

# Informe Escrito

## Autor

Nombre y Apellidos	Grupo	Correo
Ariel Plasencia Díaz	C-412	<a href="mailto:arielplasencia00@gmail.com">arielplasencia00@gmail.com</a>

## Orientación del problema

### 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*, *corrales* y *agentes*, estos últimos son llamados robots de casa. A continuación se precisan las características de los elementos del ambiente:

### Obstáculo

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 moverlos. No pueden ser movidos a 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, es decir, no tiene suciedad, no hay un corral, no hay un robot de casa), y aleatoriamente (puede que no ocurra movimiento), a una de las casilla 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  $3 \times 3$  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 celdas. 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 en 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.

## Objetivos

El objetivo del robot de casa es mantener la casa limpia. Se considera la casa limpia si el 60% de las casillas vacías no están sucias.

## Principales ideas seguidas para la solución del problema

---

Para darle solución al problema se hizo uso del lenguaje python como herramienta. Se modeló el ambiente como una clase con ciertas funcionalidades y los elementos que interactúan en este como una jerarquía de clases. Al inicio de una simulación se garantiza que el ambiente construido es factible. Primero se colocan los corrales de los niños, luego los niños garantizando que ninguno se encuentre inicialmente en el corral. Después se coloca la suciedad y, por último, los obstáculos teniendo en cuenta que estos no bloqueen el camino del robot hacia ningún elemento del ambiente. El robot va a existir independientemente del ambiente, este se garantiza que inicia en una casilla vacía y sin cargar a ningún niño.

Se proponen dos modelos de agentes para describir el comportamiento del robot. La estrategia de ambos modelos puede ser considerada híbrida entre reactivo y proactivo, pero cada uno presenta matices más marcados de alguno de los rasgos. El desempeño de ambos modelos fue evaluado en 10 configuraciones de ambientes iniciales diferentes y para cada una de ellas se ejecutaron 30 simulaciones.

## Modelos de agentes considerados

---

### Modelo 1

El primer modelo que se propone se puede clasificar como un híbrido entre proactivo y reactivo, donde destaca el comportamiento proactivo. La estrategia que sigue se basa en priorizar la ubicación de los niños en el corral por encima de limpiar la suciedad. Si la suciedad se ha acumulado demasiado entonces el robot priorizará la limpieza.

Para recoger un niño analiza la distancia a estos y selecciona el más cercano a su posición. Una vez ha recogido un niño selecciona el corral mas distante a su posición (esto para evitar que al colocar niños en el corral estos bloqueen la entrada a el robot a los corrales que estén detrás y sean inaccesibles desde otras posiciones) y se dedica exclusivamente a llevarlo a ese corral. En el

proceso de moverse hacia un niño para recogerlo o guardarlo en el corral, si el robot se encuentra encima de una casilla sucia, este no la limpiará porque prioriza por encima de todo a los niños.

La mayor manifestación de reactividad está dada en que, si en algún turno detecta que la suciedad en el ambiente ha sobrepasado el 50 %, entonces pasa a priorizar la limpieza de suciedad hasta que esta vuelva a estar por debajo del 50 % y entonces vuelve a retomar la recogida de niños. Si en el momento en que detecta este aumento de suciedad está cargando un niño primero culminará su objetivo de llevarlo al corral y después se dedicará al control de la suciedad.

Para limpiar la suciedad el robot simplemente encuentra la casilla más cercana que esté sucia y se mueve hacia ella para limpiarla. Este modelo va a estar encapsulado en la clase *ChildsFirstRobot*.

## Modelo 2

El segundo modelo que se propone también se puede clasificar como un híbrido entre proactivo y reactivo, pero en este caso destaca el comportamiento reactivo. La estrategia de este robot va a estar basada en la distancia de su posición al resto de los elementos del ambiente.

En cada turno se compara la distancia hacia las suciedades y los niños en el ambiente. Si lo más cercano al robot es un niño entonces se va a dedicar a buscar ese niño y luego llevarlo al corral mas distante de su posición. En el caso de que lo más cercano sea una suciedad entonces se va a centrar en ir y limpiar esa casilla sucia.

Si en el trayecto del robot hacia el niño o la suciedad que está más cerca de su posición ocurre un cambio en el ambiente que hace que algún otro elemento esté más cercano entonces el robot reajustará el objetivo hacia el cual se mueve. Este modelo va a estar encapsulado en la clase *NearFirstRobot*.

## Ideas seguidas para la implementación

---

La modelación del ambiente se encuentra en el archivo *environment.py*. Se tiene una clase *Environment* que contiene los siguientes atributos y métodos que permiten describir un ambiente:

- `matrix` : es un diccionario que tiene como llave una tupla (i,j) y como valor el objeto que representa al elemento que se encuentra en la posición (i,j) del ambiente o None en caso de que esté vacía.
- `robot` : contiene el objeto Robot que actúa como agente en el ambiente.
- `dirty_count` : sirve para registrar los niveles de suciedad en cada turno y luego calcular el nivel de suciedad promedio.
- `set_playpen()` : coloca el corral en el ambiente.
- `initialize()` : coloca niños, suciedad y objetos en el ambiente.
- `initialize_robot()` : crea el robot y lo coloca en una posición vacía del ambiente.
- `generate_dirtiness()` : se encarga de seleccionar las casillas que van a ser ensuciadas según la ubicación de los niños en el ambiente.
- `natural_change()` : realiza las acciones que describen un cambio natural del ambiente.
- `random_change()` : realiza las acciones que describen un cambio aleatorio del ambiente.

Los elementos del ambiente se encuentran modelados en el archivo *elements.py*. El diseño consiste en una clase base *Element*. Esta clase está formada por los siguientes atributos y métodos:

- `pos` : es una tupla (i,j) que describe la posición del elemento.
- `environment` : es un objeto de tipo Environment que describe el ambiente al que pertenece el elemento.
- `find_next_step()` : dada una dirección a la que moverse calcula la posición en la que te colocarías.
- `step()` : cambia la posición del elemento en el ambiente.

Element va a ser la clase base para las siguientes:

- `Child` : Representa un niño. Esta clase además va a definir el siguiente método:
  - `move` : realiza el movimiento de los niños en el ambiente.
- `Obstacle` : Representa un obstáculo. Esta clase además va a definir el siguiente método:
  - `move` : realiza el movimiento de los obstáculos cuando son empujados por los niños.
- `Dirty` : Representa una suciedad.
- `Playpen` : Representa un corral. Esta clase además va a definir el siguiente atributo:
  - `child` : es un bool que indica si el corral contiene o no un niño.
- Robot: Representa el robot. Esta clase va a definir los siguientes atributos y métodos:
  - `child` : es un bool que indica si el robot tiene cargado o no un niño.
  - `bfs()` : calcula la distancia y el camino hacia todos los elementos del ambiente.
  - `get_path()` : devuelve el camino hacia un elemento determinado del ambiente.
  - `find_near_element()` : dado un conjunto de elementos devuelve el más cercano de ellos.
  - `find_far_element()` : dado un conjunto de elementos devuelve el más lejano de ellos.

Los modelos de agentes implementados *NearFirstRobot* y *ChildsFirstRobot* van a heredar de esta clase y definir sus estrategias en un método llamado *move()*. En el archivo *main.py* es donde se definen los ambientes iniciales, se ejecutan las simulaciones y se guarda la información que describe los resultados.

## Consideraciones obtenidas a partir de las simulaciones

---

Se construyeron ambientes iniciales con las características que se muestran en la siguiente tabla.

No. Ambiente	N	M	Niños	Por ciento Suciedad	Por ciento Obstáculos	t
1	10	10	5	25	15	10
2	7	8	3	15	10	5
3	7	8	3	15	10	20
4	15	15	10	15	20	50
5	5	5	2	10	5	5
6	10	5	3	20	20	10
7	10	10	5	30	10	20
8	10	10	5	10	30	30
9	9	9	3	15	20	20
10	10	10	4	20	20	20

Cada uno de los modelos de agente implementados se colocó en ambientes con cada una de las características descritas en la tabla anterior y se realizaron 30 simulaciones en cada tipo de ambiente. En las tablas siguientes se reportan los resultados obtenidos con el Modelo 1 y el Modelo 2 respectivamente. Además esta información se encuentra en el archivo *output.txt*, hacia donde se guarda la salida de la ejecución de las simulaciones.

No. Ambiente	Casa limpia y Niños en el corral	Despedido	Tiempo Agotado	Media del Por ciento de Casillas Sucias
1	1	29	0	51.57
2	5	23	2	40.04
3	6	21	3	38.82
4	5	24	1	50.04
5	7	22	1	36.25
6	13	12	5	35.89
7	1	29	0	50.03
8	24	3	3	25.58
9	21	7	2	29.00
10	12	15	3	38.55

No. Ambiente	Casa limpia y Niños en el corral	Despedido	Tiempo Agotado	Media del Por ciento de Casillas Sucias
1	0	29	1	51.45
2	0	30	0	44.00
3	0	30	0	46.56
4	0	29	1	53.78
5	0	29	1	39.77
6	1	26	3	49.58
7	0	30	0	48.71
8	0	7	23	54.04
9	0	25	5	49.42
10	0	28	2	53.11

Al analizar los resultados obtenidos se observa que el Modelo 1 presentó mejor desempeño que el Modelo 2, pues este último fue despedido la mayoría de las veces, mientras que el Modelo 1 logró limpiar la casa y poner todos los niños en el corral al menos una vez en todas las configuraciones de ambiente probadas. Además, como es de esperarse si es despedido tantas veces, el Modelo 2 presenta un porcentaje medio de casillas sucias que en la mayoría de los casos supera al del Modelo 1.