

---

# INFORME DEL PROYECTO DE SIMULACIÓN – AGENTES

---

## Generales del Estudiante

Estudiante: Gelin Eguinosa Rosique  
Grupo: C-411  
Correo: [eguinosa@gmail.com](mailto:eguinosa@gmail.com)  
Móvil: +535-869-4330

## Orden del Problema Asignado

### 1. Marco General

El ambiente en el cual intervienen los agentes es discreto y tiene la forma de un rectángulo de  $N \times M$ . El ambiente es de información completa, por tanto, todos los agentes conocen toda la información sobre el agente. El ambiente puede variar aleatoriamente cada  $t$  unidades de tiempo. El valor de  $t$  es conocido.

Las acciones que realizan los agentes ocurren por turnos. En un turno, los agentes realizan sus acciones, una sola por cada agente, y modifican el medio sin que este varíe a no ser que cambie por una acción de los agentes. En el siguiente, el ambiente puede variar. Si es el momento de cambio del ambiente, ocurre primero el cambio natural del ambiente y luego la variación aleatoria. En una unidad de tiempo ocurren el turno del agente y el turno de cambio del ambiente.

Los elementos que pueden existir en el ambiente son obstáculos, suciedad, niños, el corral y los agentes que son llamados Robots de Casa. A continuación se precisan las características de los elementos del ambiente:

**Obstáculos:** estos ocupan una única casilla en el ambiente. Ellos pueden ser movidos, empujándolos, por los niños, una única casilla. El Robot de Casa, sin embargo, no puede moverlo. No pueden ser movidos ninguna de las casillas ocupadas por cualquier otro elemento del ambiente.

**Suciedad:** la suciedad es por cada casilla del ambiente. Solo puede aparecer en casillas que previamente estuvieron vacías. Esta, o aparece en el estado inicial o es creada por los niños.

**Corral:** el corral ocupa casillas adyacentes en número igual al del total de niños presentes en el ambiente. El corral no puede moverse. En una casilla del corral solo puede coexistir un niño. En una casilla del corral, que esté vacía, puede entrar un robot. En una misma casilla del corral pueden coexistir un niño y un robot solo si el robot lo carga, o si acaba de dejar al niño.

**Niño:** los niños ocupan solo una casilla. Ellos en el turno del ambiente se mueven, si es posible (si la casilla no está ocupada: no tiene suciedad, no está el corral, no hay un Robot de Casa), y aleatoriamente (puede que no ocurra movimiento), a una de las casillas adyacentes. Si esa casilla está ocupada por un obstáculo este es empujado por el niño, si en la dirección hay más de un obstáculo, entonces se desplazan todos. Si el obstáculo está en una posición donde no puede ser empujado y el niño lo intenta, entonces el obstáculo no se mueve y el niño ocupa la misma posición.

Los niños son los responsables de que aparezca suciedad. Si en una cuadrícula de 3 por 3 hay un solo niño, entonces, luego de que él se mueva aleatoriamente, una de las casillas de la cuadrícula anterior que esté vacía puede haber sido ensuciada. Si hay dos niños se pueden ensuciar hasta 3. Si hay tres niños o más pueden resultar sucias hasta 6.

Los niños cuando están en una casilla del corral, ni se mueven ni ensucian.

Si un niño es capturado por un Robot de Casa tampoco se mueve ni ensucia.

**Robot de Casa:** El Robot de Casa se encarga de limpiar y de controlar a los niños. El Robot se mueve a una de las casillas adyacentes, las que decida. Solo se mueve una casilla sino carga un niño. Si carga un niño puede moverse hasta dos casillas consecutivas.

También puede realizar las acciones de limpiar y cargar niños. Si se mueve a una casilla con suciedad, en el próximo turno puede decidir limpiar o moverse. Si se mueve a una casilla donde está un niño, inmediatamente lo carga. En ese momento, coexisten en la casilla Robot y niño.

Si se mueve a una casilla del corral que está vacía, y carga un niño, puede decidir si lo deja esta casilla o se sigue moviendo. El Robot puede dejar al niño que carga en cualquier casilla. En ese momento cesa el movimiento del Robot en el turno, y coexisten hasta el próximo turno, en la misma casilla, Robot y niño.

## 2. Objetivos

El objetivo del Robot de Casa es mantener la casa (a.k.a el ambiente) limpia. Se considera la casa limpia si el 60 % de las casillas vacías no están sucias.

Se sabe que si la casa llega al 60 % de casillas sucias el Robot es despedido e inmediatamente cesa la simulación. Si el Robot ubica a todos los niños en el corral y el 100 % de las casillas están limpias también cesa la simulación. Estos son llamados estados finales.

Debe programar el comportamiento del robot por cada turno así como las posibles variaciones del ambiente.

## 3. Ambiente inicial

Como ambiente inicial se especifica el tamaño del ambiente, el porcentaje de casillas que aparecen sucias, el porcentaje de obstáculos y el número de niños. El Robot de Casa parte de una posición aleatoria y es el que realiza el primer turno. Igual, se especifica el valor del tiempo de unidades de cambio (t). Con estos datos se genera un ambiente inicial que cumpla las restricciones previamente planteadas en el Marco General. El ambiente inicial debe ser factible.

En caso de que no se logre uno de los estados finales del ambiente, la simulación debe detenerse cuando hayan transcurrido 100 veces t.

## 4. Experimentos

Para complementar el trabajo, deben realizarse un conjunto de simulaciones (30 por cada ambiente) partiendo de distintos escenarios iniciales (no menos de 10). Se debe reportar, por ambiente generado, el porcentaje de casillas sucias medio, el número de veces que el Robot fue despedido y el número de veces que ubicó a los niños en el corral y limpió toda la casa. Estos datos deben reportarse en el informe.

## Principales Ideas Seguidas

Para solucionar el problema, el robot divide la función de decisión en dos subsistemas: Percepción y Acción.

### Percepción

*See: Ambiente → Percepción*

La función de percepción del Robot analiza el ambiente y guarda la ubicación de todos los niños, corrales y suciedad que encuentre.

Por cada niño y suciedad que encuentre, determina si existe un camino que pueda llevar al robot desde su ubicación a la de ellos. Si no existe un camino, el robot ignora este elemento durante el transcurso del presente ciclo. Si existe un camino, entonces lo guarda junto con la ubicación del elemento y el largo de este camino. Una vez que se haya revisado todo el mapa, el robot agrupa cada elemento por su tipo, y los ordena por el largo del camino que los lleva hasta ellos, de menor a mayor.

En el caso de los corrales, estos son organizados por la probabilidad que tengan de que el camino del robot a ellos sea bloqueado, para esto el robot analiza cuantos lados del corral son inaccesibles en el momento (si se encuentran en el borde del mapa, tienen como vecino un obstáculo o un corral con un niño). Una vez que tenga esta información para cada corral, los organiza de mayores a menores lados cubiertos.

### Acción

*Action: Percepción → Acción*

Para determinar qué acción realizar, la decisión del robot está determinada por las siguientes prioridades:

1. Si no tiene ningún niño, y hay niños sueltos en el mapa, avanzar hacia el niño con el menor camino factible hacia su ubicación y atraparlo.
2. Si tiene a un niño atrapado consigo, entonces buscar entre los corrales libres aquel que tiene las mayores probabilidades de ser bloqueado y dirigirse a este para dejar allí al niño.

3. Si no se encuentra ningún niño libre en el mapa y el robot no tiene a ninguno consigo atrapado, entonces su primera prioridad pasaría a ser limpiar la suciedad del ambiente. Escoge la casilla sucia con el camino factible mas corto y se dirige a esta para limpiarla

## Modelos de Agentes Considerados

Para representar la toma de decisión del Robot, dos modelos de agentes fueron considerados, [AgentA](#), similar a la Arquitectura Brooks para agentes reactivos y, [AgentB](#), similar a las arquitecturas por capas para agentes.

### AgentA

Este agente selecciona una acción directamente de la entrada perceptual, con los comportamientos asignados a cada entrada ordenados por prioridades para evitar que varios comportamientos se disparen simultáneamente.

Las reglas de este agente son las siguientes:

**Regla 1:** *Si* el Robot tiene un niño *entonces* llevar el niño hacia un corral.

**Regla 2:** *Si* Hay niños libres en el mapa *entonces* caminar hacia el niño más cercano y recogerlo.

**Regla 3:** *Si* Hay suciedad en el mapa *entonces* caminar hacia la casilla sucia más cercana y limpiarla.

Siendo **Regla 1** la de mayor prioridad, y **Regla 3**, la de menor prioridad.

### AgentB

En este agente, para mostrar un comportamiento más balanceado entre reactivo y proactivo, se organiza el sistema para tratar de llevar a cabo los objetivos de mayor prioridad, encontrar y llevar los niños a los corrales, intercalados con el objetivo de menor prioridad del agente, limpiar las casillas con suciedad en el mapa.

Para llevar a cabo este comportamiento, el agente sigue las siguientes reglas:

**Regla 1:** *Si* el Robot tiene un niño:

*Si* hay alguna casilla con suciedad en el camino hacia el corral donde se llevara el niño

*entonces* avanzar hacia esta casilla y limpiarla

*sino* llevar el niño hacia el corral.

**Regla 2:** *Si* Hay niños libres en el mapa:

*Si* hay alguna casilla con suciedad en el camino hacia el niño más cercano

*entonces* avanzar hacia esta casilla y limpiarla

*sino* caminar hacia el niño más cercano y recogerlo.

**Regla 3:** *Si* Hay suciedad en el mapa *entonces* caminar hacia la casilla sucia más cercana y limpiarla.

# Ideas Seguidas para la implementación

El Proyecto se implementó en C#. A continuación, se explica que función tiene cada clase en el proyecto.

## Elementos del Ambiente:

- `abstract class Element`: Esta es la clase base de todos los elementos del ambiente.
- `class Obstacle`: representa los Obstáculos en el ambiente y su comportamiento.
- `class Dirt`: representa la Suciedad en el ambiente.
- `class Corral`: representa los Corrales en el ambiente y su comportamiento.
- `class Kid`: representa los niños en el ambiente y su comportamiento.
- `abstract class Robot`: esta es la clase base de las dos implementaciones del agente Robot.
- `class AgentA`: esta es la implementación más reactiva del agente Robot.
- `class AgentB`: esta es la implementación reactiva y proactiva del agente Robot.

## Ambiente

- `class Cell`: esta clase representa las casillas dentro del ambiente, y es la encargada que evitar de que ocurra cualquier tipo de conflictos dentro de una casilla, asegurándose que siempre se cumplan las restricciones del ambiente en cuanto a los tipos de elementos que pueden ocupar juntos la misma casilla.
- `class Environment`: es la clase encargada de manejar el ambiente y los ciclos que ocurren en él.

Para la implementación de este proyecto de simulación, se usa un array bidimensional (`class Cell`), en este es donde se encuentran todos los elementos y donde ocurren todas las interacciones.

Este array bidimensional es manejado por la clase `Cell` quien se encarga de manejar los ciclos del ambiente y las interacciones entre los elementos y se encarga de realizar los cambios en el mapa que realicen el Robot y los niños. Esta también realiza los cambios necesarios al final de cada ciclo, como la suciedad que dejan los niños cada vez que se mueven.

## Consideraciones

En la práctica ambos agentes tuvieron un desempeño similar, con pocas diferencias en cuanto a sus desempeños en situaciones semejantes.

A continuación, los 10 escenarios a los cuales se les realizaron 30 simulaciones. Cada escenario viene acompañado con sus resultados:

### Escenario 1:

Dimensiones: 5 x 5, Número de niños: 1, % Casillas Sucias: 5, % Obstáculos: 11.

- **Estadísticas Agente A:**
  - % Casillas Sucias Medio – 13.86 %
  - Número de veces que fue despedido - 0
  - Número de veces que ubico a los niños en el Corral y limpio toda la casa – 30

- **Estadísticas Agente B:**

% Casillas Sucias Medio – 16.33 %

Número de veces que fue despedido - 0

Número de veces que ubico a los niños en el Corral y limpio toda la casa - 30

## Escenario 2:

Dimensiones: 6 x 6, Número de niños: 3, % Casillas Sucias: 15, % Obstáculos: 20.

- **Estadísticas Agente A:**

% Casillas Sucias Medio – 25.71 %

Número de veces que fue despedido - 10

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 13

- **Estadísticas Agente B:**

% Casillas Sucias Medio – 33.14 %

Número de veces que fue despedido - 16

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 11

## Escenario 3:

Dimensiones: 6 x 8, Número de niños: 3, % Casillas Sucias: 15, % Obstáculos: 20.

- **Estadísticas Agente A:**

% Casillas Sucias Medio – 21.51 %

Número de veces que fue despedido - 6

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 10

- **Estadísticas Agente B:**

% Casillas Sucias Medio – 19.81 %

Número de veces que fue despedido - 8

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 12

## Escenario 4:

Dimensiones: 7 x 8, Número de niños: 3, % Casillas Sucias: 15, % Obstáculos: 15.

- **Estadísticas Agente A:**

% Casillas Sucias Medio – 18.70 %

Número de veces que fue despedido - 8

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 19

- **Estadísticas Agente B:**

% Casillas Sucias Medio – 10.25 %

Número de veces que fue despedido - 4

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 23

### Escenario 5:

Dimensiones: 9 x 7, Número de niños: 5, % Casillas Sucias: 15, % Obstáculos: 20.

- **Estadísticas Agente A:**  
% Casillas Sucias Medio – 53.89 %  
Número de veces que fue despedido - 24  
Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0
- **Estadísticas Agente B:**  
% Casillas Sucias Medio – 54.65 %  
Número de veces que fue despedido - 26  
Número de veces que ubico a los niños en el Corral y limpio toda la casa – 1

### Escenario 6:

Dimensiones: 9 x 9, Número de niños: 6, % Casillas Sucias: 20, % Obstáculos: 15.

- **Estadísticas Agente A:**  
% Casillas Sucias Medio – 53.32 %  
Número de veces que fue despedido - 23  
Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0
- **Estadísticas Agente B:**  
% Casillas Sucias Medio – 48.62 %  
Número de veces que fue despedido - 21  
Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0

### Escenario 7:

Dimensiones: 10 x 10, Número de niños: 3, % Casillas Sucias: 15, % Obstáculos: 20.

- **Estadísticas Agente A:**  
% Casillas Sucias Medio – 12.52 %  
Número de veces que fue despedido - 2  
Número de veces que ubico a los niños en el Corral y limpio toda la casa – 1
- **Estadísticas Agente B:**  
% Casillas Sucias Medio – 11.52 %  
Número de veces que fue despedido - 1  
Número de veces que ubico a los niños en el Corral y limpio toda la casa – 3

### Escenario 8:

Dimensiones: 12 x 12, Número de niños: 3, % Casillas Sucias: 15, % Obstáculos: 15.

- **Estadísticas Agente A:**  
% Casillas Sucias Medio – 17.92 %  
Número de veces que fue despedido - 0  
Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0

- **Estadísticas Agente B:**

% Casillas Sucias Medio – 14.86 %

Número de veces que fue despedido - 0

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0

### Escenario 9:

Dimensiones: 12 x 12, Número de niños: 5, % Casillas Sucias: 20, % Obstáculos: 15.

- **Estadísticas Agente A:**

% Casillas Sucias Medio – 40.64 %

Número de veces que fue despedido - 4

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0

- **Estadísticas Agente B:**

% Casillas Sucias Medio – 31.34 %

Número de veces que fue despedido - 2

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0

### Escenario 10:

Dimensiones: 13 x 14, Número de niños: 10, % Casillas Sucias: 20, % Obstáculos: 15.

- **Estadísticas Agente A:**

% Casillas Sucias Medio – 58.23 %

Número de veces que fue despedido - 17

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0

- **Estadísticas Agente B:**

% Casillas Sucias Medio – 56.23 %

Número de veces que fue despedido - 23

Número de veces que ubico a los niños en el Corral y limpio toda la casa – 0