM).

Os modelos baseados em regras, como o Boids, foram desconsiderados para esta simulação por não integrarem nativamente interações de contacto físico entre agentes e obstáculos (como repulsão a curtas distâncias). Consideramos que estas interações são essenciais para simular de forma credível o comportamento de multidões em áreas de afunilamento ou de grande densidade, como as que observamos no parque [37, 39, 42].

Os Modelos de Autómatos Celulares, embora bastante eficientes, têm uma característica que limita o movimento dos agentes, que é a discretização do espaço. Esta característica torna o movimento dos agentes menos fluído e mais “quadriculado”. Numa perspetiva de analisar as micro interações entre agentes, a continuidade do espaço foi considerada essencial e por esse motivo nem sempre é o mais indicado [44, 45, 46].

A abordagem do SFM é a solução para estas necessidades. Primeiramente, funciona num espaço contínuo que permite um movimento mais realista dos agentes, mas, mais importante ainda, é o funcionamento baseado em interações de forças, que permite gerar comportamentos complexos e realistas [34, 35, 36]. A implementação abordada neste trabalho será descrita de forma detalhada no Capítulo 4.

Planeamento do Caminho Global

Este processo é executado apenas uma vez, no início de cada simulação, para determinar o caminho que o agente irá percorrer. Para calcular o caminho, foi utilizado o algoritmo A* que foi mostrado e analisado no capítulo 2.

Primeiro, o agente identifica o nó mais próximo dele (nó inicial), de seguida escolhe o nó de destino e calcula o caminho mais curto desde o nó inicial até ao nó de destino e guarda o caminho para percorrer.

Movimento Local e Interação

A cada passo de tempo cada agente ATIVO executa uma lógica de movimento para se deslocar em direção ao próximo ponto intermédio, o *waypoint*, enquanto reage à presença de outros agentes.

O cálculo do movimento de cada agente a cada passo de tempo é a implementação prática do Modelo de Força Social, discutido no capítulo anterior. O vetor de movimento de um agente é o resultado da soma de duas forças principais: uma força de atração em direção ao seu objetivo imediato (nó a que se desloca) e uma força de repulsão em relação aos agentes vizinhos.

Força de Atração (*goal_force*): Esta força representa a intenção do agente de seguir o seu caminho pré-calculado. É calculada como um vetor unitário que aponta da posição atual do agente para o próximo ponto intermédio do seu trajeto (Figura 13). Este vetor é depois multiplicado pela velocidade do agente.

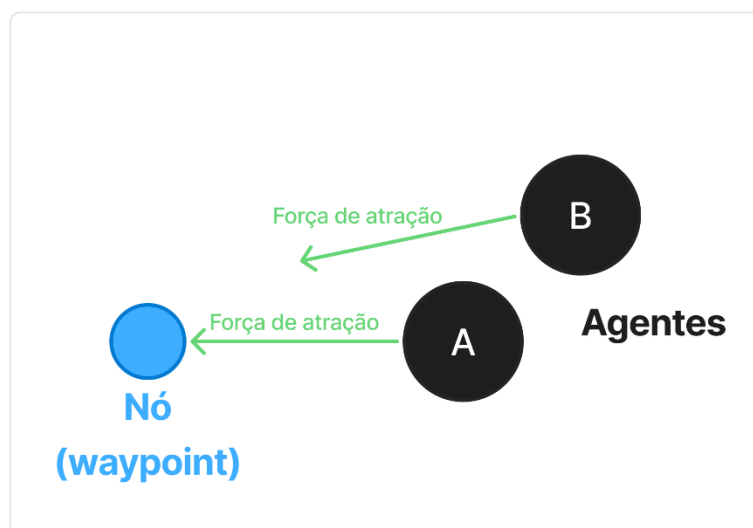


Figura 13 – Representação exemplificativa da força de atração dos agentes ao nó.

Força de Repulsão (repulsion_force): Esta força simula o comportamento social de manter um espaço pessoal e evitar colisões (Figura 14). Para cada agente, o algoritmo identifica todos os outros agentes que se encontram dentro de um raio definido. Para cada vizinho encontrado, é calculada uma força de repulsão que:

1. Atua na direção oposta à do vizinho.
2. A sua intensidade é inversamente proporcional à distância, ou seja, quanto mais próximo o vizinho, mais forte a repulsão.

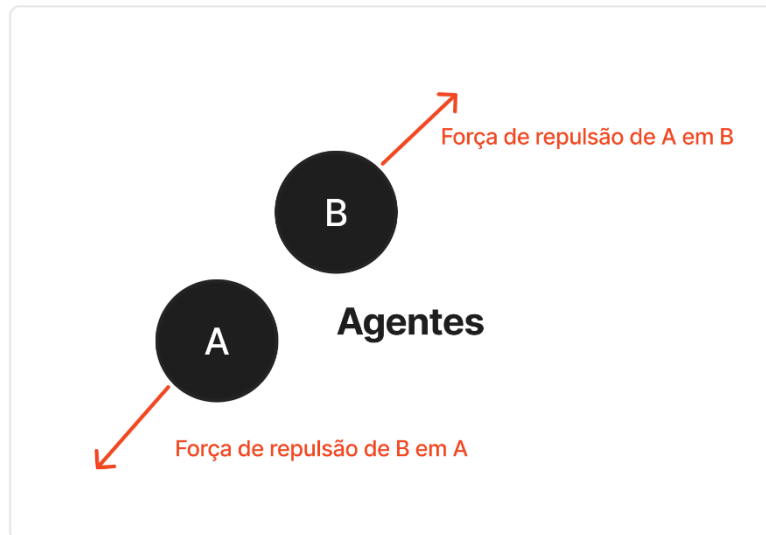


Figura 14 - Representação exemplificativa da força de repulsão entre agentes.

A força final que atua sobre o agente é a soma vetorial da força de atração e da força de repulsão combinada de todos os vizinhos, como se observa na Figura 15.

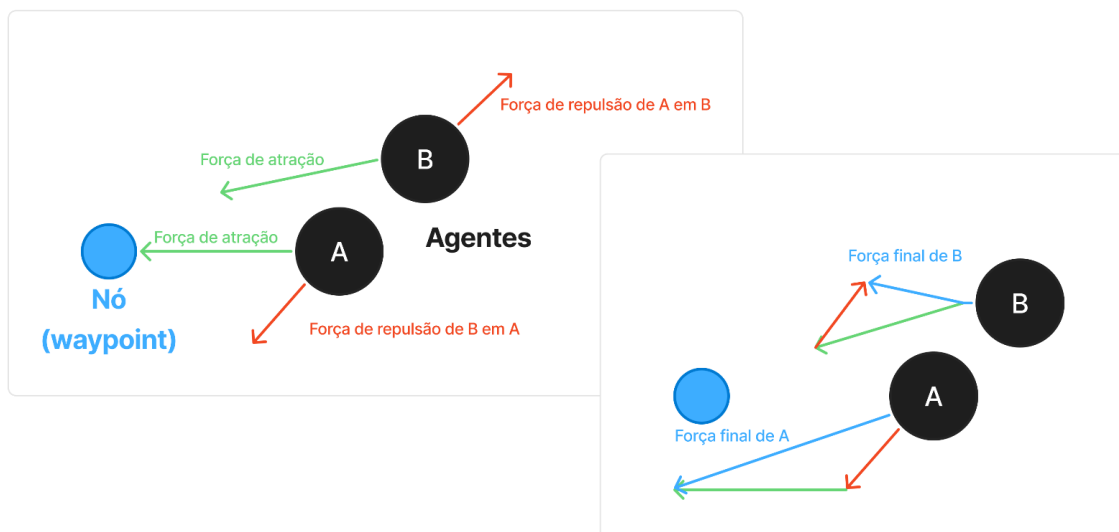


Figura 15 - Representação exemplificativa e aproximada da soma das forças de atração e repulsão.

No fim a posição dos agentes é deslocada segundo a força final, como se observa na Figura 16.

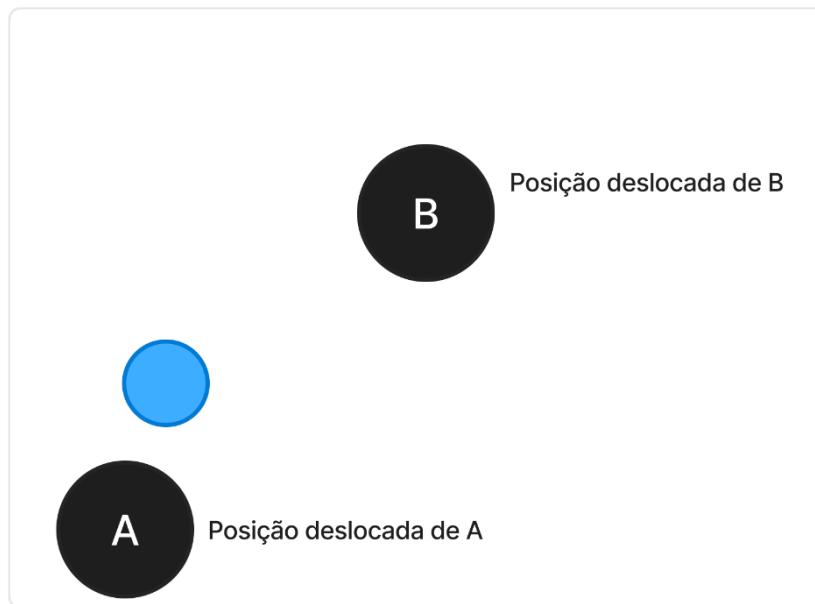


Figura 16 - Representação exemplificativa e aproximada do deslocamento dos agentes A e B segundo as suas forças finais.

CAPÍTULO 4 IMPLEMENTAÇÃO DO MODELO DE SIMULAÇÃO

O presente capítulo aborda a implementação prática do simulador. O seu objetivo é detalhar a forma como o software foi desenvolvido e de que forma é executado. Para tal serão explicadas as funções primordiais do programa, mostrando a sua lógica e o modo como interagem entre si.

4.1 Software e Ferramentas

4.1.1 Sistema de Informação Geográfica (GIS): QGIS

O QGIS é um sistema de informação geográfica open-source que permite desenhar, visualizar, editar e analisar dados geográficos (Figura 17). Com esta ferramenta é possível criar e gerir dados geoespaciais como shapefiles que são essenciais nesta investigação [52].

Este software foi utilizado para desenhar a planta do parque através de polígonos vetoriais, desta forma possibilitando definir diferentes áreas como “praia”, “escadas” e “relvado”. Esta representação é a base sobre a qual toda a simulação executa.

Os polígonos criados têm alguns dados associados: a forma geométrica e um atributo chamado “nome”. Este atributo permite categorizar as áreas (por exemplo, “porta”, “safe_zone”) que posteriormente irá ser lido pelo simulador para criar regras específicas, como restringir a navegação para que os agentes tomem o caminho desejado, ou até identificar zonas com características específicas.

Em suma, o QGIS atua como um transformador do mundo real para o mundo virtual, criando um cenário geograficamente preciso para que os agentes o possam usar posteriormente.

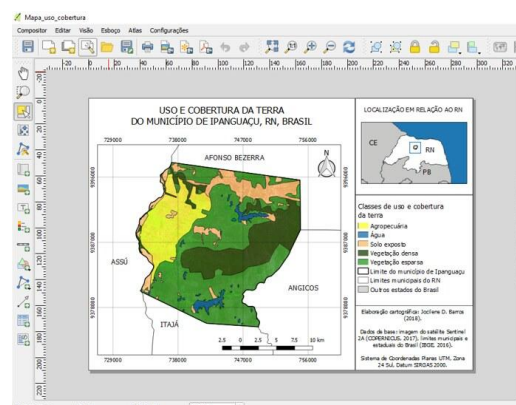
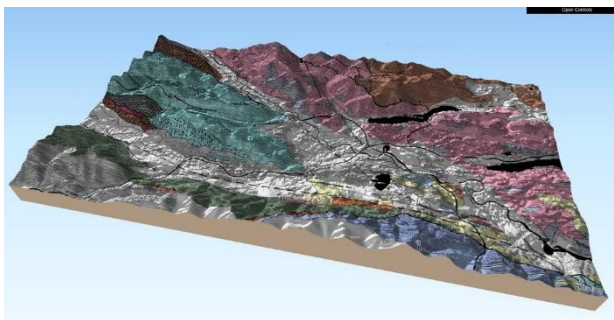


Figura 17 - Exemplos de utilização do QGIS [53].

4.1.2 Ambiente de Desenvolvimento: Python e Bibliotecas

A implementação da simulação foi realizada em Python, uma linguagem de programação amplamente utilizada em contextos científicos e acadêmicos. A escolha desta linguagem justifica-se pela sua versatilidade e, sobretudo, pela extensa disponibilidade de bibliotecas que suportam o desenvolvimento de modelos de simulação e a integração com ferramentas de análise e visualização de dados. Entre as bibliotecas utilizadas destacam-se:

- GeoPandas e Shapely: Estas duas bibliotecas são as responsáveis por fazer a ponte entre o QGIS e o Python. O GeoPandas é uma extensão da biblioteca Pandas que torna possível operações espaciais em DataFrames, que facilita a leitura e manipulação dos ficheiros shapefile gerados no QGIS [54]. O Shapely permite a funcionalidade de alteração e análise dos objetos geométricos, sendo necessário para verificar, por exemplo, se um ponto está contido numa geometria (através de `.contains`) [55].
- NetworkX: Esta é a biblioteca necessária para a criação do ambiente com um grafo. Foi utilizado para criar a estrutura de dados do grafo de navegação, adicionar os nós, ligá-los através de arestas e executar o algoritmo A*, através da função `nx.astar_path`, que calcula o caminho ótimo para os agentes [56].

Folium: Esta é a biblioteca que permite visualizar os resultados da simulação. É utilizada para criar mapas interativos em HTML. Tem um plugin chamado `TimestampedGeoJson` que permitiu criar uma visualização dinâmica do movimento dos agentes ao longo do tempo, facilitando a análise dos resultados [57].

4.1.3 Ambiente de Execução

A análise aos tempos de execução da simulação em diferentes cenários é impactada diretamente pelo *hardware* do computador que a executa. Para que se possa garantir transparência dos resultados apresentados, é importante documentar as especificações do computador. As especificações são as seguintes:

- Processador (CPU): Intel(R) Core(TM) i7-10750H CPU @ 2,60GHz (Nº de cores = 12)
- Memória RAM: 16 GB
- Placa gráfica (GPU): NVIDIA GeForce GTX 1650 Ti (VRAM = 12GB)
- Sistema operativo: Windows 11

Com a especificação destes dados é possível ter um ponto de referência para os tempos de execução e viabilidade de escalabilidade da simulação.

4.2 Construção do Ambiente de Simulação

O passo inicial da execução do simulador é a conversão do mapa geográfico, ou seja, de um ficheiro de tipo shapefile gerado pelo QGIS, numa estrutura de dados que permita o cálculo de caminhos. Este processo consiste na discretização do espaço contínuo num espaço discreto através de nós de uma grelha, seguido da construção de um grafo de navegação que incorpora certas restrições.

4.2.1 Representação Geoespacial e Grafo de Navegação

A criação de um grafo navegável é iniciada pela função `create_grid_nodes`. O processo começa pela leitura do shapefile, seguido da filtragem de algumas geometrias. Esta filtragem acontece através do atributo “nome” presente nas geometrias e pretende excluir as geometrias cujo nome tenha a string “edif” contida nele. Desta forma os nós criados do grafo não irão ser criados onde existem geometrias com “edif” no seu nome, com o objetivo de representar áreas intransponíveis como edifícios ou arbustos. Estas geometrias ficam a cor cinza na visualização do mapa como mostra a Figura 18.

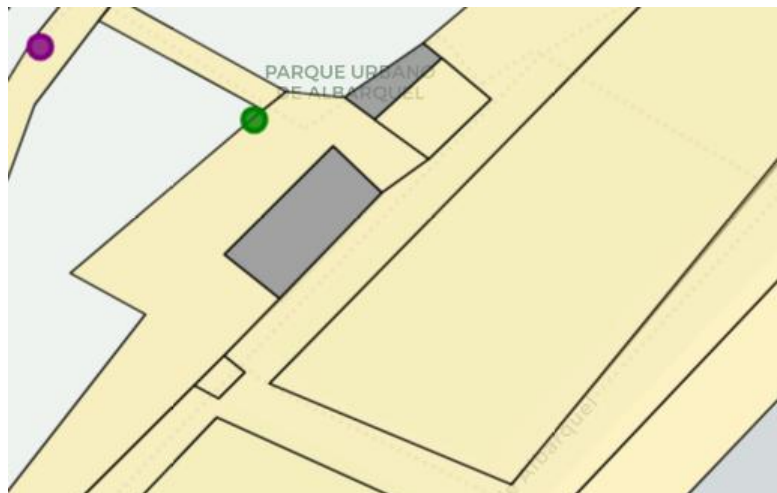


Figura 18 - Visualização do mapa com edifícios (cor cinza).

Subsequentemente, uma grelha de pontos é gerada e sobreposta aos limites das áreas navegáveis. A distância entre os pontos adjacentes desta grelha é definida pela variável global `GRID_RESOLUTION_METERS`. Este parâmetro é fundamental pois define a resolução do movimento dos agentes: valores mais baixos tornam o grafo mais denso e trajetórias mais precisas, em detrimento de um maior custo computacional. Nesta dissertação e tendo em conta o hardware utilizado e algumas experiências piloto realizadas, foi escolhido o valor de três metros para este parâmetro (Figura 19). Consideramos que este valor é adequado para ter uma simulação realista e ao mesmo tempo não muito complexa computacionalmente.

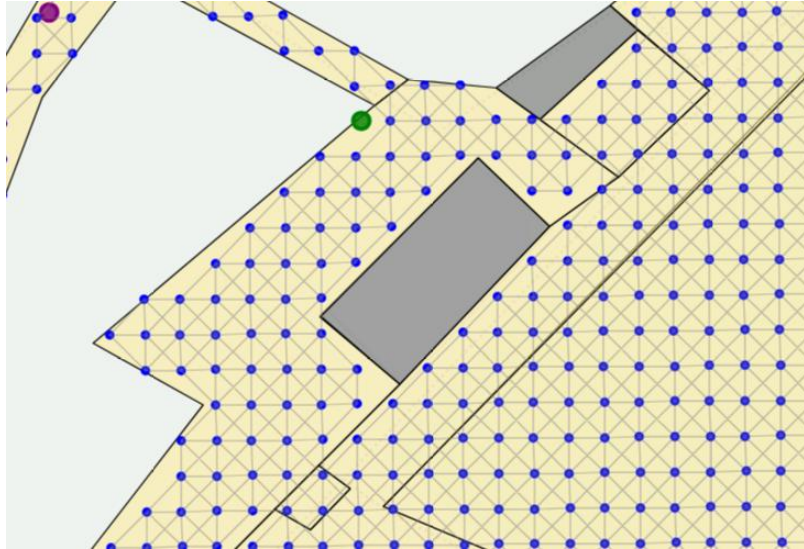


Figura 19 - Visualização do mapa com edifícios (cor cinza) e nós do grafo.

Através de uma operação de junção espacial (spatial join), são mantidos exclusivamente os pontos que se encontram contidos num polígono navegável, ou seja sem “edif” no seu nome. Cada um destes pontos é então instanciado como um nó no grafo de navegação, guardando o nome da área onde está inserido (e.g., "praia", "relvado1"), como se vê através da Figura 20.



Figura 20 - Exemplo de informação em um nó.

Este procedimento resulta numa transformação do problema de pesquisa de caminho num espaço contínuo para um problema computacionalmente menos exigente de pesquisa numa estrutura de grafo discreta.

4.2.2 Aplicação de Restrições ao Grafo

Após a criação dos nós, o próximo passo é ligá-los entre si para formar um grafo que represente os caminhos possíveis de percorrer por parte dos agentes. Esta tarefa é da responsabilidade da função `build_graph`.

A função utiliza a biblioteca `NetworkX` para instanciar um grafo vazio, depois itera sobre cada nó e identifica todos os nós vizinhos dentro de um raio definido pela resolução da grelha, com a ajuda de uma estrutura de dados `KDTree`. Após a identificação dos nós vizinhos são criadas arestas entre eles – nas direções vertical, horizontal e diagonal – sendo que é atribuído um peso a cada uma das arestas baseado na distância Euclidiana entre os nós que conecta.

Visto que o grafo criado não representa o ambiente de forma exata tal como está, temos de criar restrições para que não existam ligações entre nós, que seriam inacessíveis por uma pessoa. A título de exemplo consideramos a Figura 21, que ilustra a diferença de alturas, sendo apenas acessível através de escadas. Ambas as imagens são da autoria do autor deste trabalho. A imagem da direita foi criada através do software QGIS com a utilização de dados Lidar, disponibilizados pela Direção Geral do Território, que indicam a cota do terreno (MDT – Modelo Digital de Terreno) [62].



Figura 21 - Diferença de altura (a vermelho) entre praia e paredão em metros.

Conseguimos perceber que nesta localização a diferença de altitude é por volta de dois metros, que faz com que uma pessoa só se consiga deslocar de um ponto para outro através das escadas. Por este motivo é essencial que no grafo se quebre a ligação entre os nós da praia e do paredão (chamado “`caminho_ind1`” no shapefile criado no QGIS), apenas restando ligações pelos nós presentes em zonas de escadas, como mostra a seguinte figura. A marcação a

vermelho representa a quebra de ligações e a zona a amarelo forte são escadas (com ligação entre nós).

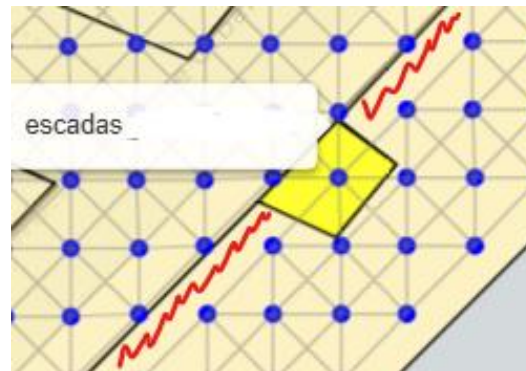


Figura 22 - Exemplo de nós sem ligação.

São também cortadas ligações entre nós noutros sítios do mapa. Para tal, a função percorre a totalidade de arestas do grafo e remove aquelas que cumprem certos requisitos. É utilizado o nome dos diversos shapefiles para perceber que ligações (arestas) os nós sobrepostos a esse shapefile devem ter. Por exemplo, caso o nó esteja num shapefile com o nome “praia” deve quebrar todas as ligações com nós que estejam num shapefile sem “escadas” ou “praia” no nome (Figura 22). Dessa forma quebra-se todas as ligações entre os nós da praia com outros que não sejam escadas ou a praia, fazendo com que os agentes saiam da praia através de escadas. Com esta implementação é possível também a criação de áreas específicas por onde os agentes devem passar, às quais chamamos “portas”, que obriga uma percentagem de agentes a passar por essa zona, para que não escolham todos o caminho mais curto ao destino. O produto final é um grafo de navegação que não só representa a topologia do espaço, mas também tem em conta as determinadas restrições ou regras de funcionamento.

4.3 Implementação do Agente e Respetivos Comportamentos

Após a construção do ambiente da simulação, o foco da implementação está nos agentes. Este subcapítulo detalha a forma como o agente toma decisões, bem como o seu processo de criação e posicionamento inicial no ambiente.

4.3.1 Processo de Criação e Posicionamento

A posição inicial dos agentes na simulação é um fator determinante para os seus resultados. No modelo implementado, o que controla este processo é o parâmetro `AGENT_SPAWN_DISTRIBUTION`. Este parâmetro consiste num dicionário Python

constituído por uma string que indica o nome da shape definida no shapefile e por um número de 0 a 1, que corresponde à percentagem de agentes que serão posicionados naquela shape em relação ao número total de agentes. Um exemplo do formato do dicionário é o seguinte: `AGENT_SPAWN_DISTRIBUTION = [{"area_eventos1", 0,19}, {"praia", 0,50}]`. Neste caso apareceriam 19% do total de agentes na shape chamada “area_eventos1”, 50% do total na shape “praia” e a restante percentagem espalhada aleatoriamente pelo resto do ambiente. A Figura 23 mostra um exemplo de diferentes distribuições de agentes no ambiente.

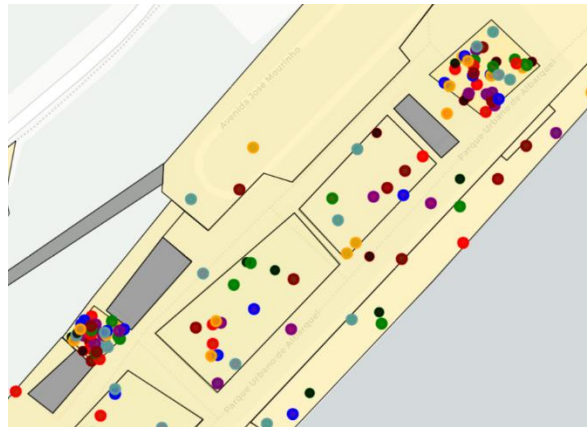


Figura 23 - Exemplo de posicionamento mais concentrado de agentes em shapes específicas.

O processo de posicionamento de agentes numa área específica é feito da seguinte forma. Primeiramente, é selecionada uma das áreas de partida com base na distribuição definida. Depois são obtidos os limites (bounds) do polígono que corresponde à área selecionada. Seguidamente é gerado um ponto com coordenadas aleatórias dentro desses limites e depois verifica-se se o ponto gerado está efetivamente dentro do polígono e não numa área vazia do retângulo que envolve o polígono. Se o ponto for inválido é gerado um novo ponto até que uma posição válida seja encontrada.

Desta forma é possível controlar a posição inicial dos agentes, que permite testar diferentes cenários da simulação e perceber o impacto de cada uma.

4.3.2 Lógica de Decisão de Caminhos

Uma vez posicionados, os agentes necessitam de calcular o caminho de evacuação que cada um irá percorrer. Esta “decisão” de caminho não é homogênea na população inteira de agentes. Foi implementada uma lógica que pretende tornar a simulação mais realista, fazendo com que os agentes não escolham todos o mesmo destino e caminho. A função principal neste processo é a

`get_shortest_path`, que apesar do nome, foi alterada para que não seja apenas uma pesquisa pelo caminho mais curto.

O processo de decisão de um agente começa por o agente identificar todas as áreas designadas como `safe_zone` no mapa. Após a identificação destas áreas o agente irá, por defeito, calcular e escolher, através do algoritmo A*, o caminho mais curto para a `safe_zone` mais próxima. No entanto, há comportamentos probabilísticos que podem fazer com que o agente se comporte de formas diferentes. Para simular agentes que, por falta de informação ou por seguirem um fluxo contrário, não se dirigem à saída mais próxima, foi introduzido o parâmetro `PROB_FARTHEST_TARGET`. Este valor define a probabilidade de um agente ignorar a `safe_zone` mais próxima e calcular o caminho para a `safe_zone` mais distante de si. Caso o valor seja 0, todos os agentes seguem para a saída mais próxima de si, caso seja 1 todos os agentes seguem para a saída mais distante de si.

Existe outro comportamento probabilístico que vem resolver um problema sentido ao executar a simulação. Este problema surgia quando muitos agentes queriam ir para a `safe_zone` mais distante de si. Ao calcularem o caminho mais curto, todos os agentes iam pelo mesmo caminho, apesar de haver alternativas adequadas. Na Figura 24 mostra este comportamento a acontecer onde os agentes que pretendem ir para cima não usam o caminho a azul, e tentam todos ir pelo caminho mais curto, gerando um aglomerado de agentes, fazendo com que muitos não se consigam deslocar.



Figura 24 - Aglomerado de agentes causado por não utilizarem caminhos alternativos.

Para resolver esta situação implementou-se o que chamamos de “porta” para que os agentes sejam obrigados a passar por lá. Através do parâmetro `PROB_FARTHEST_THROUGH_ALL_DOORS`, é definida a probabilidade, entre os agentes que escolhem a `safe_zone` mais distante, de seguirem pelo caminho alternativo através de todas as portas. Na Figura 25 está assinalado a azul uma porta colocada na rampa que os agentes não estavam a utilizar (mostrado na Figura 24). Através da implementação da porta, os agentes já se deslocam por lá, dispersando o fluxo por outros caminhos.

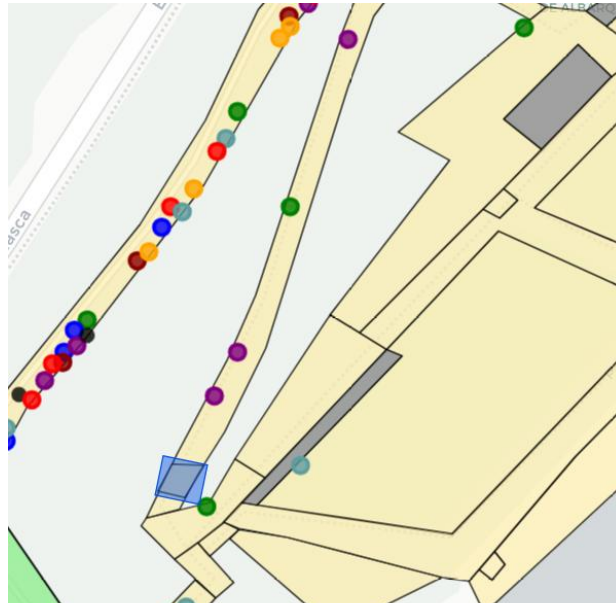


Figura 25 - Utilização de porta para controlar o fluxo dos agentes.

Quando um agente é selecionado para seguir pela porta através da probabilidade definida no parâmetro referido, o seu caminho é calculado pela função `get_path_through_doors`. Esta função identifica todas as áreas com nome “porta” e calcula, através do algoritmo A*, o melhor caminho para o agente ir da posição em que começa até à primeira porta; depois calcula o melhor caminho dessa porta até à porta seguinte, e assim sucessivamente até chegar à `safe_zone`. Neste caso, uma só porta foi suficiente para que os agentes se desloquem através do caminho pretendido. Desta forma é possível controlar o caminho que os agentes tomam, bem como a quantidade de agentes que o fazem.

A adição e combinação destes diferentes comportamentos permitem flexibilizar o funcionamento do modelo e acabam por resultar num modelo mais rico e variado.

4.4 Ciclo de Execução

O ciclo de execução é o componente principal onde o tempo avança de forma discreta e o estado de cada agente é atualizado iterativamente. Esta secção descreve a arquitetura deste ciclo, detalha a mecânica de movimento e interação baseada no Modelo de Força Social, e explica como o estado de cada agente é alterado ao longo do tempo.

4.4.1 Inicialização e Estrutura Temporal

A simulação inicia-se com a definição de um ciclo temporal (for step in range), que limita a execução ao número máximo de passos definido. Como se observa no topo da Figura 26, a primeira fase de cada passo consiste na preparação do estado global:

- Identificação de agentes ativos: O sistema verifica quais os agentes que ainda permanecem no recinto.
- Reset do Buffer: É limpo o buffer temporário. Este componente é usado para garantir que as decisões de um agente num determinado instante não influenciem as decisões de outros agentes no mesmo passo de tempo.
-

4.4.2 Ciclo Interno do Agente e Cálculo de Forças

Após a preparação do passo, inicia-se o Ciclo do Agente. Para cada indivíduo, o sistema realiza uma verificação inicial: se o agente já terminou o seu percurso, o ciclo avança para o próximo; caso contrário, procede-se ao cálculo do movimento:

- Cálculo de Forças SFM: É calculada a soma vetorial da Força de Objetivo (atração ao próximo nó) e da Força de Repulsão (afastamento de outros agentes), resultando na Força Total. Neste ponto, o sistema verifica também se o agente se encontra numa "área lenta", ajustando a velocidade base.
- Determinação do Deslocamento: Com base na Força Total e na velocidade desejada, é calculada a posição potencial (potential_pos).

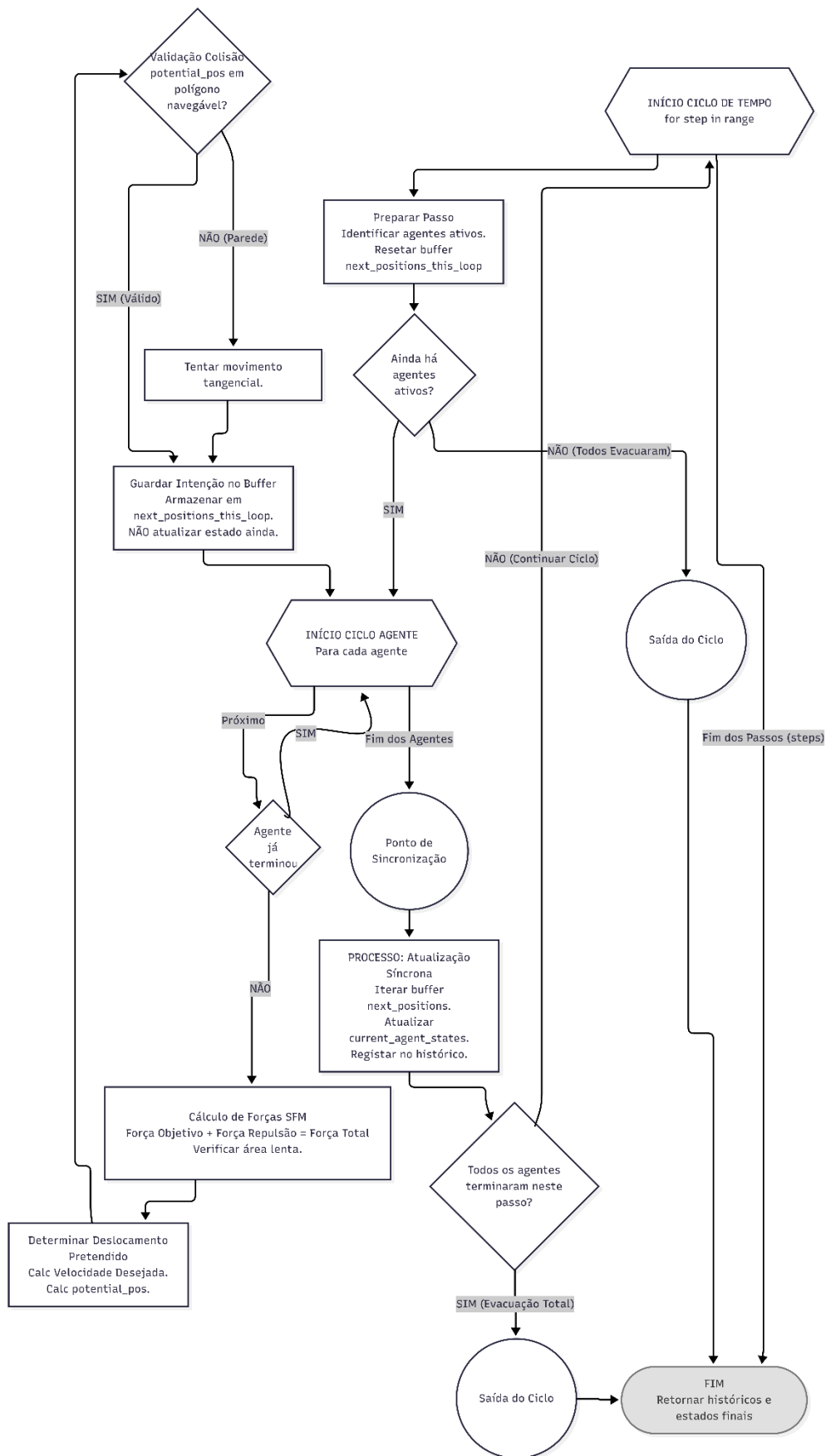


Figura 26 - Fluxograma representativo da lógica de execução e sincronização do modelo de simulação.

4.4.3 Validação Espacial e Movimento Tangencial

Um dos aspetos críticos visíveis no fluxograma é a Validação de Colisão. Antes de qualquer movimento ser confirmado, a posição potencial é testada contra as geometrias do Sistema de Informação Geográfica (GIS):

- SIM (Válido): Se a posição se encontra dentro de um polígono navegável, a intenção é guardada diretamente no buffer.
- NÃO (Parede/Obstáculo): Caso o agente tente atravessar uma área intransponível, o algoritmo executa um teste de movimento tangencial. Esta lógica permite que o agente "deslize" ao longo da parede em vez de ficar totalmente imobilizado, procurando um vetor alternativo que ainda o aproxime do objetivo.

4.4.4 Sincronização e Atualização do Estado

Concluído o processamento de todos os agentes individuais, a simulação atinge o Ponto de Sincronização. É neste momento que ocorre a transição do estado "pretendido" para o estado "real":

- Atualização Síncrona: O sistema itera sobre o buffer, atualiza efetivamente as coordenadas em `current_agent_states` e regista os novos dados no histórico da simulação.
- Condições de Paragem: O ciclo termina se for atingida a Evacuação Total (todos os agentes terminaram) ou se o contador de passos atingir o limite máximo (Fim dos Passos).

Finalmente, o simulador retorna os históricos e os estados finais para posterior análise estatística e visualização.

4.5 Desafios e Soluções no Movimento de Agentes

Neste subcapítulo, são abordados alguns desafios encontrados durante as simulações de evacuação, especificamente aqueles relacionados ao movimento dos agentes no ambiente simulado.

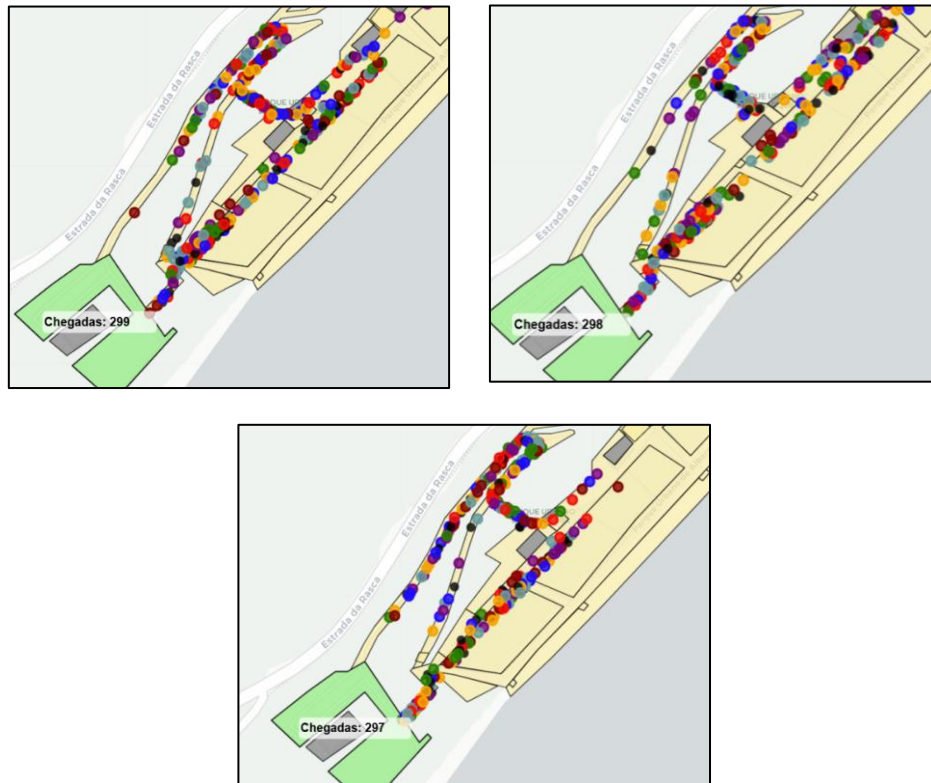


Figura 27 - Simulação de cenário base com 300 pessoas. Caminhos percorridos e zonas de congestão são observáveis.

Através da Figura 27 conseguimos perceber que nas três simulações não chegaram todos os agentes à zona segura até ao final dos 20 minutos, em resultado dos agentes ficarem presos em certos sítios sem que se consigam libertar. Isto acontece por dois motivos que explicamos com maior detalhe mais à frente:

1. O primeiro motivo é a resolução do grafo que por vezes pode fazer com que o agente não se consiga deslocar, dependendo da geometria.
2. O segundo motivo é mais complexo e ocorre quando um agente atinge um nó, mas acaba por ser empurrado para trás por outros agentes. Em certas situações, o agente alcança um determinado nó e define o nó seguinte como o seu novo alvo de deslocamento; contudo, durante o movimento em direção a esse nó, é empurrado pela força social exercida pelos restantes agentes, regressando assim a uma zona da qual já não consegue prosseguir para o próximo nó.

Nas seguintes figuras são ilustrados exemplos de ambas as situações.

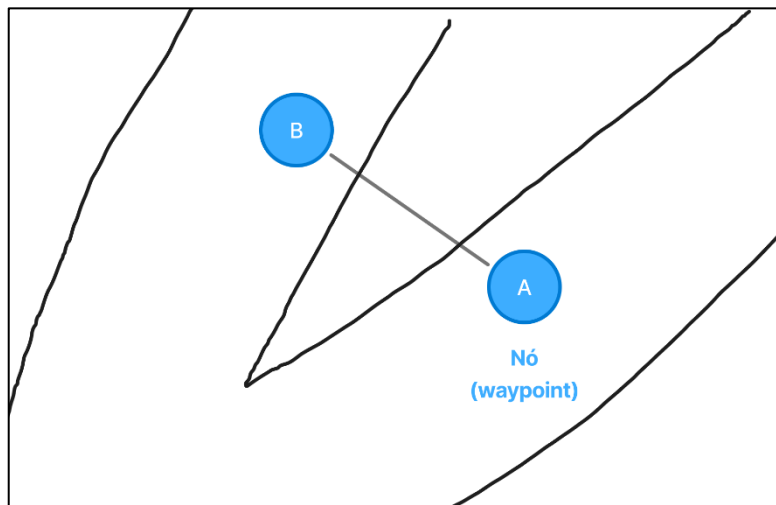


Figura 28 – Ligação entre nós por cima de zona inacessível pelo agente.

Pela Figura 28 observamos que o grafo tem uma ligação entre estes dois nós, mas quando o agente se tenta deslocar não consegue, porque encontra um limite fora da zona onde pode andar, ficando, por conseguinte, encalhado.

Na Figura 29 vemos os vários passos que levam ao agente ficar preso por ser empurrado para trás por outros agentes.

1. Agente verde chegou ao nó A e pretende deslocar-se para o nó B.
2. Agentes acumulam-se por cima do nó B por causa de uma zona de afunilamento.
3. Agentes acumulam-se ainda mais e empurram-se mutuamente, fazendo com que alguns sejam empurrados em direções indesejáveis.
4. Agente verde ainda pretende ir para o nó B, mas agora tem um obstáculo à frente.
5. Já não existe concentração excessiva de agentes, no entanto o agente verde está a tentar ir para o nó B sem contornar o obstáculo, ficando por isso preso.

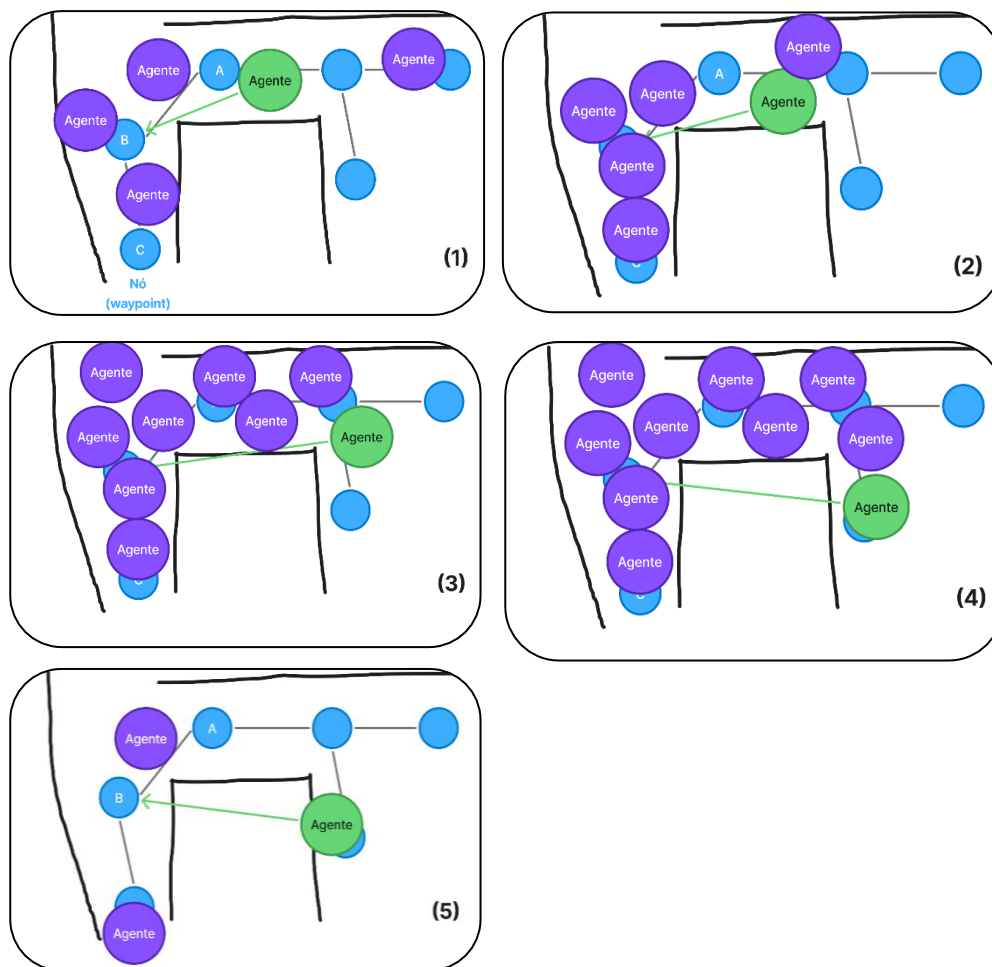


Figura 29 - Demonstração exemplificativa de agente a ficar preso.

Para solucionar esta situação foi implementada uma movimentação aleatória aos agentes caso se tentem deslocar para um sítio impossível (Figura 30). Isto resulta para alguns agentes, fazendo com que saiam do sítio onde ficaram presos. Não sendo uma estratégia infalível, foi considerada útil por ser de baixo custo computacional.

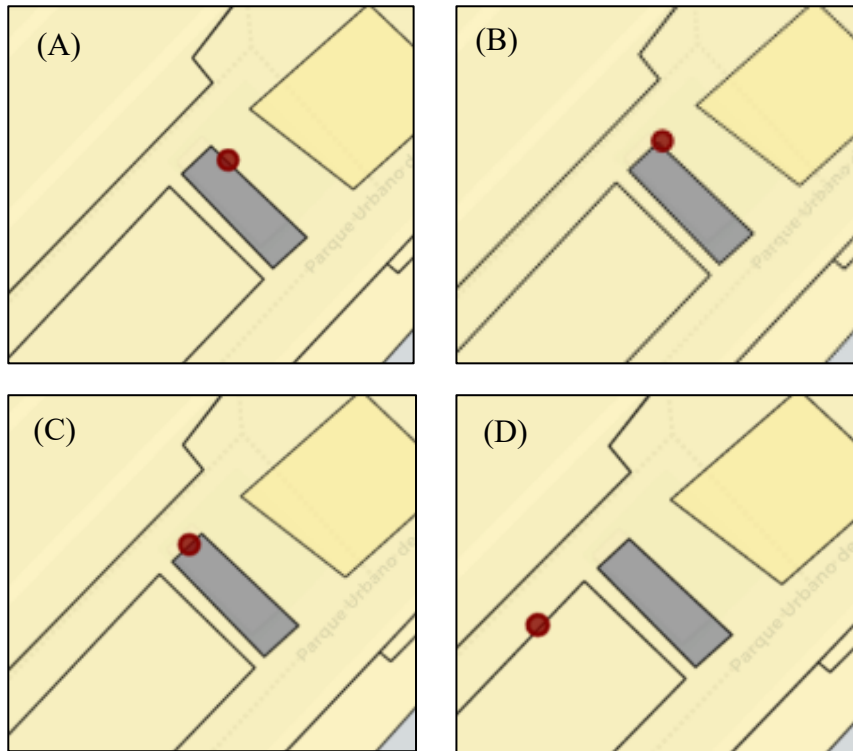


Figura 30 - Agente a desprender-se através da movimentação aleatória.

4.6 Geração e Formatação dos Resultados

Uma simulação produz um grande volume de dados que necessitam de ser tratados para se tornarem em informação útil para quem os usa. O fim do ciclo de execução do simulador trata-se disto mesmo, a recolha, processamento e exportação dos dados gerados.

A geração de resultados foi concebida para dar origem a dois artefactos: um ficheiro em formato CSV com o registo (*log*) da simulação que serve para análise estatística e um mapa interativo em HTML para fazer uma visualização dinâmica da simulação (Figura 31).

A1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	agent_id,initial_pos_x_3763,initial_pos_y_3763,chosen_target_node_id_3763,target_node_pos_x_3763,target_node_pos_y_3763,astar_path_found,astar_path_nodes_count,astar_path_length_meters,r																	
2	Agente_1,-67612.96570846987,-127718.24841963654,3904,-67902.9154508677,-127938.85470836368,109,402.28636328851445,True,373,373,-67901.3189826571,-127938.82232161812																	
3	Agente_2,-67599.73947227548,-127718.64919161405,3904,-67902.9154508677,-127938.85470836368,True,113,413.04372260139513,True,491,491,-67902.09933646568,-127938.29543440173																	
4	Agente_3,-67600.2310754936,-127716.94534190072,3904,-67902.9154508677,-127938.85470836368,True,113,414.28636328851445,True,398,398,-67902.33458770961,-127938.31984843296																	
5	Agente_4,-67621.4300872849,-127717.62447652765,3904,-67902.9154508677,-127938.85470836368,True,106,393.2863632885145,True,391,391,-67902.27099461004,-127937.82075371241																	
6	Agente_5,-67607.95687057056,-127713.49887746185,3904,-67902.9154508677,-127938.85470836368,True,110,406.5290039756337,False,600,600,-67630.10411232819,-127723.5787364352																	
7	Agente_6,-67619.69702944063,-127718.37674758483,3904,-67902.9154508677,-127938.85470836368,True,176,691.5138520739847,False,600,600,-67814.59997026926,-127820.51996417617																	
8	Agente_7,-67606.54265104467,-127716.55023848386,3904,-67902.9154508677,-127938.85470836368,True,111,408.28636328851445,True,451,451,-67901.2156431619,-127938.73029466976																	
9	Agente_8,-67614.16668640153,-127724.95982716237,3904,-67902.9154508677,-127938.85470836368,True,108,398.04372260139525,True,393,393,-67901.55328622181,-127938.30231945621																	
10	Agente_9,-67612.86138560885,-127714.34462355498,2475,-67869.9154508677,-127968.85470836368,True,91,370.6538956566618,False,600,600,-67851.99869336507,-127943.99404410202																	
11	Agente_10,-67610.23714840678,-127727.25115926303,2475,-67869.9154508677,-127968.85470836368,True,92,368.68333290818464,True,474,474,-67868.69817686755,-127967.63678212423																	
12	Agente_11,-67609.45612256283,-127717.79312207246,3904,-67902.9154508677,-127938.85470836368,True,110,405.28636328851445,True,489,489,-67901.83489284724,-127939.12910043311																	
13	Agente_12,-67603.69560822369,-127722.32777140006,2475,-67869.9154508677,-127968.85470836368,True,93,374.16861428242316,True,528,528,-67868.40690407224,-127967.93851666621																	
14	Agente_13,-67615.48836927406,-127708.68322925104,3904,-67902.9154508677,-127938.85470836368,True,108,403.01428534987235,True,395,395,-67902.11511310565,-127937.5737694649																	
15	Agente_14,-67604.90860141205,-127705.41657027396,2475,-67869.9154508677,-127968.85470836368,True,93,380.3818177180196,False,600,600,-67799.84627774477,-127892.78553524074																	
16	Agente_15,-67612.45855839434,-127726.7364760237,2475,-67869.9154508677,-127968.85470836368,True,92,367.4406922210653,True,546,546,-67868.37185220349,-127968.62847281282																	
17	Agente_16,-67597.73861883994,-127715.49746673816,3904,-67902.9154508677,-127938.85470836368,True,114,417.28636328851445,True,531,531,-67902.14541892352,-127938.0820128044																	
18	Agente_17,-67612.5450708211,-127713.23811887155,2475,-67869.9154508677,-127968.85470836368,True,91,370.6538956566618,False,600,600,-67839.7469962028,-127932.68625369875																	
19	Agente_18,-67602.8850334233,-127722.1711714445,2475,-67869.9154508677,-127968.85470836368,True,93,374.16861428242316,False,600,600,-67865.09263682297,-127965.35622379242																	
20	Agente_19,-67624.7545862027,-127716.17059409402,3904,-67902.9154508677,-127938.85470836368,True,105,390.2863632885145,True,378,378,-67901.637303131,-127939.1679252155																	
21	Agente_20,-67768.16724838142,-127909.53242316957,2475,-67869.9154508677,-127968.85470836368,True,38,132.12489168102786,True,220,220,-67868.54313422104,-127967.70736317452																	

Figura 31 - Ficheiro CSV com o registo de uma simulação.

No fim da simulação, os dados de cada agente (identificador, tempo total de evacuação, distância percorrida, estado final ("finished" ou "active") e o número de passos em que esteve bloqueado) são agregados numa estrutura de dados DataFrame da biblioteca Pandas. Esta estrutura é depois exportada para um ficheiro CSV, que servirá para fazer uma análise quantitativa dos resultados como, por exemplo, saber o tempo médio de evacuação ou a percentagem de agentes evacuados com sucesso.

Para a análise qualitativa, é essencial poder observar o comportamento da multidão. Para tal, é necessário processar o histórico de trajetórias de todos os agentes e formatá-lo de acordo com a estrutura exigida pelo plugin TimestampedGeoJson da biblioteca Folium. Fazemo-lo através da função `generate_features_from_simulation`. Este formato associa a cada ponto da trajetória de um agente um *timestamp*. Ao ser renderizado no mapa Folium, este plugin adiciona uma barra de tempo interativa que permite ao utilizador visualizar a posição de todos os agentes num instante específico, bem como reproduzir a simulação completa de forma animada (Figura 32). Desta forma é possível a identificação de padrões no fluxo da multidão, pontos de afunilamento e outros comportamentos coletivos complexos que seriam difíceis de perceber apenas a partir dos dados numéricos.

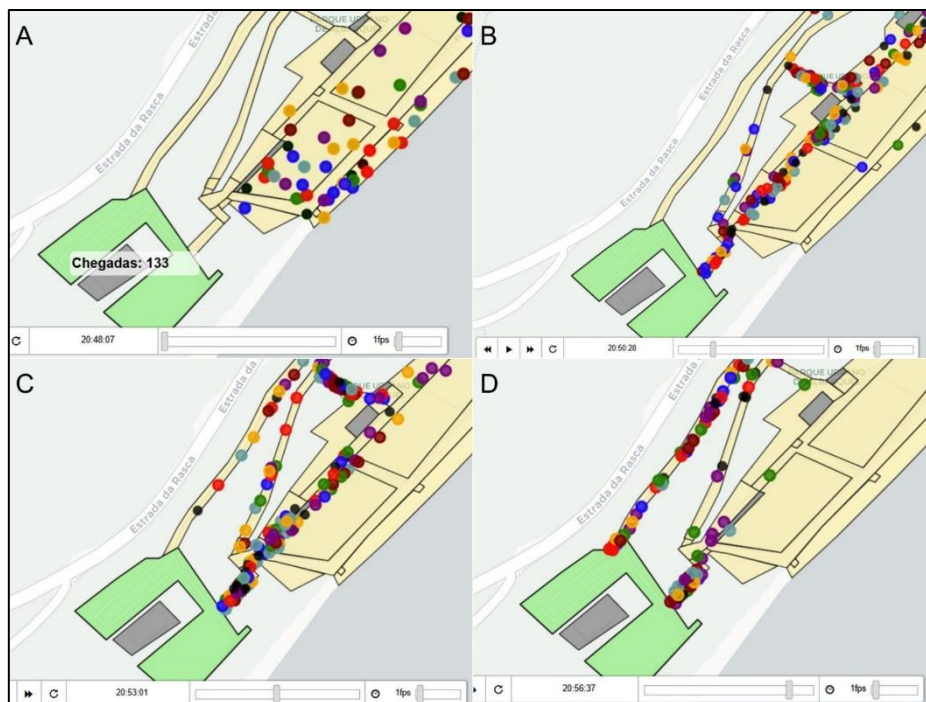


Figura 32 - Demonstração do deslocamento dos agentes com barra de tempo interativa. Hora de cada imagem no formato hh:mm:ss.
A – 20:48:07. B – 20:50:26. C – 20:53:01. D – 20:56:37.

CAPÍTULO 5 EXPERIÊNCIAS E RESULTADOS

O capítulo 4 mostrou a infraestrutura necessária para a execução das simulações, abordando desde a configuração do ambiente de desenvolvimento até a implementação da lógica de comportamento dos agentes. Agora, no capítulo 5, será realizada uma análise dos resultados obtidos a partir dos cenários propostos, com o objetivo de avaliar a eficácia do modelo.

Este capítulo irá abordar a configuração experimental utilizada, discutir as métricas de avaliação e analisar os resultados, terminando na resposta às questões de investigação propostas.

5.1 Configuração Experimental

Esta secção detalha os parâmetros e os cenários que constituem a base de toda a análise subsequente, de forma a assegurar a transparência e a replicabilidade do estudo.

5.1.1 Parâmetros do Modelo

Para que a simulação se ajuste à realidade que se pretende simular, é necessária a calibração de múltiplos parâmetros. A escolha dos valores utilizados nesta investigação é fundamentada por literatura científica, observação empírica e testes preliminares para garantir a plausibilidade do comportamento dos agentes.

Velocidade de Deslocação: A velocidade base dos agentes (`AGENT_SPEED_MPS`) foi definida com base no estudo da praia da Figueirinha [58]. Neste artigo é representada a velocidade de andamento de adultos através de uma distribuição normal, com velocidade de 1,12 m/s e com um desvio padrão de 0,17 m/s, com base em alguns estudos, como o “Estimation of pedestrian walking speeds on footways” [59].

Neste trabalho iremos então utilizar os mesmos valores referidos acima para proceder com as simulações, fazendo com que a velocidade de cada agente seja definida por uma distribuição normal com média em 1,12 e com desvio padrão de 0,17.

Redução de velocidade em escadas: Numa situação real as pessoas tendem a reduzir a sua velocidade nas escadas e para além disso, como o ambiente dos agentes é plano, torna-se necessário simular a redução de velocidade dos agentes nestas zonas. Para tal existe o parâmetro `SLOW_AREA_SPEED_FACTOR` cujo objetivo é multiplicar o valor deste parâmetro pela velocidade base calculada anteriormente. Por exemplo, se a velocidade base for 1 e este parâmetro for 0,2, a velocidade ao andar numa zona de escadas torna-se 0,2.

Para chegar a uma conclusão sobre a escolha do valor deste parâmetro, utilizou-se as conclusões do estudo de Fujiyama e Tyler [60]. Através da Figura 33 é possível perceber a tendência entre a inclinação das escadas e a velocidade das pessoas nessas escadas, quanto mais inclinado, menos velocidade. Nesta figura existem três gráficos, um com a velocidade horizontal, outro com a velocidade vertical e outro com a velocidade inclinada (junção da horizontal com vertical). Como o ambiente dos agentes é plano, apenas necessitamos dos dados sobre velocidade horizontal.

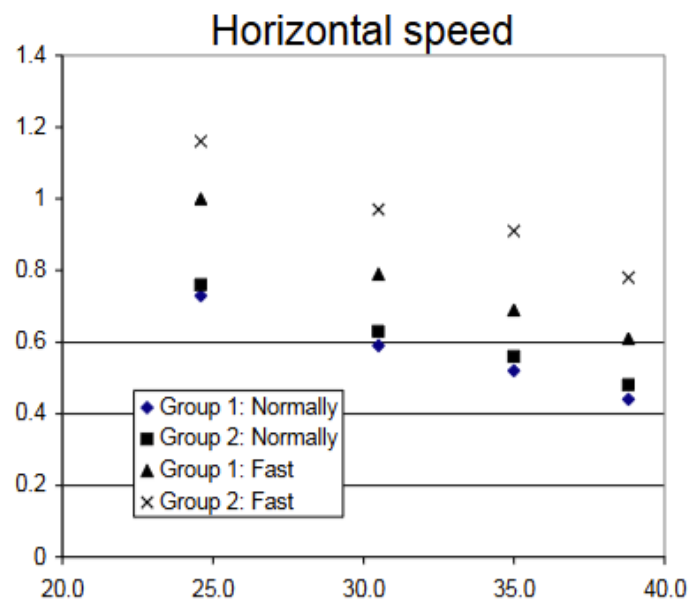


Figura 33 - Relação entre os graus de inclinação das escadas e as velocidades de subida [60].

Para perceber a inclinação das escadas necessitamos da diferença de altura e da distância horizontal entre o começo e o fim das escadas. Para descobrir a distância horizontal, recorreremos ao *website* “Google Maps” [61].



Figura 34 - Medição de distância horizontal das escadas [61].

A distância horizontal é aproximadamente 26 metros (Figura 33). Agora é necessário medir a distância vertical. Para tal iremos usar os mesmos dados apresentados anteriormente, fornecidos pela Direção Geral do Território que indicam a cota do terreno (MDT – Modelo Digital de Terreno) [62] e visualizados através do software QGIS.



Figura 35 - Medição da distância vertical das escadas [62].

Concluimos então que a distância vertical é aproximadamente oito metros (Figura 35). Assim, podemos calcular a inclinação do ângulo da seguinte forma:

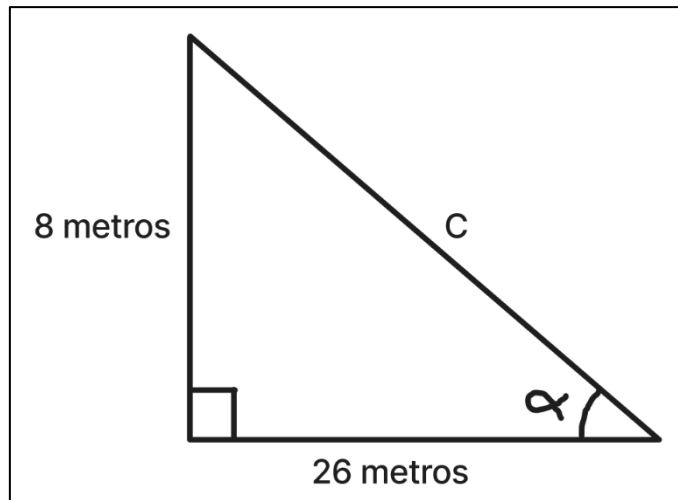


Figura 36 - Representação das escadas (lado C).

Através da análise da Figura 36 fica mais claro a forma de calcular o ângulo α . Como apenas temos conhecimento sobre o ângulo reto e dois lados do triângulo, podemos aplicar a regra dos senos que nos permite calcular o ângulo α .

Primeiramente teremos de usar o Teorema de Pitágoras para calcular o valor de C:

$$\begin{aligned}
 a^2 + b^2 &= c^2 \\
 26^2 + 8^2 &= c^2 \\
 c &\approx 27,2
 \end{aligned}
 \tag{1}$$

Sabendo o comprimento do lado C, podemos então aplicar a regra dos senos:

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}
 \tag{2}$$

Alterando pelos nossos valores:

$$\begin{aligned}
 \frac{8}{\sin \alpha} &= \frac{27,2}{\sin 90} \\
 \sin \alpha &= 0,294 \\
 \sin^{-1} 0,294 &= \alpha
 \end{aligned}$$

Resultado:

$$\alpha \approx 17,1^\circ$$

Com este ângulo podemos agora determinar a velocidade horizontal dos agentes nas escadas, através da Figura 33, retirada do estudo de Fujiyama e Tyler [60]. No entanto, o gráfico fornecido no estudo não abrange ângulos de escadas tão reduzidos, por isso vamos assumir que a velocidade dos agentes neste ângulo é igual à velocidade do ângulo mais baixo registado. Este ângulo é o de aproximadamente 24° e vamos escolher o grupo com velocidade normal (representado por quadrados na Figura 33), por ser o grupo que tem dados de pessoas com idades perto de 30 anos.

Percebemos então que a velocidade é aproximadamente 0,75 vezes a velocidade base, que será o valor utilizado nas simulações.

Parâmetros de Interação Social: Os valores para a distância a que os agentes começam a repelir-se em metros (`SOCIAL_RADIUS_METERS = 2,5m`) e para a força de repulsão (`REPULSION_STRENGTH = 4,0`) foram ajustados através de uma série de testes iniciais, nos quais se observou o comportamento dos agentes. Os valores finais foram seleccionados por serem aqueles que produziram as interações mais realistas, calibradas com base nas observações apresentadas no Capítulo 3 — por exemplo, o número de pessoas que conseguem permanecer lado a lado nas escadas — evitando tanto a sobreposição excessiva de agentes como comportamentos de repulsão exagerados.

Parâmetros de Decisão de Caminho: Para introduzir heterogeneidade comportamental, definiram-se duas probabilidades. A `PROB_FARTHEST_TARGET` foi ajustada em 0,5, para que metade dos agentes se dirija para uma saída e a outra metade para a outra saída. A `PROB_FARTHEST_THROUGH_ALL_DOORS` foi definida com um valor de 0,2, para fazer com que, dentro do conjunto de agentes que se deslocam para a saída mais afastada, alguns vão por um caminho específico de modo a passarem por pontos intermédios pré-definidos no QGIS.

Resolução do Grafo: O parâmetro `GRID_RESOLUTION_METERS` foi fixado em três metros. Esta escolha advém de um balanço entre a precisão do caminho e a performance computacional, garantindo que o grafo tem detalhe suficiente para representar os percursos e para impedir muitos agentes de ficarem presos em cantos e também evitando um número excessivo de nós que tornaria a simulação demasiado lenta.

Passo de Tempo: O valor de `TIME_STEP_SECONDS` foi definido como 1 segundo. Este valor faz com que cada *tick* represente um segundo da simulação, significando que o estado de todos os agentes é recalculado a cada segundo de tempo simulado.

Duração da Simulação: No Capítulo 1 vimos que o *tsunami* demorou cerca de 30 minutos a chegar à costa de Setúbal no sismo de 1755. Para retirar tempo de execução da simulação, mantendo um valor inferior aos 30 minutos, foi estabelecido um tempo máximo de simulação de 20 minutos (1200 passos). Este valor funciona como o critério temporal para o sucesso da evacuação, considerando-se que qualquer agente que não tenha alcançado uma zona segura neste período não conseguiu evacuar com sucesso.

5.1.2 Métricas de Avaliação

Para avaliar e comparar o desempenho dos diferentes cenários de simulação de forma objetiva, foram definidas as seguintes métricas de avaliação:

Tempo Total de Evacuação: Corresponde ao instante de tempo em que o último agente alcança uma zona segura. É a principal métrica para determinar a duração global do processo de evacuação.

Porcentagem de Agentes Evacuados: Mede a eficácia da evacuação, calculando a percentagem de agentes que conseguiram chegar a uma zona segura dentro do limite de tempo de 20 minutos.

Análise de Congestionamento: Esta análise é realizada de forma qualitativa, através da visualização dos mapas interativos para identificar os pontos de estrangulamento e observar padrões.

Mapa de calor dos agentes: Consiste num mapa de calor que mostra o sucesso de evacuação de cada agente. O mapa de calor é parte de uma camada no mapa de visualização e é representado através de círculos verdes e vermelhos. Caso o agente evacue com sucesso, aparece um círculo verde na sua posição inicial. Caso o agente não consiga evacuar com sucesso, é representado um círculo vermelho na sua posição inicial. Desta forma é fácil de visualizar que zonas têm uma mortalidade mais elevada (Figura 37).



Figura 37 - Exemplo do mapa de calor (layer do mapa interativo).

5.2 Análise dos Cenários de Simulação

Nesta secção, são apresentados e discutidos os resultados obtidos para cada um dos cenários de simulação definidos. A análise é feita combinando uma abordagem quantitativa, baseada nas métricas de avaliação, com uma abordagem qualitativa, através da visualização do mapa interativo. O objetivo é avaliar o desempenho da evacuação em cada cenário e perceber o impacto das alterações propostas, realizando três execuções da simulação para cada cenário. Existem três cenários distintos, o cenário base com 300 agentes, o cenário com 500 agentes e outro com 700 agentes.

5.2.1 Visualização da simulação

A visualização das simulações é suportada por um mapa interativo (Figura 38), desenhado para facilitar a inspeção visual dos comportamentos emergentes. A ferramenta incorpora mecanismos de controlo temporal, permitindo acelerar a visualização para uma análise macroscópica ou reduzir a velocidade para examinar interações locais entre agentes.

Através de um sistema de camadas seleccionáveis, é possível sobrepor diferentes níveis de informação sobre o mapa base. O utilizador pode optar por uma visualização com o grafo de navegação, as delimitações físicas do parque, contador de evacuação e o mapa de calor para identificação de zonas com uma mortalidade mais elevada.

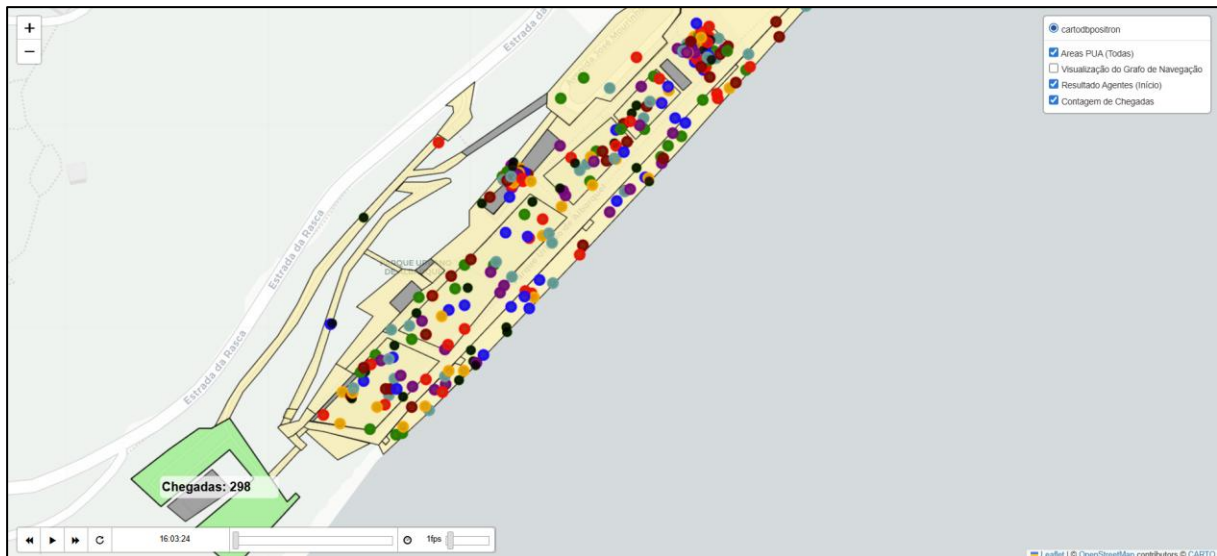


Figura 38 - Mapa interativo para visualização da simulação.

5.2.2 Análise do Cenário Base

O cenário base serve como ponto de referência para todas as comparações seguintes. As suas condições representam uma aproximação a um dia de afluência normal, com base na observação empírica. O dia observado foi um sábado, dia 07/06/2025, entre as 15h30 e as 16h30. Neste dia estava muito calor e sol e foi observado o seguinte número de pessoas:

1. Praia ~ 50 pessoas (19%)
2. Pequeno evento ~ 50 pessoas (19%)
3. Campo basquete ~ 15 pessoas (5,6%)
4. Relva 1 ~ 20 pessoas (7,5%)
5. Relva 2 ~ 25 pessoas (9,4%)
6. Relva 3 (parque crianças) ~ 25 pessoas (9,4%)
7. Evento entre edifícios ~ 50 pessoas (19%)
8. Pessoas a andar ~ 30 pessoas (11%)

O total de pessoas ascende a 265 que iremos arredondar para 300 no cenário base (Figura 39). Este cenário simula a evacuação de 300 agentes distribuídos pelo parque de acordo com as densidades observadas. A disposição espacial corresponde à realidade do parque atual, com apenas uma zona segura designada (o restaurante) e os caminhos atuais sem quaisquer alterações. Os parâmetros do modelo são os que foram detalhados na secção 5.1.1.

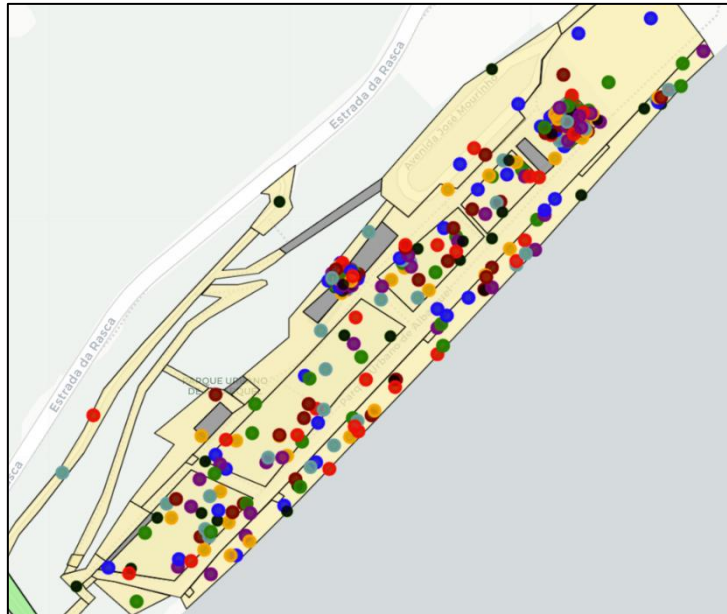


Figura 39 - Exemplo da distribuição dos agentes no instante inicial. Cenário base com 300 pessoas.

A Figura 27 mostra as três simulações no seu pico de congestionamento, feitas para o cenário base com 300 pessoas. Aí é possível perceber o aparecimento de fluxos de evacuação em direção à zona segura. Foram também identificados pontos críticos de afunilamento, nomeadamente na base das escadas que sobem diretamente para a zona segura e nas escadas que sobem para a rampa. Nestes pontos a densidade dos agentes aumenta significativamente, o que causa uma diminuição acentuada na velocidade e progresso dos agentes.

Um padrão interessante de observar nestas simulações foi o aparecimento de um caminho ótimo seguido pelos agentes que vêm da praia. Estes agentes “sobem” as escadas da praia e seguem um caminho que “corta” a relva, como mostra a Figura 40 (caminhos destacados a azul).

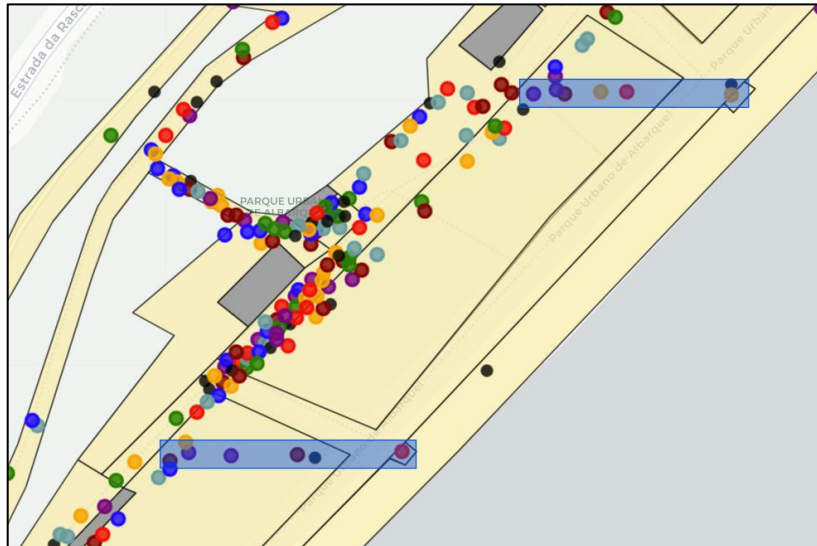


Figura 40 - Caminhos ótimos surgem ao sair da praia.

Estes caminhos ótimos deverão ficar mais aparentes com o aumento do número de agentes na simulação, apresentado mais à frente.

A Tabela 2 refere-se aos dados quantitativos retirados das três simulações feitas com o cenário base e com 300 agentes no total. O número de agentes não evacuados exclui os agentes presos quando todos os outros já terminaram a evacuação. O tempo até segurança total é o tempo que demora até todos os agentes ficarem acima da cota de inundação, ou seja, refere-se ao tempo que os agentes demoram a todos subirem as escadas.

Tabela 2 - Dados de simulações de cenário base com 300 agentes.

	Simulação 1	Simulação 2	Simulação 3	Média	Desvio Padrão
# agentes evacuados	298	295	297	296,66	1,24
# agentes presos	2	5	3	3,33	1,25
# agentes não evacuados	0	0	0	0	0,00
% agentes evacuados	99,33%	98,33%	99,00%	98,88%	0,41%
Tempo até segurança total (ignorando agentes presos)	10m20s	9m36s	10m18s	10m4s	±20s

Tempo total de evacuação (ignorando agentes presos)	18m36s	15m54s	14m40s	16m23s	±1m38s
-----------------------------------------------------	--------	--------	--------	--------	--------

Os resultados mostram que, para uma população de 300 pessoas, a evacuação foi concluída com sucesso dentro do limite imposto de 20 minutos. Apesar de todos os agentes conseguirem evacuar no tempo recomendável, conseguimos perceber duas grandes zonas de afunilamento que podem levar a lesões em caso de queda de pessoas por causa de empurrões. A zona junto ao café e a zona imediatamente antes das escadas a oeste (Figura 41).



Figura 41 - Zona junto ao café (imagem da esquerda) e zona junto às escadas a oeste (imagem da direita)

Na primeira simulação o tempo total de evacuação foi bastante mais elevado que nas outras porque dois agentes ficaram presos devido aos motivos explicados no subcapítulo 4.5, no entanto a movimentação aleatória conseguiu libertar os agentes antes que os outros concluíssem a evacuação, permitindo-lhes alcançar a área segura e contando para a estatística de agentes evacuados.

Vamos então estudar o impacto que o aumento do número de agentes tem na evacuação, fazendo mais simulações com 500 pessoas e 700 pessoas.

5.2.3 Estudo da Escalabilidade da Evacuação

Para compreender como o sistema reage a um aumento de agentes, o cenário base foi replicado com populações de 500 e 700 agentes. As condições de simulação são idênticas às do cenário base, variando apenas o número total de agentes.

Nas simulações com 500 pessoas vemos ainda mais facilmente a aparição de um caminho ótimo utilizado pelas pessoas que vêm da praia (Figura 42).

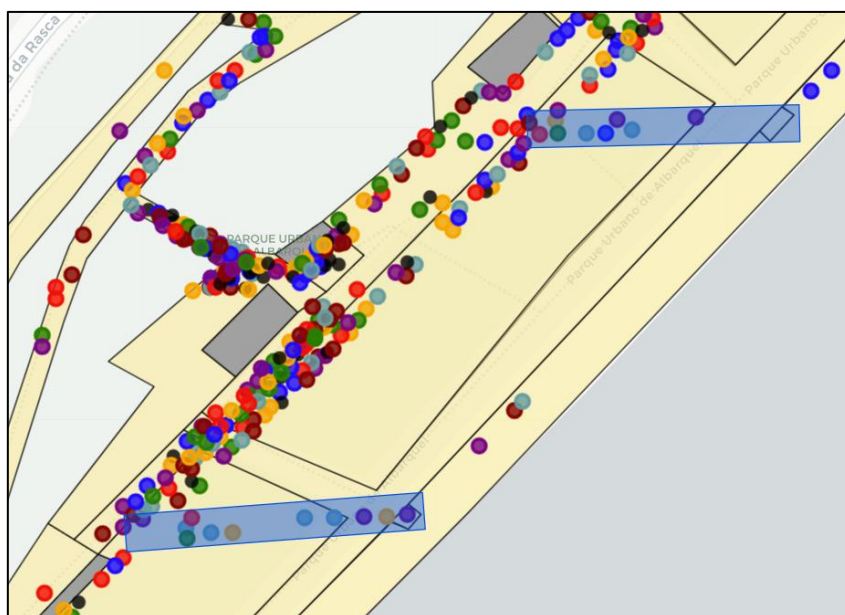


Figura 42 - Caminhos ótimos mais acentuados ao simular com 500 agentes.

Tal como foi feito no caso anterior, as seguintes tabelas referem-se aos dados quantitativos retirados das três simulações feitas com o cenário base e com 500 agentes no total, assim como com 700 agentes no total. O número de agentes não evacuados exclui os agentes presos.

Tabela 3 - Dados de simulações de cenário base com 500 agentes.

	Simulação 1	Simulação 2	Simulação 3	Média	Desvio Padrão
# agentes evacuados	494	494	494	494	0,00
# agentes presos	6	6	6	6	0,00
# agentes não evacuados	0	0	0	0	0,00
% agentes evacuados	98,80%	98,80%	98,80%	98,80%	0,00%
Tempo até segurança total (ignorando agentes presos)	14m6s	14m29s	15m34s	14m43s	±33s

Tempo total de evacuação (ignorando agentes presos)	18m31s	19m47s	18m00s	18m46s	±47s
-----------------------------------------------------	--------	--------	--------	--------	------

Nestas simulações, através da Tabela 3, é possível observar que, apesar do tempo total de evacuação ter aumentado cerca de dois minutos, continua menor que 20 minutos.

Vemos também que, tal como nas simulações de 300 pessoas, há uma que demora significativamente mais tempo que as outras, a segunda simulação com 19m47s. Analogamente ao observado na Tabela 2, nesta simulação um dos agentes ficou temporariamente retido, mas conseguiu libertar-se a tempo de alcançar a área de segurança. No entanto, chegou mais tarde do que os restantes, o que contribuiu para o aumento do tempo total de evacuação por já não ser contabilizado como um agente preso.

Tabela 4 - Dados de simulações de cenário base com 700 agentes.

	Simulação 1	Simulação 2	Simulação 3	Média	Desvio Padrão
# agentes evacuados	639	623	639	633,66	7,54
# agentes presos	12	6	10	9,33	2,49
# agentes não evacuados	49	71	51	57	9,93
% agentes evacuados	91,28%	89,00%	91,28%	90,52%	1,04%
Tempo até segurança total (ignorando agentes presos)	17m48s	19m30s	18m58s	18m45s	±42s
Tempo total de evacuação (ignorando agentes presos)	20m00s	20m00s	20m00s	20m00s	0s

Na Tabela 4 vemos que, com 700 agentes, há uma grande quantidade de agentes que não evacuaram totalmente em 20 minutos. Apesar de todos os agentes já se encontrarem em sítios seguros (com a altitude recomendada) em média aos 18 minutos e 45 segundos de simulação. Vemos que, aos 20 minutos, em média 57 agentes ainda não tinham chegado à zona segura de evacuação e por isso a percentagem de agentes evacuados decresce drasticamente. Isto pode ser perigoso por diversos motivos, um deles é a possibilidade de derrocadas que pode atingir pessoas fora da zona segura de evacuação.



Figura 43 - Mapa de calor da primeira simulação com 700 agentes.

Verificou-se que a evacuação é bem-sucedida mesmo em cenários extremos, com a presença de 700 pessoas no Parque. Embora os dados indiquem que nem todos os agentes concluíram totalmente a evacuação, como é observável pelo mapa de calor da Figura 43, constatou-se que estes se encontravam já em zonas seguras, não afetadas pelo tsunami. Este estudo considera uma evacuação controlada e organizada, em que os agentes se deslocam com velocidade constante, exceto nas escadas, sem paragens nem situações de caos ou pânico. Por outro lado, as simulações não consideraram crianças, idosos ou pessoas de mobilidade reduzida, que podem eventualmente atrasar toda a evacuação. Assim, recomenda-se que os exercícios de evacuação e simulacros no Parque devam continuar para as pessoas treinarem o caminho de evacuação.

5.2.4 Discussão do Realismo dos Resultados

A análise dos tempos obtidos nas diversas simulações exige uma reflexão sobre a sua aplicabilidade prática e realismo face ao contexto histórico e geográfico de Setúbal. O limite temporal de 20 minutos (1200 passos) definido para as experiências revela-se um critério de sucesso conservador mas adequado, quando comparado com os dados históricos de tsunamis na região. No sismo de 1755, estima-se que as primeiras ondas tenham atingido a costa de Setúbal cerca de 30 minutos após o abalo, enquanto no evento de 1969 esse intervalo foi de aproximadamente 40 minutos.

No cenário base de 300 agentes, o tempo médio de evacuação de 16m23s oferece uma margem de segurança de aproximadamente 13 minutos em relação ao cenário de 1755. Mesmo no cenário de sobrelotação com 700 agentes, embora a conclusão total da evacuação para a zona segura tenha ultrapassado os 20 minutos, a métrica de "Tempo até segurança total" fixou-se numa média de 18m45s. Este dado é crucial, pois sugere que, sob os parâmetros do modelo, a maioria da população conseguiria evitar o impacto direto da massa de água dentro da janela temporal histórica de 30 minutos.

O realismo destes resultados é sustentado pela calibração das velocidades de deslocação. A utilização de uma velocidade média de 1,12 m/s para caminhada horizontal, fundamentada em estudos de fluxos pedestres, aliada à penalização de 0,75 para o movimento em escadas, calculada com base na inclinação real do terreno ($\sim 17,1^\circ$) e na literatura, permite que o modelo replique com fidelidade o esforço físico e o abrandamento inerentes à topografia do Parque Urbano de Albarquel.

Contudo, é imperativo notar que os tempos apresentados pressupõem uma evacuação controlada e organizada. A ausência de variáveis como o pânico generalizado, obstáculos imprevistos (derrocadas ou queda de rochas) e a presença de indivíduos com mobilidade reduzida significa que os valores obtidos representam um cenário de eficiência otimizada. Numa situação real, a degradação do fluxo em pontos de afunilamento críticos, como a zona do café e as escadas a Oeste, poderia ser mais acentuada, aproximando os tempos de evacuação do limite crítico de chegada da primeira onda.

5.3 Respostas às Questões de Investigação

Relembrando as questões de investigação apresentadas no Capítulo 1, nesta secção apresentam-se as respostas obtidas com base nos resultados e análises realizadas.

O modelo desenvolvido simula a evacuação de forma realista?

O modelo simula a evacuação através da aplicação de três critérios fundamentais: a integridade espacial, garantindo que os agentes respeitam os limites físicos do ambiente e a volumetria dos restantes indivíduos sem que ocorram sobreposições; a adaptação ao meio, visível nas variações realistas de velocidade em locais como escadas; e a consistência de fluxos, que permite replicar comportamentos coletivos como o efeito de afunilamento em congestionamentos e a transmissão de forças físicas entre agentes através de empurrões.

Qual o tempo médio de evacuação total do Parque Urbano de Albarquel?

Vimos que o tempo médio de evacuação total do PUA é dependente do número de pessoas presentes, por isso esta resposta deve ser dividida em várias. Para o caso de haver 300 pessoas, o tempo médio de evacuação foi de 16m23s. Com 500 pessoas, o tempo aumentou para 18m46s. No caso extremo de 700 pessoas, o tempo de evacuação ultrapassou os 20 minutos, apesar de todas já estarem numa altitude considerada segura.

Haverá locais do parque em que a possibilidade de fuga das pessoas que lá se encontrem seja menor?

Não foi encontrada uma relação entre a localização inicial do agente e a sua probabilidade de evacuação bem-sucedida. Vemos nas simulações, através do mapa de calor da Figura 43, que os agentes que não foram bem-sucedidos a evacuar estão espalhados por todo o parque. Mesmo aqueles que começam perto de uma zona segura podem acabar por não conseguir evacuar devido às interações com outros agentes ou à escolha do caminho a percorrer.

CAPÍTULO 6 CONCLUSÃO E TRABALHO FUTURO

6.1 Conclusão

O trabalho nesta dissertação produziu um modelo que permitiu simular a evacuação controlada do Parque Urbano de Albarquel, em Setúbal. Através deste modelo foi possível simular a deslocação e interação de um número elevado de pessoas e perceber, através da visualização da simulação, como as pessoas se comportam e quais as zonas de maior congestionamento que atrasam a evacuação e a tornam mais perigosa.

A implementação deste modelo utiliza um sistema de informação geográfica para a criação do ambiente, que permite uma flexibilidade elevada no caso de o ambiente ser alterado, sendo simples adicionar ou remover caminhos novos, edifícios ou escadas. No deslocamento de agentes foram implementadas algoritmos e mecanismos representativos de interações entre agentes, já testados e empiricamente reconhecidos na literatura, cuja aplicação ao caso concreto em estudo mostrou ser capaz de simular interações entre pessoas em multidões de forma realista.

Através dos dados recolhidos e das animações geradas pelas diversas simulações realizadas, concluiu-se que é possível realizar-se uma evacuação total do parque antes do tempo limite estabelecido, identificando-se, contudo, algumas zonas de afunilamento, suscetíveis de resultar em atrasos e, possivelmente, interações de maior risco entre pessoas, como por exemplo empurrões, passíveis de provocarem, por exemplo, quedas perigosas ou atropelamentos da multidão. Estes resultados oferecem à Proteção Civil a capacidade de antecipar riscos nos planos vigentes, permitindo priorizar intervenções de reforço estrutural em zonas críticas, nomeadamente nas escadas junto ao café e a Oeste, onde a densidade de agentes detetada sugere a necessidade de alargamento ou de criação de barreiras para controlo de fluxo.

Podemos também concluir que os caminhos ótimos observados nas simulações sugerem onde os caminhos pavimentados deveriam estar localizados no parque, de forma a permitir uma deslocação mais célere e intuitiva das pessoas em caso de necessidade de evacuação. Para a Proteção Civil, esta informação é fundamental para o reposicionamento da sinalética de evacuação, assegurando que as indicações visuais estão colocadas nos fluxos naturais de fuga e em locais de visibilidade desimpedida.

Finalmente, o modelo constitui um suporte essencial para a organização de simulacros e exercícios de campo, permitindo definir cenários baseados em dados científicos e prever novas zonas de congestionamento sempre que ocorrerem alterações na infraestrutura ou na disposição dos edifícios do parque.

6.2 Limitações e Trabalho Futuro

Como sugestão para trabalhos futuros, seria relevante abordar a resolução de alguns dos problemas atuais, como, por exemplo, o processo manual de remoção das ligações entre nós. Atualmente, esse procedimento é feito de forma explícita no código, onde o nome das *shapes* é especificado para desconectar os nós de outras *shapes*. Tornar esse processo mais dinâmico poderia melhorar a flexibilidade e a escalabilidade da simulação.

Outro aspeto interessante seria melhorar o comportamento dos agentes, permitindo a simulação de grupos de agentes e de diferentes perfis de indivíduos, como crianças, idosos ou pessoas com deficiência. Atualmente, a simulação apenas varia a velocidade dos agentes com base numa distribuição normal, mas a introdução de perfis específicos alteraria de forma significativa a velocidade e o comportamento de cada agente. Investigar como esses diferentes perfis e interações em grupo influenciam o desempenho e os resultados das simulações poderia trazer informação e conclusões valiosas sobre o comportamento coletivo e a eficiência dos sistemas de evacuação.

Referências Bibliográficas

- [1] “Parques e Jardins,” *Município De Setúbal*. <https://www.mun-setubal.pt/parques-e-jardins/> (accessed Jan. 25, 2025).
- [2] A. Santos, N. Fonseca, M. Queirós, J. Zêzere, and J. Bucho, “Implementation of tsunami evacuation maps at Setubal Municipality, Portugal,” *Geosciences*, vol. 7, no. 4, p. 116, Nov. 2017, doi: 10.3390/geosciences7040116.
- [3] Município de Setúbal, ““Memória do Sismo de 1969 – 28 de fevereiro, a noite em que Portugal tremeu,”” YouTube. Feb. 28, 2019. [Online]. Available: <https://www.youtube.com/watch?v=xU3cSCuZiTY>
- [4] S. Ângela, Q. Margarida, and C. Luís, “Fire and seismic risk perception at Lisbon University – Faculty of Letters,” *Territorium*, no. 24, pp. 15–27, Mar. 2017, doi: 10.14195/1647-7723_24_2.
- [5] “Overview of the MESA library — Mesa .1 documentation.” <https://mesa.readthedocs.io/latest/overview.html> (accessed Nov. 12, 2024).
- [6] A. R. Hevner, “Design Science in Information Systems Research,” *MIS Quarterly*, vol. 28–28, no. 1, pp. 75–105, Mar. 2004.
- [7] E. Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems,” *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl_3, pp. 7280–7287, May 2002, doi: 10.1073/pnas.082080899.
- [8] E. M. Gallagher and J. J. Bryson, “Agent-Based modelling,” in *Springer eBooks*, 2017, pp. 1–9. doi: 10.1007/978-3-319-47829-6_224-1.
- [9] L.-A. Cotfas, C. Delcea, L.-D. Iancu, and C. Ioanăș, “Large event halls evacuation using an Agent-Based Modeling approach,” journal-article, May 2022. doi: 10.1109/ACCESS.2022.3172285.
- [10] E. Encarnação Cândido, “Modelação Baseada em Agentes da Evacuação da Praia da Figueirinha,” Dissertação de mestrado, Iscte - Instituto Universitário de Lisboa. [Online]. Available: <http://hdl.handle.net/10071/27712>
- [11] Y. Liu and T. Kaneda, “Using agent-based simulation for public space design based on the Shanghai Bund waterfront crowd disaster,” *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, vol. 34, no. 2, pp. 176–190, Jan. 2020, doi: 10.1017/s0890060420000049.
- [12] S. Yang, T. Li, X. Gong, B. Peng, and J. Hu, “A review on crowd simulation and modeling,” *Graphical Models*, vol. 111, p. 101081, Jul. 2020, doi: 10.1016/j.gmod.2020.101081.
- [13] M. Xiong, S. Tang, and D. Zhao, “A hybrid model for simulating crowd evacuation,” *New Generation Computing*, vol. 31, no. 3, pp. 211–235, Jul. 2013, doi: 10.1007/s00354-013-0304-2.
- [14] W. Van Toll and J. Pettré, “Algorithms for Microscopic Crowd Simulation: Advancements in the 2010s,” *Computer Graphics Forum*, vol. 40, no. 2, pp. 731–754, May 2021, doi: 10.1111/cgf.142664.
- [15] M. A. Aljamal, H. A. Rakha, J. Du, and I. El-Shawarby, “Comparison of microscopic and mesoscopic traffic modeling tools for evacuation analysis,” *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2321–2326, Nov. 2018, doi: 10.1109/itsc.2018.8569290.
- [16] E. Ronchi and R. Lovreglio, “EXPOSED: An occupant exposure model for confined spaces to retrofit crowd models during a pandemic,” *Safety Science*, vol. 130, p. 104834, Jul. 2020, doi: 10.1016/j.ssci.2020.104834.

- [17] Y. Yuan, B. Goñi-Ros, H. H. Bui, W. Daamen, H. L. Vu, and S. P. Hoogendoorn, “Macroscopic pedestrian flow simulation using Smoothed Particle Hydrodynamics (SPH),” *Transportation Research Part C Emerging Technologies*, vol. 111, pp. 334–351, Jan. 2020, doi: 10.1016/j.trc.2019.12.017.
- [18] N. Bellomo, J. Liao, A. Quaini, L. Russo, and C. Siettos, “Human behavioral crowds review, critical analysis and research perspectives,” *Mathematical Models and Methods in Applied Sciences*, vol. 33, no. 08, pp. 1611–1659, Apr. 2023, doi: 10.1142/s0218202523500379.
- [19] D. Foad, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, “A Systematic Literature review of A* pathfinding,” *Procedia Computer Science*, vol. 179, pp. 507–514, Jan. 2021, doi: 10.1016/j.procs.2021.01.034.
- [20] Z. Liu, H. Liu, Z. Lu, and Q. Zeng, “A dynamic fusion pathfinding algorithm using Delaunay triangulation and improved A-Star for mobile robots,” *IEEE Access*, vol. 9, pp. 20602–20621, Jan. 2021, doi: 10.1109/access.2021.3055231.
- [21] T. Standley and R. Korf, “Complete algorithms for cooperative pathfinding problems,” *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 668–673, Jul. 2011, doi: 10.5591/978-1-57735-516-8/ijcai11-118.
- [22] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959, doi: 10.1007/bf01386390.
- [23] A. Schrijver, “On the history of the shortest path problem,” in *EMS Press eBooks*, 2012, pp. 155–167. doi: 10.4171/dms/6/19.
- [24] A. R. Soltani, H. Tawfik, J. Y. Goulermas, and T. Fernando, “Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms,” *Advanced Engineering Informatics*, vol. 16, no. 4, pp. 291–303, Oct. 2002, doi: 10.1016/s1474-0346(03)00018-1.
- [25] A. Ammar, H. Bennaceur, I. Châari, A. Koubâa, and M. Alajlan, “Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments,” *Soft Computing*, vol. 20, no. 10, pp. 4149–4171, Jul. 2015, doi: 10.1007/s00500-015-1750-1.
- [26] Wikipedia contributors, “Dijkstra’s algorithm,” *Wikipedia*, Jul. 21, 2025. https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm#/media/File:Dijkstras_progress_animation.gif (accessed Aug. 05, 2025).
- [27] X. Wang, Z. Liu, and J. Liu, “Mobile robot path planning based on an improved A* algorithm,” *International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2021)*, May 2023, doi: 10.1117/12.2674526.
- [28] A. Felner et al., “Search-Based Optimal Solvers for the Multi-Agent Pathfinding Problem: Summary and Challenges,” *Proceedings of the International Symposium on Combinatorial Search*, vol. 8, no. 1, pp. 29–37, Sep. 2021, doi: 10.1609/socs.v8i1.18423.
- [29] Wikipedia contributors, “A* search algorithm,” *Wikipedia*, Jun. 20, 2025. https://en.wikipedia.org/wiki/A*_search_algorithm#/media/File:Astar_progress_animation.gif (accessed Aug. 05, 2025).
- [30] J. Pak, J. Kim, Y. Park, and H. I. Son, “Field evaluation of Path-Planning algorithms for autonomous mobile robot in smart farms,” *IEEE Access*, vol. 10, pp. 60253–60266, Jan. 2022, doi: 10.1109/access.2022.3181131.

- [31] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo, “Walking Ahead: the headed Social Force model,” *PLoS ONE*, vol. 12, no. 1, p. e0169734, Jan. 2017, doi: 10.1371/journal.pone.0169734.
- [32] B. Liu, H. Liu, H. Zhang, and X. Qin, “A social force evacuation model driven by video data,” *Simulation Modelling Practice and Theory*, vol. 84, pp. 190–203, Feb. 2018, doi: 10.1016/j.simpat.2018.02.007.
- [33] X. Chen, M. Treiber, V. Kanagaraj, and H. Li, “Social force models for pedestrian traffic – state of the art,” *Transport Reviews*, vol. 38, no. 5, pp. 625–653, Nov. 2017, doi: 10.1080/01441647.2017.1396265.
- [34] Y. Jiang, B. Chen, X. Li, and Z. Ding, “Dynamic navigation field in the social force model for pedestrian evacuation,” *Applied Mathematical Modelling*, vol. 80, pp. 815–826, Oct. 2019, doi: 10.1016/j.apm.2019.10.016.
- [35] N. Taherifar, H. Hamedmoghadam, S. Sree, and M. Saberi, “A macroscopic approach for calibration and validation of a modified social force model for bidirectional pedestrian streams,” *Transportmetrica a Transport Science*, vol. 15, no. 2, pp. 1637–1661, Jun. 2019, doi: 10.1080/23249935.2019.1636156.
- [36] G. Vizzari, L. Manenti, K. Ohtsuka, and K. Shimura, “An Agent-Based pedestrian and group dynamics model applied to experimental and Real-World scenarios,” *Journal of Intelligent Transportation Systems*, vol. 19, no. 1, pp. 32–45, Oct. 2013, doi: 10.1080/15472450.2013.856718.
- [37] C. Wang, S. Zhang, T. Ma, Y. Xiao, M. Z. Chen, and L. Wang, “Swarm intelligence: A survey of model classification and applications,” *Chinese Journal of Aeronautics*, Mar. 2024, doi: 10.1016/j.cja.2024.03.019.
- [38] N. Maruyama, D. Saito, Y. Hashimoto, and T. Ikegami, “Dynamic organization of flocking behaviors in a large-scale boids model,” *Journal of Computational Social Science*, vol. 2, no. 1, pp. 77–84, Jan. 2019, doi: 10.1007/s42001-019-00037-9.
- [39] R. P. Ramos, S. M. Oliveira, S. M. Vieira, and A. L. Christensen, “Evolving flocking in embodied agents based on local and global application of Reynolds’ rules,” *PLoS ONE*, vol. 14, no. 10, p. e0224376, Oct. 2019, doi: 10.1371/journal.pone.0224376.
- [40] Z. Dong, Q. Wu, and L. Chen, “Reinforcement Learning-Based Formation pinning and shape transformation for swarms,” *Drones*, vol. 7, no. 11, p. 673, Nov. 2023, doi: 10.3390/drones7110673.
- [41] J. Tang, G. Leu, and H. A. Abbass, “Networking the Boids is more robust against adversarial learning,” *IEEE Transactions on Network Science and Engineering*, vol. 5, no. 2, pp. 141–155, Aug. 2017, doi: 10.1109/tNSE.2017.2745108.
- [42] N. T. Ouellette and D. M. Gordon, “Goals and limitations of modeling collective behavior in biological systems,” *Frontiers in Physics*, vol. 9, Jun. 2021, doi: 10.3389/fphy.2021.687823.
- [43] C. Kniewel, A. Pejic, L. Krüger, C. Ziegler, and J. Adamy, “BOIDs flocking algorithm for situation assessment of driver Assistance Systems,” *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 71–82, Jan. 2023, doi: 10.1109/ojits.2023.3236985.
- [44] R.-Y. Guo and H.-J. Huang, “A modified floor field cellular automata model for pedestrian evacuation simulation,” *Journal of Physics a Mathematical and Theoretical*, vol. 41, no. 38, p. 385104, Aug. 2008, doi: 10.1088/1751-8113/41/38/385104.

- [45] N. Pelechano and A. Malkawi, "Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches," *Automation in Construction*, vol. 17, no. 4, pp. 377–385, Aug. 2007, doi: 10.1016/j.autcon.2007.06.005.
- [46] G. Zhong, G. Zhai, and W. Chen, "Evacuation simulation of multi-story buildings during earthquakes based on improved cellular automata model," *Journal of Asian Architecture and Building Engineering*, vol. 22, no. 2, pp. 1007–1027, Apr. 2022, doi: 10.1080/13467581.2022.2070491.
- [47] M. Chen, K. Wu, H. Zhang, D. Han, and M. Guo, "A ship evacuation model considering the interaction between pedestrians based on cellular automata," *Ocean Engineering*, vol. 281, p. 114644, May 2023, doi: 10.1016/j.oceaneng.2023.114644.
- [48] B. Zou, C. Lu, and Y. Li, "Simulation of a hospital evacuation including wheelchairs based on modified cellular automata," *Simulation Modelling Practice and Theory*, vol. 99, p. 102018, Nov. 2019, doi: 10.1016/j.simpat.2019.102018.
- [49] D. Chang, L. Cui, and Z. Huang, "A Cellular-Automaton Agent-Hybrid model for emergency evacuation of people in public places," *IEEE Access*, vol. 8, pp. 79541–79551, Jan. 2020, doi: 10.1109/access.2020.2986012.
- [50] Y. Ji, W. Wang, M. Zheng, and S. Chen, "Real Time Building Evacuation Modeling with an Improved Cellular Automata Method and Corresponding IoT System Implementation," *Buildings*, vol. 12, no. 6, p. 718, May 2022, doi: 10.3390/buildings12060718.
- [51] T. Liu, X. Yang, Q. Wang, M. Zhou, and S. Xia, "A Fuzzy-Theory-Based cellular automata model for pedestrian evacuation from a Multiple-Exit room," *IEEE Access*, vol. 8, pp. 106334–106345, Jan. 2020, doi: 10.1109/access.2020.3000606.
- [52] "Spatial without Compromise · QGIS Web Site." <https://qgis.org/> (accessed Aug. 08, 2025).
- [53] "QGIS Maps · QGIS Web Site." <https://qgis.org/project/overview/screenshots/> (accessed Aug. 07, 2025).
- [54] "Documentation - GeoPandas 1.1.1 documentation." <https://geopandas.org/en/stable/docs.html> (accessed Aug. 09, 2025).
- [55] "Shapely — Shapely 2.1.1 documentation." <https://shapely.readthedocs.io/en/stable/> (accessed Aug. 09, 2025).
- [56] A. A. Hagberg, D. A. Schult, and P. J. Swart, 'Exploring Network Structure, Dynamics, and Function using NetworkX', in *Proceedings of the 7th Python in Science Conference*, 2008, pp. 11–15.
- [57] "Folium — Folium 0.20.0 documentation." <https://python-visualization.github.io/folium/latest> (accessed Aug. 09, 2025).
- [58] A. Santos, N. David, N. Perdigão, and E. Cândido, "Agent-Based modeling of tsunami evacuation at Figueirinha Beach, Setubal, Portugal," *Geosciences*, vol. 13, no. 11, p. 327, Oct. 2023, doi: 10.3390/geosciences13110327.
- [59] A. M. C. B. Silva, J. R. R. Da Cunha, and J. P. C. Da Silva, "Estimation of pedestrian walking speeds on footways," *Proceedings of the Institution of Civil Engineers - Municipal Engineer*, vol. 167, no. 1, pp. 32–43, Oct. 2013, doi: 10.1680/muen.12.00048.
- [60] T. Fujiyama and N. Tyler, 'An explicit study on walking speeds of pedestrians on stairs', Fujiyama, T. and Tyler, N. (2004) An explicit study on walking speeds of pedestrians on stairs. In: 10th International

Conference on Mobility and Transport for Elderly and Disabled People, 23-26 May 2004, Hamamatsu, Japan., 01 2004.

[61]“Google Maps,” Google Maps. https://www.google.com/maps/@38.5141134,-8.9105055,97a,35y,359.3h/data=!3m1!1e3?entry=ttu&g_ep=EgoyMDI1MDgyNS4wIKXMDSoASAFQAw%3D%3D (accessed Aug. 29, 2025).

[62]“Levantamento LiDAR de Portugal continental | DGT.” <https://www.dgterritorio.gov.pt/Levantamento-LiDAR-de-Portugal-continental-0> (accessed Aug. 14, 2025).