



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Master Universitario en Inteligencia Artificial, Reconocimiento de Formas e  
Imagen Digital  
Universidad Politécnica de Valencia

## **Simulación de Evacuación mediante Sistemas Multi-Agentes con MESA**

**TRABAJO SISTEMAS MULTI-AGENTES**

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e  
Imagen Digital

*Autor:* Miquel Gómez



# Resumen

Este trabajo presenta el desarrollo e implementación de un sistema de simulación multi-agente para el estudio de dinámicas de evacuación en situaciones de emergencia. Utilizando el framework MESA en Python, se ha creado un modelo discreto basado en malla que permite analizar el impacto de comportamientos individuales en procesos de evacuación de grupos de personas.

Se han implementado tres tipologías de agentes (educados, agresivos y lentos) que interactúan en diferentes escenarios. Estos escenarios representan espacios y situaciones reales. Los resultados experimentales revelan conclusiones contraintuitivas respecto a la literatura clásica, demostrando que en modelos discretos la agresividad puede mejorar la eficiencia global del sistema de evacuación.

Se presenta el sistema con todos sus componentes, la metodología de evaluación y los resultados obtenidos, destacando las implicaciones prácticas para el diseño de espacios seguros y eficientes.

**Palabras clave:** Sistemas Multi-Agentes; Agentes; Simulación; Evacuación; Simulación de evacuación; Multitudes.

---

# Índice general

---

<b>Índice general</b>	<b>IV</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Alcance y Objetivos . . . . .	1
1.1.1 Enfoque en Evacuación de Emergencia . . . . .	2
1.2 Uso de MESA . . . . .	2
1.3 Experimentación . . . . .	2
1.4 Estructura del Documento . . . . .	3
<b>2 Estado del Arte</b>	<b>5</b>
2.1 Objetivos de la Simulación de Evacuación . . . . .	5
2.1.1 Diseño Basado en Prestaciones (PBD) y Seguridad . . . . .	5
2.1.2 Nota sobre Nivel de Servicio (LOS) . . . . .	6
2.2 Marcos Regulatorios y Validación . . . . .	6
2.3 Arquitecturas de Navegación y Micro-comportamientos . . . . .	6
2.3.1 Modelos de Espacio Continuo . . . . .	6
2.3.2 Modelos Discretos y Grid-Based . . . . .	6
<b>3 Análisis del problema</b>	<b>7</b>
3.1 Modelo: Discrete Grid-Based . . . . .	7
3.2 Agentes: Comportamiento en Evacuación . . . . .	8
3.3 Movimiento . . . . .	8
3.3.1 Dead locks . . . . .	8
3.3.2 Preferencias . . . . .	9
3.3.3 Altas densidades . . . . .	9
<b>4 Evaluación</b>	<b>11</b>
4.1 Métricas de Evaluación . . . . .	11
4.2 Diseño de Experimentos . . . . .	12
4.2.1 Escenarios . . . . .	12
4.2.2 Variables Experimentales . . . . .	14
4.3 Configuración y Protocolo Experimental . . . . .	14
4.3.1 Valores por Defecto . . . . .	14
4.3.2 Definición de Experimentos . . . . .	15
4.4 Deadlocks . . . . .	16
4.5 Nota sobre Flujos Continuos . . . . .	16
<b>5 Implementación</b>	<b>17</b>
5.1 Visualización y Recolección de Datos . . . . .	17
5.1.1 Visualización de Agentes y Entorno . . . . .	17
5.1.2 Visualización de Gráficas y Datos . . . . .	18
5.2 Estructura del Modelo . . . . .	20
5.2.1 Entorno . . . . .	20
5.3 Navegación y Cálculo de Caminos . . . . .	20
5.3.1 Heurística en A* . . . . .	21
5.3.2 Diferenciación de Salidas . . . . .	21
5.4 Lógica del Agente . . . . .	21

---

5.4.1	1. Probabilidad de Movimiento . . . . .	21
5.4.2	2. Resolución Prioridad . . . . .	21
5.4.3	3. Gestión de Bloqueos (Deadlock) . . . . .	22
5.5	Flujo continuo . . . . .	22
<b>6</b>	<b>Resultados Experimentales</b>	<b>23</b>
6.1	Análisis de Resultados Agregados . . . . .	24
6.1.1	Densidad de Agentes . . . . .	24
6.1.2	Comportamiento Agresivo . . . . .	25
6.1.3	Agentes Lentos . . . . .	26
6.1.4	Algoritmos de Pathfinding . . . . .	27
6.1.5	Preferencias de Salida . . . . .	27
6.1.6	Número de Salidas . . . . .	28
6.1.7	Ancho de Salida . . . . .	28
6.2	Análisis de Evolución Temporal . . . . .	29
6.2.1	Metodología de Normalización Temporal . . . . .	29
6.2.2	Dinámica de las Preferencias de Salida . . . . .	30
6.2.3	Dinámica del Ancho de Salida . . . . .	31
<b>7</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>33</b>
7.1	Conclusiones . . . . .	33
7.2	Conclusiones Generales . . . . .	33
7.3	Trabajo Futuro . . . . .	34



---

# CAPÍTULO 1

## Introducción

---

La gestión de evacuaciones y la seguridad en situaciones de emergencia son preocupaciones crecientes en la planificación urbana y arquitectónica. Entender cómo las personas evacuan espacios durante emergencias, cómo reaccionan bajo presión y cómo interactúan entre sí en procesos de evacuación es fundamental para diseñar espacios seguros y eficientes.

La simulación informática surge como una alternativa ética, económica y segura a los experimentos con personas reales, permitiendo probar escenarios hipotéticos de riesgo sin poner en peligro vidas humanas.

Este trabajo surge de la asignatura de Sistemas Multi-Agente, con el objetivo de experimentar con el framework MESA en Python para crear un entorno de experimentación sobre evacuación de multitudes. Se centra en el desarrollo y análisis del sistema, así como en la evaluación de su comportamiento bajo diferentes configuraciones y escenarios.

### 1.1 Alcance y Objetivos

---

El objetivo principal de este trabajo es desarrollar un entorno de simulación multi-agente (MAS) flexible utilizando la librería MESA en Python, que permita estudiar el impacto de los comportamientos individuales heterogéneos en la dinámica de evacuación de grupos de personas.

- **Abstracción frente a Precisión Física:** Este proyecto **no pretende** crear una herramienta de ingeniería civil certificable bajo la norma ISO 20414 (estándar del que se habla más tarde) para validación legal de planos de evacuación, ni replicar con exactitud milimétrica unidades físicas reales.
- **Enfoque en Tendencias y Comportamiento de Evacuación:** El objetivo es simular tendencias emergentes y relaciones causales en procesos de evacuación (p.ej. "¿Cómo afecta un 10 % de agentes agresivos al tiempo total de evacuación?") en lugar de predecir tiempos exactos de evacuación.
- **Tipología de Agentes:** Se busca simular la variabilidad de las personas usando perfiles de agentes (*Polite*, *Aggressive*, *Slow*) para observar la composición de la multitud altera el comportamiento de esta.

### 1.1.1. Enfoque en Evacuación de Emergencia

El sistema se centra exclusivamente en el estudio de situaciones de evacuación de emergencia, donde la rapidez y eficiencia del proceso de evacuación son críticas para la supervivencia. Se pretende analizar cómo diferentes configuraciones de agentes y escenarios de evacuación afectan métricas clave como el tiempo total de evacuación, la tasa de flujo hacia las salidas y la congestión durante la evacuación.

Estos sistemas de evacuación son relevantes en contextos como:

- **Eventos Masivos:** Evacuación de conciertos, festivales y eventos deportivos en situaciones de emergencia.
- **Infraestructuras Públicas:** Planes de evacuación para aeropuertos, estaciones de tren y centros comerciales ante emergencias.
- **Diseño Urbano:** Planificación de rutas de evacuación en espacios públicos y minimización de riesgos durante evacuaciones.

Aunque el sistema también tiene la capacidad técnica de simular **flujos continuos de personas** en situaciones no urgentes, este aspecto no se ha explorado en profundidad en este trabajo. El enfoque principal es la evacuación de emergencia.

El enfoque principal y único de este trabajo es la evacuación de emergencia, validando si un espacio permite una evacuación segura y eficiente bajo diferentes condiciones y composiciones de agentes durante situaciones críticas. El sistema permite modelar de forma sencilla distintos escenarios de evacuación y composiciones de agentes en emergencias.

## 1.2 Uso de MESA

---

MESA es un framework de código abierto en Python diseñado para facilitar la creación, simulación y visualización de modelos basados en agentes. Proporciona una estructura modular que permite a los desarrolladores definir agentes, entornos y reglas de interacción de manera sencilla.

Además, una vez el sistema está implementado, MESA ofrece herramientas integradas para la recolección de datos y la visualización en tiempo real, lo que facilita el análisis de los resultados de las simulaciones.

En este trabajo, MESA se utiliza como la base para construir el modelo de simulación de multitudes, aprovechando sus capacidades para gestionar agentes heterogéneos y sus interacciones en un entorno discreto basado en malla.

Se ha implementado todo el sistema siguiendo la documentación oficial de MESA [1] y adaptando sus componentes a las necesidades específicas del estudio de evacuación.

## 1.3 Experimentación

---

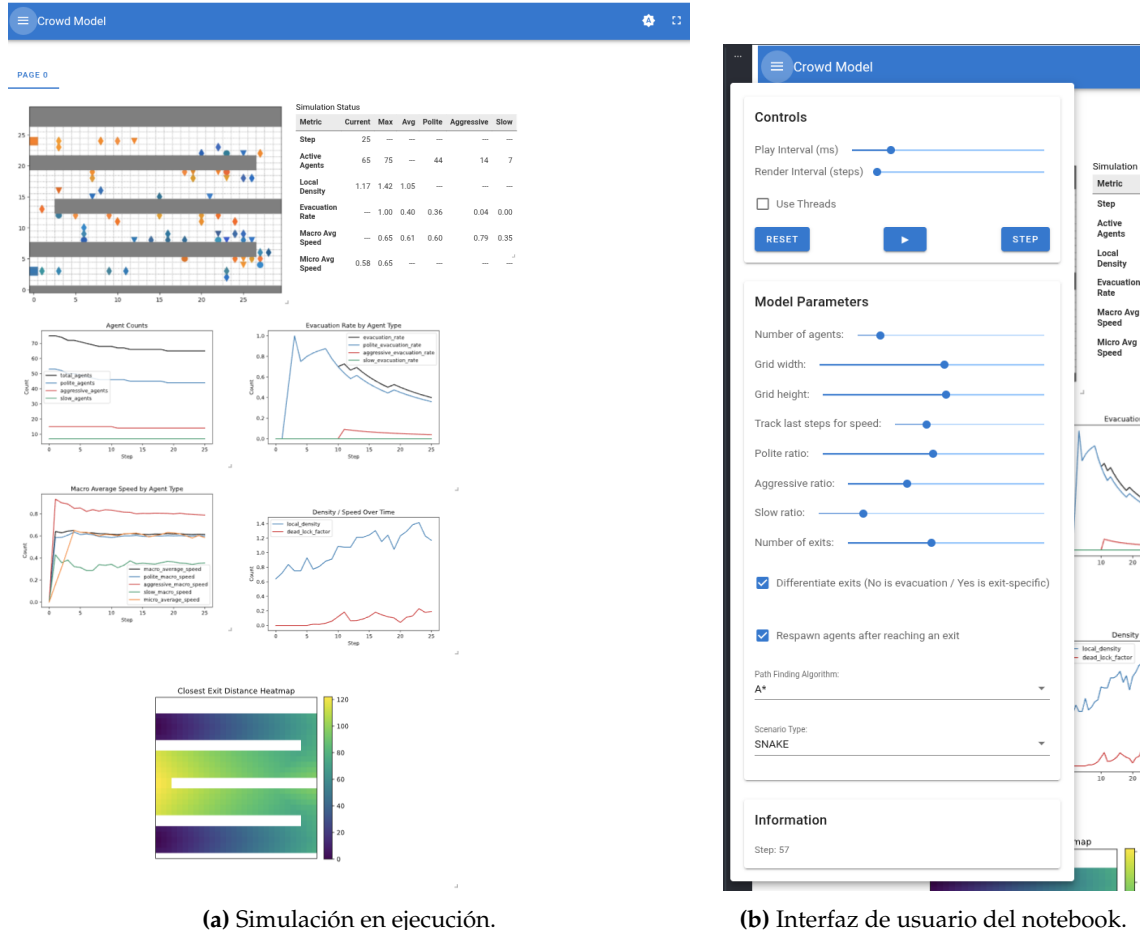
El trabajo incluye un notebook de Jupyter que permite ejecutar simulaciones con diferentes configuraciones y escenarios, cambiarlos y visualizar los resultados y las estadísticas en tiempo real.

Este notebook facilita la experimentación al permitir modificar parámetros como la densidad de agentes, la proporción de tipos de agentes y las características del entorno.



Además, incluye gráficos y tablas que resumen las métricas clave de cada simulación, permitiendo un análisis comparativo entre diferentes configuraciones.

Cualquier persona interesada puede clonar el repositorio del proyecto, instalar las dependencias necesarias y ejecutar el notebook para replicar los experimentos o realizar nuevas pruebas con diferentes parámetros.



(a) Simulación en ejecución.

(b) Interfaz de usuario del notebook.

Figura 1.1: Capturas de pantalla del sistema de simulación final.

## 1.4 Estructura del Documento

El documento se estructura en: estado del arte, donde se revisan las tecnologías actuales; análisis del problema, donde se define la estrategia de modelado; evaluación experimental, detallando métricas y escenarios; implementación técnica; y finalmente los resultados y conclusiones obtenidos.

El desarrollo del código y la redacción de este trabajo se ha realizado totalmente a mano. Se ha hecho uso de asistentes como Copilot para la revisión y análisis de resultados, pero sin generar ningún contenido automáticamente. La única excepción es la generación automática de gráficos y tablas a partir de los datos experimentales, pero siempre bajo la supervisión y control de los autores.



---

## CAPÍTULO 2

# Estado del Arte

---

La simulación de evacuación ha evolucionado de ser una disciplina académica minoritaria, a convertirse en una herramienta crítica para garantizar la seguridad en situaciones de emergencia en infraestructuras modernas.

Este capítulo evalúa el estado del arte actual en simulación de evacuación, dividiendo el análisis en los objetivos que persiguen estos sistemas: los marcos regulatorios de seguridad en evacuaciones y las arquitecturas de micro-comportamiento utilizadas para modelar la navegación de agentes durante evacuaciones.

### 2.1 Objetivos de la Simulación de Evacuación

---

En el panorama actual de seguridad en edificaciones, la simulación de evacuación responde principalmente a regulaciones de seguridad y Diseño Basado en Prestaciones para garantizar evacuaciones efectivas en situaciones de emergencia.

#### 2.1.1. Diseño Basado en Prestaciones (PBD) y Seguridad

Históricamente, los códigos de edificación eran toscos, dictando reglas rígidas sobre anchos de escalera y distancias de evacuación. El enfoque PBD permite diseños flexibles siempre que se demuestre, mediante simulación, que se cumplen los criterios de seguridad [2].

La métrica fundamental en PBD es la relación entre el Tiempo Disponible para la Evacuación Segura (ASET) y el Tiempo Requerido para la Evacuación Segura (RSET). Para que un diseño sea válido, debe cumplirse que:

$$ASET > RSET + Margen\_de\_Seguridad \quad (2.1)$$

Mientras que el ASET depende de la dinámica del fuego (toxicidad, calor, humo), el RSET es calculado por el simulador de multitudes y se compone de:

- **Tiempo de Detección y Tiempo de Notificación:** Darse cuenta del peligro y alertar.
- **Tiempo de Pre-evacuación:** El intervalo entre la alarma y el primer movimiento.
- **Tiempo de Viaje:** La fase física del movimiento desde el origen hasta una zona segura. Este es el componente principal que modela el sistema propuesto.

### 2.1.2. Nota sobre Nivel de Servicio (LOS)

Aunque existen metodologías como el Nivel de Servicio (LOS) para evaluar el confort peatonal en situaciones no urgentes [3], este trabajo se centra exclusivamente en evacuación de emergencia. Por tanto, las métricas de confort no son consideradas.

## 2.2 Marcos Regulatorios y Validación

---

Para que un modelo de simulación sea considerado una herramienta válida y no un mero ejercicio teórico, debe adherirse a estándares de validación internacionales.

- **ISO 20414:2020:** Esta norma establece protocolos rigurosos de verificación y validación para modelos de evacuación. Exige pruebas de componentes (p.ej., verificar que un agente se mueve a la velocidad asignada), verificación funcional (capacidad de flujo en puertas, contraflujo en pasillos) y validación cualitativa de comportamientos emergentes como la formación de arcos en las salidas [4].
- **NIST y SFPE:** El National Institute of Standards and Technology y la Society of Fire Protection Engineers proporcionan los datos demográficos y de comportamiento estándar (velocidades según edad, dimensiones corporales) que deben utilizarse para configurar los agentes [5].

Dado el alcance académico de este trabajo, no se pretende certificar el modelo bajo ISO 20414, pero se han seguido sus directrices para asegurar una base sólida y reproducible.

## 2.3 Arquitecturas de Navegación y Micro-comportamientos

---

La lógica de un simulador de multitudes reside en sus algoritmos de navegación local y resolución de conflictos. Existen dos tipos principales en la literatura:

### 2.3.1. Modelos de Espacio Continuo

Dominantes en la industria de la animación y la robótica por su fidelidad visual.

- **Social Force Model (SFM):** Propuesto por Helbing [6], trata a los agentes como partículas sometidas a fuerzas newtonianas. Combina una fuerza impulsora hacia la meta con fuerzas repulsivas para mantener la distancia personal.
- **Optimal Reciprocal Collision Avoidance (ORCA):** Opera en el espacio de velocidades para garantizar matemáticamente trayectorias libres de colisiones [7]. Aunque eficiente, puede resultar en comportamientos 'perfectos' o robóticos.

### 2.3.2. Modelos Discretos y Grid-Based

Los modelos basados en Autómatas Celulares (CA) o mallas discretas dividen el espacio en celdas. Son la base del sistema implementado en este trabajo. El desafío principal es la resolución de conflictos (cuando dos agentes compiten por la misma celda). El estado del arte actual emplea *Floor Fields* (campos de potencial) estáticos y dinámicos para guiar a los agentes, y utiliza teoría de juegos o heurísticas de "paciencia" para resolver bloqueos, superando las reglas simples de exclusión [8].

Ya que este es el elegido para el sistema, se profundiza más adelante.

---

## CAPÍTULO 3

# Análisis del problema

---

El desarrollo de simuladores de evacuación implica equilibrar el realismo físico con la complejidad computacional. Dado el objetivo de estudiar comportamientos emergentes durante evacuaciones de grandes grupos, se requiere una aproximación que permita simular cientos de personas evacuando en tiempo real o cercano al tiempo real, por lo que se prioriza la ejecución del trabajo para la asignatura antes que la precisión milimétrica.

Además, el framework MESA en Python, aunque flexible y modular, no está optimizado para simulaciones de alta fidelidad física. Por lo tanto, esto nos deja como mejor opción un modelo discreto basado en rejilla que simplifica la representación espacial y las interacciones entre agentes.

### 3.1 Modelo: Discrete Grid-Based

---

Para llevar a cabo las simulaciones, se ha optado por un **Modelo Discreto basado en Rejilla (Grid-Based)**, implementado sobre el framework MESA en Python.

Comparado con los modelos planteados en el estado del arte, esta elección presenta las siguientes cualidades:

1. **Espacio Discreto:** El entorno se divide en celdas cuadradas, donde cada celda haremos que solo pueda contener un agente, una pared (obstáculo) o una salida. Esto simplifica la detección de colisiones y elimina los costosos cálculos de geometrías complejas realistas.
2. **Tiempo Discreto:** La simulación avanza en pasos discretos o "steps". La velocidad de los agentes se modela probabilísticamente: un agente con "mayor velocidad" simplemente tiene una mayor probabilidad de moverse en cada step.
3. **Navegación:** La estructura de malla facilita la implementación directa de algoritmos de búsqueda de caminos en grafos como BFS y A\*, esenciales para la planificación de rutas de los agentes. Con estos algoritmos, se pueden modelar las rutas óptimas hacia las salidas teniendo en cuenta los obstáculos.

BFS es simple y representa NO tener conocimiento previo del entorno (seguir solo las señales de evacuación), mientras que A\* permite incorporar heurísticas para simular conocimiento previo o preferencias de ruta (sé que por aquí había una salida). Esto permite estudiar como afecta el conocimiento el que las personas hagan caso a las señales de evacuación o a su instinto.

## 3.2 Agentes: Comportamiento en Evacuación

---

Si nos fijamos en análisis reales de evacuaciones, las personas exhiben comportamientos heterogéneos bajo situaciones de emergencia. Para obtener resultados significativos sobre evacuación, el sistema debe ser capaz de modelar distintos perfiles de comportamiento de evacuación que interactúen entre sí.

En concreto, para este trabajo se han definido tres tipos de agentes con comportamientos diferenciados durante evacuaciones. Estos pretenden agrupar las características más relevantes observadas en emergencias:

- **Polite (Educados):** Serían la gente por defecto, educados y cooperativos. Deben ser capaces de ceder el paso y esperar, evitando conflictos activos y no imponerse ante los demás.
- **Aggressive (Agresivos):** Modelan la competencia y el pánico. Sería la gente que corre y se pone por delante. Su lógica ha de priorizar su movimiento sobre el de los demás, ocupando espacios libres sin consideración.
- **Slow (Lentos):** Modelan la diversidad física, gente con dificultades, niños o personas mayores. Actúan como obstáculos dinámicos y permiten estudiar el impacto de los usuarios más vulnerables en el flujo general.

La lógica e interacción de estos agentes será muy similar, pero con pequeñas variaciones que los cualifican. Esto permitirá observar dinámicas emergentes interesantes.

En concreto, vamos a evaluar comportamientos es choques de flujo, cuellos de botella y congestión en pasillos estrechos.

## 3.3 Movimiento

---

Para el movimiento de los agentes, se implementa una lógica simple basada en la búsqueda del camino óptimo hacia la salida más cercana utilizando algoritmos como A\* o BFS.

De forma eficiente y únicamente al inicio de la simulación, el sistema creará 'rutas precomputadas' en la que los agentes sepan donde deben ir en cada momento, teniendo en cuenta los obstáculos y las paredes: el suelo o las celda, tendrán un valor asociado que indica la distancia a la salida más cercana.

Se menciona esto porque es un aspecto crucial del sistema, ya que los agentes no actualizarán sus rutas de forma dinámica en función de la posición de los demás agentes. Se seguirán rutas precomputada hacia la salida más cercana.

Además, se pretende que los agentes resuelvan estas situaciones con una serie de reglas y comportamientos adicionales. Por esto introducimos los siguientes conceptos:

### 3.3.1. Dead locks

En las simulaciones basadas en mallas discretas, un desafío común es la aparición de **dead locks** o bloqueos mutuos entre agentes. Estos ocurren cuando varios agentes intentan moverse en direcciones opuestas en un espacio limitado, resultando en una situación donde ninguno puede avanzar: Uno va a la celda del otro y viceversa, pero para poder moverse, la celda de algún agente debe estar libre.

Para mitigar estas situaciones, se definen una serie de reglas y mecanismos dentro de la lógica de los agentes. Se mantendrá un contador de bloqueo que se incrementa cada vez que un agente no puede moverse y el comportamiento del agente cambiará ligeramente en función de este contador.

Cada tipo de agente se verá más o menos afectado por los dead locks. Lo buscamos provocar es que los agentes dejen de ir en la dirección más próxima a la salida, y se muevan de forma 'asustada' a celdas vecinas para intentar aliviar la congestión y / o encontrar una ruta alternativa.

También se implementa un mecanismo de influencia entre agentes: si un agente está bloqueado y sus vecinos también lo están, su contador de bloqueo aumentará más rápido. Esto simula la propagación del pánico o la frustración en situaciones de congestión.

### 3.3.2. Preferencias

De forma general se hará que los agentes escaneen el entorno en una dirección concreta, de forma que en caso de empate, se priorice la celda que esté en esa dirección. Esto simula el comportamiento humano de 'siempre ir a la derecha' o 'seguir recto'.

Si una celda mejor o igual a la actual está libre, el agente se moverá a esa celda. En caso de empate, se moverá a la primera celda escaneada simulando una dirección preferida.

#### Salidas preferidas

Además, los agentes también pueden tener preferencias de salida. En lugar de dirigirse siempre a la salida más cercana, algunos agentes pueden tener una salida asignada o preferida.

Esto simula situaciones donde las personas tienen conocimiento previo del entorno o están siguiendo 'por donde han entrado'. Esto puede afectar la dinámica de la multitud, especialmente en escenarios con múltiples salidas alejadas entre ellas.

### 3.3.3. Altas densidades

Cuando la densidad de agentes aumenta, el movimiento se vuelve más difícil y la gente no camina de forma fluida.

Para simular este fenómeno, se implementa una mecánica donde los agentes tienen una probabilidad creciente de no moverse en función del número de vecinos que tengan en las 8 celdas adyacentes. Esto simula el efecto de 'pisar con cuidado' o 'moverse lentamente' en situaciones de alta densidad.

En concreto, esta probabilidad solo empezará a contar cuando hayan más de 3 vecinos, y no afectará igual a todos los tipos de agentes. Los agresivos serán menos propensos a verse afectados, mientras que los amables serán más sensibles a la congestión.





---

## CAPÍTULO 4

# Evaluación

---

La validación y análisis del modelo propuesto se realiza mediante una serie de experimentos controlados. En este capítulo se detallan las métricas diseñadas para cuantificar el comportamiento de la multitud y la metodología de los experimentos.

### 4.1 Métricas de Evaluación

---

Se han implementado monitores de datos (Data Collectors, clase de MESA) que registran paso a paso el estado de la simulación.

Aunque las unidades son abstractas (celdas, ticks), estas métricas son análogas a las utilizadas en estudios de seguridad profesional:

- **Tiempo de Evacuación (Total Steps):** Cantidad de ticks necesarios para que el último agente abandone el escenario. Representa el RSET (Required Safe Egress Time) y es la métrica principal de eficiencia.
- **Tasa de Evacuación (Evacuation Rate):** Cantidad de agentes que alcanzan una salida por tick. Permite visualizar la constancia del flujo e identificar cuellos de botella (caídas en el flow rate).
- **Densidad Local (Local density):** Se calcula para cada agente la ocupación de sus 8 celdas vecinas. El promedio global de esta métrica indica el nivel de congestión del sistema.
- **Factor de Bloqueo (Deadlock Factor):** Métrica específica diseñada para este sistema discreto. Mide la proporción de agentes que *intentaron* moverse pero no han podido debido a que sus celdas objetivo estaban ocupadas. Un alto factor de bloqueo indica fricción ineficiente o colapso del flujo.

Esta métrica se usa para gestionar el comportamiento de los agentes en situaciones de congestión y alterar su lógica de movimiento.

- **Velocidad Macro y Micro:**
  - *Velocidad Macro:* Velocidad promedio de todo el conjunto de agentes en el sistema a lo largo de toda la simulación.
  - *Velocidad Micro:* Velocidad individual efectiva 'actual'. Mide la velocidad media pero solo teniendo en cuenta los últimos 4 ticks.

Luego, también se recogen métricas de cada una por cada tipo de agente (agresivos, educados y lentos) además de las globales.

## 4.2 Diseño de Experimentos

---

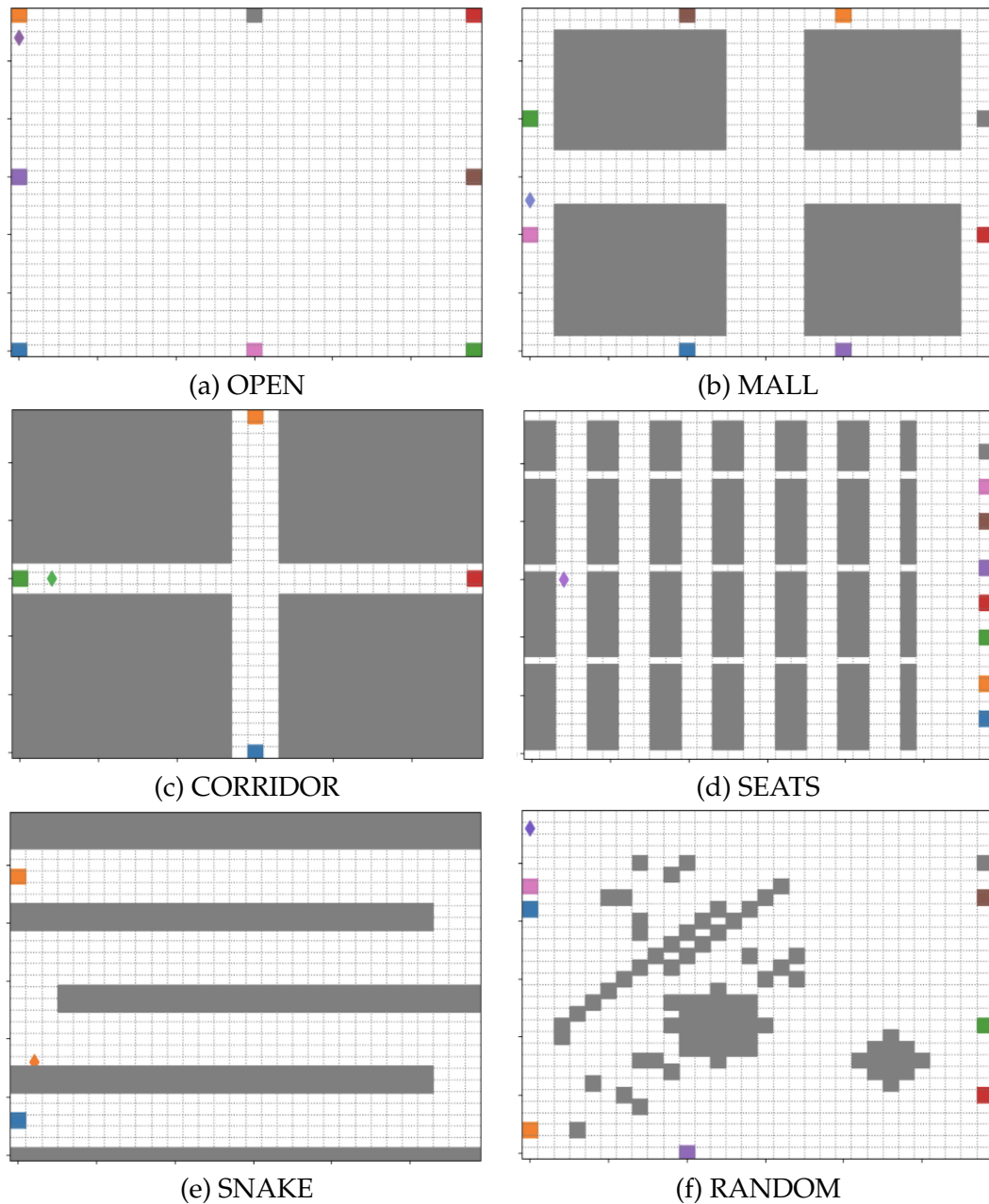
Se han diseñado escenarios específicos para poner a prueba distintas hipótesis sobre la dinámica de multitudes. Aunque el sistema que se ha creado soporta simulación de flujo continuo, este estudio se centra exclusivamente en el **Escenario de Evacuación**, donde el objetivo es vaciar el recinto.

### 4.2.1. Escenarios

En total, se han definido 6 escenarios diferentes 4.1, de los cuales nos interesan 4 para este trabajo:

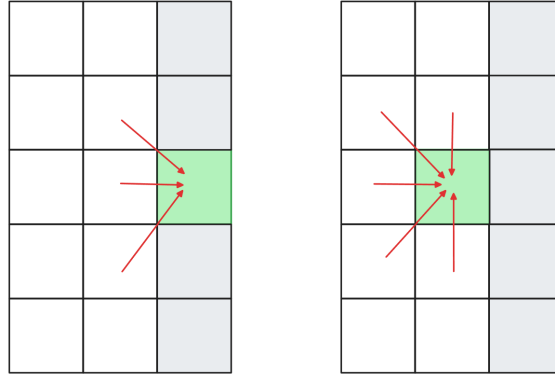
- **OPEN:** Un espacio abierto sin obstáculos. Hay salidas en los bordes de la malla. No es de mucho interés para este trabajo, pero sirve como referencia y espacio de experimentación.
- **MALL:** Pretende simular un centro comercial donde se juntan varios pasillos con tiendas al rededor. Los agentes pueden moverse por los pasillos y por el anillo exterior, pero no pueden atravesar las tiendas. Las salidas están de dos a dos en los bordes de la malla.
- **CORRIDOR:** Similar a MALL, pretende simular la intersección de varios pasillos con poco espacio para moverse. Las salidas están de dos a dos en los finales de cada pasillo.
- **SEATS:** Pretende simular un auditorio, cine o concierto, en el que todas las salidas están en un lado y el resto de la sala está llena de filas de asientos con 'pasillos' entre medias.
- **SNAKE:** Simula un pasillo zigzagante con una o dos salidas en los extremos. La idea detrás de este escenario es ver el comportamiento de los agentes en espacios estrechos y con giros. Podrían simular una cola de personas.
- **RANDOM:** Por último, este escenario genera obstáculos (paredes, círculos y cuadrados) de forma aleatoria en la malla. Las salidas están en los bordes de la malla. Al igual que OPEN, este escenario no es de mucho interés para el trabajo, pero sirve como referencia y espacio de experimentación.

La cantidad de salidas en cada escenario se puede configurar, yendo desde una sola salida hasta una cierta cantidad definida para cada uno. El tamaño de estos se puede cambiar y los obstáculos y salidas se adaptan de forma dinámica al nuevo tamaño.



**Figura 4.1:** Escenarios implementados en el sistema. Cada uno presenta diferentes configuraciones de obstáculos y salidas para evaluar distintos aspectos del comportamiento de multitudes.

Mencionar que las salidas NO están incrustadas en las paredes, sino que son celdas libres en los bordes de la malla. Los agentes pueden salir del escenario al llegar a una celda de salida y esto se puede hacer de 5 en 5. Esto quiere decir que el flujo máximo que cada salida puede soportar es de 5 agentes por tick.



**Figura 4.2:** Posición de las salidas importa y puede afectar a la cantidad de agentes que pueden evacuar rápidamente.

Es caso de haberlas incrustado, sería de tres. Es algo a mencionar con lo que NO se experimenta en este trabajo, pero que podría ser interesante para futuros estudios. Somos generosos y cualquier agente al lado de una salida puede evacuar, lo que puede no ser realista en escenarios de alta densidad ya que las puertas físicas tienen un ancho limitado.

#### 4.2.2. Variables Experimentales

En cada escenario, se modifican las siguientes variables independientes para observar su impacto en las métricas:

- **Densidad Poblacional:** Se varía el número de agentes iniciales (de 10 a 300) para observar la transición de flujo libre a congestión.
- **Composición (Polite vs Aggressive vs Slow):** Se alteran los ratios de agentes para responder preguntas como: ¿Es una multitud cooperativa siempre más rápida? ¿Cómo penalizan los agentes lentos al grupo?
- **Estrategia de Navegación (A\* vs BFS):** Se comparan algoritmos para ver si el cálculo de rutas óptimas (A\*) compensa su coste computacional frente a la simple (BFS).
- **Configuración de Salidas:** Se prueba el impacto de asignar salidas fijas (preferencias) frente a dejar que los agentes elijan la más cercana (oportunismo).

### 4.3 Configuración y Protocolo Experimental

Para garantizar la reproducibilidad de los resultados, a continuación se detallan los valores por defecto utilizados en todas las simulaciones, así como las modificaciones específicas realizadas para cada conjunto de experimentos.

#### 4.3.1. Valores por Defecto

Salvo que se especifique lo contrario en la descripción de cada experimento, las simulaciones parten de una configuración base establecida (Tabla 4.1). Cada simulación se repite **1000 veces** para asegurar la robustez estadística de los datos.

Parámetro	Valor por Defecto
Dimensiones de la Malla	30 x 30 celdas
Población Inicial	125 agentes
Algoritmo de Pathfinding	A* (con heurística Euclidiana)
Escenario Base	MALL
Número de Salidas	4 salidas (distribuidas en los bordes)
Comportamiento de Salida	Oportunista (Salida más cercana)
Reaparición	Desactivada (Respawn = False)
Distribución de Agentes	70 % <b>Polite</b> (Educados), 20 % <b>Aggressive</b> (Agresivos), 10 % <b>Slow</b> (Lentos)

Tabla 4.1: Configuración base para las simulaciones

### 4.3.2. Definición de Experimentos

Se han llevado a cabo las siguientes baterías de experimentos, modificando parámetros específicos respecto a la configuración base:

#### 1. Impacto de la Densidad (Puro y Mixto):

- **Objetivo:** Evaluar la saturación en distintos escenarios.
- **Variaciones:** Población [10, 25, 50, 75, 100, 125, 150, 175, 200, 300].
- **Casos:** Se ejecutan dos series, una con población 100 % Polite y otra con la distribución mixta por defecto (70/20/10).

#### 2. Impacto de la Agresividad:

- **Escenario Focal:** SEATS (Entorno estructurado).
- **Variaciones:** Ratio de agentes agresivos [0 %, 10 %, 25 %, 50 %, 80 %, 100 %] manteniendo el resto como Polite sin lentos.
- **Densidades Probadas:** 125, 200 y 300 agentes.

#### 3. Impacto de Agentes Lentos:

- **Escenario Focal:** SEATS.
- **Variaciones:** Ratio de agentes lentos [0 %, 10 %, 25 %, 50 %, 80 %, 100 %] manteniendo el resto como Polite sin agresivos.
- **Densidades Probadas:** 125, 200 y 300 agentes.

#### 4. Comparativa de Pathfinding (A\* vs BFS):

- **Alcance:** Todos los escenarios.
- **Variación:** Se alterna el algoritmo de navegación entre A\* y BFS manteniendo la población mixta por defecto.

#### 5. Preferencia de Salidas:

- **Alcance:** Todos los escenarios.
- **Variación:** Activación del parámetro *differentiate\_exits*, forzando a los agentes a tener una salida preasignada aleatoria en lugar de la óptima.

#### 6. Número de Salidas:

- **Variación:** Se modifica el número de salidas disponibles  $N$  en función de la geometría del escenario (Mall/Seats: 1-8, Corridor: 1-4, Snake: 1-2).

---

## 4.4 Deadlocks

---

Los deadlocks serán estudiados mediante la métrica de **Factor de Bloqueo**, observando su evolución temporal y su correlación con la densidad y composición de la multitud.

Se analizarán situaciones donde los deadlocks son más prevalentes, como en pasillos estrechos o intersecciones, y se evaluará la efectividad de las estrategias implementadas para mitigarlos.

---

## 4.5 Nota sobre Flujos Continuos

---

Aunque el sistema tiene capacidades técnicas para simular flujos continuos de personas en situaciones no urgentes, este aspecto queda completamente fuera del alcance del presente trabajo.

Este trabajo se centra exclusivamente en escenarios de evacuación de emergencia donde el objetivo es vaciar completamente el espacio, no en situaciones de flujo diario o confort peatonal.

Los flujos continuos podrían estudiarse en trabajos futuros activando la opción de *reaparición de agentes*, pero no se consideran relevantes para el estudio de evacuación que nos ocupa.

---

## CAPÍTULO 5

# Implementación

---

El sistema ha sido implementado utilizando Python y la librería **MESA**. Como se ha mencionado, MESA es un framework modular para el modelado basado en agentes y es bastante popular porque permite una rápida prototipación, una fácil extensibilidad de comportamientos y la recolección eficiente de métricas estadísticas.

Los principales componentes que hay que definir en MESA son los mencionados hasta ahora: el modelo (entorno) y los agentes (comportamiento). Además, MESA ofrece herramientas para la visualización y recolección de datos que se han aprovechado en este trabajo.

### 5.1 Visualización y Recolección de Datos

---

MESA incluye un módulo de visualización basado en navegador web que permite observar la simulación en tiempo real.

Aprovechamos esta funcionalidad para crear una interfaz interactiva donde se pueden ajustar parámetros como la cantidad de agentes, la proporción de tipos de agentes o las características del entorno antes de iniciar la simulación entre otros.

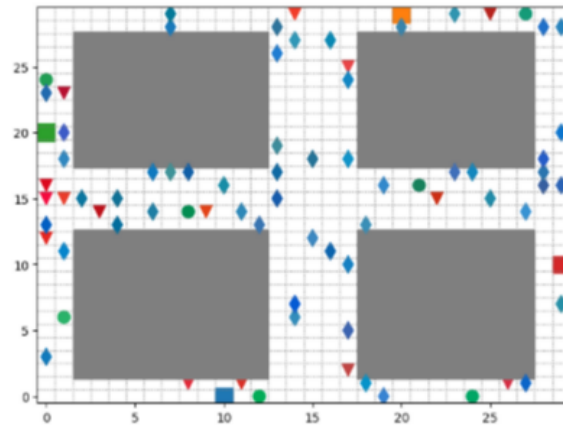
Se han implementado visualizadores personalizadas, en las que hemos recolectado los datos de la simulación y se muestran en gráficos con tablas en tiempo real.

#### 5.1.1. Visualización de Agentes y Entorno

Además, para hacer visibles los agentes, coloreamos de diferentes tonos de ciertos colores y formas según su tipo: los agentes *Polite* serán rombos de color azul, los *Aggressive* serán triángulos rojos y los *Slow* círculos verdes.

También, los obstáculos y las salidas serán cuadrados, los obstáculos se representarán en gris oscuro y las salidas cada una de un color.

En las simulaciones en las que los agentes tengan una salida favorita, se usarán los colores de cada una en lugar de los colores por defecto.



**Figura 5.1:** Visualización de los diferentes tipos de agentes y elementos del entorno con sus respectivas formas y colores.

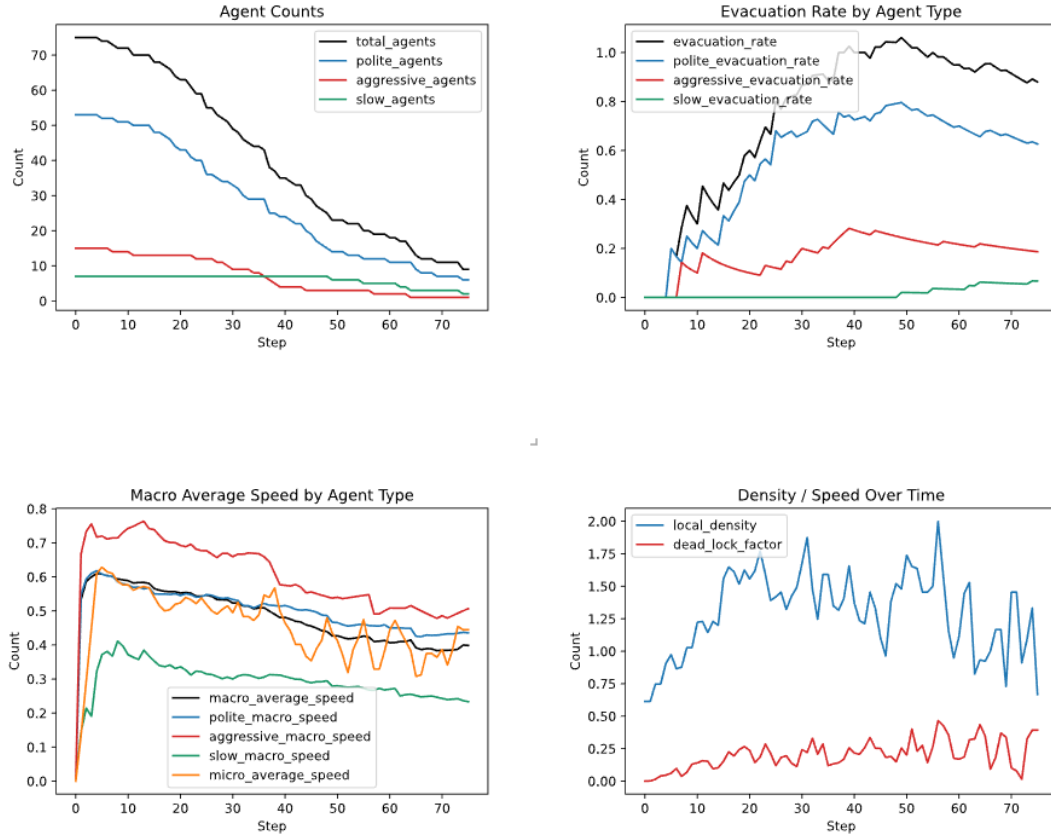
Esto se ha hecho a partir de la definición de una función de `portrayal` personalizada que asigna colores y formas según el tipo de agente o elemento del entorno.

### 5.1.2. Visualización de Gráficas y Datos

Se han creado gráficas en tiempo real de las métricas, una tabla resumen de los datos actuales de la simulación y una visualización del entorno con la distancia a la salida más cercana de cada celda.

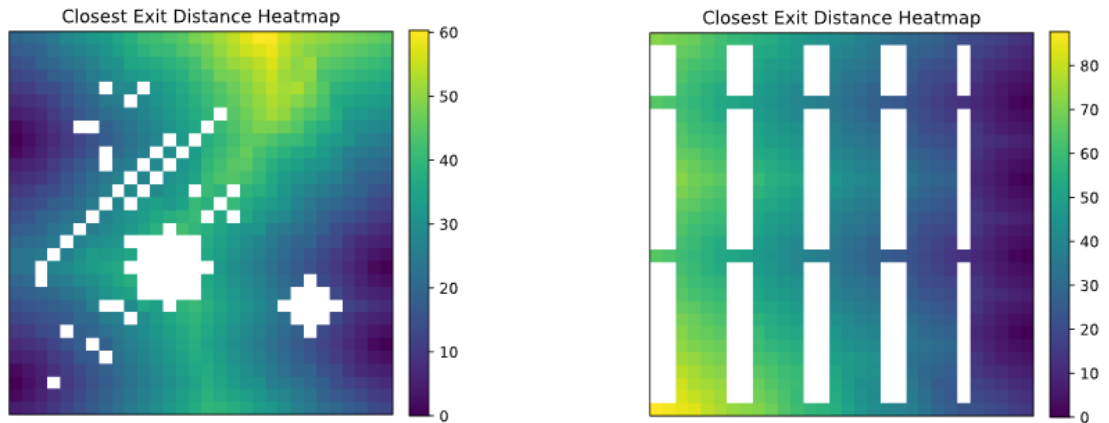
La Figura 5.2 muestra las principales métricas recolectadas durante una simulación, incluyendo el conteo de agentes a lo largo del tiempo, las tasas de evacuación por tipo de agente, la velocidad macro promedio y la evolución de la densidad y el factor de bloqueo. Estas gráficas permiten observar en tiempo real el comportamiento emergente del sistema y identificar patrones como cuellos de botella o situaciones de deadlock.





**Figura 5.2:** Visualización de métricas en tiempo real durante una simulación de evacuación. Se muestran cuatro gráficas principales: conteo de agentes por tipo, tasas de evacuación diferenciadas, velocidades macro promedio por tipo de agente, y evolución de la densidad local junto con el factor de bloqueo.

Adicionalmente, el sistema genera mapas de calor que visualizan la distancia desde cada celda hasta la salida más cercana, como se muestra en las Figuras 5.3a y 5.3b. Estos mapas de distancia son fundamentales para la navegación de los agentes, ya que constituyen los *floor fields* que guían su movimiento hacia las salidas durante la evacuación.



(a) Mapa de distancias en escenario RANDOM con obstáculos aleatorios.

(b) Mapa de distancias en escenario SEATS tipo auditorio.

**Figura 5.3:** Mapas de calor mostrando la distancia a la salida más cercana en diferentes escenarios. Los colores más claros (amarillo) indican mayor distancia, mientras que los más oscuros (púrpura) indican proximidad a las salidas. Las zonas blancas representan obstáculos o salidas.

En la Figura 1.1 se pueden ver capturas de pantalla del sistema completo de simulación con ejemplos de lo mencionado.

## 5.2 Estructura del Modelo

En MESA, un modelo es una clase que hereda de `mesa.Model` y que contiene la lógica principal de la simulación.

En concreto, los modelos de MESA se encargaran principalmente de inicializar el entorno, crear los agentes, definir la función `Step` que mueve a los agentes y actualizar y controlar el estado del entorno.

### 5.2.1. Entorno

En nuestro caso, el entorno será una malla bidimensional que representa los espacios por los que se moverán los agentes. Esta estará compuesta por agentes, obstáculos y salidas.

Esta malla define por defecto una serie de funcionalidades que facilitan la gestión del espacio y la interacción entre agentes. Usaremos una malla de tipo `OrthogonalMooreGrid`, la cual permite a los agentes moverse en las cuatro direcciones cardinales y en las cuatro diagonales. No permitiremos que más de un agente ocupe la misma celda al mismo tiempo, y haremos que los obstáculos y las salidas ocupen celdas enteras en la malla.

Se define un ancho y alto para la malla, y según el tipo de escenario seleccionado para la simulación, se generarán los obstáculos y las salidas en posiciones y formas específicas adaptándose al tamaño definido. Luego, los agentes aparecerán en el resto de celdas disponibles de forma aleatoria.

Se han implementado 6 tipos de escenarios distintos (OPEN, MALL, CORRIDOR, SEATS, SNAKE y RANDOM), cada uno con características topológicas específicas que permiten evaluar diferentes aspectos del comportamiento de multitudes, como se puede observar en la Figura 4.1.

## 5.3 Navegación y Cálculo de Caminos

Para optimizar el rendimiento y evitar costosas búsquedas de ruta por cada agente en cada paso, se utiliza un enfoque de **Mapas de Distancia Estáticos (Floor Fields)**.

Al inicio de la simulación, el entorno pre-calcula la distancia desde cada celda transitable hasta la salida más cercana utilizando un algoritmo de búsqueda en anchura (BFS) o uno con heurística A\*. Esto genera un 'mapa de calor' o gradiente donde cada celda contiene el coste de movimiento  $d(c)$  hasta la meta.

$$d(c_{actual}) \geq d(c_{siguiente}) \quad (5.1)$$

Los agentes simplemente deben inspeccionar sus celdas vecinas y escoger aquella con el valor  $d$  más bajo, realizando un descenso de gradiente.

No se requiere que los agentes recalculen rutas dinámicamente, lo que reduce significativamente la carga computacional y permite simular grandes multitudes en tiempo real. Y mencionar que se hace  $\geq$  y no solo  $>$  para permitir movimientos laterales en caso de deadlocks.

### 5.3.1. Heurística en A\*

Ya que el movimiento de los agentes permite movimientos diagonales, la heurística utilizada en A\* es la **Distancia Euclidiana**, que calcula el coste estimado con una pequeña optimización para evitar agentes atascados en esquinas:

$$H(current, goal) = 0,5 \times \sqrt{(x_{goal} - x_{current})^2 + (y_{goal} - y_{current})^2}$$

Ese factor de 0.5 se ha añadido para reducir la influencia de la heurística y favorecer caminos más directos en lugar de diagonales excesivas que podrían llevar a bloqueos. Se ha llegado al valor tras experimentación.

### 5.3.2. Diferenciación de Salidas

En escenarios complejos donde existen preferencias de salida (p.ej., "salir por donde entré"), el sistema genera múltiples mapas de distancia, uno por cada salida. Cada agente recibe una referencia al mapa que debe seguir, permitiendo flujos cruzados y comportamientos heterogéneos en la selección de rutas.

En caso de ir a la más cercana, el agente recibe todos los mapas y evalúa el mínimo en cada paso.

## 5.4 Lógica del Agente

La clase `CrowdAgent` descende de la clase `mesa.Agent` y encapsula la lógica de decisión. En cada paso de simulación (`step`), el agente ejecuta el siguiente ciclo:

### 5.4.1. 1. Probabilidad de Movimiento

No todos los agentes se mueven en cada tick. La decisión de intentar moverse depende de dos factores:

- **Velocidad Base:** Un valor aleatorio que se elige para cada agente entre un rango concreto para cada tipo de agente. Este valor representa la probabilidad base de que el agente intente moverse en cada tick. Los agentes *Slow* tendrán una probabilidad base menor que los *Polite* y estos a su vez menor que los *Aggressive*.
- **Factor de Multitud (Crowd Slowdown):** Si el agente está rodeado por muchos vecinos (densidad alta), su probabilidad de movimiento disminuye, simulando la fricción física y la necesidad de ajustar el paso en aglomeraciones. Este efecto es más pronunciado en agentes *Polite* y menos en *Aggressive* y en los *Slow*.

### 5.4.2. 2. Resolución Prioridad

Para crear la ilusión de diferentes niveles de agresividad, se implementa una lógica simple en la que, para cada `step`, primero los agentes *Aggressive* se moverán a las celdas que deseen ocupar, seguidos por los *Polite* y los *Slow* en orden aleatorio.

### 5.4.3. 3. Gestión de Bloqueos (Deadlock)

Se ha implementado un mecanismo de **Contador de Bloqueo** ( $C_{dl}$ ) para situaciones donde el flujo se detiene (p.ej., dos flujos opuestos en un pasillo estrecho).

El factor de bloqueo es la métrica clave para entender la congestión. Se calcula como la cantidad de ticks en los que se ha intentado mover pero no ha podido, multiplicado por un factor específico para cada tipo de agente más un componente de influencia de los vecinos.

- Si un agente intenta moverse pero no puede, incrementa su contador  $C_{dl}$ .
- Cuando  $C_{dl}$  supera un umbral, el agente entra en estado de "Pánico/Presión".
- En este estado, el agente relaja sus reglas de movimiento: puede aceptar moverse a celdas que *no* reducen la distancia óptima y puede incluso retroceder o moverse lateralmente para aliviar la congestión, dejando pasar a otros agentes.
- El estado de bloqueo se contagia "localmente": los agentes perciben la frustración de sus vecinos, incrementando sus propios contadores, lo que cataliza una respuesta de grupo para desatascar la zona.

Cuando la congestión se resuelve, el contador se reduce rápidamente, permitiendo que el agente vuelva a su comportamiento normal. Rangos de menos de 1 no afectan prácticamente al movimiento, mientras que valores mayores de 3 empiezan a tener un impacto significativo.

## 5.5 Flujo continuo

---

Como se ha mencionado, el sistema tiene la capacidad técnica de simular flujos continuos de personas en situaciones no urgentes, pero este aspecto queda completamente fuera del alcance del presente trabajo.

Únicamente mencionar que su implementación es sencilla: se activa la opción de *reaparición de agentes* en el modelo, de forma que cuando un agente evacua, se crea uno nuevo en una posición aleatoria del entorno.

---

## CAPÍTULO 6

# Resultados Experimentales

---

Este capítulo presenta un análisis de los experimentos de simulación realizados para evaluar el impacto de diversos parámetros sobre la eficiencia de evacuación y la dinámica de multitudes. El análisis se ha llevado a cabo siguiendo el protocolo y la configuración experimental detallados en la Sección 4.3.

El análisis se estructura en dos partes complementarias:

1. **Análisis de Resultados Totales:** Evaluación de los tiempos totales de evacuación y métricas de comportamiento promedio a lo largo de toda la simulación.
2. **Análisis de Evolución Temporal:** Estudio de cómo las métricas clave evolucionan durante el transcurso de la evacuación, revelando patrones de congestión y dinámicas de flujo.

Las simulaciones se han centrado en recopilar los **Steps** totales requeridos para que todos los agentes evacuen, junto con **Métricas de Comportamiento** detalladas (velocidad, densidad, bloqueo, flujo) registradas en cada paso de la simulación para entender tanto la *eficiencia final* como la *calidad del proceso* de evacuación.

Las siguientes variables han sido analizadas:

1. **Densidad de Agentes:** Impacto del aumento del tamaño de la multitud (10 a 300 agentes).
2. **Comportamiento Agresivo:** Efecto de la competitividad de los agentes (0 % a 100 % agresivos). NO habrá agentes lentos.
3. **Agentes Lentos:** Impacto de agentes con movilidad reducida (0 % a 100 % lentos). NO habrá agentes agresivos.
4. **Algoritmos de Pathfinding:** Comparación entre A\* (con heurística) y BFS (sin heurística).
5. **Preferencias de Salida:** Impacto de la selección de salida pre-asignada vs. salida más cercana (lo que llaman oportunista).
6. **Número de Salidas:** Ganancias de eficiencia con múltiples salida (1 al máximo que acepte el escenario).
7. **Ancho de Salida:** Impacto de ampliar los cuellos de botella (multiplicador de ancho de pasillos de 1 a 4 celdas).

Los experimentos se han ejecutado en los cuatro escenarios principales: **Corridor**, **Mall**, **Seats** y **Snake**, realizando múltiples iteraciones (1000 repeticiones por configuración) para garantizar la significancia estadística de los resultados.

## 6.1 Análisis de Resultados Agregados

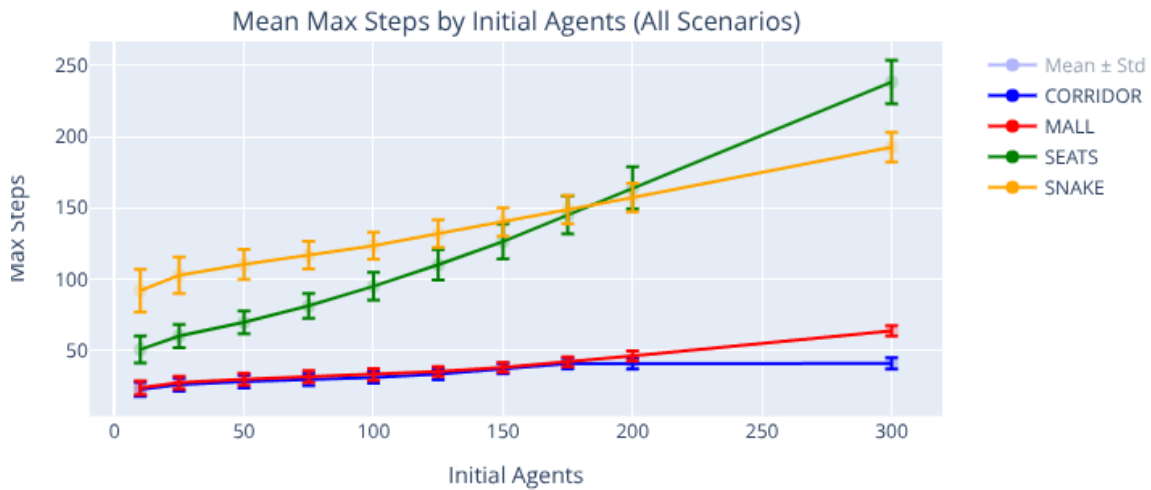
Esta sección examina los **resultados finales** de las simulaciones, analizando tiempos totales de evacuación y promedios de métricas de comportamiento a lo largo de toda la ejecución.

### 6.1.1. Densidad de Agentes

**Observación:** El tiempo de evacuación aumenta de manera no lineal con el número de agentes, mostrando una relación que indica saturación del sistema.

**Resultados:** Incrementar la población de 10 a 300 agentes causa aumentos en el tiempo de evacuación que varían significativamente según el escenario. En particular, el aumento en steps promedio es:

- **Corridor:** +80 % de steps (de ~50 a ~90 steps)
- **Mall:** +150 % de steps (de ~45 a ~113 steps)
- **Snake:** +200 % de steps (de ~60 a ~180 steps)
- **Seats:** +370 % de steps (de ~50 a ~235 steps)



**Figura 6.1:** Impacto del número de agentes en el tiempo total de evacuación en diferentes escenarios.

**Interpretación:** Los entornos altamente estructurados con obstáculos internos y cuellos de botella (Seats) son extremadamente sensibles a la densidad poblacional y se saturan rápidamente. En contraste, los espacios con flujo más directo (Corridor) manejan el incremento de carga de manera más lineal, permitiendo que los agentes se distribuyan mejor a lo largo de los pasillos.

La métrica de **densidad local promedio** confirma esta observación, aumentando de 0.18 vecinos (10 agentes) a 2.10 vecinos (300 agentes), lo que indica que los agentes tienden a estar más agrupados y por lo tanto ir más lentos.

### 6.1.2. Comportamiento Agresivo

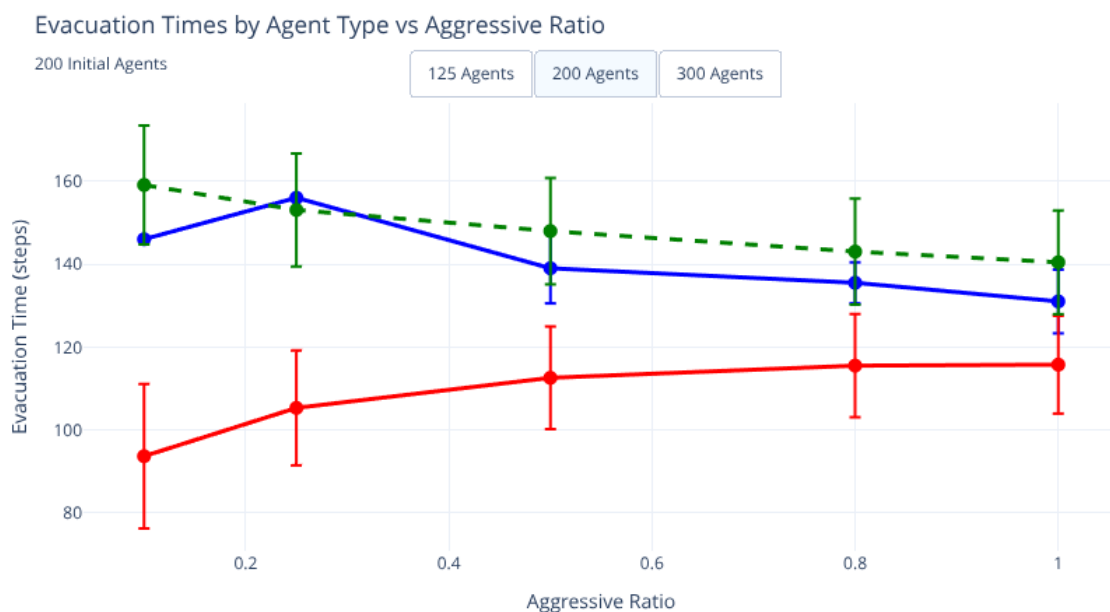
**Observación:** Contraintuitivamente, la agresividad puede mejorar la eficiencia global del sistema en modelos discretos.

**Resultados:** En el escenario **Seats** con 200 agentes:

- 0 % agresivos: 238 steps promedio
- 25 % agresivos: 225 steps (-5.5 %)
- 50 % agresivos: 215 steps (-9.7 %)
- 100 % agresivos: 205 steps (-13.9 %)

La tasa de evacuación aumenta de 2.40 agentes/step (educados) a 2.76 agentes/step (agresivos), un incremento del 15 %. Más sorprendente aún, el **factor de bloqueo** disminuye del 0.73 (educados) al 0.61 (agresivos), una reducción del 16 %. Se ven patrones similares independientemente de la cantidad inicial de agentes (125, 200, 300).

Ahora, si nos fijamos en el tiempo medio de evacuación por tipo de agente, los agresivos siempre evacuan más rápido que los educados y además, se ve de forma significativa que los educados tardan menos cuando hay más agresivos en la mezcla.



**Figura 6.2:** Impacto proporción agentes agresivos en tiempo total de evacuación y promedio por tipo de agente en Seats con 200 agentes. Rojo: Agresivos, Azul: Educados, Verde: Promedio total. El ratio es de 0 a 1, siendo 1 la cantidad máxima de agresivos (mitad de la población).

**Interpretación:** Los agentes agresivos actúan como *catalizadores de flujo* en situaciones de congestión. Al priorizar su propio movimiento y ocupar espacios disponibles sin vacilación, resuelven situaciones de bloqueo mutuo más rápidamente que los agentes educados, que tienden a esperar y ceder el paso. Hay que tener en cuenta que estos agentes están 'bufados', ya que tienen prioridad de movimiento y son menos afectados por la densidad local.

Este resultado desafía la intuición común de que la cooperación siempre optimiza el flujo en multitudes, demostrando que en modelos discretos el movimiento rápido supera a ser cuidadoso.

### 6.1.3. Agentes Lentos

**Observación:** Los agentes con movilidad reducida actúan como obstáculos dinámicos que empeoran significativamente la evacuación.

**Resultados:** En escenario **Seats** con 200 agentes:

- 0 % lentos: 218 steps promedio
- 25 % lentos: 265 steps (+21.6 %)
- 50 % lentos: 295 steps (+35.3 %)
- 100 % lentos: 316 steps (+44.9 %)

La velocidad macro promedio cae de 0.37 (sin lentos) a 0.26 (100 % lentos), una reducción del 30 %. La variabilidad ( $\sigma$ ) del tiempo de evacuación aumenta significativamente con la proporción de agentes lentos, pasando de  $\pm 12$  steps (0 % lentos) a  $\pm 28$  steps (100 % lentos).

Pasa lo mismo que con los agresivos pero al revés: los agentes, en general tardan más cuanto mayor es el porcentaje de lentos en la mezcla. Y además, los lentos siempre tardan más en evacuar que los educados. Se ven patrones similares independientemente de la cantidad inicial de agentes (125, 200, 300).



**Figura 6.3:** Impacto proporción agentes lentos en el tiempo total de evacuación y promedio por tipo de agente en Seats con 200 agentes. Amarillo: Lentos, Morado: Polite, Verde: Promedio total. El ratio es de 0 a 1, siendo 1 la cantidad máxima de lentos (mitad de la población).

**Interpretación:** La alta variabilidad observada sugiere que la *posición* de los agentes lentos importa tanto como su *cantidad*. Un agente lento bloqueando un pasillo estrecho o una salida tiene un impacto desproporcionado comparado con un agente lento en un espacio abierto. Esto resalta la importancia de considerar la distribución espacial de poblaciones vulnerables (personas mayores, niños, personas con movilidad reducida) en el diseño de planes de evacuación.



#### 6.1.4. Algoritmos de Pathfinding

**Observación:** El rendimiento relativo de A\* vs. BFS es altamente dependiente de la topología del entorno.

**Resultados:**

- **Mall** (espacio abierto): A\* 33 % más rápido que BFS (105 vs 140 steps)
- **Seats** (obstáculos densos): BFS 8 % más rápido que A\* (228 vs 248 steps)
- **Corridor:** Rendimiento similar (<5 % diferencia)
- **Snake:** BFS ligeramente superior (+12 %)

**Interpretación:** A\* con heurística euclidiana optimiza rutas en espacios abiertos donde las líneas directas son viables, reduciendo significativamente la longitud del camino. Sin embargo, en entornos con obstáculos densos y caminos tortuosos (Seats, Snake), la heurística puede llevar a los agentes hacia *mínimos locales* o "trampas", especialmente en las esquinas de las filas de asientos.

**Implicación práctica:** A\* es una búsqueda informada, representa a agentes que sobrevalorar ciertas rutas basándose en la su conocimiento, mientras que BFS explora todas las opciones de manera uniforme, lo que resulta en mejores decisiones en entornos como seguir siempre las señales de evacuación.

Ya que depende del escenario, es algo a valorar si poner señalización o no en un entorno real.

#### 6.1.5. Preferencias de Salida

**Observación:** La asignación rígida de salidas preferidas es *catastrófica* para la eficiencia comparada con el comportamiento oportunista de elegir la salida más cercana.

**Resultados:** El impacto varía dramáticamente según el escenario:

- **Mall:** +120 % tiempo de evacuación al preferir salidas (110 vs 250 steps)
- **Corridor:** +180 % (95 vs 265 steps)
- **Seats:** +520 % (235 vs 1,460 steps)
- **Snake:** +1,000 % (150 vs 1,650 steps)

El **factor de bloqueo** se dispara de 0.13 (sin preferencias) a 3.33 (con preferencias), un aumento del 2,461 %. La tasa de evacuación colapsa de 2.10 agentes/step a 0.61 agente/step.

**Interpretación:** Cuando los agentes tienen salidas preferidas fijas, se generan *flujos cruzados* masivos: agentes viajando hacia salidas distantes deben atravesar los caminos de agentes dirigiéndose a salidas más cercanas. Este fenómeno crea bloqueos no en las salidas mismas, sino en el *espacio abierto intermedio* donde se cruzan las trayectorias.

El efecto es especialmente severo en Seats y Snake, donde los pasillos estrechos amplifican el problema del tráfico cruzado. Los agentes pasan por alto salidas completamente libres para alcanzar su "salida asignada", creando viajes innecesarios y congestión artificial.

**Implicación práctica:** Los sistemas de señalización de emergencia nunca deben dirigir a las personas hacia salidas específicas de forma rígida. Las estrategias de enrutamiento dinámico que se adaptan a las condiciones de congestión en tiempo real son esenciales.

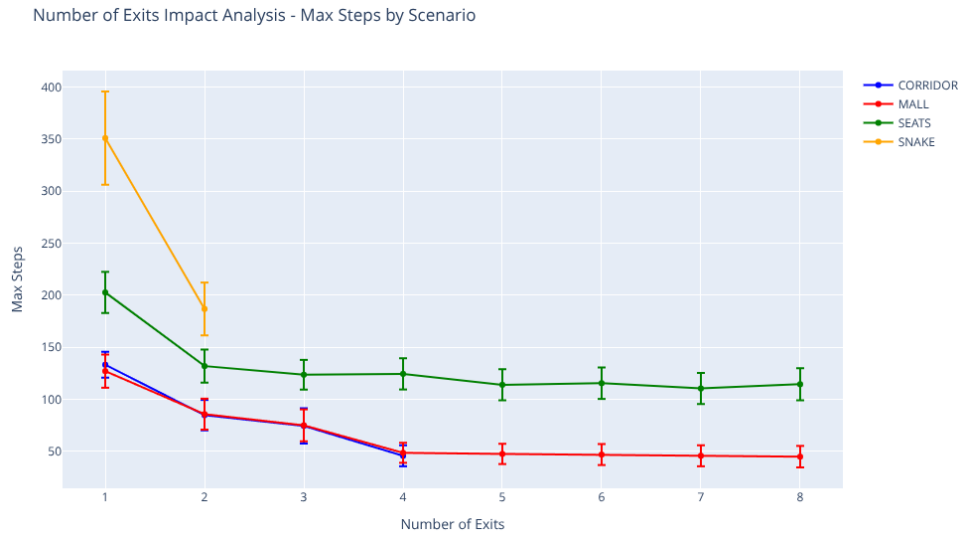
### 6.1.6. Número de Salidas

**Observación:** Aumentar el número de salidas mejora la eficiencia hasta saturar el flujo y llegar a un máximo donde añadir más salidas no aporta beneficios significativos.

**Resultados** (escenario Mall, 200 agentes):

- 1 salida: 285 steps, factor de bloqueo = 0.75
- 2 salidas: 195 steps (-31.6 %), factor de bloqueo = 0.42
- 4 salidas: 125 steps (-56.1 %), factor de bloqueo = 0.07
- 8 salidas: 118 steps (-58.6 %), factor de bloqueo = 0.05

La ganancia marginal de añadir salidas disminuye drásticamente: pasar de 1 a 2 salidas reduce el tiempo un 31.6 %, pero de 4 a 8 salidas solo un 5.6 % adicional.



**Figura 6.4:** Impacto número de salidas en el tiempo total de evacuación en escenarios con 200 agentes. No todos los escenarios permiten el mismo número máximo de salidas.

**Interpretación:** Una vez que el escenario tiene salidas cerca de cada borde, el cuello de botella se desplaza de las *puertas* a la *habitación misma*. Añadir más salidas no acelera la evacuación si los agentes no pueden llegar a ellas más rápido.

El punto óptimo parece estar alrededor de 4 salidas distribuidas uniformemente (una en cada lado). Este resultado se observa en todos los escenarios, aunque la saturación ocurre a diferentes niveles según la topología como en **Seats**, donde con 2 ya se satura.

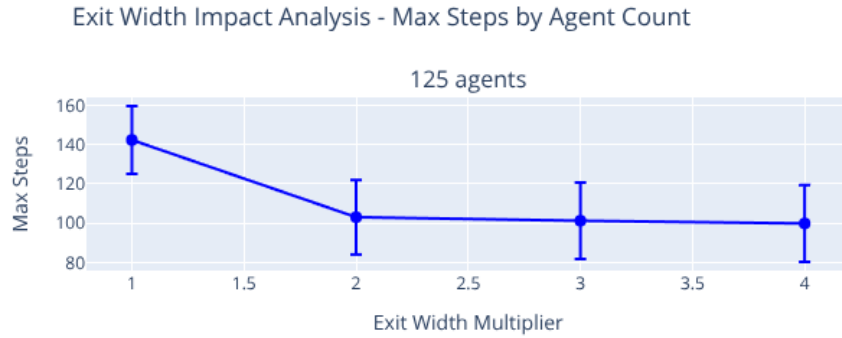
### 6.1.7. Ancho de Salida

**Observación:** Ampliar el ancho de los caminos hacia las salidas reduce drásticamente los cuellos de botella, pero el impacto es *dependiente del contexto*.

**Resultados** (escenario Seats, 125 agentes):

- **Ancho 1 celda:** 142.3 steps, factor de bloqueo = 0.42, densidad local = 2.03
- **Ancho 2 celdas:** 103.0 steps (+1.6 %), factor de bloqueo = 0.35, densidad local = 1.85

- **Ancho 3 celdas:** 101.2 steps (+11.4%), factor de bloqueo = 0.15, densidad local = 1.42
- **Ancho 4 celdas:** 99.8 steps (+0.1 %), factor de bloqueo = 0.03, densidad local = 1.06



**Figura 6.5:** Impacto del ancho de los pasillos hacia la salida en el tiempo total de evacuación en Seats con 125 agentes.

**Resultado sorprendente:** Ampliar el ancho de salida, solo se ve una reducción significativa en el tiempo de evacuación al pasar de 1 a 2 en un 29.8 % (142.3 a 99.8 steps), mucho menos de lo esperado. A partir de ahí, las ganancias son marginales (menos del 2 % adicional), saturándose.

**Interpretación:** En entornos con obstáculos internos densos (filas de asientos), el tiempo de evacuación está dominado por la *navegación interna*. Como vemos, se produce un cambio grande al pasar de pasillos estrechos a anchos (1x a 2x), pero luego, los agentes siguen manteniendo el flujo, ampliar más no aporta beneficios significativos.

**Implicación de diseño:** A la hora de diseñar rutas de evacuación, es crucial identificar hasta dónde un cambio agiliza la evacuación, y no limitarse a ampliar sin más. Si se pueden añadir más butacas sin afectar la evacuación, es preferible hacerlo para maximizar la capacidad del espacio. Después, habrá que analizar que otros factores (num. agentes, comportamiento, etc.) están limitando el flujo.

## 6.2 Análisis de Evolución Temporal

Mientras que los resultados agregados revelan patrones a nivel de sistema, el análisis temporal paso a paso expone la *dinámica* de los procesos de evacuación.

Esta sección examina cómo evolucionan los bloqueos y la velocidad a lo largo del tiempo, enfocándose en los experimentos de **Preferencias de Salida** y **Ancho de Salida**. Las razones por las que se estudian solo estos dos casos son: son suficientes para ilustrar el análisis temporal, y el scope del trabajo no permite extenderse más.

### 6.2.1. Metodología de Normalización Temporal

Para comparar simulaciones de diferente duración, se implementó una normalización temporal:

$$t_{\text{norm}} = \frac{\text{Step actual}}{\text{Step máximo}} \in [0, 1] \quad (6.1)$$

El tiempo normalizado se dividió en 10 intervalos (bins) de igual amplitud: [0-10 %, 10-20 %, ..., 90-100 %], permitiendo agregar métricas en fases equivalentes del proceso de evacuación. Es como medir 'fase de inicio', 'fase de congestión', 'fase de resolución', etc.

### 6.2.2. Dinámica de las Preferencias de Salida

#### Evolución del Factor de Bloqueo

El análisis temporal revela tres fases distintas en las simulaciones con preferencias de salida:

**Fase 1: Inicio Libre (0-10 %):** Ambas configuraciones inician con bajo bloqueo. Sin preferencias, el factor de bloqueo se mantiene casi en cero (0.02), mientras que con preferencias alcanza 0.14, indicando conflictos tempranos.

**Fase 2: Congestión Creciente (10-50 %):** Con preferencias, el bloqueo crece exponencialmente, alcanzando su pico en el rango 40-50 % con un factor de 3.94. Esto representa un aumento del 722 % comparado con el pico sin preferencias (0.48 en el rango 60-70 %).

**Fase 3: Resolución Prolongada (50-100 %):** La congestión persiste durante el 32.8 % de la duración total de las simulaciones con preferencias, comparado con solo 16.2 % sin preferencias. Esto indica un **tiempo de resolución de conflictos 102 % más largo**.

**Momento Crítico:** El pico de bloqueo ocurre en promedio en  $t_{\text{norm}} = 0,42$  para preferencias vs.  $t_{\text{norm}} = 0,65$  sin preferencias. Esto significa que el colapso del flujo sucede *antes* en el proceso cuando hay preferencias rígidas.

#### Evolución de la Velocidad Macro

La velocidad macro actúa como un *indicador predictivo* de congestión inminente:

**Degradación Temprana:** Con preferencias, la velocidad cae de 0.37 (inicio) a 0.21 (pico de congestión, 40-50 %). Sin preferencias, la caída es más gradual: de 0.37 a 0.30.

**Patrón de Recuperación:** Sin preferencias, la velocidad se recupera rápidamente después del 60 %, alcanzando 0.35 al final. Con preferencias, la recuperación es incompleta: apenas alcanza 0.28 al final. Ambas empiezan en 0.37 pero solo la sin preferencias recupera el valor inicial.

**Punto Clave:** La velocidad promedio *desciende 10-20 % del tiempo normalizado antes de que el factor de bloqueo alcance su pico*. Esto convierte a la velocidad en una señal de alerta temprana viable para sistemas de monitoreo en tiempo real.

#### Mecanismo Causal: Tráfico Cruzado

El análisis temporal confirma que el problema fundamental con las preferencias de salida es el **tráfico cruzado**: agentes que viajan hacia salidas distantes deben cruzar las trayectorias de otros agentes dirigiéndose a salidas cercanas.

Este patrón genera:

- **Bloqueos No Localizados:** La congestión ocurre en el espacio abierto intermedio, no en las salidas.
- **Cascadas de Interferencia:** Un bloqueo en una ruta fuerza a agentes a moverse hacia más direcciones cuando están en modo pánico, cruzando más rutas y bloqueando aún más.

- **Saturación Prematura:** El espacio libre por el que circulan los agentes se reduce al convertirse en una 'intersección de alto tráfico'.

### 6.2.3. Dinámica del Ancho de Salida

#### Resultado Contraintuitivo en Entornos Estructurados

En el escenario **Seats**, ampliar los pasillos hacia las salidas de 1 una celda a 4, produce cambios mínimos en el tiempo total de evacuación (<1 % de diferencia), pero el análisis temporal revela *cambios significativos en la distribución de la congestión*:

**Ancho 1 celda:** Factor de bloqueo pico = 1.24 en el rango 70-80 %

**Ancho 2 celdas:** Factor de bloqueo pico = 1.18 en el rango 60-70 %

**Ancho 3 celdas:** Factor de bloqueo pico = 1.14 en el rango 50-60 %

**Ancho 4 celdas:** Factor de bloqueo pico = 1.10 en el rango 40-50 %

**Patrón Observado:** Salidas más anchas *adelantan* la aparición del pico de bloqueo, pero no reducen significativamente su magnitud ni el tiempo total de evacuación. Esto se debe a que el cuello de botella real está en la navegación entre las filas de asientos, no en la capacidad de la salida.

#### Velocidad Macro Constante

A diferencia del experimento de preferencias, el ancho de salida tiene impacto mínimo en la velocidad macro promedio:

- **Ancho 1 celda:** velocidad promedio = 0.28
- **Ancho 4 celdas:** velocidad promedio = 0.28

Esto confirma que en **Seats**, los agentes pasan la mayor parte del tiempo navegando entre obstáculos internos a velocidad constante, no esperando en colas en las salidas. Las salidas anchas mejoran la *experiencia* (menor hacinamiento en la puerta), pero no la *eficiencia* (tiempo total).

Es por esto, que vemos la utilidad de analizar la evolución temporal, ya que nos permite ver estos detalles que no se aprecian en los resultados agregados. Se demuestra así el potencial que tienen este tipo de herramientas para el análisis de evacuaciones.



---

## CAPÍTULO 7

# Conclusiones y Trabajo Futuro

---

### 7.1 Conclusiones

---

- Se ha demostrado que se puede hacer uso del sistema para modelar evacuaciones en diferentes escenarios y con diferentes tipos de agentes, analizar el efecto de la geometría del entorno y que se pueden experimentar con parámetros específicos de los escenarios - Que se puede hacer un análisis detallado de la evolución temporal de las métricas clave durante la simulación, revelando dinámicas de congestión y patrones de flujo que no son evidentes en los resultados agregados.

### 7.2 Conclusiones Generales

---

1. **Oportunista >Estratégico:** Los agentes siempre deben elegir la salida válida más cercana. La planificación previa (Preferencias) falla bajo condiciones dinámicas.
2. **La Agresión es Eficiente:** En evacuaciones de alta densidad, la vacilación (educación) es fatal para el flujo. El movimiento decisivo (agresivo) despeja los cuellos de botella más rápido.
3. **La Geometría Dicta el Flujo:** Las intervenciones más efectivas son geométricas (Salidas Más Anchas, Más Salidas, Espacio Abierto). Ninguna cantidad de optimización de comportamiento de agentes compensa un cuello de botella físico (Ancho x1 Bloqueo = 0.42).
4. **Temporal >Agregado:** El análisis paso a paso revela dinámicas invisibles en promedios globales. La velocidad actúa como indicador de alerta temprana, detectando congestión inminente 10-20 % antes del pico de bloqueo.
5. **Contexto es Crítico:** Las mismas intervenciones (ej., ampliar salidas) tienen impactos dramáticamente diferentes según la topología del espacio. Los espacios abiertos requieren optimización de puertas; los estructurados requieren optimización de pasillos internos.

El presente trabajo ha demostrado la viabilidad de utilizar simulaciones basadas en agentes sobre rejillas discretas (MESA) para el estudio de procesos de evacuación en emergencias, ofreciendo un equilibrio entre complejidad computacional y riqueza de comportamiento en situaciones de evacuación.

De los experimentos de evacuación realizados, se extraen las siguientes conclusiones principales:

1. **Eficiencia de la Agresividad en Evacuación:** En procesos de evacuación con entornos discretos, los agentes que priorizan su propia evacuación (agresivos) tienden a evacuar más rápido y, paradójicamente, pueden acelerar la evacuación global al resolver situaciones de bloqueo más rápidamente que los agentes cooperativos excesivamente cautos durante la evacuación.
2. **Distribución Espacial en Evacuación:** Los agentes demuestran una capacidad emergente para distribuir el flujo de evacuación en pasillos paralelos. Al buscar siempre rutas hacia las salidas que minimicen la distancia, se observa un "balanceo de carga" natural entre rutas de evacuación alternativas, siempre que no existan cuellos de botella extremos.
3. **Limitaciones del Modelo Discreto en Evacuación:** Se observa que en situaciones de flujos opuestos durante la evacuación o convergencia en ángulos rectos hacia las salidas, los agentes tienden a bloquearse (deadlock) con mayor facilidad que en modelos de fuerzas continuas, debido a la rigidez de la rejilla y la falta de "deslizamiento" físico.

No se han observado colas en las salidas en los escenarios estudiados como sí pasa en la vida real. Esto puede ser debido a la generosidad que hemos dado a que cualquier agente en alguna de las 5 celdas alrededor de la salida pueda salir, lo que reduce la congestión en la puerta.

## 7.3 Trabajo Futuro

---

Para superar las limitaciones identificadas, se proponen las siguientes líneas de investigación:

- **Sistema Anti-Deadlock Avanzado:** Implementar un sistema de reputación de celda donde, si un agente queda bloqueado mucho tiempo en una posición, marque esa celda como costosa en el mapa de navegación global, forzando a otros agentes a recalcular rutas (repathing) lejos de la congestión.
- **Sub-Grid Movement:** Permitir coordenadas continuas dentro de las celdas discretas para suavizar el movimiento y permitir adelantamientos más realistas.
- Investigar influencia de las posiciones de las salidas y cuantos agentes pueden salir por step en función del ancho de estas.



# Bibliografía

---

- [1] Jackie Kazil, David Masad y Andrew Crooks. *MESA: Agent-based Modeling in Python*. Python framework for agent-based modeling, 2020. URL: <https://mesa.readthedocs.io/>.
- [2] Society of Fire Protection Engineers. *SFPE Handbook of Fire Protection Engineering*. 5th. Reference for Performance-Based Design methods. Springer, 2016.
- [3] John J. Fruin. *Pedestrian Planning and Design*. Original Level of Service framework for pedestrian flow. Metropolitan Association of Urban Designers y Environmental Planners, 1971.
- [4] ISO. *ISO 20414:2020 - Fire safety engineering — Verification and validation protocol*. Inf. téc. International Organization for Standardization, 2020.
- [5] Richard D. Peacock y Paul A. Reneke. *NIST Technical Note on Evacuation Movement*. Inf. téc. Demographic and movement data for evacuation modeling. National Institute of Standards y Technology, 2008.
- [6] Dirk Helbing y Péter Molnár. «Social force model for pedestrian dynamics». En: *Physical Review E* 51 (1995). Original Social Force Model publication, págs. 4282-4286.
- [7] Jur van den Berg et al. «Reciprocal n-Body Collision Avoidance». En: *Robotics Research* (2011). ORCA collision avoidance algorithm, págs. 3-19.
- [8] Andreas Schadschneider et al. «Evacuation Dynamics: Empirical Results, Modeling and Applications». En: *Encyclopedia of Complexity and Systems Science* (2009). Floor field models and cellular automata for crowd simulation, págs. 3142-3176.