

En las conferencias de la asignatura, abordamos temas como el proceso de levantamiento de requerimientos, la planificación inicial y las iteraciones, enfatizando la importancia de entregar algo funcional aunque no esté perfecto, así como la retroalimentación con el cliente. Aprendimos a organizar el proceso de recopilación de información sobre un tema a partir de investigaciones y papers, y a crear una tabla con esta información. También tuvimos la oportunidad de escuchar a profesionales de diferentes ramas que compartieron sus experiencias personales y consejos sobre la gestión de proyectos.

Inicialmente, el cliente quería un sistema que permitiera la simulación de la propagación de una enfermedad infecciosa de la manera más cercana posible a la realidad, en contraste con los modelos epidemiológicos existentes. En la primera iteración, se decidió crear un modelo basado en agentes donde cada agente representa a una persona, haciendo así más realista el comportamiento y luego compararlo con los resultados dados por el modelo SIR, este último ampliamente utilizado en epidemiología. En la segunda iteración, se planea agregar al modelo un cierto grado de inteligencia que lo acerque al comportamiento humano, mediante la implementación de inteligencia artificial con varias estrategias que dictarán el comportamiento de la persona dentro del entorno.

¿Que tema trataré en mi proyecto?

El objetivo de este proyecto es simular cómo se propaga una enfermedad infecciosa utilizando un **modelo basado en agentes (ABM del inglés Agent Based Model)**. Luego examinaremos cómo cambiando los valores de los parámetros en el modelo se afecta el resultado final y los compararemos con los obtenidos por el **modelo SIR**, modelo ampliamente utilizado para simular este fenómeno.

¿Qué es una Simulación Basada en Agentes?

Una simulación basada en agentes (**ABS**) es un enfoque de modelado computacional que simula las acciones e interacciones de individuos autónomos (llamados "agentes") para evaluar sus efectos en el sistema en su conjunto. Este tipo de simulación es especialmente útil para estudiar sistemas complejos donde el comportamiento emergente resulta de las interacciones de muchos agentes individuales.

¿Qué es una Simulación Basada en Agentes Inteligentes?

Una **Simulación Basada en Agentes Inteligentes (Intelligent Agent-Based Simulation, IABS)** es una extensión avanzada de la Simulación Basada en Agentes (ABS) tradicional. En una IABS, los agentes no solo siguen reglas estáticas predefinidas, sino que también incorporan capacidades de inteligencia artificial (IA) para aprender, adaptarse y tomar decisiones óptimas en respuesta a su entorno cambiante.

El modelo SIR

El **modelo SIR** utiliza un sistema de ecuaciones diferenciales generadas a partir de un modelo compartimental para determinar los cambios en el número de individuos **susceptibles**, **infectados** y **recuperados** a lo largo del tiempo.

El modelo se basa en tres compartimentos:

-(S): Número de individuos susceptibles.

-(I): Número de individuos infectados.

-(R): Número de individuos recuperados.

Las ecuaciones diferenciales del modelo SIR son las siguientes:

$$\frac{dS}{dt} = -\beta SI$$

$$\frac{dI}{dt} = \beta SI - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

Donde: β es la tasa de transmisión del contacto entre susceptibles e infectados. γ es la tasa de recuperación, es decir, la proporción de infectados que se recuperan por unidad de tiempo.

Estas ecuaciones representan la dinámica del modelo SIR, describiendo cómo las poblaciones de susceptibles (S), infectados (I) y recuperados (R) cambian con el tiempo.

Sin embargo, el **modelo SIR** tiene varias limitaciones fundamentales. Supone que la población es lo suficientemente grande como para que todas las curvas y tasas puedan considerarse continuas en lugar de discretas. Para una enfermedad que tiene muy pocos infectados o que solo está presente en una población pequeña, esta suposición puede ser falsa.

Además, el modelo asume que las tasas en cualquier punto en el tiempo son proporcionales al número de personas infectadas o susceptibles en ese mismo punto en el tiempo, cuando en realidad, típicamente hay un retraso. Por ejemplo, si en un momento particular muchas personas están infectadas, solo esperaríamos ver una alta tasa de recuperación varios días/semanas después, ya que lleva tiempo que una persona se recupere.

En este proyecto, intentamos codificar un modelo que no dependa de estas suposiciones. Específicamente, utilizamos un **modelo basado en agentes (ABM)**, donde cada agente representa a una persona individual. Esto permite que la transmisión de la enfermedad se modele de forma discreta, de modo que solo números enteros de personas se muevan entre grupos. Del mismo modo, los agentes individuales pueden rastrear variables relacionadas con ellos mismos, como el tiempo que han tenido la enfermedad, qué tan cerca están de individuos infectados o cuánto tiempo han estado expuestos. Esto permite que las reglas que dictan la transmisión de enfermedades en el **ABM** incorporen más variables (incluidos los retrasos en las tasas mencionadas anteriormente).

Modelo Basado en Agentes

En este modelo de simulación, inspirado en el **modelo SIR**, los agentes se dividen en tres categorías: **susceptibles**, **infectados** y **recuperados**. Los **susceptibles** pueden contagiarse al interactuar con **infectados**, mientras que los **recuperados** pueden o no volver a enfermarse en dependencia del tipo de virus que se simule. También, en dependencia de la enfermedad a simular, pueden haber **fallecidos**, que son agentes que no se recuperan y fallecen a causa de la enfermedad. Los agentes se desplazan aleatoriamente en una cuadrícula que cuando un agente sale de un borde, reaparece en el lado opuesto.

Exposición y Probabilidad de Infección

Para determinar cuán probable es que un agente contraiga la enfermedad, definimos un atributo llamado exposición. Cuanto mayor sea la exposición de un agente susceptible

en cualquier momento, mayor será la probabilidad de que el agente contraiga la enfermedad en ese momento. La exposición de un agente susceptible aumenta cuando está cerca de un individuo infectado. Específicamente, la cantidad en que aumenta su exposición está dictada por dos variables: cuánto tiempo ha estado infectado el individuo infectado y cuán cerca están los dos agentes.

Dependiendo de cuánto tiempo ha estado infectado el individuo infectado, su infectividad cambia. Para muchas enfermedades, la infectividad sigue una distribución normal sesgada a la derecha. Una distribución normal sesgada a la derecha (también conocida como distribución normal sesgada positiva) es una modificación de la distribución normal que introduce asimetría, desplazando la mayoría de los valores hacia la izquierda con una cola más larga hacia la derecha. Específicamente, la infectividad es baja cuando se infectan por primera vez, aumenta rápidamente a un máximo y luego disminuye lentamente hasta que se recuperan. Definimos una función para representar esto llamada función de infectividad, que toma el tiempo desde que el agente se infectó por primera vez como entrada y devuelve la cantidad de exposición que "da" a los agentes susceptibles cercanos.

- **Exposición** : Es la medida de cuánto riesgo tiene un agente **susceptible** de contraer la enfermedad. Aumenta cuando está cerca de un **infectado**. Es el producto de la función entre la función de infectividad y la de distancia.
- **Infectividad** : Es la capacidad de un **infectado** para contagiar. Depende del tiempo que ha estado infectado y sigue una **distribución normal sesgada o asimétrica**.
- **Función de distancia** : Reduce o aumenta la exposición según la distancia entre el **infectado** y el **susceptible** . Usa una **curva logística generalizada de Tipo I** para calcular la proporción de exposición.
- La **probabilidad de infección** se determina aplicando otra curva logística a la exposición recibida.

Recuperación y Aleatoriedad

- Un agente **infectado** se considera **recuperado** cuando su infectividad cae por debajo de 1.
- El modelo incluye variaciones aleatorias en los parámetros de las funciones para simular diferencias individuales, como el estado inmunológico.

Distribución Normal Sesgada

La ecuación de infectividad que utilizamos en nuestro modelo :

$$\frac{a}{s\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-m}{s}\right)^2} \left(1 + \operatorname{erf}\left(\frac{k\left(\frac{x-m}{s}\right)}{\sqrt{2}}\right)\right)$$

Aquí, cada parámetro tiene un significado específico:

- (x): La variable aleatoria.
- (a): Un factor de escala que normaliza la función.
- (m) (mu): La media de la distribución.
- (s) (sigma): La desviación estándar de la distribución.
- (k): El parámetro de asimetría .
- (erf): La función de error, definida como:

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

La función de error, erf , se utiliza para ajustar la asimetría de la distribución.

Dependiendo del valor de (k), la distribución se puede sesgar hacia la derecha (si ($k > 0$)) o hacia la izquierda (si ($k < 0$)).

Es una extensión de la distribución normal que incorpora asimetría en la distribución, permitiendo que sea sesgada hacia la derecha o la izquierda. Esta distribución es útil en la modelización de datos donde la simetría no es una suposición razonable

Desgloce de la función:

1. Parte Gaussiana Estándar:

$$\frac{a}{s\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-m}{s}\right)^2}$$

Esta parte corresponde a la función de densidad de probabilidad de una distribución normal estándar con media (m) y desviación estándar (s), ajustada por el factor de escala (a).

1. Parte de Asimetría:

$$1 + \operatorname{erf}\left(\frac{k\left(\frac{x-m}{s}\right)}{\sqrt{2}}\right)$$

Esta parte introduce la asimetría. La función de error (erf) modula la probabilidad según el parámetro de asimetría (k).

Propiedades

- **Asimetría:** Permite modelar distribuciones con colas de diferentes longitudes.
- **Flexibilidad:** Puede ajustarse a formas de datos no simétricas.
- **Aplicaciones:** Utilizada en finanzas, ingeniería y otras áreas con datos no simétricos.

Función Logística

Una **función logística** es un tipo de función sigmoide que se utiliza para modelar situaciones de crecimiento que empiezan con un crecimiento exponencial, pero luego se estabilizan a medida que alcanzan una capacidad máxima o límite. Esta función tiene aplicaciones en diversas áreas, como la biología, la estadística, la economía y la inteligencia artificial.

En nuestro caso utilizaremos la de **tipo 1** que es una forma más específica y estandarizada de la función logística general que se utiliza comúnmente en el análisis de crecimiento biológico, modelado de difusión tecnológica y otros campos donde se observa crecimiento sigmoidal.

Definición Matemática

La función logística de tipo 1 está definida por la siguiente fórmula:

$$L(x) = \frac{1}{1 + e^{-k(x-x_0)}}$$

Aquí:

- (x) es la variable independiente.
- (k) es la tasa de crecimiento de la función.
- (x_0) es el valor en x en el cual la función alcanza el punto medio (0.5).

Propiedades

Rango de Valores:

- La función mapea cualquier valor real (x) al intervalo $(0, 1)$, lo que la hace adecuada para modelar proporciones o probabilidades.

Simetría:

- La curva es simétrica respecto al punto $(x = x_0)$.

Crecimiento Sigmoidal:

- Tiene un crecimiento lento en sus extremos y más rápido en la región central cerca de $(x = x_0)$.

Asintóticas:

- Se aproxima asintóticamente a 0 cuando $x \rightarrow -\infty$ y a 1 cuando $x \rightarrow \infty$

Punto de Inflexión:

- La tasa de crecimiento de la función es máxima en $(x = x_0)$.

Interpretación de los Parámetros

- (k) : Controla la pendiente de la curva en el punto $(x = x_0)$. Un valor mayor de (k) produce una transición más abrupta entre los valores bajos y altos de la función.
- (x_0) : Desplaza la curva a lo largo del eje (x) . Es el valor de (x) donde la función alcanza el 50% de su valor máximo.

Simulación

La simulación modela la propagación de una enfermedad infecciosa en una población con agentes móviles. Los agentes se dividen en diferentes grupos con estados específicos: susceptibles, infectados, recuperados y, opcionalmente, muertos. La simulación visualiza cómo cambian estos estados a lo largo del tiempo y cómo interactúan los agentes en un entorno bidimensional.

Detalles de la Simulación

Inicialización del Entorno:

- Se crea un entorno donde se distribuyen los agentes. Los agentes pueden moverse y cambiar su estado de salud con el tiempo.
- Se establece un agente infectado inicialmente para iniciar la propagación de la enfermedad.
- Los parámetros como la inmunidad permanente, la disminución de la exposición, y la posibilidad de muerte se configuran al inicio.

Estados de los Agentes:

- **Susceptibles** (S) : Agentes que pueden contraer la enfermedad.
- **Infectados** (I) : Agentes que han contraído la enfermedad y pueden transmitirla.

- **Recuperados (R)**: Agentes que se han recuperado y, dependiendo de la configuración, pueden o no volver a ser susceptibles.
- **Muertos (D)**: Agentes que han muerto debido a la enfermedad.

Ciclo de Simulación:

- La simulación avanza en pasos de tiempo donde se actualizan las posiciones de los agentes y se evalúan sus estados.
- En cada paso, se determina si los agentes susceptibles se infectan, si los infectados se recuperan o mueren, y se ajustan sus exposiciones e infectividades.
- Los agentes se mueven aleatoriamente dentro del entorno.

Parámetros del Modelo

- **Inmunidad Permanente**: Define si los agentes recuperados pueden volver a ser susceptibles.
- **Disminución de la Exposición**: La tasa a la que la exposición de los agentes disminuye con el tiempo.
- **Muerte**: Si los agentes pueden morir debido a la enfermedad.
- **Tiempo de Inmunidad**: El tiempo durante el cual los agentes recuperados permanecen inmunes antes de volverse susceptibles nuevamente (si no hay inmunidad permanente).
- **Distribuciones de Infectividad**: Controlan cómo varía la capacidad de los agentes para infectar a otros a lo largo del tiempo.

Flujo de la Simulación

1. **Iniciación**: Se configura el entorno y los agentes, estableciendo el estado inicial.
2. **Evolución Temporal**: En cada paso de tiempo:
 - Los agentes se mueven aleatoriamente.
 - Se evalúa la probabilidad de que los susceptibles se infecten al entrar en contacto con infectados.
 - Se actualizan los estados de los infectados, ya sea recuperándose o muriendo.
 - Se ajustan las propiedades como la exposición e infectividad.
3. **Visualización y Estadísticas**: Se actualizan y muestran las gráficas y estadísticas que permiten observar la evolución de la epidemia.

Clase **Person**

Propósito

La clase **Person** representa a un individuo en la simulación. Cada instancia de esta clase es un agente que puede moverse y cambiar de estado de salud (susceptible, infectado, recuperado o muerto). Sus responsabilidades principales incluyen:

Estado de Salud:

- Maneja el estado de salud de una persona, que puede ser:
 - Susceptible (S)
 - Infectado (I)
 - Recuperado (R)
 - Muerto (D)
- El estado de cada persona determina cómo interactúa con otros agentes y cómo puede cambiar a lo largo del tiempo.

Infectividad:

- La clase calcula la capacidad de una persona infectada para contagiar a otros (**infectivity**). Esta infectividad cambia con el tiempo, generalmente aumentando hasta un pico y luego disminuyendo.
- La fórmula de infectividad se basa en parámetros que pueden ajustarse para modelar diferentes características de la enfermedad.

Probabilidad de Infección:

- Para las personas susceptibles, la clase determina la probabilidad de que se infecten cuando entran en contacto con personas infectadas.
- Esta probabilidad depende tanto de la exposición de la persona susceptible como de la infectividad de la persona infectada.

Movimiento:

- Cada persona puede moverse aleatoriamente dentro del entorno simulado, lo que modela el desplazamiento en un espacio bidimensional.
- El movimiento puede ser restringido o alterado en función de ciertos parámetros, como la velocidad.

Tiempo de Infección y Recuperación:

- La clase también maneja el tiempo desde la infección (`caught_time`) y, si la recuperación no es permanente, el tiempo de inmunidad (`immunity_time`).

1. Gestión de Agentes :

- Mantiene un registro de todas las personas en la simulación, categorizadas por su estado de salud (S, I, R, D).
- Permite la adición de nuevas personas y la actualización de su estado a lo largo del tiempo.

1. Simulación del Tiempo :

- Gestiona el paso del tiempo en la simulación y controla cómo las interacciones entre los agentes evolucionan con cada unidad de tiempo.
- Avanza el tiempo mediante la función `advance_time`, que actualiza las posiciones de los agentes y evalúa las interacciones (como contactos e infecciones).

Movimiento de Agentes :

- Simula el movimiento aleatorio de los agentes en el entorno, respetando los límites del espacio definido.
- El movimiento se modela como un desplazamiento aleatorio en un espacio bidimensional, con restricciones basadas en la velocidad máxima permitida.

1. Evaluación de Contactos e Infecciones :

- Evalúa los contactos entre los agentes para determinar si los susceptibles se infectan, considerando su proximidad y la probabilidad de infección.
- Calcula las nuevas infecciones basándose en las distancias entre agentes y las probabilidades de infección derivadas de la exposición y la infectividad.

Actualización de Estados de Salud :

- Actualiza el estado de los agentes a medida que avanzan en el tiempo, manejando transiciones como de infectado a recuperado o muerto.

Parámetros de Simulación :

- Permite la configuración de parámetros clave como la velocidad de los agentes, el decremento de exposición y la duración de la inmunidad.
- Estos parámetros pueden ajustarse para modelar diferentes escenarios de propagación de la enfermedad.

- `advance_time` : Maneja el avance del tiempo en la simulación, moviendo a los agentes y evaluando los contactos e infecciones.
- `move_all` : Controla el movimiento aleatorio de los agentes dentro del entorno.

Para un funcionamiento más realista de el modelo he agregado inteligencia a los agentes atribuyendole comportamientos parecidos a los del ser humano.

Estrategias de la IA

Se han definido tres estrategias principales para los agentes inteligentes (IA) que modelan la propagación de un virus: "random", "social" y "awareness". Cada estrategia tiene un enfoque diferente en términos de cómo los agentes toman decisiones y se comportan en el entorno simulado. Aquí hay una descripción de cada una de las estrategias:

Estrategia "Random":

En esta estrategia, los agentes toman decisiones de forma aleatoria sin considerar factores específicos del entorno o de otros agentes.

La dirección y velocidad de movimiento de los agentes son determinadas al azar, lo que puede resultar en interacciones impredecibles con otros individuos.

Esta estrategia puede simular comportamientos caóticos o poco predecibles en la propagación del virus.

Estrategia "Social":

En esta estrategia, los agentes evitan el contacto con otras personas ajustando su dirección y velocidad basándose en la presencia y comportamiento de otros individuos cercanos.

Se espera que esta estrategia refleje comportamientos más realistas relacionados con la interacción social en la propagación del virus.

Estrategia "Awareness":

En esta estrategia, los agentes tienen conciencia del riesgo asociado con la propagación del virus y ajustan su comportamiento en consecuencia.

La conciencia del riesgo puede influir en las decisiones de movimiento de los agentes, priorizando acciones más cautelosas cuando perciben un mayor peligro.

Esta estrategia simula un escenario donde los individuos son conscientes de las implicaciones del virus y actúan para minimizar su exposición y propagación.

Cada estrategia tiene sus propias ventajas y desventajas en términos de modelar diferentes aspectos del comportamiento humano durante una pandemia. Al implementar estas estrategias en la simulación con agentes inteligentes, se pueden explorar diversos escenarios y analizar cómo afectan a la propagación del virus y a las medidas preventivas tomadas por los individuos.

Función de Conciencia

La función de conciencia es un componente clave en la simulación, diseñada para modelar cómo los agentes perciben y responden al riesgo de propagación del virus en su entorno. La conciencia afecta las decisiones de los agentes, determinando si se comportan de manera más cautelosa o aleatoria.

Bases Matemáticas

La **función de percepción del riesgo** con una distribución logarítmica modificada se utiliza principalmente en modelos de simulación epidemiológica y comportamental para capturar cómo los individuos o agentes en una población ajustan su comportamiento en respuesta a la propagación de una enfermedad ya que simula comportamientos como

Distanciamiento Social : A medida que aumenta la percepción del riesgo, los individuos pueden optar por mantener una mayor distancia entre ellos y otros, reduciendo la probabilidad de contagio.

Uso de Mascarillas y Otros Equipos de Protección : Una mayor percepción del riesgo puede llevar a un aumento en el uso de medidas de protección personal.

Decisiones de Movilidad : Los individuos pueden modificar su patrón de movilidad, evitando áreas concurridas o reduciendo su interacción social.

Formula

$$\text{conciencia} = \text{maximo} \times \frac{\log(1 + ((\text{infestados} + \text{fallecidos}) * \text{riesgo/población}) \times \text{ajuste})}{\log(1 + \text{ajuste})}$$

Componentes Clave

1. **Percepción del Riesgo (risk_perception):** Este factor multiplica el número combinado de individuos infectados y fallecidos para ajustar su impacto en la conciencia.
2. **Valor Combinado:** Calculado como $(\text{infectados} + \text{fallecidos}) * \text{riesgo_perception}$.
3. **Valor Escalado:** Este valor combinado se divide por el total de la población para obtener una medida relativa.
4. **Transformación Logarítmica Normalizada:** La transformación logarítmica aplicada al valor escalado permite modelar un crecimiento inicial rápido de la conciencia que se estabiliza con altos números de casos.
5. **Máximo de Conciencia (max_awareness):** Define el valor máximo que puede alcanzar la conciencia.

Pasos en la Función awareness_func

1. **Cálculo de la Población Total:** Se considera el número total de individuos en diferentes estados (infectados, fallecidos, susceptibles, etc.).
2. **Cálculo del Valor Combinado:** $(\text{infectados} + \text{fallecidos}) * \text{riesgo_perception}$.
3. **Escalado del Valor Combinado:** Dividir el valor combinado por la población total.
4. **Aplicación de la Transformación Logarítmica Normalizada:** Utilizar la fórmula $\log(1 + x)$ escalada y limitada por max_awareness para obtener el nivel de conciencia.
5. **Ajuste del Nivel de Conciencia:** Asegurar que el nivel de conciencia no exceda el valor máximo definido.

AI Controller

La clase AIController se encarga de controlar el comportamiento de los agentes inteligentes en la simulación. Los parámetros de inicialización incluyen el número de personas, el radio de interacción, la forma del entorno, la estrategia de comportamiento (que puede ser "random", "social" o "awareness"), entre otros. Cada agente tiene cierto nivel de aleatoriedad en sus decisiones, tolerancia a la proximidad de otras personas, conciencia del riesgo, entre otros atributos.

Propósito General de la Clase **AIController**

- **Controlar Estrategias de Movimiento:** Determina cómo se moverán los agentes en el entorno (aleatoriamente, siguiendo reglas sociales o con conciencia del riesgo).
- **Gestionar Decisiones Relacionadas con la Enfermedad:** Incluye la toma de decisiones sobre evitar a otros agentes, ajustarse a la proximidad de otros y percibir el riesgo.

Métodos y Funciones de la Clase

__init__

- **Propósito:** Inicializar los parámetros del controlador de IA.
- **Parámetros:**
 - **people:** Diccionario que contiene grupos de personas en diferentes estados de salud.
 - **radius:** Radio dentro del cual los agentes pueden detectar a otros.
 - **shape:** Forma (dimensiones) del entorno.
 - **strategy:** Estrategia de movimiento y decisiones (puede ser "random", "social", "awareness", "neural").
 - **randomness:** Nivel de aleatoriedad en las decisiones.
 - **tolerance:** Tolerancia a la proximidad de otras personas.
 - **awareness:** Nivel de conciencia del riesgo.
 - **risk_perception:** Percepción del riesgo.
 - **adjustment_rate:** Tasa de ajuste de la conciencia.
 - **max_awareness:** Nivel máximo de conciencia.

change_strategy

- **Propósito:** Cambiar la estrategia de movimiento y decisiones.
- **Parámetros:**
 - **strategy:** Nueva estrategia a utilizar.

initial_angle

- **Propósito:** Determinar el ángulo inicial de movimiento de los agentes.
- **Retorna:** Un valor aleatorio entre 0 y 2π .

decide_angle

- **Propósito:** Decidir el ángulo de movimiento de un agente basado en la estrategia.
- **Parámetros:**
 - **current_person:** El agente para el cual se está decidiendo el ángulo de movimiento.
- **Retorna:** Un ángulo de movimiento calculado según la estrategia.

decide_angle_cautious

- **Propósito:** Decidir un ángulo de movimiento cauteloso para evitar la proximidad de otros agentes.
- **Parámetros:**
 - **current_person:** El agente para el cual se está decidiendo el ángulo.
- **Retorna:** Un ángulo que evita a los agentes cercanos.

decide_angle_random

- **Propósito:** Decidir un ángulo de movimiento aleatorio.
- **Parámetros:**
 - **current_person:** El agente para el cual se está decidiendo el ángulo.
- **Retorna:** Un ángulo aleatorio modificado por una desviación aleatoria.

decide_speed

- **Propósito:** Decidir la velocidad de movimiento de un agente.
- **Parámetros:**
 - **current_person:** El agente para el cual se está decidiendo la velocidad.
- **Retorna:** La velocidad del agente (actualmente no varía con la estrategia).

find_nearby_people

- **Propósito:** Encontrar personas cercanas a un agente dentro de un radio específico.
- **Parámetros:**
 - **current_person:** El agente para el cual se están buscando personas cercanas.
- **Retorna:** Una lista de agentes cercanos.

calculate_avoidance_vector

- **Propósito:** Calcular un vector de evitación para alejarse de los agentes cercanos.
- **Parámetros:**
 - **current_person:** El agente que desea evitar a otros.

- **nearby_people**: Lista de agentes cercanos.
- **Retorna**: Un vector que indica la dirección para evitar a otros agentes.

calculate_avoidance_vector_closer

- **Propósito**: Calcular un vector de evitación basado en el agente más cercano.
- **Parámetros**:
 - **current_person**: El agente que desea evitar a otros.
 - **nearby_people**: Lista de agentes cercanos.
- **Retorna**: Un vector que indica la dirección para evitar al agente más cercano.

can_die

- **Propósito**: Decidir si un agente puede morir basado en una probabilidad.
- **Parámetros**:
 - **current_person**: El agente para el cual se está decidiendo.
- **Retorna**: **True** si el agente muere, **False** de lo contrario.

can_sick

- **Propósito**: Decidir si un agente se enferma basado en su nivel de exposición.
- **Parámetros**:
 - **current_person**: El agente para el cual se está decidiendo.
- **Retorna**: **True** si el agente se enferma, **False** de lo contrario.

can_increase_exposure

- **Propósito**: Decidir si un agente aumenta su nivel de exposición.
- **Parámetros**:
 - **current_person**: El agente para el cual se está decidiendo.
- **Retorna**: **True** si el agente aumenta su exposición, **False** de lo contrario.

awareness_func

- **Propósito**: Calcular el nivel de conciencia del riesgo basado en la proporción de infectados y muertos en la población.
- **Retorna**: Un valor de conciencia normalizado.