



Instituto Politécnico Nacional

Escuela Superior de Cómputo ESCOM

Trabajo Terminal

Comportamiento colectivo no trivial implementado en robots de bajo costo: el caso de “La hormiga de Langton”.

Trabajo Terminal No. 2016–A058

Presentan

**De Felipe Vargas Dalia
Sánchez Salazar Carmen Maytee**

Directores

**Dr. en C. Genaro Juárez Martínez
Dr. en C. Luz Noé Oliva Moreno**

RESUMEN

En la presente propuesta se describe la aproximación inicial para la implementación de comportamiento colectivo no trivial en organismos artificiales, empleando como caso de estudio el autómata celular “La hormiga de Langton”, inventado por Chris Langton en 1986. El estudio de dicho sistema nos permitirá describir parte de la complejidad así como el comportamiento colectivo de esta función en robots. Con base en las propiedades globales del sistema, las reglas propias derivadas de este autómata se implementarán en los robots para explorar y validar el comportamiento esperado de auto organización que presenta la función de Langton.

Palabras clave: Autómata celular, Comportamiento colectivo no trivial, Microcontroladores, Sistemas complejos.

Junio, 2017



1. Introducción	1
2. Marco teórico	3
2.1. Autómatas celulares	3
2.1.1. Definición de un autómata celular	4
2.1.1.1. Elementos de un autómata celular	4
2.1.2. Clases de Autómatas celulares	4
2.1.3. El juego de la vida	6
2.1.3.1. Reglas del juego de la vida	6
2.1.4. La hormiga de Langton	8
2.1.4.1. Modelo clásico	8
2.1.4.2. Reglas del modelo clásico	8
2.1.4.3. Extensiones del modelo clásico	10
2.2. Sistemas complejos	11
2.2.1. Características de los sistemas complejos	11
2.3. Comportamiento colectivo	12
2.3.1. Comportamiento colectivo en la naturaleza	12
2.3.2. Autoorganización	13
2.4. Inteligencia de enjambre	14
2.4.1. Enjambre robótico	15
2.4.2. Desafíos	15
2.4.3. Aplicaciones	15
2.4.4. Hormigas robóticas	15
3. Estado del arte	17
3.1. Autómata celular: La hormiga de Langton	17
3.2. Comportamiento colectivo implementado en robots	18
4. Análisis del simulador de la Hormiga de Langton	21
4.1. Análisis de riesgos	22
4.1.1. Identificación de riesgos	22
4.1.1.1. Riesgos de la curva de aprendizaje	22
4.1.1.2. Riesgos de los integrantes del equipo.	22
4.1.2. Valoración de los riesgos.	22

4.1.3.	Administración de los riesgos	23
4.2.	Herramientas	24
4.2.1.	Lenguaje para desarrollo del simulador	24
4.2.1.1.	Framework Qt	24
4.3.	Requerimientos del simulador	25
4.3.1.	Definición de actores	25
4.3.1.1.	Usuario final	25
4.3.2.	Requerimientos funcionales	25
4.3.3.	Requerimientos no funcionales	26
4.3.3.1.	Tecnologías de desarrollo	26
4.3.3.2.	Usabilidad	26
4.3.3.3.	Sistema Operativo	26
4.3.3.4.	Mantenibilidad	26
4.3.3.5.	Requerimientos de Hardware	26
4.3.4.	Reglas de negocio	27
5.	Diseño del simulador de la Hormiga de Langton	29
5.1.	Vista de escenarios	30
5.1.1.	Diagrama de casos de uso del actor Usuario Final para la Regla original	30
5.1.2.	Documentación del diagrama de casos de uso general del simulador.	30
5.1.2.1.	CU01 Simular	31
5.2.	Diseño del algoritmo de la regla de Langton	33
5.2.1.	Pseudocódigo: Regla de la hormiga de Langton	33
5.2.2.	Diagrama de flujo del movimiento de una hormiga	34
5.2.2.1.	Subrutinas correspondientes al algoritmo de la regla de Langton	35
5.2.2.2.	Subrutinas correspondientes al algoritmo de los antBot	37
5.3.	Diseño de pantallas	38
5.3.1.	Catálogo de menús	38
5.3.1.1.	MENU01. Acciones correspondientes a la configuración	38
5.3.1.2.	MENU02. Editar configuración	38
5.3.1.3.	MENU03. Vista de simulación	39
5.3.2.	Interfaz del simulador	39
5.3.2.1.	IUS01 Iniciar configuración a simular	39
5.3.2.2.	IUS02. Visualizar Simulación	40
5.3.2.3.	IUS03. Visualizar Diagrama de Frecuencias	41
5.3.3.	Diagrama de clases.	42
5.3.4.	Diagrama de secuencia.	43
6.	Análisis y Arquitectura del robot	45
6.1.	Diagrama de módulos del prototipo del robot	45
6.2.	Circuito eléctrico del robot	46
6.2.1.	Descripción de los bloques	47
6.2.1.1.	Sensado	47
6.2.1.2.	Control	47
6.2.1.3.	Indicador	47
6.2.1.4.	Movimiento	47
6.2.1.5.	Potencia	48
6.3.	Programación del robot	49

⁴TT 2016-A058 Comportamiento colectivo no trivial implementado en robots de bajo costo: el caso de "La hormiga de Langton".

7. Pruebas y resultados	51
7.1. Simulador de la regla clásica “La hormiga de Langton ”	51
7.1.1. Fenomenología de las hormigas de Langton	51
7.1.1.1. Configuración 033 (000100001)	52
7.1.1.2. Configuración 068 (001000100)	53
7.1.1.3. Configuración 096 (001100000)	54
7.1.1.4. Configuración 160 (010100000)	55
7.1.1.5. Configuración 257 (100000001)	56
7.1.1.6. Configuración 260 (100000100)	57
7.1.1.7. Configuración 272 (100010000)	58
7.1.1.8. Configuración 288 (100100000)	59
7.1.1.9. Configuración 320 (101000000)	60
7.1.1.10. Configuración 384 (110000000)	61
7.1.2. Patrones resultantes	62
7.1.2.1. Patrones del arreglo 033	62
7.1.2.2. Patrones del arreglo 068	63
7.1.2.3. Patrones del arreglo 096	64
7.1.2.4. Patrones del arreglo 160	65
7.1.2.5. Patrones del arreglo 257	66
7.1.2.6. Patrones del arreglo 260	67
7.1.2.7. Patrones del arreglo 272	68
7.1.2.8. Patrones del arreglo 288	69
7.1.2.9. Patrones del arreglo 320	70
7.1.2.10. Patrones del arreglo 384	71
7.1.3. Matrices de incidencia	72
7.1.3.1. Arreglo 033	72
7.1.3.2. Arreglo 068	72
7.1.3.3. Arreglo 096	72
7.1.3.4. Arreglo 160	73
7.1.3.5. Arreglo 257	73
7.1.3.6. Arreglo 260	73
7.1.3.7. Arreglo 272	74
7.1.3.8. Arreglo 288	74
7.1.3.9. Arreglo 320	74
7.1.3.10. Arreglo 384	75
7.1.4. Periodos de patrones oscilatorios	76
7.2. Prototipo de robot: antBot	77
7.2.1. Esquema de pruebas	77
7.2.1.1. Prueba 1	77
7.2.1.2. Prueba 2	78
7.2.1.3. Prueba 3	78
7.2.2. Prototipo final antBot	79
7.2.3. Prototipo de carcasa para un antBot	82
7.2.4. Análisis de la dinámica de la colonia de robots	83
7.2.4.1. Análisis de movimientos	83
7.2.4.1.1. Cálculo probabilidad de giros.	83
7.2.4.2. Análisis de trayectorias	84
7.3. Simulador de la regla adaptada de “La hormiga de Langton ”	87

7.3.1. Giros válidos del antBot	87
7.3.1.1. Giro de 45° a la Derecha	87
7.3.1.2. Giro de 45° a la Izquierda	88
7.3.1.3. Giro de 90° a la Derecha	88
7.3.1.4. Giro de 90° a la Izquierda	89
7.3.2. Casos a considerar por el antBot	90
7.3.2.1. Choque en la frontera	90
7.3.2.2. Colision de antBots	91
7.3.3. Comportamiento colectivo	92
8. Conclusiones	95
9. Bibliografía	97
10. Glosario de términos	101

⁶TT 2016-A058 Comportamiento colectivo no trivial implementado en robots de bajo costo: el caso de "La hormiga de Langton".

CAPÍTULO 1

Introducción

Es notable la gran practicidad que tienen los autómatas celulares (AC) para simular fenómenos naturales, ayudando a una interpretación más inmediata de los resultados de una investigación a través de su simulación por computadora. De este modo, la proyección del autómata celular "La hormiga de Langton" en una colonia de robots de bajo costo, permitirá presentar resultados relevantes acerca de su comportamiento, observando así la posible existencia de una relación directa entre la regla de Langton y algunos de los comportamientos reales de las hormigas.

Los autómatas celulares han sido utilizados con éxito en distintas disciplinas. Por ejemplo, en el área de la física es una de las técnicas más interesantes y prácticas para simular fenómenos concretos en dinámica de fluidos. En el estudio de los sistemas complejos en Biología, los AC representan, desde mediados de los 80's, una alternativa a la modelización con ecuaciones diferenciales. Entre 1992 y 2002, Stephen Wolfram trabajó en su libro "A new kind of Science" [1], donde hace un estudio extenso de los autómatas celulares y su aplicación en campos de simulación y estudio de la realidad, es decir, introduce la idea de que se puede modelar y entender cualquier fenómeno natural mediante el uso de autómatas celulares. Mientras que Langton propone la utilización de los AC como técnica principal para el estudio de la vida artificial, expresado en su artículo "Studying Artificial Life with Cellular Automata" [2].

Actualmente la computación ha llegado prácticamente a todo el mundo, sin embargo existen ciertos campos dentro de dicha ciencia que no han sido tan explorados como otros. Tal es el caso de los autómatas celulares, que sin ser un tema nuevo sigue siendo un área desconocida por muchos.

Los autómatas celulares surgen en la década de los 40's con John von Neumann, quien intentaba modelar una máquina capaz de autoreplicarse, llegando así a un modelo matemático de dicha máquina con reglas complicadas sobre una red regular. Inicialmente fueron interpretados como conjunto de células que crecían, se reproducían y morían a medida que pasaba el tiempo. Su nombre se debe a esta similitud con el crecimiento de las células.

Un autómata celular es un sistema dinámico compuesto por un conjunto de celdas o células que adquieren distintos estados o valores. Estos estados son transformados de un instante a otro en unidades de tiempo discreto y de manera simultánea, es decir, que se puede cuantificar con valores enteros a intervalos regulares. De esta manera, este conjunto de células logran una evolución a través de una función, que es sensible a los estados de las células vecinas, y que se conoce como regla de transición local.

El aspecto que más caracteriza a los AC es su capacidad de lograr una serie de propiedades que surgen de la propia dinámica global a través del paso del tiempo y no desde un inicio, aplicándose a todo el sistema en general. Por lo tanto, no es fácil analizar las propiedades globales de un AC desde su comienzo si no es por medio de una simulación, partiendo de un estado o configuración inicial de células y cambiando en cada instante los estados de todas ellas de forma síncrona. Los autómatas celulares, a pesar de ser descritos por reglas sencillas, frecuentemente dan lugar a comportamientos complejos.

Los sistemas complejos se caracterizan fundamentalmente porque su comportamiento es imprevisible. En la actualidad no existe una definición formal para un sistema complejo, pero se conoce una lista de propiedades presentes en ellos. Entre ellas, un sistema complejo debe estar compuesto por una gran cantidad de elementos primitivos. Otra propiedad hace referencia a la interacción entre dichos elementos, la cual es local y origina un comportamiento emergente que no puede explicarse a partir de su análisis individual. Además, es muy difícil predecir su evolución dinámica futura.

En la naturaleza se pueden encontrar una gran cantidad de ejemplos de sistemas complejos que se extienden desde la física hasta la neurología, desde la economía hasta la biología molecular, desde la sociología hasta las matemáticas. Por ese motivo, esta clase de sistemas no constituye un caso raro ni excepcional sino que se manifiesta en la inmensa mayoría de los fenómenos que se observan a diario. Sin embargo, y a pesar de su gran diversidad y abundancia, se pueden identificar conductas dinámicas genéricas, no importa su naturaleza (física, química, biológica o social); entre ellas, las leyes de crecimiento, la autoorganización y los procesos colectivos emergentes.

La regla objeto de estudio en este trabajo terminal, “La hormiga de Langton”, es un autómata celular de dos dimensiones el cual presenta un conjunto de reglas muy sencillo, que sin embargo da lugar a comportamiento complejo. La hormiga de Langton opera sobre espacio conformado por una rejilla uniforme, en la que cada celda se encuentra en un estado del alfabeto binario posible. Fue inventada por Chris Langton en 1986.

La implementación de dicho autómata celular sucederá de manera adaptada a la física de los robot pero con cierta correspondencia al modelo clásico.

En este capítulo se muestra la información correspondiente a la investigación documental realizada con el fin de comprender el contexto del desarrollo del presente Trabajo Terminal.

2.1. Autómatas celulares

Los autómatas celulares (AC) nacen gracias a John von Neumann en la década de los 40's. En esa época John von Neumann se dio a la tarea de crear una máquina capaz de la auto réplica y además soportar comportamiento complejo, el modelo matemático de dicha máquina se convirtió en el primer autómata celular.

En un principio los autómatas celulares fueron interpretados como conjunto de células que crecían, se reproducían y morían en lapsos de tiempo discreto. Su nombre se debe a esta similitud con el desarrollo de las células.

El posterior estudio de los autómatas celulares fue retomado por Stephen Wolfram quien en 1983 presenta los llamados autómatas celulares elementales, una serie de 256 reglas que es posible encontrar en el dominio de una dimensión.

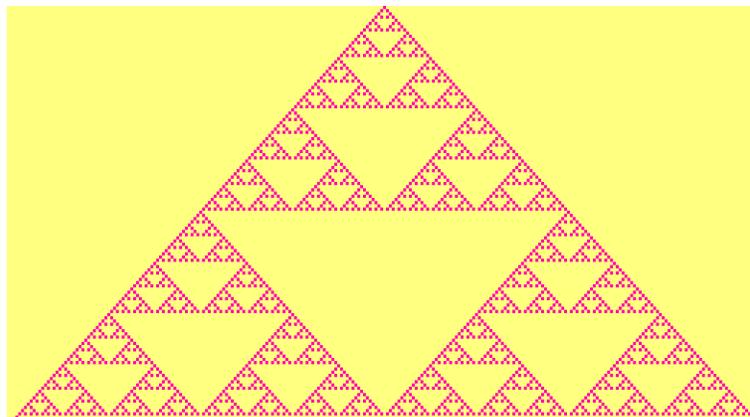


Figura 2.1: Regla 90 de los Automátas celulares elementales

2.1.1. Definición de un autómata celular

Un autómata celular puede entenderse simplemente como un modelo matemático de un sistema natural. Está compuesto por una rejilla en donde cada celda adquiere distintos valores. Estos estados son transformados de un instante a otro en unidades de tiempo discreto y de manera simultánea. De esta manera, este conjunto de células logran una evolución a través de una función, que es sensible a los estados de las células vecinas, y que se conoce como regla de transición local.

El aspecto característico de los AC es su capacidad de lograr una serie de propiedades que surgen de la propia dinámica global a través del paso del tiempo y no desde un inicio, aplicándose a todo el sistema en general. Por lo tanto, no es fácil analizar las propiedades globales de un AC desde su comienzo si no es por medio de una simulación, partiendo de un estado o configuración inicial de células y cambiando en cada instante los estados de todas ellas de forma síncrona. Los autómatas celulares, a pesar de ser descritos por reglas sencillas, frecuentemente dan lugar a comportamientos complejos.[3]

2.1.1.1. Elementos de un autómata celular

Es importante conocer los elementos base de un autómata celular para así entender su dinámica.

- Arreglo Regular. Es el espacio de evoluciones, y cada división homogénea del arreglo es llamada celda.
- Conjunto de Estados. Es finito y cada elemento del arreglo toma un valor de este conjunto de estados. También se denomina alfabeto. Puede ser expresado en valores o colores.
- Configuración Inicial. Consiste en asignar un estado a cada una de las células del espacio de evolución inicial del sistema vecindades.
- Función Local. Es la regla de evolución que determina el comportamiento del AC. Se conforma de una célula central y sus vecindades. Define como debe cambiar de estado cada célula dependiendo de los estados anteriores de sus vecindades. Puede ser una expresión algebraica o un grupo de ecuaciones.

Se puede entonces definir formalmente un autómata celular de la siguiente forma:

Un autómata celular es una 4-tupla $A = \langle \Sigma, u, \varphi, c_0 \rangle$ evolucionando en $d \in \mathbb{Z}^+$ dimensión, con un conjunto finito de estados Σ , una conexión local v tal que $v = \{ x_{i,j}, \dots, n:d \}$, la función de transición $\varphi : v \rightarrow \Sigma$ y la condición inicial del sistema c_0 . [13]

2.1.2. Clases de Autómatas celulares

En el estudio de los autómatas celulares existe una clasificación[4] propuesta por Stephen Wolfram la cual da un marco de referencia a distintas preguntas sobre el comportamiento de dichos modelos.

Stephen Wolfram fue uno de los primeros investigadores que realizaron estudios más sistemáticos sobre los AC. Wolfram clasificó el comportamiento cualitativo de los autómatas celulares unidimensionales. De acuerdo con esto, un AC pertenece a una de las siguientes clases:

- Clase I. La evolución lleva a una configuración estable y homogénea, es decir, todas las células terminan por llegar al mismo valor.
- Clase II. La evolución lleva a un conjunto de estructuras simples que son estables o periódicas.
- Clase III. La evolución lleva a un patrón caótico.

⁴TT 2016-A058 Comportamiento colectivo no trivial implementado en robots de bajo costo: el caso de "La hormiga de Langton".

- Clase IV. La evolución lleva a estructuras aisladas que muestran un comportamiento complejo (es decir, ni completamente caótico, ni completamente ordenado) este suele ser el tipo de comportamiento más interesante que un sistema dinámico puede presentar.

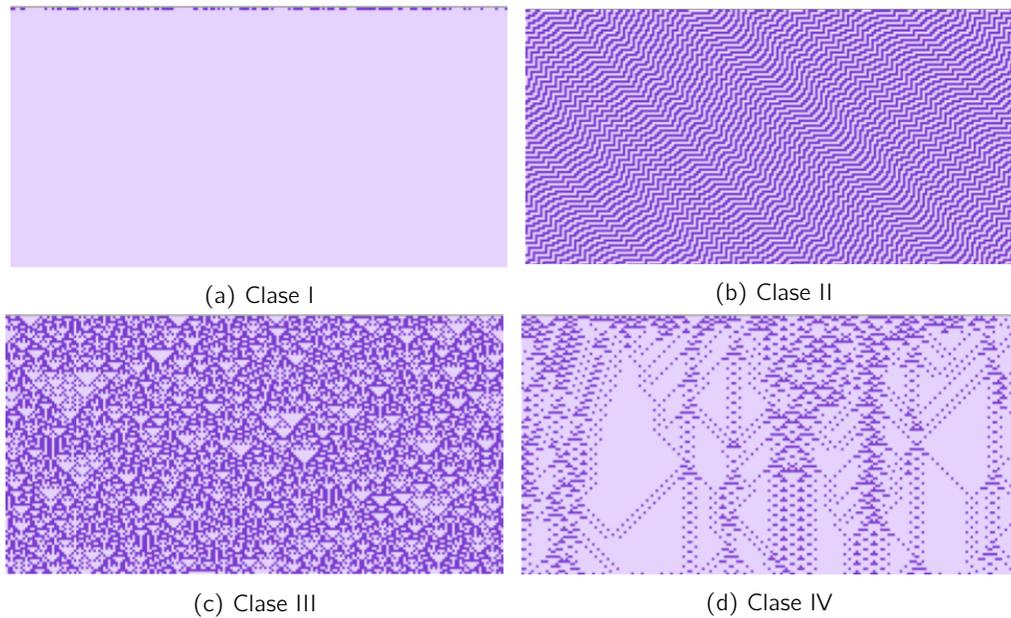


Figura 2.2: Clases de autómatas celulares propuestas por Stephen Wolfram [5].

2.1.3. El juego de la vida

El juego de la vida (GOL) es el autómata celular más popular de todos los tiempos. Fue inventado por John Horton Conway, un matemático británico y definido y presentado por Martin Gardner en 1970 en el Journal Scientific American^[6]. Su relevancia recae en el hecho de contener una máquina universal, es el autómata celular más estudiado y de él se han obtenido una cantidad considerable de resultados interesantes.

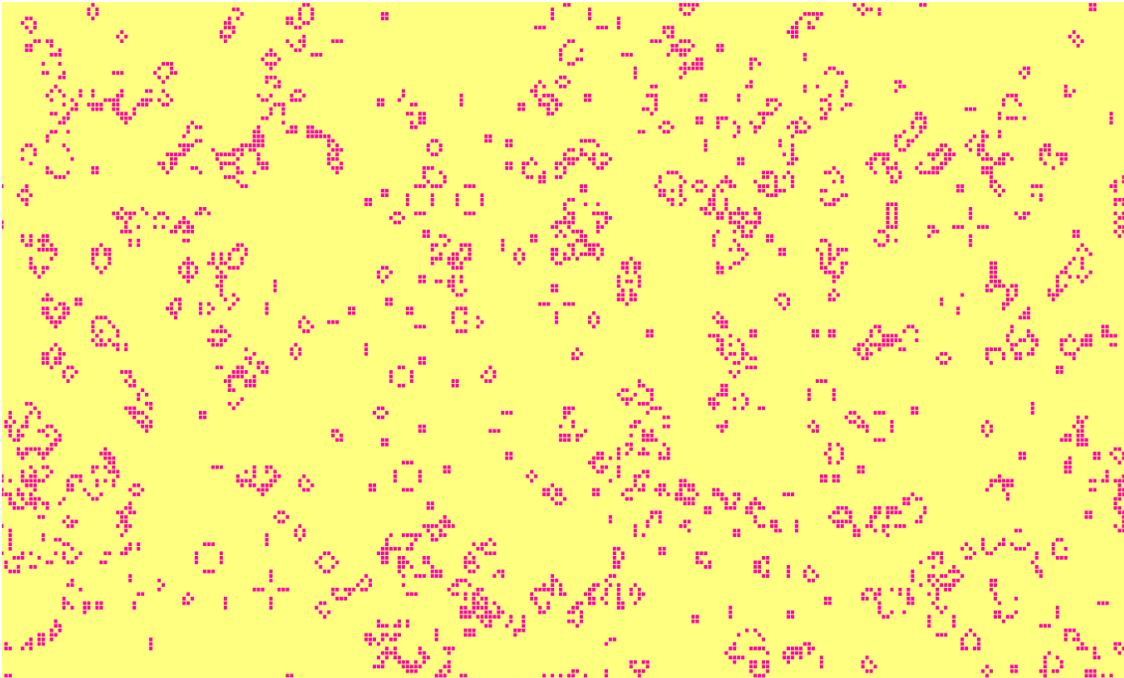


Figura 2.3: El juego de la vida

2.1.3.1. Reglas del juego de la vida

La dinámica de este autómata está basada en la evolución de estados sucesivos, en los cuales las condiciones del estado futuro dependen solamente de las condiciones del estado anterior, por tanto no se requiere de la entrada de datos durante el desarrollo del mismo, y es tan sólo el estado inicial y las normas del juego lo que determinan el desarrollo del mismo.

El juego se desarrolla en un tablero infinito teóricamente, y de unas dimensiones prefijadas en la práctica, subdividido en forma de una rejilla mediante filas y columnas en celdas. Cada una de las celdas de la rejilla quedan por tanto rodeadas por otras ocho casillas que determinan su “vecindad”. Estas celdas albergan lo que se denominan “células” que pueden encontrarse en dos estados distintos, célula viva (podría considerarse un “uno lógico”) o célula muerta (“cero lógico”) el estado de una célula en el futuro dependerá tanto de su estado actual como del estado actual de sus células vecinas, siguiendo unas simples reglas establecidas. Una célula al morir deja vacía la celda que ocupaba.

⁶TT 2016-A058 Comportamiento colectivo no trivial implementado en robots de bajo costo: el caso de “La hormiga de Langton”.

Las reglas en las que se basa este autómata celular son descritas a continuación:

- Supervivencia: Cada individuo o célula que cumpla el requisito de tener 2 ó 3 vecinos vivos sobrevive a la siguiente generación (por tanto su estado se mantiene inalterado en el siguiente turno).
- Fallecimiento: Una célula viva que tenga menos de 2 vecinos fallece por aislamiento en el siguiente estado o turno. Una célula viva que tenga más de tres vecinos vivos muere por superpoblación en el siguiente estado o turno.
- Nacimiento: Si una celda vacía pasa a tener en su vecindad exactamente 3 células vivas su estado futuro en el siguiente turno será el de célula viva.

2.1.4. La hormiga de Langton

2.1.4.1. Modelo clásico

En 1986 Chris Langton, inspirado por la bioquímica, exploró la posibilidad de implementar la lógica molecular del estado viviente generando así una bioquímica artificial basada en la interacción entre moléculas artificiales, a través de un un autómata virtual. Con el fin de entender cómo construir moléculas artificiales, Langton comenzó por entender los roles funcionales a nivel biológico que deben llevar a cabo si estas partículas participan en algo como la bioquímica artificial.

Señaló que la lógica molecular de la vida es una lógica distribuida dinámica, donde un conjunto de operadores y operandos trabajan para producir más y más de ellos. Algunos de ellos se encuentran distribuidos dando origen al proceso de auto reproducción. Esta característica dinámica no tiene algún parecido con alguna lógica formal típica. Esta lógica dinámica masiva y continua es magníficamente regulada por los mismos procesos que la constituyen. Este tipo de interacción dinámica es la que buscó implementar en un medio computacional adecuado. Como sabemos, a principios de los años 50, John von Neumann tratando de modelar el proceso natural de auto reproducción propuso los autómatas celulares.

Una colonia de insectos artificial. Langton dijo que el comportamiento global de una sociedad es un fenómeno emergente, que surge de todas las interacciones locales de sus miembros. El comportamiento complejo puede surgir de la interacción de las partes muy simples. Mencionó también que las colonias de insectos sociales proporcionan buen material para el estudio de la vida artificial debido a que presentan fácilmente comportamiento complejo emergente en la interacción de partes de vida muy simples.

Creyó que podría utilizar una colonia de hormigas como modelo para una forma variante de un autómata celular, una en la que cada célula individual es móvil y puede moverse semiautónomamente. Cada célula puede cambiar de estado, en virtud de los estados de las otras células en su vecindad inmediata. Sin embargo, en lugar de consistir en un conjunto fijo de células, la idea fue que esta colonia consistiera en un conjunto de células en constante cambio. La colonia resultante dependerá de las muchas células individuales que están siendo impulsadas por el territorio que ocupa la colonia, encontrándose una con otra y participar así en interacciones complejas.

Este autómata celular tomó popularidad debido a la simplicidad de sus reglas pues su dinámica resulta de una función de transición la cual es incluso más simple que la utilizada en el juego de la vida pero resultando también en un autómata celular universal.

2.1.4.2. Reglas del modelo clásico

A las primeras hormigas, Langton las llamo vants (virtual ants). Las normas que rigen el movimiento de una vant son simples. Las hormigas virtuales residen en un entorno que consta de células uniformemente espaciadas, fijas que se encuentran en uno de los dos estados posibles. (viva o muerta, 0 o 1). Una hormiga viaja en línea recta en el espacio vacío.

- Si se encuentra con una célula muerta, hace un giro a la derecha y sale de la célula invirtiendo el estado de la misma.
- Si se encuentra con una célula viva, gira a la izquierda y sale de la célula invirtiendo su estado.

Por lo tanto, la hormiga deja un "rastros", donde quiera que va. la Figura 2.4 muestra los seis primeros movimientos que la hormiga realiza según las reglas descritas. La flecha roja representa la dirección hacia donde la hormiga se encuentra mirando.

Sabiendo que un autómata celular se define formalmente como una 4-tupla, se puede definir a “La hormiga de Langton” como :

$$LA = \langle \Sigma, u, \varphi, c0 \rangle \text{ evolucionando en } d \in \mathbb{Z}^+ \text{ dimensión}$$

Entonces:

● Σ (Conjunto finito de estados): $\{ 0,1 \}$

● u (Conexión local)

● φ (Función de transición):

$\{ 1 \rightarrow 0 \} \& \{ 0 \rightarrow 1 \}$

● $c0$ (Condición inicial del sistema): $t0$

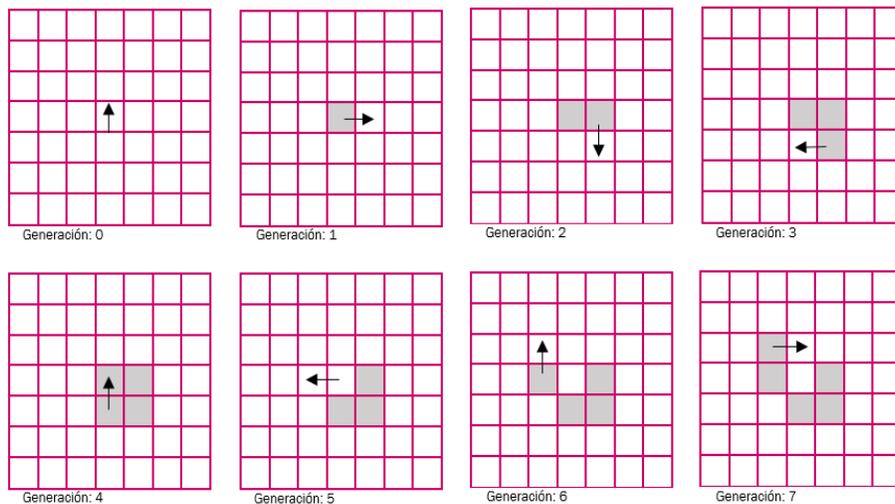


Figura 2.4: Primeros siete movimientos de una hormiga que inicia mirando hacia el norte.

Pronto notó Langton que incluso para el caso de una hormiga solitaria, el comportamiento resultante es bastante complejo. La complejidad se debe al hecho de que la hormiga se mantiene corriendo en su propio camino. Su propio comportamiento pasado sirve para complicar el entorno en que se encuentra en su comportamiento actual, siempre es parcialmente un reflejo de su comportamiento en el pasado. Por lo tanto, opera como una máquina de Turing virtual.

Para las interacciones entre muchas hormigas, el comportamiento se vuelve mucho más complejo. Ahora, cada una de las hormigas responde, no sólo a su, propio comportamiento, sino al comportamiento pasado del resto de las hormigas colectivamente. Por lo tanto, las hormigas pueden cooperar en la realización de tareas en masa que están más allá de las capacidades de cualquiera de ellas individualmente. No obstante, lo que se observó en ese entonces, mostró interesantes maneras en que los comportamientos comunes a las hormigas pueden ser solitarios, cuando se suman esfuerzos, se convierten en bloques de construcción para producir comportamientos de orden superior.

La figura 2.8 muestra cómo dos hormigas pueden cooperar en la construcción de una pista “circular” cada

vez más amplia. Este comportamiento implica que dos hormigas participen conjuntamente para formar una 'pista de desplazamiento', una actividad que no se puede observar en las hormigas solitarias.

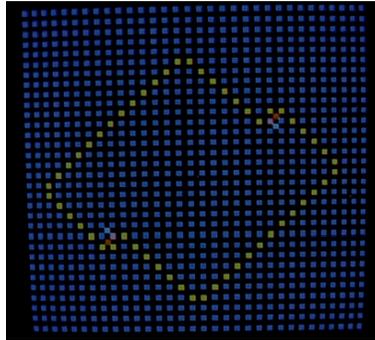


Figura 2.5: Dos hormigas interactuando en la construcción de una pista [2].

Hay tantas formas en que estas hormigas virtuales pueden encontrarse unas con otras que las reglas de transición aún no se han elaborado para todos los posibles encuentros. Los únicos encuentros resueltos hasta la fecha en que Langton publicó este estudio implican pares de hormigas que chocan en una de las celdas de entorno uniformemente espaciadas.

Después de realizar una serie de pruebas, Langton menciona que estos sistemas de hormigas que interactúan exhiben algunos de los fenómenos agregados conocidos a partir del estudio de las sociedades de insectos. El comportamiento a nivel macro depende altamente de la conducta a nivel micro. Además comprobó que un pequeño cambio evolutivo en el patrón de comportamiento de los individuos puede ser amplificado en un importante efecto social.

2.1.4.3. Extensiones del modelo clásico

El término Turmite se refiere a máquinas de Turing de 2 dimensiones y nacen a partir del caso base "La hormiga de Langton", existen múltiples variaciones debido a que las posibilidades son prácticamente infinitas. A continuación se presentan las 3 más populares.

Múltiples colores

Greg Turk y Jim Propp consideraron una simple extensión la cual consiste en utilizar más de dos colores creando cadenas para identificar el comportamiento de la hormiga. Considerando este formato el caso clásico de "La hormiga de Langton" sería una cadena RL.

Múltiples estados

Otra extensión considera más de dos estados para una hormiga, su comportamiento tiende a formar avenidas, espirales o crecimiento caótico.

Múltiples hormigas

Es posible que más de dos hormigas coexistan en el mismo plano, e incluso que exista más de una Turmite sin embargo es necesario definir las reglas que seguirán al encontrarse.

2.2. Sistemas complejos

2.2.1. Características de los sistemas complejos

Un sistema complejo es un grupo u organización que es construido desde la interacción de muchos elementos primitivos.

En tales sistemas, las partes individuales (conocidos como componentes), y la interacción entre ellos, frecuentemente conducen a comportamientos en gran escala, que no son fáciles de predecir desde un conocimiento dado de sus componentes individuales, tal efecto de este comportamiento es llamado emergente.

Los sistemas complejos se caracterizan fundamentalmente porque su comportamiento es imprevisible. En la actualidad no existe una definición formal para un sistema complejo, pero se conoce una lista de propiedades presentes en los sistemas complejos. Entre ellas:

- Gran cantidad de elementos primitivos.
- Interacción local la cual origina un comportamiento emergente que no puede explicarse a partir de su análisis individual.
- Evolución futura muy difícil de predecir.

En la naturaleza se pueden encontrar una gran cantidad de ejemplos de sistemas complejos que se extienden desde la física hasta la neurología, desde la economía hasta la biología molecular, desde la sociología hasta las matemáticas. Por ese motivo, esta clase de sistemas no constituye un caso raro ni excepcional sino que se manifiesta en la inmensa mayoría de los fenómenos que se observan a diario. Sin embargo, y a pesar de su gran diversidad y abundancia, se pueden identificar conductas dinámicas genéricas, no importa su naturaleza (física, química, biológica o social); entre ellas, las leyes de crecimiento, la auto organización y los procesos colectivos emergentes.

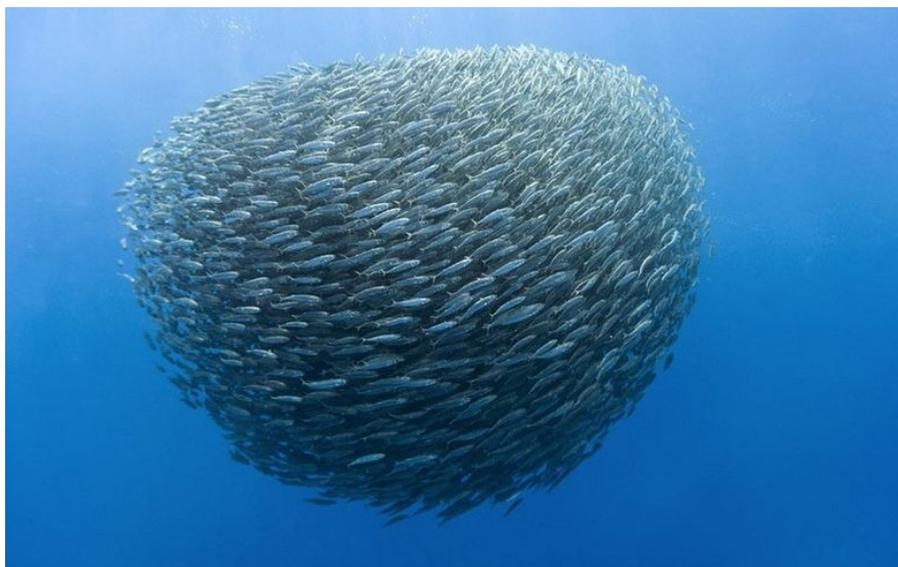


Figura 2.6: Cardumen[7].

2.3. Comportamiento colectivo

El comportamiento colectivo fue estudiado en seres humanos como la conducta que se produce en las multitudes o masas, que sale de las normas establecidas. Dentro de este enfoque sociológico, el comportamiento colectivo se diferencia de cualquier otro tipo de comportamiento grupal por el hecho de que siempre es corrupto, desviado.

Entre otras características que destacan este comportamiento, desde el punto de vista sociológico, encontramos que debe ser relativamente inesperado, poco estructurado, desorganizado, espontáneo. Es decir, tiene que emerger de las condiciones precisas de la situación, de las variables del sistema en un momento determinado.

2.3.1. Comportamiento colectivo en la naturaleza

Es fascinante e intrigante cómo y por qué algunas especies de animales coordinan sus movimientos de tal manera que es observable el comportamiento colectivo en ellos.

El fenómeno colectivo en grupos de animales ha atraído la atención en los últimos años, convirtiéndose en uno de los temas más destacados en la etología. Existen diferentes razones para ello. Por un lado, los grupos de animales proveen un ejemplo paradigmático de auto organización, donde el comportamiento colectivo surge en ausencia de un control centralizado. El mecanismo de la formación de grupos, donde reglas locales de los individuos dirigen a un estado global coherente, es muy general y trasciende los detalles naturales de estos componentes.

El comportamiento colectivo en grupos de animales es un fenómeno generalizado en los sistemas biológicos, a muy diferentes escalas y niveles de complejidad. En muchos casos se forman grupos cohesivos, que actúan en conjunto con notable coordinación y capacidad de adaptación, y se sostienen de forma espontánea por la atracción mutua de los miembros.

El comportamiento colectivo surge en ausencia de un control centralizado: miembros individuales actúan sobre la base de información local limitada procedente de la interacción con sus vecinos.



Figura 2.7: Comportamiento colectivo manifestado en una parvada de aves [8].

Un ejemplo claro de ello son las parvadas de aves, donde cada ave individualmente va en la misma dirección que sus vecinos, y esta tendencia local, da lugar a una multitud en movimiento coherente. Este mecanismo, que produce patrones globales de las normas locales, se conoce como autoorganización.

Otro ejemplo claro de autoorganización en el mundo natural se produce en las colonias de hormigas. La reina no da órdenes directas y no le dice a cada hormiga qué hacer. En su lugar, cada hormiga reacciona a estímulos provocados por un rastro químico que a su vez van dejando. Resulta que cada hormiga es una unidad autónoma que reacciona dependiendo sólo de su entorno local y de las reglas codificadas genéticamente y a pesar de la falta de toma de decisiones centralizada, las colonias de hormigas exhiben un comportamiento complejo, incluso han sido capaces de demostrar la capacidad de resolver problemas geométricos.

2.3.2. Autoorganización

La autoorganización significa la espontánea emergencia del orden en sistemas físicos y naturales.

La autoorganización sin la presencia de un líder, como se mencionó en líneas anteriores es una característica común en el comportamiento colectivo de ciertas especies. Insectos, peces y pájaros en lo que animales concierne y también en algunos microorganismos. Las razones biológicas a este sorprendente comportamiento incluyen, entre algunas otras: la protección contra algún depredador, el ahorro de energía en cada individuo y por supuesto la búsqueda de alimento, pero entonces nos preguntamos: ¿Cómo se convierten las interacciones de comportamiento individual, no correlacionado y desordenado en un comportamiento más ordenado y coherente?, ¿Cómo surge el comportamiento social colectivo a partir de individuos que cooperan en interacción a nivel local?.

En los últimos años, ha existido un interés renovado para hacer frente a este tipo de preguntas. Modelos basados en agentes han demostrado ser herramientas poderosas para explorar aspectos de estas preguntas, ya que incorporan reglas básicas de comportamiento individual que capturan la esencia del problema que se está explorando. Por lo tanto, mediante la construcción de simulaciones con gran número de agentes que interactúan de manera paralela, ofrecen una buena aproximación para obtener respuestas pertinentes.

El fenómeno enjambre es un comportamiento colectivo exhibido por un gran número de entidades, particularmente animales de tamaño semejante. El término se aplica particularmente a insectos.

En las últimas décadas, los científicos han recurrido a modelar dicho comportamiento buscando comprender de mejor manera tal conducta. Los modelos matemáticos empleados para simular enjambres generalmente representan animales individuales de los cuales su movimiento se rige bajo tres principios. El primero de ellos dice que cada individuo debe moverse en la misma dirección que sus vecinos, el segundo principio resalta que cada individuo debe permanecer cerca de éstos y finalmente el tercer principio dice que el individuo debe evitar colisiones con sus vecinos. Muchos modelos en la actualidad usan variaciones de estas reglas

2.4. Inteligencia de enjambre

En la naturaleza los insectos parecen sincronizarse para realizar ciertas tareas en conjunto sin contar con un control central y presentando un comportamiento altamente coordinado, localizando, por ejemplo, la ruta más corta en una red para llegar una fuente de comida.

Este término es adaptado del inglés “Swarm Intelligence” y se trata de un enjambre/grupo que se define como una población de elementos interactuando entre si capaces de optimizar un objetivo global a través de la búsqueda de colaboración de un espacio. Los elementos o agentes pueden ser máquinas muy simples o muy complejas. Existen dos restricciones que deben observarse: se suelen limitar a las interacciones locales, por lo general la interacción no se realiza directamente, sino indirectamente a través del entorno. La propiedad básica que los hace enjambres es su comportamiento de autoorganización.

Algunos de los más flexibles y poderosos algoritmos de optimización desarrollados en los últimos años han sido inspirados por sistemas naturales, estos incluyen: algoritmos genéticos, redes neuronales artificiales y algoritmos de optimización tipo colonia de hormigas.



Figura 2.8: Colonia de hormigas argentinas [9].

El comportamiento de enjambre es un tipo de conducta exhibida por entidades de tamaño similar que se agrupan entre sí, reuniéndose alrededor de un punto o bien moviéndose en masa. Cada miembro del enjambre deberá obedecer tres reglas:

- Moverse en la misma dirección que sus vecinos.
- Permanecer cerca de sus vecinos.
- Evitar choques con sus vecinos.

Así también existen 3 zonas críticas alrededor de cada individuo:

- Zona de repulsión: un área muy cercana al sujeto en la cual este buscara distanciarse de sus vecinos para evitar choques.
- Zona de alineación: el miembro del enjambre trata alinearse en la misma dirección en la que se mueven sus vecinos.
- Zona de atracción: este espacio se extiende tan lejos como se lo permite la habilidad de sensado del individuo y este buscar permanecer cerca de sus vecinos.

2.4.1. Enjambre robótico

Inspirados en colonias de insectos que en conjunto llevan a cabo tareas útiles como encontrar algo, limpiar o espiar, el enjambre robótico es un nuevo enfoque en la coordinación de sistemas multirobot, los cuales consisten en un gran número de, en su mayoría, robots simples. Se espera que un comportamiento colectivo emerge de las interacciones entre los robots y de sus interacciones con el medio ambiente.

2.4.2. Desafíos

- Miniaturización
- Bajo costo
- Simplicidad de los individuos del equipo

Tener la posibilidad de usar hardware en la investigación del enjambre robótico en lugar de simulaciones permite encontrar y resolver muchos más problemas y así extender el alcance de la investigación de gran manera. La capacidad de conseguir un bajo costo permite escalar el enjambre, hacer a cada individuo de baja demanda en términos de energía es también importante así como reducir lo más posible el consumo de recursos.

2.4.3. Aplicaciones

- Misiones de rescate en zonas de desastre
- Actividades en minas
- Actividades en agricultura
- Participación en arte interactivo
- Formación de ejércitos autónomos

2.4.4. Hormigas robóticas

Las hormigas robóticas son un caso especial del enjambre robótico y cuentan con características como:

- Ser simples
- Ser baratos
- Tener sensado limitado
- Contar con capacidad computacional
- Presentar comportamiento dirigido por interacciones locales
- Comunicación mediante marcas

Un solo robot hormiga (modelado como una máquina finita de estados) es capaz de simular la ejecución de cualquier máquina arbitraria de Turing. Esto comprueba que un solo robot hormiga, utilizando feromonas, puede ejecutar arbitrariamente algoritmos complejos de un solo robot. Sin embargo esto no se mantiene para n robots.

3.1. Autómata celular: La hormiga de Langton

Con el paso de los años, el comportamiento complejo de un sistema de tipo “simple” ha motivado varios estudios tanto experimentales como analíticos.

Moreira, Gajardo y Goles, interesados en el autómata celular “La hormiga de Langton” han trabajado en colaboración por varios años publicando sus resultados en la última década. “Complexity of Langton’s Ant” [10] es uno de los trabajos donde se aborda exclusivamente la regla de Langton en la malla cuadrículada, presentando en detalle la construcción de circuitos lógicos y con ello mostrando su universalidad computacional. Inician con una descripción general de la hormiga virtual creada por Langton y sus bien conocidos comportamientos, posteriormente dan paso a uno de los principales resultados:

”Para cualquier configuración inicial, la trayectoria de la hormiga es ilimitada.”

Desafortunadamente esto no dice nada más acerca del comportamiento de la hormiga a largo plazo. Así que se planteó la siguiente conjetura:

”Para cualquier configuración inicial con soporte finito, la hormiga, finalmente, comienza la construcción de la “avenida” periódica, en alguna dirección sin obstáculos.”

Entendiendo como soporte finito un número total de celdas finito, iniciando todas y cada una de las celdas en un mismo estado. Si la conjetura resultara ser verdadera, entonces cualquier problema asociado a la hormiga, cuya entrada sea una configuración inicial con soporte finito, es decidible.

Entre su contenido, se muestra una construcción que permite la representación de cualquier circuito booleano como una configuración finita en una cuadrícula. Se muestra cómo a partir de un circuito dado se construye la configuración. Además se detalla la reducción del problema P completo a un problema concreto en la dinámica de la hormiga, y se deduce la universalidad computacional, a través de la simulación de una máquina de Turing universal.

Un par de años más tarde, estos autores trabajaron en un nuevo estudio titulado ‘Generalized Langton’s ant: Dynamical Behavior and Complexity’ [11]. Donde podemos encontrar la construcción de circuitos lógicos, un trabajo similar al estudio referenciado en líneas anteriores, pero aquí se presenta de una forma más general que

la hace aplicable también a una malla hexagonal, y además a todo grafo de grado 3 o 4. Además, se demuestra que, en particular en grafos de grado 5 o superior, la hormiga construirá siempre una “avenida”.

Treinta años después de ser inventada, la hormiga de Langton ha sido objeto de estudio de diversos artículos de divulgación, sin embargo, observamos que, a pesar de las valiosas contribuciones científicas a la regla, este autómata carece de un estudio global del sistema, es decir, un estudio del comportamiento de la regla a partir de un análisis de las estructuras básicas de la misma.

3.2. Comportamiento colectivo implementado en robots

A continuación, se muestra una rápida revisión acerca del desarrollo de robots que simulan el fenómeno referido como auto organización. Enfocado en un ejemplo particularmente interesante desarrollado en la universidad de Harvard. Además, se presenta un prototipo desarrollado en la Escuela Superior de Cómputo en colaboración con la University of the West of England en el Reino Unido.

En agosto del año 2014, se dio a conocer el primer lote de robots ensamblados en la Universidad de Harvard, que siguiendo un conjunto de reglas programadas simples, los robots autónomos se organizan en grandes formas complejas. Este enjambre fue creado en el laboratorio de Radhika Nagpal, Fred Kavli Profesor de Ciencias de la Computación de la Facultad de Ingeniería y Ciencias Aplicadas (SEAS) de Harvard. Lo magnífico de los sistemas biológicos es que son elegantemente simples y, sin embargo, logran lo que parece imposible. En cierto nivel, se vuelve imposible distinguir a cada individuo, ya que mediante la cooperación de cada elemento básico se logra una entidad única en sí.

El kilobot fue diseñado para realizar pruebas de algoritmos colectivos autoorganizantes. En teoría, el Kilobot está destinado a simular un enjambre de insectos, en la que cada Kilobot trabaja con el conjunto para llevar a cabo tareas que no funcionarían a un nivel individual. Entre ellos se comunican emitiendo señales que les ayudan a interpretar su posición con respecto al resto mientras se desplazan. Pueden comunicarse con sus compañeros si están a siete centímetros de distancia, detectan la ubicación de los demás utilizando la luz infrarroja. Se trasladan usando motores de vibración para deslizarse a través de una superficie sobre sus tres patas. La técnica en la que se basa la dinámica de estos robots es la implementación de algoritmos genéticos, que a través de la evolución logran simular el comportamiento colectivo, comenzando a comportarse como un todo.

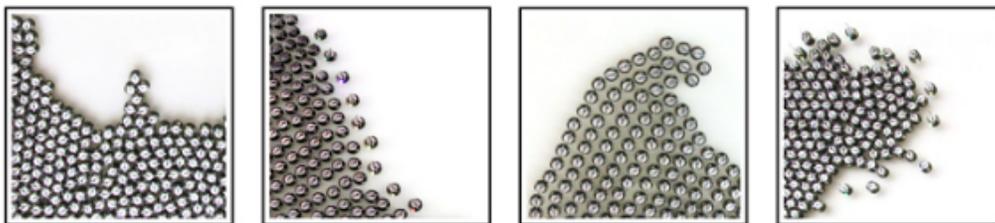


Figura 3.1: Enjambre de robots sencillos interactuando [12].

En la actualidad, el Centro de Computación de Sistemas Complejos está trabajando en la construcción de prototipos de minirobots de bajo costo.

3.2. COMPORTAMIENTO COLECTIVO IMPLEMENTADO EN ROBOTS

En la Escuela Superior de Cómputo se han realizado esfuerzos para lograr algunas implementaciones, estos proyectos han sido desarrollados bajo la tutoría del Dr. Luz Noé Oliva Moreno y el Dr. Genaro Juárez Martínez en conjunto.

El primer prototipo de robot de bajo costo que se llevó a cabo en la institución, fue el eBot, desarrollado por la alumna Rosa Graciela Chávez Barrera, durante el año 2015, quién en colaboración con los profesores buscaban simular un enjambre de robots. El proceso consistió en realizar una adaptación de la regla, para poderla aplicar a los robots, y posteriormente con dicha adaptación, observar el comportamiento colectivo, proyectando la población de la colonia a través de un simulador y reflejando así el comportamiento esperado (swarm).

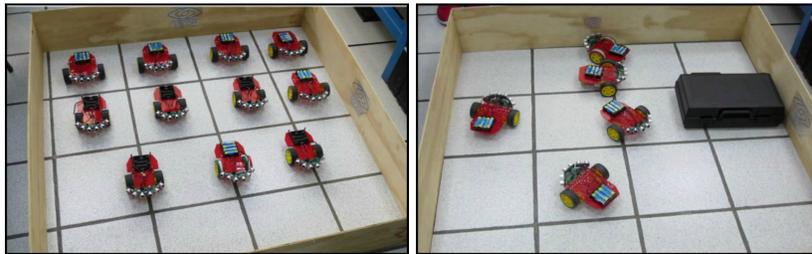


Figura 3.2: Colonia de eBots en la ESCOM [14].

La figura 3.2 muestra el conjunto de eBots construidos en la institución intercтуando. Entre los elementos principales que componen cada móvil se encuentra el microcontrolador MSP430G253, un conjunto de sensores ultrasónicos, dos motores convencionales y cuatro baterías recargables AA. Fueron ensamblados un total de 15 robots, desafortunadamente con esta cantidad de entidades es imposible la visualización del comportamiento colectivo, así que se construyó un simulador para la proyección de la colonia.

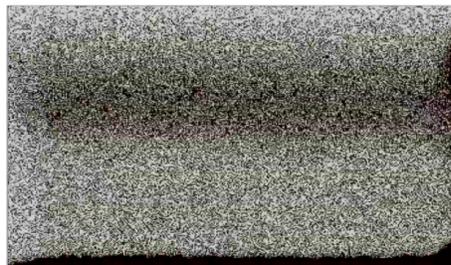


Figura 3.3: Simulación de eBots [14].

En la figura 3.3 se observa la simulación correspondiente a la regla. Es notable que a pesar de las adaptaciones, si la colonia de robots incrementara notablemente, sería evidente el enjambre de robots.

Análisis del simulador de la Hormiga de Langton

Una vez realizada la investigación documental, se procede con el análisis del simulador, el cual será la herramienta utilizada para realizar el trabajo de investigación acerca del comportamiento de “La hormiga de Langton” así como la simulación del comportamiento de la colonia de robots.

En el presente capítulo se muestran:

- Análisis de las tecnologías a usar.
- Análisis de riesgos.
- Requerimientos Funcionales.
- Requerimientos no funcionales.
- Reglas del negocio.

4.1. Análisis de riesgos

En este capítulo se determinan los riesgos potenciales que se podrían presentar durante el desarrollo del trabajo terminal, dichos riesgos tendrán asignada una prioridad, se describen los planes de contingencia para dichos riesgos y finalmente se valorarán para analizar si los riesgos pueden ser eliminados.

4.1.1. Identificación de riesgos

A continuación se muestran los riesgos identificados, los cuales se clasifican de la siguiente manera:

4.1.1.1. Riesgos de la curva de aprendizaje

A continuación se enlistarán los riesgos que pudieran existir con el proceso de la curva de aprendizaje de las tecnologías y conocimientos requeridos para el desarrollo del presente trabajo terminal.

- El desconocimiento del framework Qt como principal herramienta de desarrollo del simulador.
- El desconocimiento en el área de física, mecánica y robótica.

4.1.1.2. Riesgos de los integrantes del equipo.

A continuación se enlistarán los riesgos asociados con las personas del equipo de desarrollo.

- Que algún integrante el equipo esté enfermo y no disponible en momentos críticos.
- Si se presenta la baja de algún integrante del equipo en pleno desarrollo de la tesina.

4.1.2. Valoración de los riesgos.

A continuación se realiza la valoración de los riesgos, como se muestra en la siguiente tabla, dichos riesgos han sido ordenados según su impacto.

Riesgo	Probabilidad	Efecto
El tiempo y número de integrantes para realizar el trabajo terminal no son suficientes	Moderado	Serio
Desconocimiento del framework a utilizar en el desarrollo del simulador	Moderado	Catastrófico
Desconocimiento en el área de física, mecánica y robótica.	Alta	Catastrófico
La ausencia de un integrante del equipo por algún motivo.	Bajo	Tolerable
La ausencia de uno de los directores por algún motivo.	Bajo	Tolerable
Tiempo insuficiente para la revisión y estudio del marco teórico.	Moderado	Serio

Tabla 4.1: Valoración de riesgos.

4.1.3. Administración de los riesgos

Después de realizar la valoración de los riesgos potenciales, se proponen estrategias para administrarlos

Riesgo	Estrategia
El tiempo y número de integrantes para realizar el trabajo terminal no son suficientes	Seguir el cronograma rigurosamente y adoptar una estrategia de organización y manejo del tiempo
Desconocimiento del framework a utilizar en el desarrollo del simulador.	Documentarnos y estudiar así como practicar en la mayor medida posible el manejo del framework.
Desconocimiento en el área de física, mecánica y robótica.	Estudiar bibliografía referente a dichos temas.
La ausencia de un integrante del equipo por algún motivo emergente.	Acotar el alcance del trabajo terminal en caso de sufrir la pérdida de un miembro del equipo.
La ausencia de uno de los directores por algún motivo emergente.	En caso de ser necesario, buscar un asesor externo con conocimiento en el tema.
Tiempo insuficiente para la revisión y estudio del marco teórico.	Revisar solamente aportaciones relevantes y recientes en el área.

Tabla 4.2: Administración de riesgos.

4.2. Herramientas

Para la elección de las tecnologías que se ocuparán para el desarrollo del simulador, se realizaron comparaciones de las tecnologías y herramientas existentes, posteriormente se eligieron las que mejor se adecuan a las necesidades del proyecto, los resultados se muestran a continuación:

4.2.1. Lenguaje para desarrollo del simulador

La característica más importante a considerar es la del manejo de gráficos, es por ello que se optó por implementar el simulador en un lenguaje orientado a objetos:

CARACTERÍSTICA	JAVA	PHYTON	C++ (Qt)
Multiplataforma	X	X	X
Portabilidad	Alta	Alta	Media
Compilación	Compilado	Interpretado	Compilado
Gráficos	Amplio manejo de gráficos.	Amplio manejo de gráficos.	Amplio manejo de gráficos con diversas librerías disponibles.

Tabla 4.3: Comparación de lenguajes orientados a objetos.

La plataforma elegida para el desarrollo del proyecto es **C++ utilizando el framework Qt**, ya que la implementación en C++ es relativamente sencilla y el framework Qt cuenta con una gran cantidad de librerías orientadas a manejo de gráficos.

4.2.1.1. Framework Qt

En la siguiente tabla, se muestran algunas características y especificaciones actuales de Qt.

CARACTERÍSTICA	QT
Modelo de desarrollo	Software libre y de código abierto
Última versión estable	5.7
Programado en:	C++
Sistema operativo	Multiplataforma

Tabla 4.4: Algunas características de QT

4.3. Requerimientos del simulador

4.3.1. Definición de actores

A continuación se realiza la especificación del actor que tendrá interacción con el simulador. Debido a que el simulador es de propósito específico, existe un único usuario final.

4.3.1.1. Usuario final

Nombre: Usuario final.

Descripción: Persona que hace uso del simulador con el único objetivo de llevar a cabo pruebas que sean útiles en el proceso de investigación así como simulaciones de las interacciones posibles en la colonia de robots.

Tareas. Las tareas del usuario final son:

- Iniciar simulación.
- Guardar configuración.
- Abrir configuración.
- Editor configuración.
- Guardar imagen de la simulación.

Perfil: Conocimiento en el manejo de la computadora y la dinámica del autómeta celular “La hormiga de Langton”.

4.3.2. Requerimientos funcionales

A continuación se muestra cada uno de los requerimientos funcionales para el simulador, especificando para cada uno de ellos: un identificador único con el que se hará referencia a los mismos dentro del documento.

Requerimientos funcionales

- **RF_01.** El usuario podrá elegir mediante el puntero la posición de hormigas.
- **RF_02.** El usuario podrá elegir entre hormigas biológicas y robóticas.
- **RF_03.** El usuario podrá elegir una selección aleatoria de hormigas.
- **RF_04.** El usuario podrá guardar una configuración de simulación.
- **RF_05.** El usuario podrá abrir una configuración de simulación.
- **RF_06.** El usuario podrá editar una configuración de simulación.
- **RF_07.** El usuario podrá guardar una imagen de la simulación.
- **RF_08.** El usuario podrá visualizar una gráfica de densidades.

4.3.3. Requerimientos no funcionales

A continuación se muestran los requerimientos no funcionales, especificando para cada uno de ellos un identificador único con el cual se hará referencia a los mismos dentro del documento.

4.3.3.1. Tecnologías de desarrollo

RNF_01. Lenguajes de desarrollo.

Para el desarrollo de la simulación, se hará uso de los siguientes lenguajes:

Base: C++ como lenguaje de desarrollo.

Framework: Qt en la parte gráfica del simulador.

4.3.3.2. Usabilidad

RNF_02. Diseño de la interfaz El simulador deberá contar con una interfaz minimalista, la cual permita el rápido acceso a las herramientas necesarias.

4.3.3.3. Sistema Operativo

RNF_03. Sistema operativo. La aplicación podrá de ser utilizada en Windows 7 o superior.

4.3.3.4. Mantenibilidad

RNF_04. Se debe considerar el paradigma de programación orientada a objetos así como buenas prácticas de programación que permitan y faciliten el mantenimiento necesario en el futuro. Se proporcionará también la documentación técnica completa del simulador.

4.3.3.5. Requerimientos de Hardware

RNF_05. La aplicación no tendrá problemas de ejecución mientras se cuenten con los siguientes requerimientos básicos:

Procesador. Intel® Core® i3-2450M CPU @ 2.50GHz 2.50 GHz

RAM. 2.00 GB.

Disco duro. Desde 250 GB. **Tipo de sistema.** Desde 32 bits.

Sistema operativo. Windows 7 o superior.

4.3.4. Reglas de negocio

Regla de Negocio: RN01. Inicio a la simulación.

Descripción: Para dar inicio a la simulación será necesario al menos:

- Seleccionar un caso de estudio en el plano, su color, dirección y si es de tipo biológica o robótica.
- Si se prefiere, seleccionar una configuración aleatoria.

Regla de Negocio: RN02. Formato de la configuración.

Descripción: El archivo seleccionado para abrir una configuración debe corresponder al formato siguiente:

- Archivo de texto plano
- Contenido con el formato necesario.

Diseño del simulador de la Hormiga de Langton

Una vez concluida la etapa de análisis se procede con el diseño y arquitectura del simulador.

En este capítulo se muestra:

- El diagrama de casos de uso correspondiente al actor del simulador para el autómata celular “La hormiga de Langton” .
- La documentación de los casos de uso de alta prioridad.
- Diseño del algoritmo de la regla de Langton.
- Diseño y documentación de pantallas.
- Diagrama de secuencia.

5.1. Vista de escenarios

5.1.1. Diagrama de casos de uso del actor Usuario Final para la Regla original

En la figura 5.1 se muestra el diagrama de casos de usos general para el Usuario final.

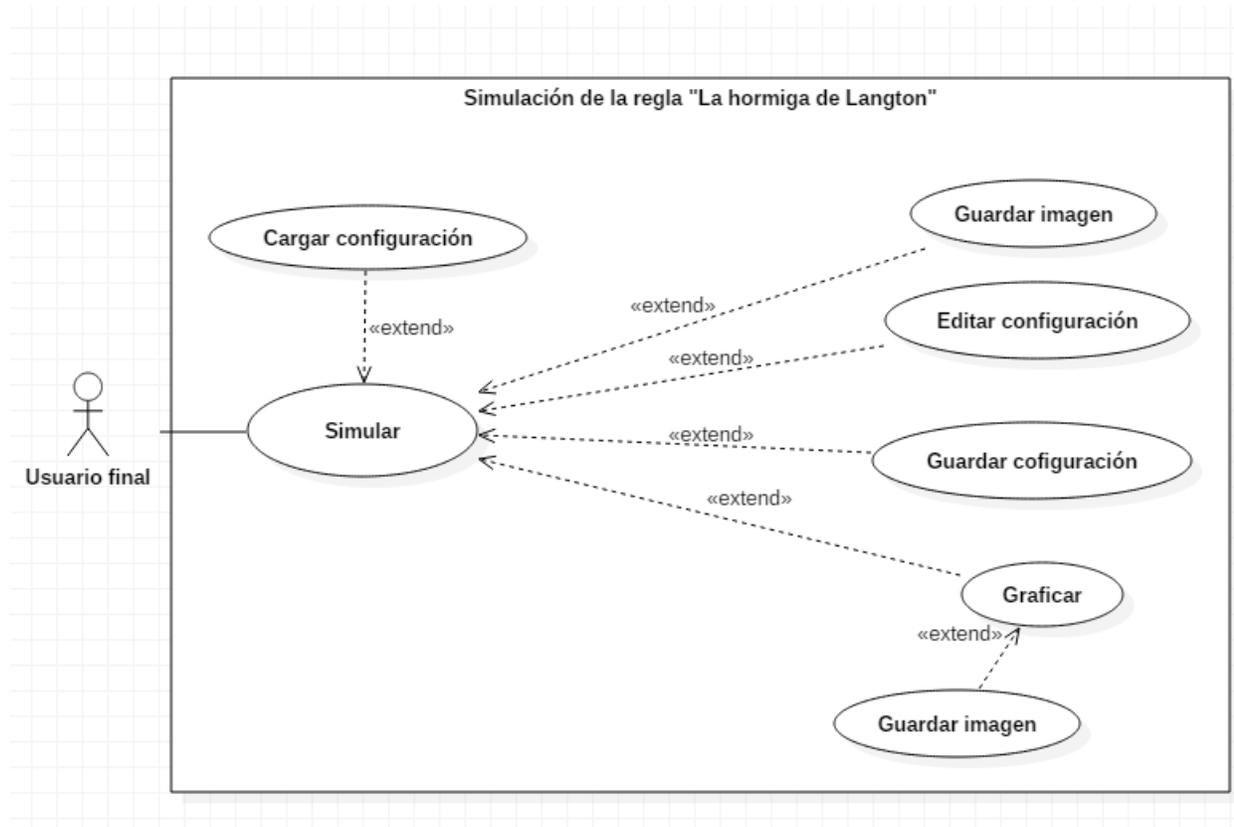


Figura 5.1: Diagrama de casos de uso del simulador para la regla original de la hormiga de Langton.

5.1.2. Documentación del diagrama de casos de uso general del simulador.

A continuación se muestra la documentación del diagrama de casos de uso general propuesto para el simulador.



5.1.2.1. CU01 Simular

Descripción completa.

Este caso de uso tiene como finalidad simular el algoritmo de “La hormiga de Langton”.

Atributos importantes.

Caso de Uso:	CU01 Simular
Actor:	Usuario final
Objetivo:	Realizar una simulación de “La hormiga de Langton”.
Entradas:	<ul style="list-style-type: none"> ● Sujeto de estudio. Se selecciona con el ratón. ● Naturaleza del sujeto de estudio. Se selecciona con el ratón. ● Color del sujeto de estudio. Se selecciona con el ratón. ● Orientación del sujeto de estudio. Se selecciona con el ratón.
Salidas:	Animación del algoritmo.
Pre-condiciones:	El actor debe seleccionar al menos una hormiga o cargar una configuración.
Post-condiciones:	El actor podrá realizar las funciones permitidas sobre la simulación.
Reglas de negocio:	RN01 Iniciar simulación. RN02 Formato del archivo.
Errores:	<ul style="list-style-type: none"> ● MSG1 Ausencia de hormigas. Se muestra en la pantalla IUS01 Iniciar configuración a simular cuando no se ha seleccionado un caso de estudio. ● MSG12 Formato incorrecto. Se muestra en la pantalla IUS01 Iniciar configuración a simular cuando el formato del archivo a cargar no es el correcto.
Tipo:	Primario.

Trayectorias del Caso de Uso

Trayectoria principal: Principal

- 1  Selecciona simular de acuerdo a la regla original.
 - 2  Muestra la pantalla IUW01 Iniciar configuración a simular.
 - 3  Selecciona al menos una hormiga así como su color, orientación y naturaleza. IUW01 Iniciar configuración a simular.
 - 4  Verifica que se haya seleccionado al menos una hormiga de acuerdo a la regla de negocio RN01 Inicio a la simulación. [Trayectoria A]
 - 5  Verifica que el archivo de carga tenga el formato correcto de acuerdo a la regla de negocio RN02 Formato del archivo. [Trayectoria B]
 - 6  Selecciona iniciar la simulación.
 - 7  Muestra la pantalla IUS02 Visualizar Simulación.
 - 8  Selecciona guardar configuración.
 - 9  Selecciona guardar imagen de la simulación.
 - 10  Selecciona graficar de acuerdo a la simulación resultante.
 - 11  Muestra la pantalla IUS03 Visualizar diagrama de frecuencias.
 - 12  Selecciona guardar imagen del diagrama de frecuencias.
- - - - *Fin del caso de uso.*

Trayectoria alternativa A:

Condición: El actor no ingreso un objeto de estudio.

- A1  Muestra el mensaje de error MSG1 FAusencia de hormigas.
 - A2  Regresa al paso 1 de la trayectoria principal.
- - - - *Fin de la trayectoria.*

Trayectoria alternativa B:

Condición: El actor ingresó archivo de carga con un formato incorrecto.

- B1  Muestra el mensaje de error MSG2 Formato incorrecto.
 - B2  Regresa al paso 1 de la trayectoria principal.
- - - - *Fin de la trayectoria.*

5.2. Diseño del algoritmo de la regla de Langton

5.2.1. Pseudocódigo: Regla de la hormiga de Langton

Algorithm 1 Regla de Langton

Require: Coordenada actual de la hormiga = $\{x, y\}$ y posición apuntando a uno de los cuatro puntos cardinales.

Ensure: Pixmap

```
1: while contadorGeneraciones < totalGeneraciones do
2:   Verifica la coordenada  $\{x, y\}$  en donde se encuentra posicionada la hormiga.
3:   if celda ha sido visitada y está encendida then
4:     Gira 90 a la izquierda
5:   else
6:     Gira 90 a la derecha
7:   end if
8:   Invierte el color de la celda
9:   Establece nueva coordenada
10:  Establece nueva posición
11: end while
12: return Pixmap
```

El algoritmo 1 muestra los giros condicionados para una hormiga que se mueve sobre un arreglo regular.

5.2.2. Diagrama de flujo del movimiento de una hormiga

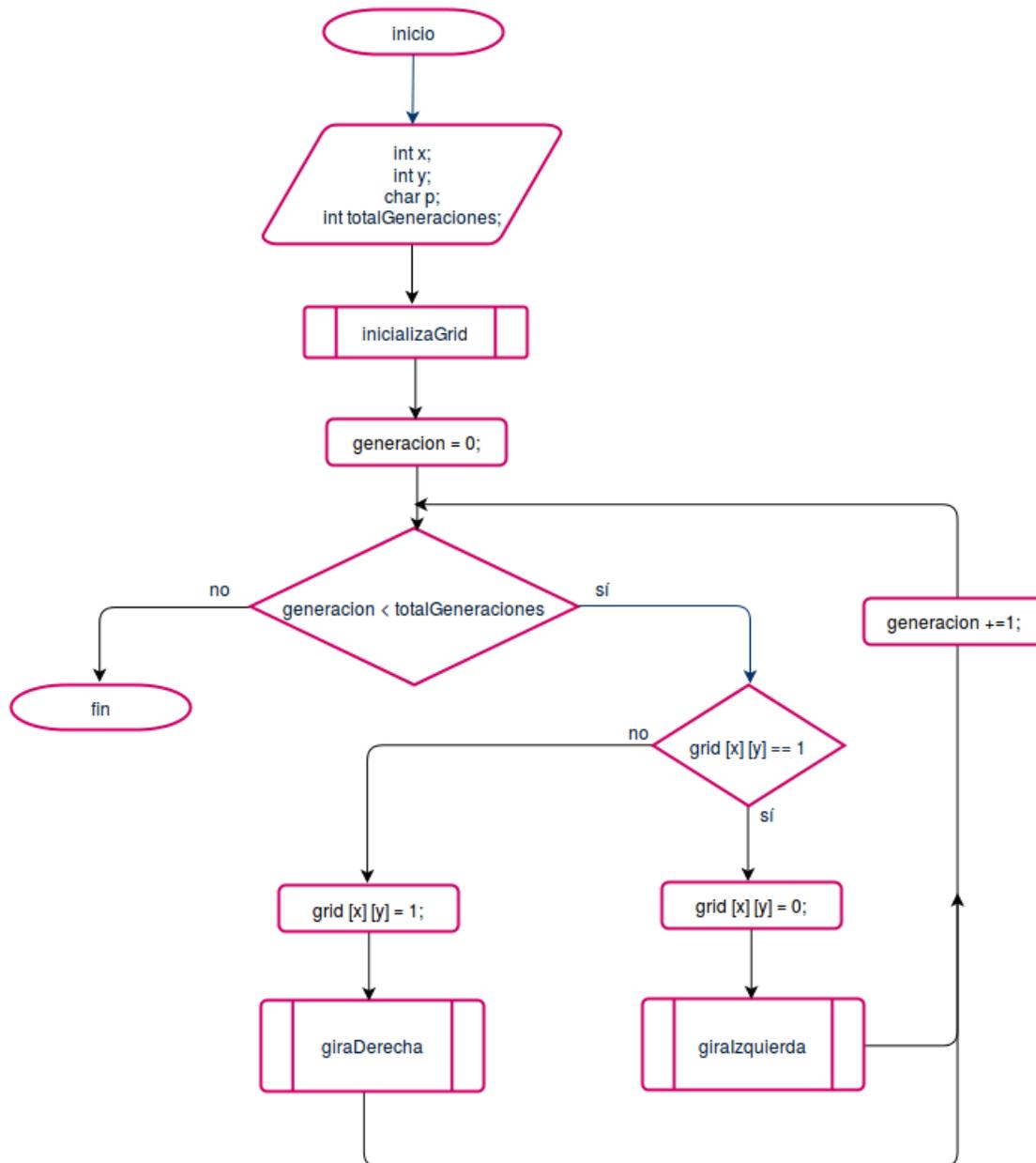
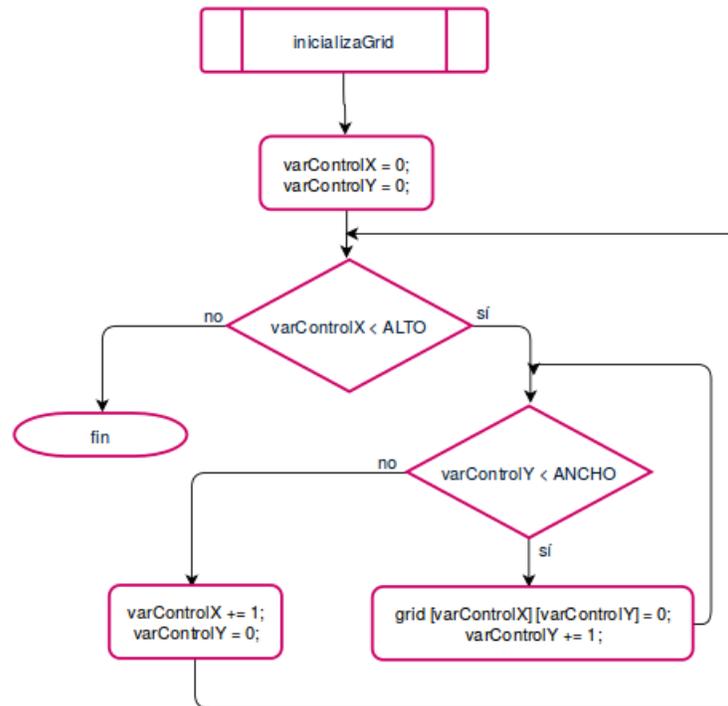


Figura 5.2: Flujo que sigue una hormiga en una rejilla de tamaño ALTO * ANCHO.

5.2.2.1. Subrutinas correspondientes al algoritmo de la regla de Langton



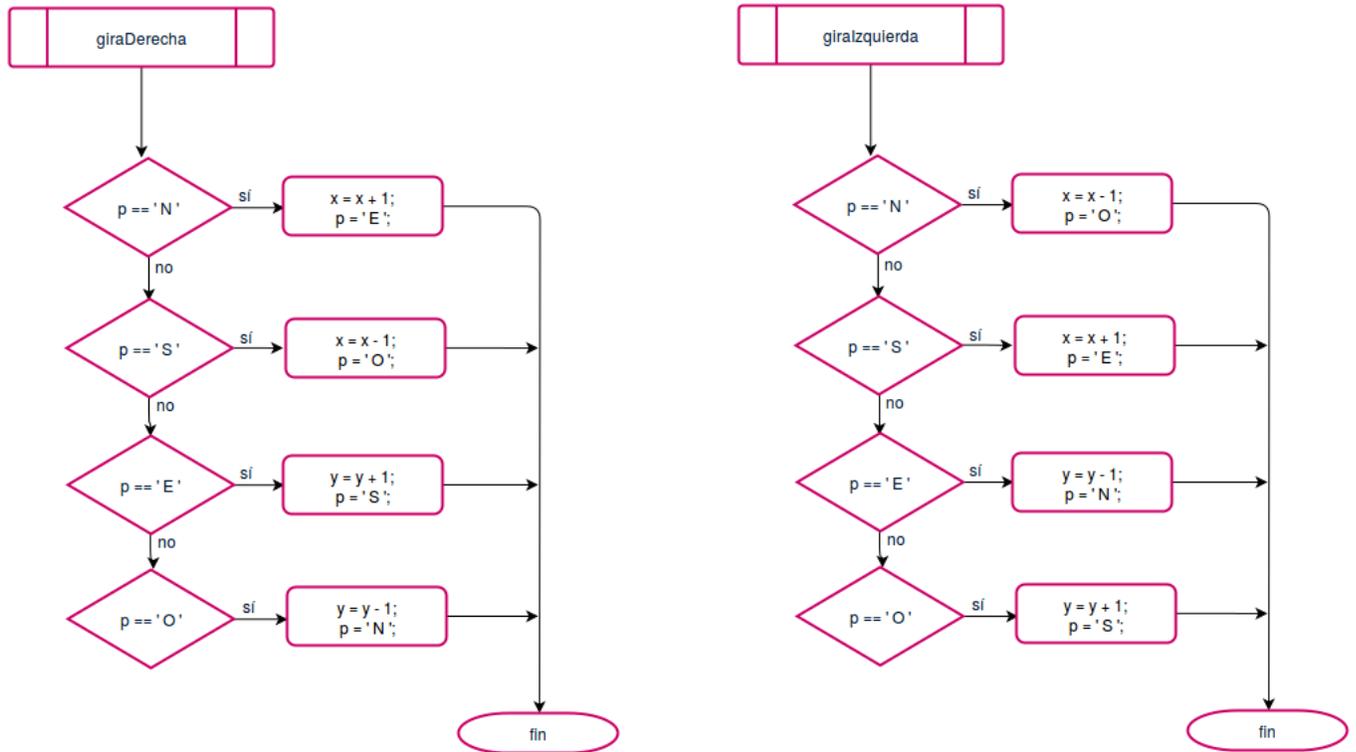


Figura 5.3: Subrutinas básicas requeridas por el algoritmo.

5.2.2.2. Subrutinas correspondientes al algoritmo de los antBot

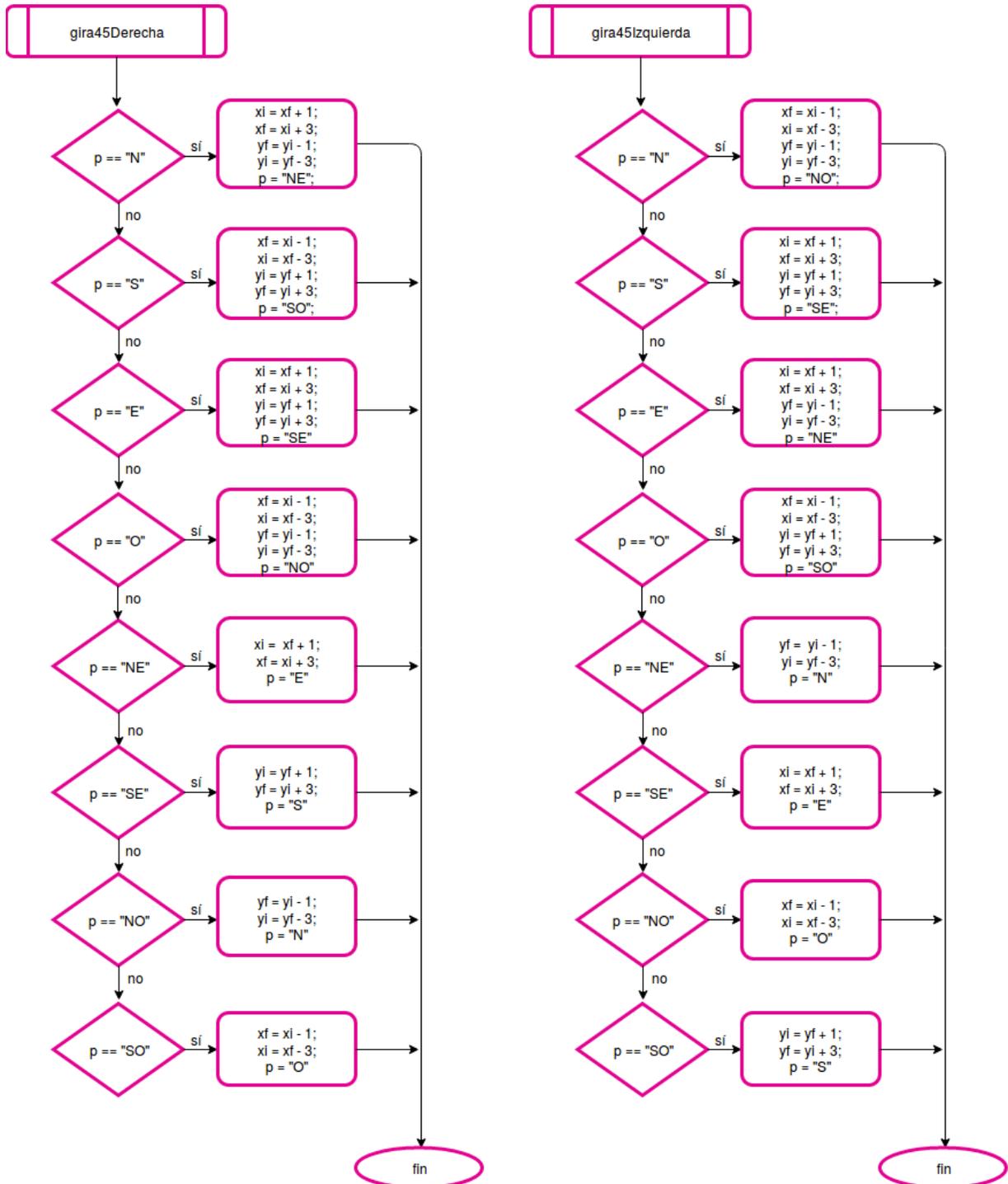


Figura 5.4: Subrutinas requeridas por el algoritmo adaptado.

5.3. Diseño de pantallas

5.3.1. Catálogo de menús

5.3.1.1. MENU01. Acciones correspondientes a la configuración

Objetivo

Este menú permite al actor Usuario Final, seleccionar una opción según sus necesidades. Es posible guardar la configuración en un archivo txt, exportar la configuración como imagen o reiniciar el espacio.

Diseño

En la Figura 5.5 se muestra la pantalla MENU01 “Configuración Inicial”.

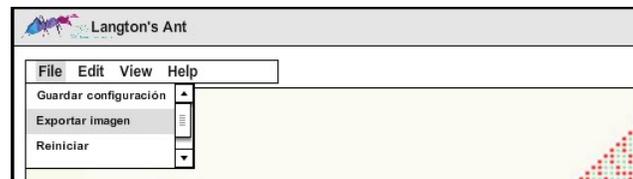


Figura 5.5: MENU01. Acciones correspondientes a la configuración.

5.3.1.2. MENU02. Editar configuración

Objetivo

Este menú permite al actor Usuario Final editar de alguna manera la configuración inicial. Es posible inicializar el espacio con una configuración aleatoria, editar el estado de alguna celda o reiniciar el espacio.

Diseño

En la Figura 5.6 se muestra la pantalla MENU02 “Editar configuración”.

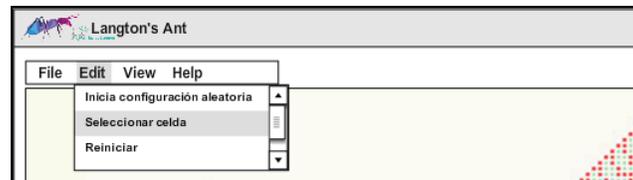


Figura 5.6: MENU02. Editar configuración.

5.3.1.3. MENU03. Vista de simulación

Objetivo

Este menú permite al actor Usuario Final definir algunas características correspondientes a la vista de la simulación, tales como el color de fondo del grid y el borde de cada celda.

Diseño

En la Figura 5.7 se muestra la pantalla MENU03 “Vista de simulación”.

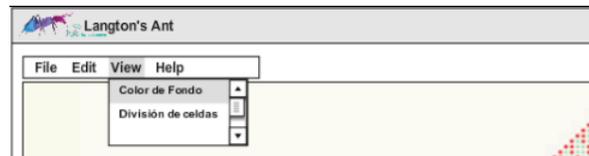


Figura 5.7: MENU03. Vista de simulación.

5.3.2. Interfaz del simulador

A continuación, se presenta el diseño de las pantallas para nuestra herramienta de análisis del autómata celular.

5.3.2.1. IUS01 Iniciar configuración a simular

Objetivo

Esta pantalla permite al actor Usuario Final, seleccionar la posición de la hormiga, así como un color asociado a cada camino para la simulación.

Diseño

En la Figura 5.8 se muestra la pantalla IUS01 “Configuración Inicial”.

Controles

Ninguno.

Comandos

Configuración Inicial: Permite al actor seleccionar los parámetros de configuración de cada hormiga de acuerdo a sus necesidades.

Mensajes

- Naturaleza por defecto: Biológica
- Posición por defecto: Norte
- Color por defecto: Rosa.

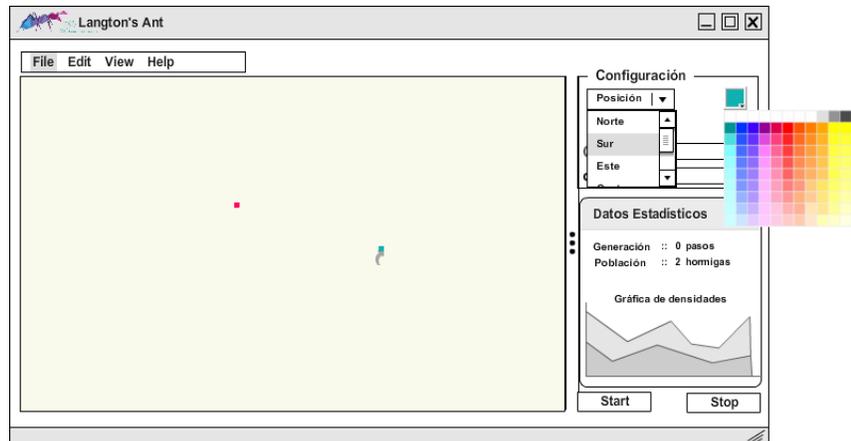


Figura 5.8: IUS01. Iniciar configuración ha simular.

5.3.2.2. IUS02. Visualizar Simulación

Objetivo

Esta pantalla permite al actor Usuario Final visualizar la dinámica de las hormigas al paso de las generaciones.

Diseño

En la Figura 5.9 se muestra la pantalla IUS02 “Visualizar Simulación”.

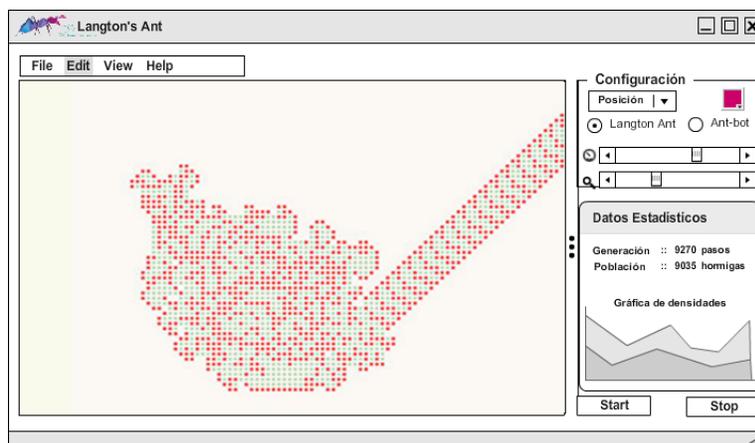


Figura 5.9: IUS02. Visualizar Simulación

Controles

Ninguno.

Comandos

Ninguno.

Mensajes

Ninguno

5.3.2.3. IUS03. Visualizar Diagrama de Frecuencias

Objetivo

Esta pantalla permite a los actores Usuario Final observar de manera detallada la dispersión de la población de hormigas con el paso de las generaciones.

Diseño

En la Figura 5.10 se muestra la pantalla IUS03 “Visualizar diagrama de frecuencias”.

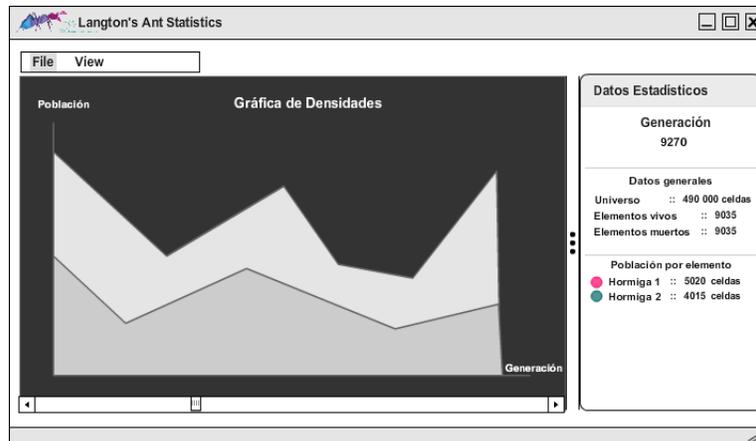


Figura 5.10: IUS03. Visualizar diagrama de frecuencias

Controles

Ninguno.

Comandos

Registrar: Permite al actor realizar el registro de densidades.

Mensajes

- Ninguno

5.3.3. Diagrama de clases.

A continuación se muestra el diagrama de clases correspondiente al diseño del simulador.

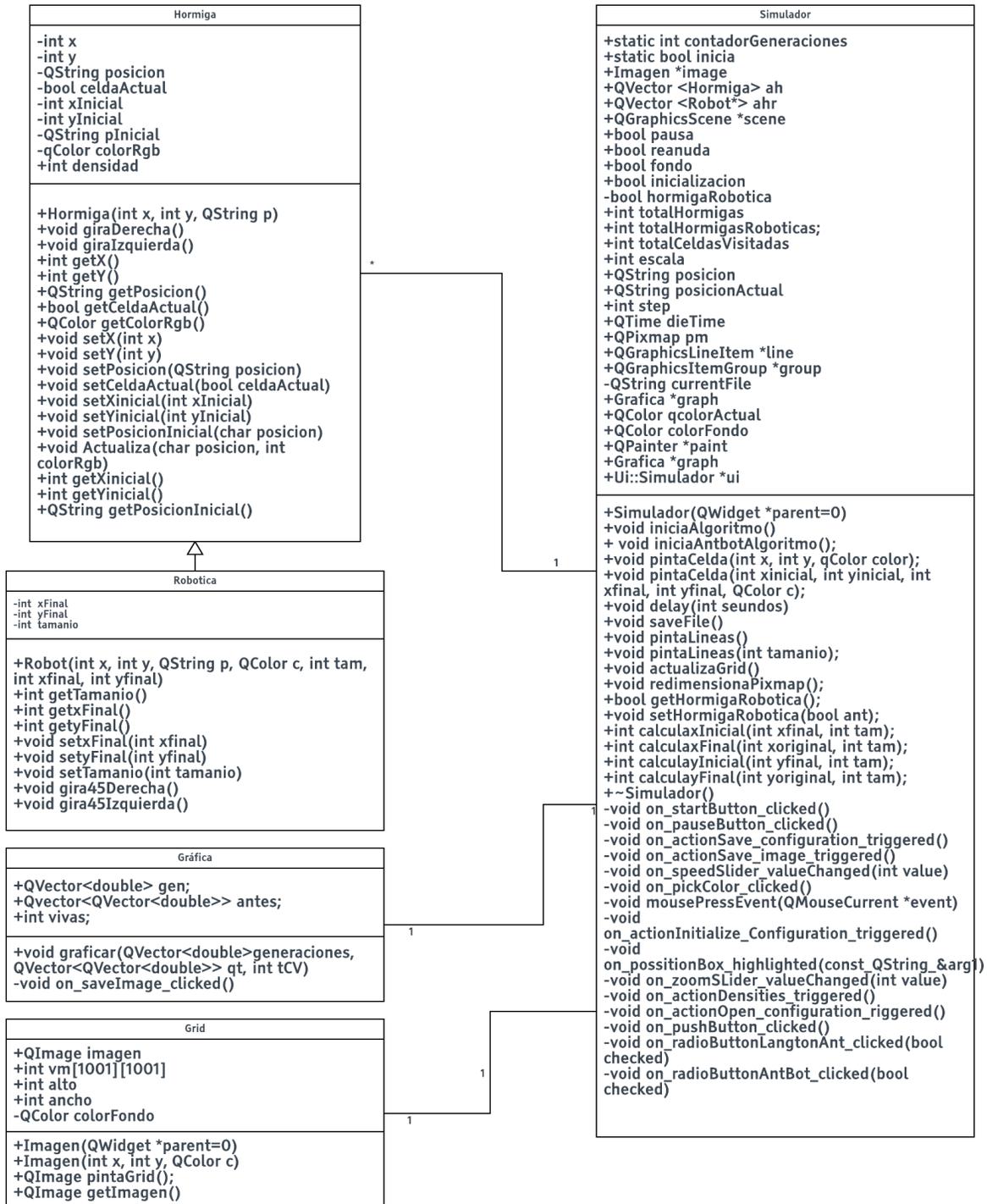


Figura 5.11: Diagrama de clases general del simulador.

5.3.4. Diagrama de secuencia.

A continuación se muestra el diagrama de secuencia correspondiente al flujo general del simulador.

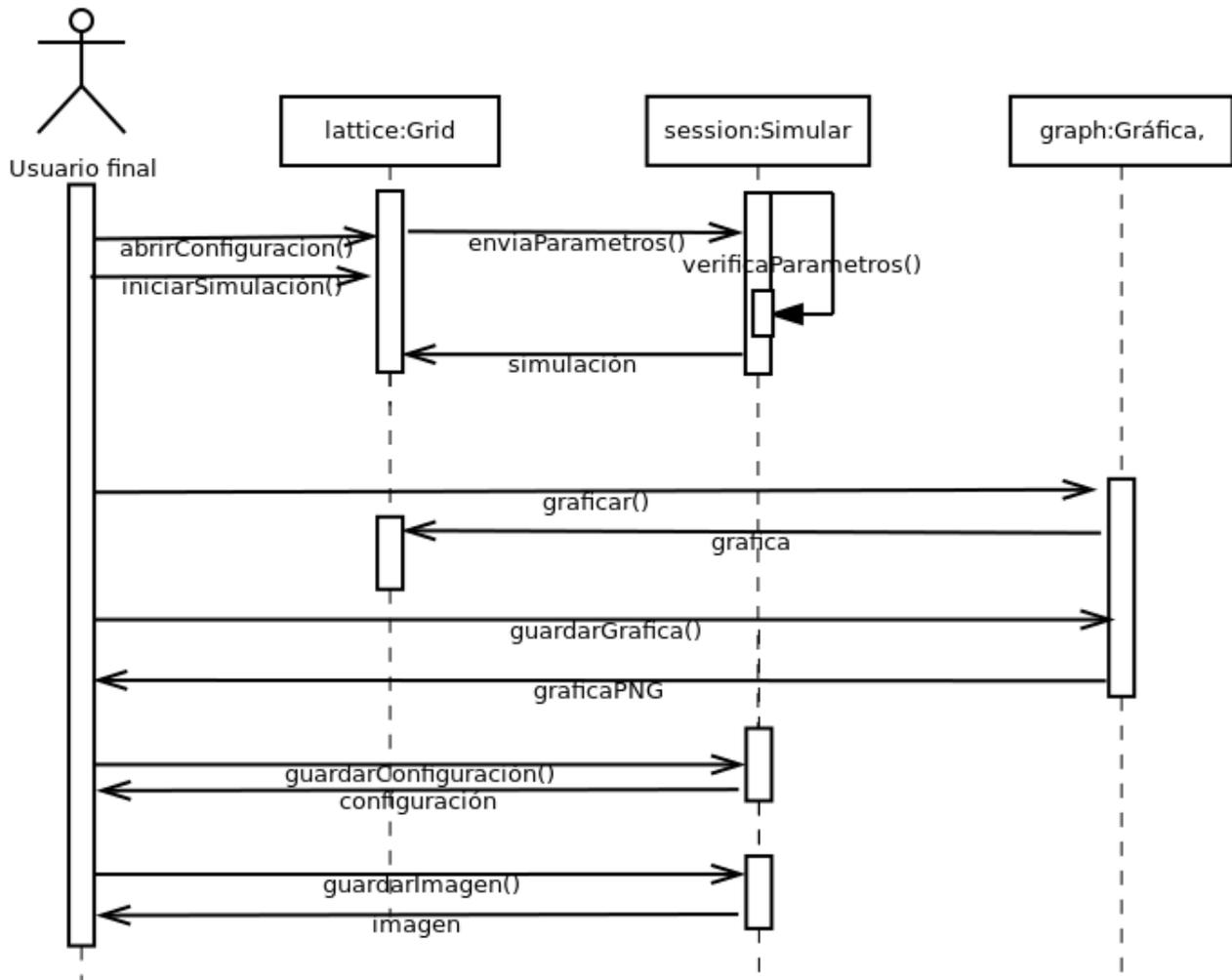


Figura 5.12: Diagrama de secuencia general del simulador.

La Figura 5.12 muestra la secuencia principal del simulador, iniciando desde arriba cuando el Usuario final decide comenzar una simulación, la cual puede ser también cargada desde un archivo de texto. La secuencia continua al ser los parámetros enviados y verificados por el simulador el cual devuelve la animación correspondiente en el caso de recibir datos validos. El Usuario final tiene la opción de graficar las densidades del universo en el plano así como guardar la configuración actual y la imagen de esta.

6.1. Diagrama de módulos del prototipo del robot

En la figura 7.1 se muestra el diagrama de módulos del robot que será programado con las reglas de la hormiga de Langton.

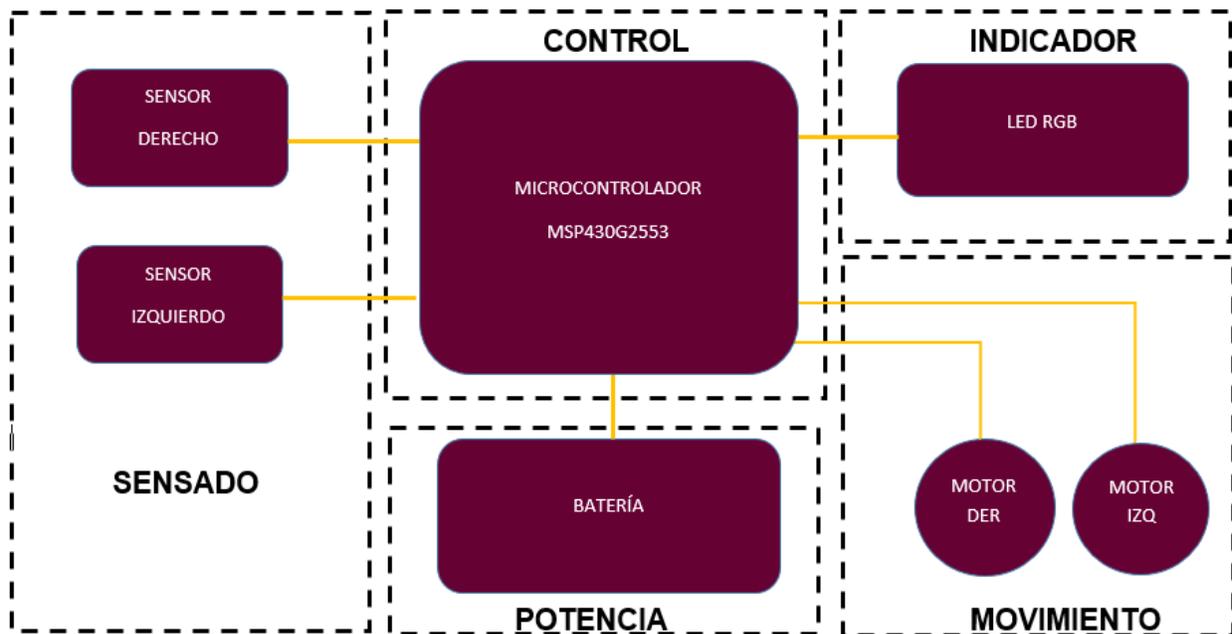


Figura 6.1: Diagrama de módulos del robot

6.2. Circuito eléctrico del robot

Con la finalidad de simular el comportamiento colectivo no trivial presentado en “La hormiga de Langton” se ensamblaran, aproximadamente, 20 robots.

El robot cuenta con un microcontrolador, una etapa de sensado (S1 y S2) de proximidad de objetos y hace uso de 2 motores de vibración (M1 y M2) que le permitirán el desplazamiento hacia enfrente, atrás, derecha e izquierda, además de poder indicar por medio de un led RGB su desplazamiento.

En la figura 7.44 se muestra el diseño del circuito eléctrico base del robot.

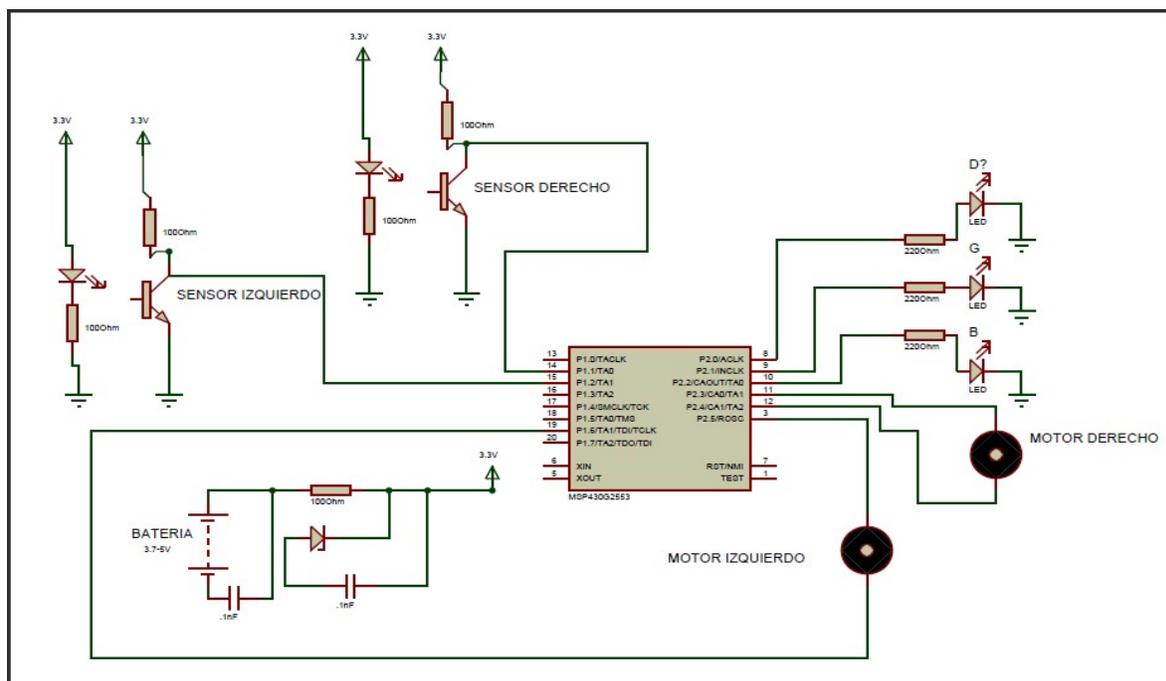


Figura 6.2: Circuito del robot

6.2.1. Descripción de los bloques



Con la finalidad de explorar su entorno, cada robot contará con dos sensores infrarrojos ubicados a cada lado del dispositivo. Esto le permitirá conocer su proximidad a objetos con el fin de evitar obstáculos. Debido a sus dimensiones (2.90mm por 3.60mm), el dispositivo propuesto es el sensor infrarrojo QRE1113, con una distancia óptima de sensado de 0.125 pulgadas equivalente a 3mm.

El sensor QRE1113 cuenta con una salida analógica, que puede variar dependiendo de la cantidad de luz infrarroja reflejada de vuelta al sensor. En la figura 7.44 se muestra la conexión de los dos sensores así como la entrada al microcontrolador, la cual se da a través del puerto ADC.

6.2.1.2. Control



Este bloque es el más importante pues se refiere al cerebro del robot, para este bloque se utilizará el microcontrolador MSP430g255. Se seleccionó dicho integrado debido a la familiaridad con su arquitectura además del bajo costo y su accesible programación.

Será utilizado el microcontrolador de 20 pines, ofreciéndonos dos puertos con 8 pines disponibles cada uno.

6.2.1.3. Indicador



Con el propósito de realizar algunas pruebas y obtener ciertos datos, así como implementar banderas, el robot contará con un led RGB smd, con una altura de 1.6mm y una longitud de 5.4mm. el cual tiene la capacidad de contener 3 colores distintos.

6.2.1.4. Movimiento



Debido al tamaño necesariamente pequeño de robot que exige el proyecto se ha optado por utilizar dos micro motores de vibración los cuales permitirán el movimiento y los giros requeridos en la implementación del algoritmo.

6.2.1.5. Potencia



Para el funcionamiento del robot, la energía necesaria que debe suministrarse al microcontrolador es de 3.3 V. A su vez, éste dispositivo se encargará de alimentar al resto de los componentes.

Los materiales considerados para satisfacer las necesidades del robot, así como su costo se muestran en la siguiente tabla. La cantidad de robots a realizar se ha fijado en 20.

COMPONENTE	PRECIO UNITARIO	CANTIDAD CONTEMPLADA
Circuito impreso	\$20	x20
Sensor infrarrojo QRD1114	\$5.5	x40
Resistencias	\$0.07	x180
Micro motor de vibración	\$37	x40
Micro controlador MSP430g255	\$45	x20
Led RGB	\$7	x20

Tabla 6.1: Tabla de componentes

Precio unitario de cada miembro de la colonia : \$ 137.

Costo total del enjambre: \$ 2740.

Cabe resaltar que los gastos implicados en la construcción de la colonia de robots son subsidiados por el proyecto de investigación titulado: Plataforma de robots móviles para el comportamiento colectivo no trivial.

6.3. Programación del robot

El cerebro del robot es el microcontrolador MSP430G255, dicho componente es programado a través de la tarjeta Launchpad en lenguaje C con asistencia de la suite de Texas Instruments: Code Composer.

Energía

LaunchPad with MSP430G2452
Revision 1.5

Flash 8 KB
Serial TimerSerial

+3.3V				1
RED_LED		A0	P1.0	2
	TXD	A1	P1.1	3
	RXD	A2	P1.2	4
PUSH2		A3	P1.3	5
		A4	P1.4	6
	SCK (B0)	A5	P1.5	7
	CS (B0)		P2.0	8
			P2.1	9
			P2.2	10

20				GROUND
19	P2.6			XIN
18	P2.7			XOUT
17				TEST
16				RESET
15	P1.7	A7	SDA	MISO (B0)
14	P1.6	A6	SCL	MOSI (B0)
13	P2.5			GREEN_LED
12	P2.4			
11	P2.3			

Hardware
Pin number

PC
TimerSerial
SPI

analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()

Rei Vilo, 2012
embeddedcomputing.weebly.com
version 1.4 2013-01-27

Figura 6.3: Launchpad de Texas Instruments

Code Composer™ Studio v5

Embedded Processor

TEXAS INSTRUMENTS

Figura 6.4: Suite de Texas Instruments CCS

7.1. Simulador de la regla clásica “La hormiga de Langton ”

7.1.1. Fenomenología de las hormigas de Langton

En esta sección se reportan los resultados de las pruebas realizadas con el autómata celular “La hormiga de Langton”, el caso de estudio fue acotado a trabajar con dos hormigas tomando en cuenta la vecindad de Moore, un espacio de 3X3 celdas. A continuación se muestran los arreglos de configuraciones iniciales, donde a cada configuración corresponde una combinación binaria de acuerdo a la presencia(1) o ausencia(0) de una hormiga en el espacio propuesto.

Dichas pruebas fueron realizadas con el simulador desarrollado durante el presente trabajo terminal.



Figura 7.1: Simulador desarrollado

7.1.1.1. Configuración 033 (000100001)

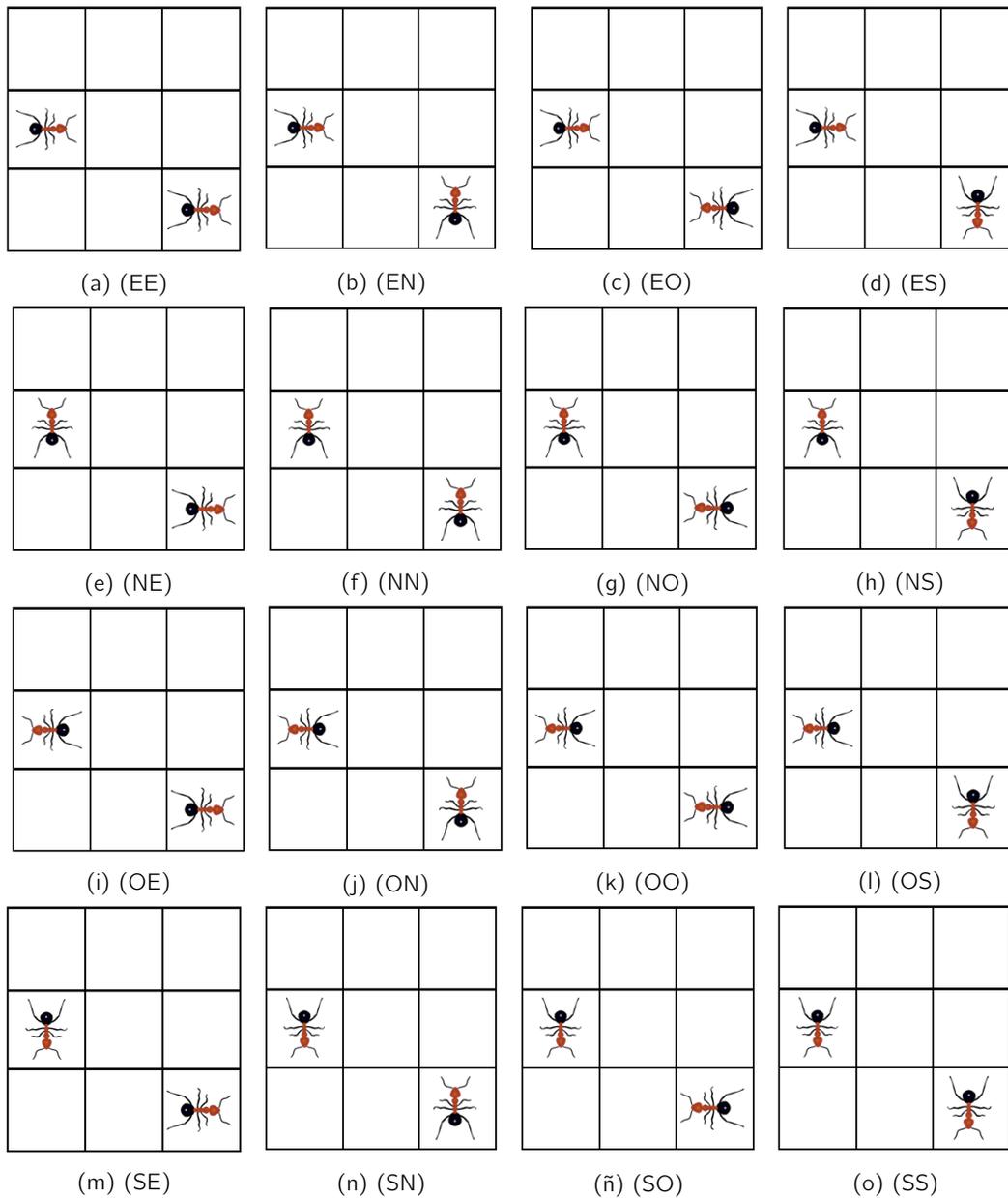


Figura 7.2: Arreglos de la combinación 000100001

7.1.1.2. Configuración 068 (001000100)

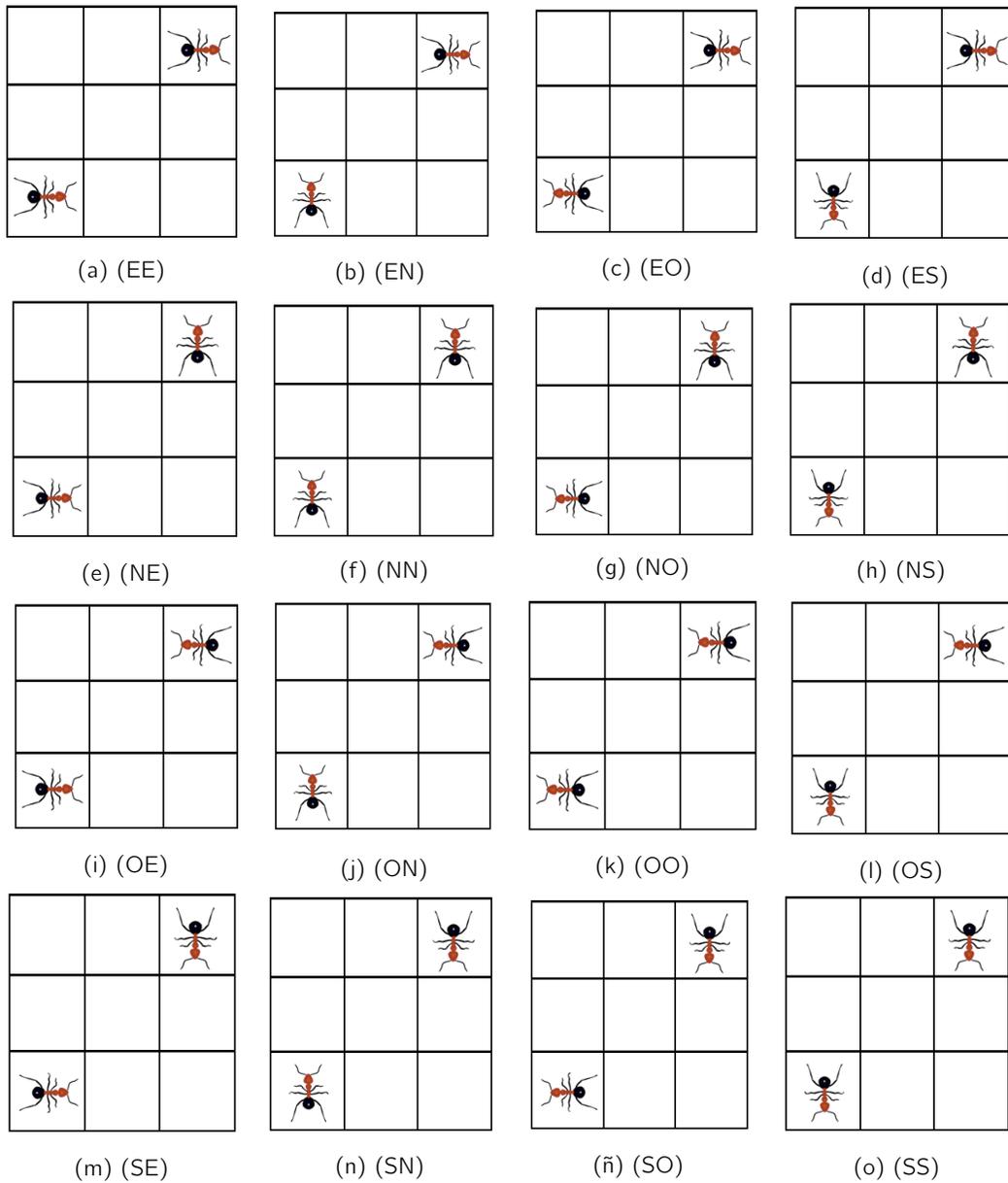


Figura 7.3: Arreglos de la combinación 001000100

7.1.1.3. Configuración 096 (00110000)

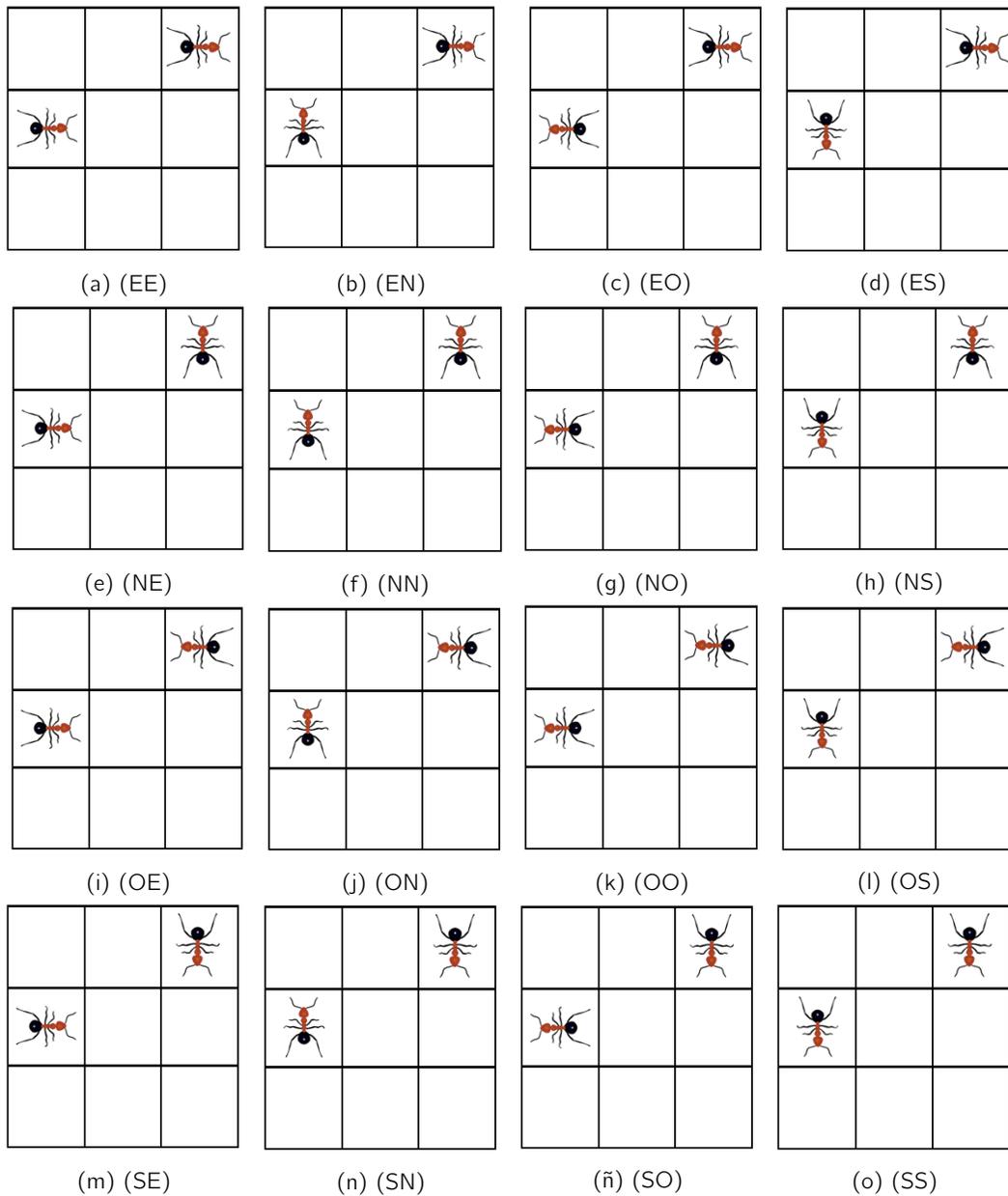


Figura 7.4: Arreglos de la combinación 00110000

7.1.1.4. Configuración 160 (01010000)

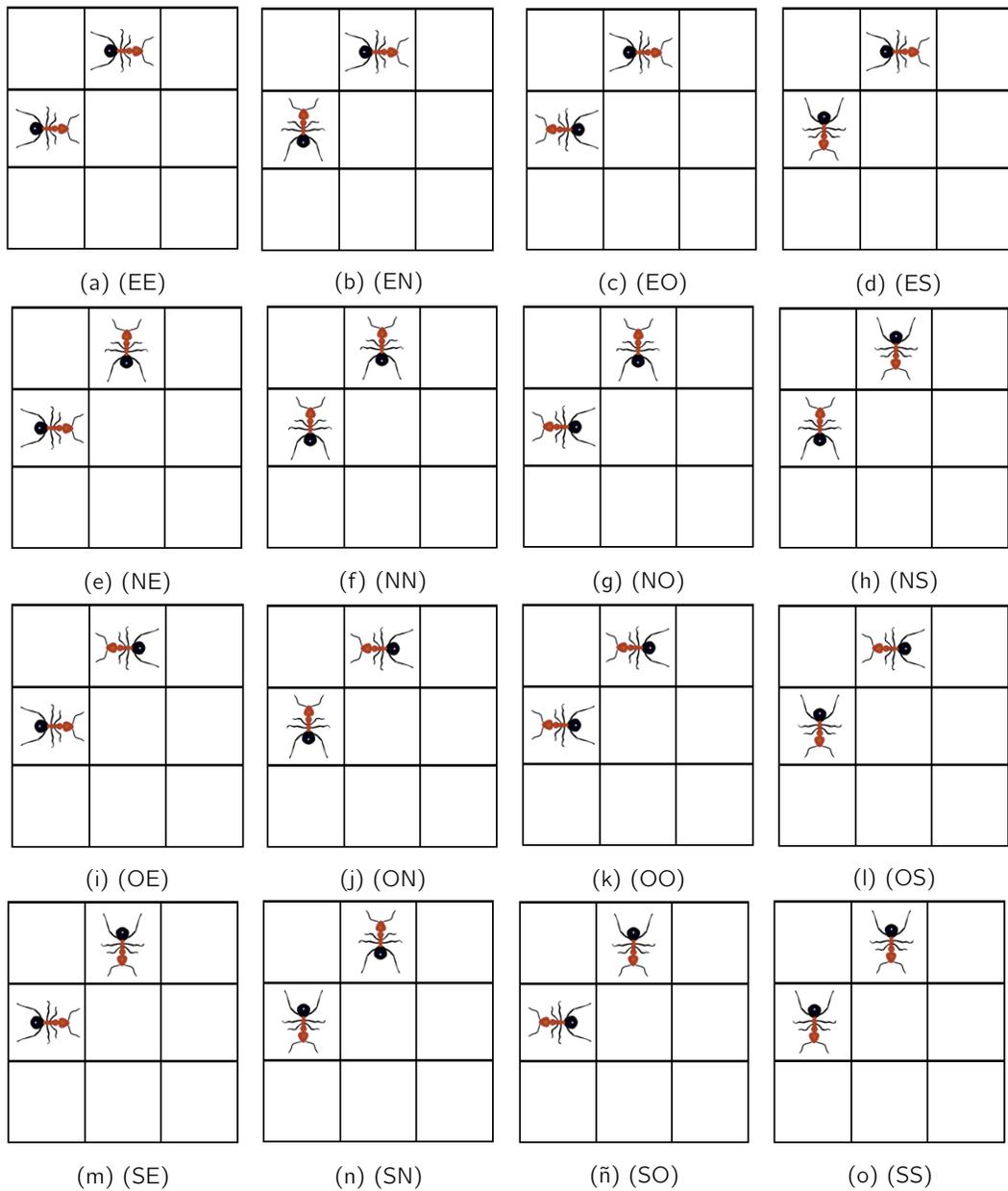


Figura 7.5: Arreglos de la combinación 01010000

7.1.1.5. Configuración 257 (10000001)

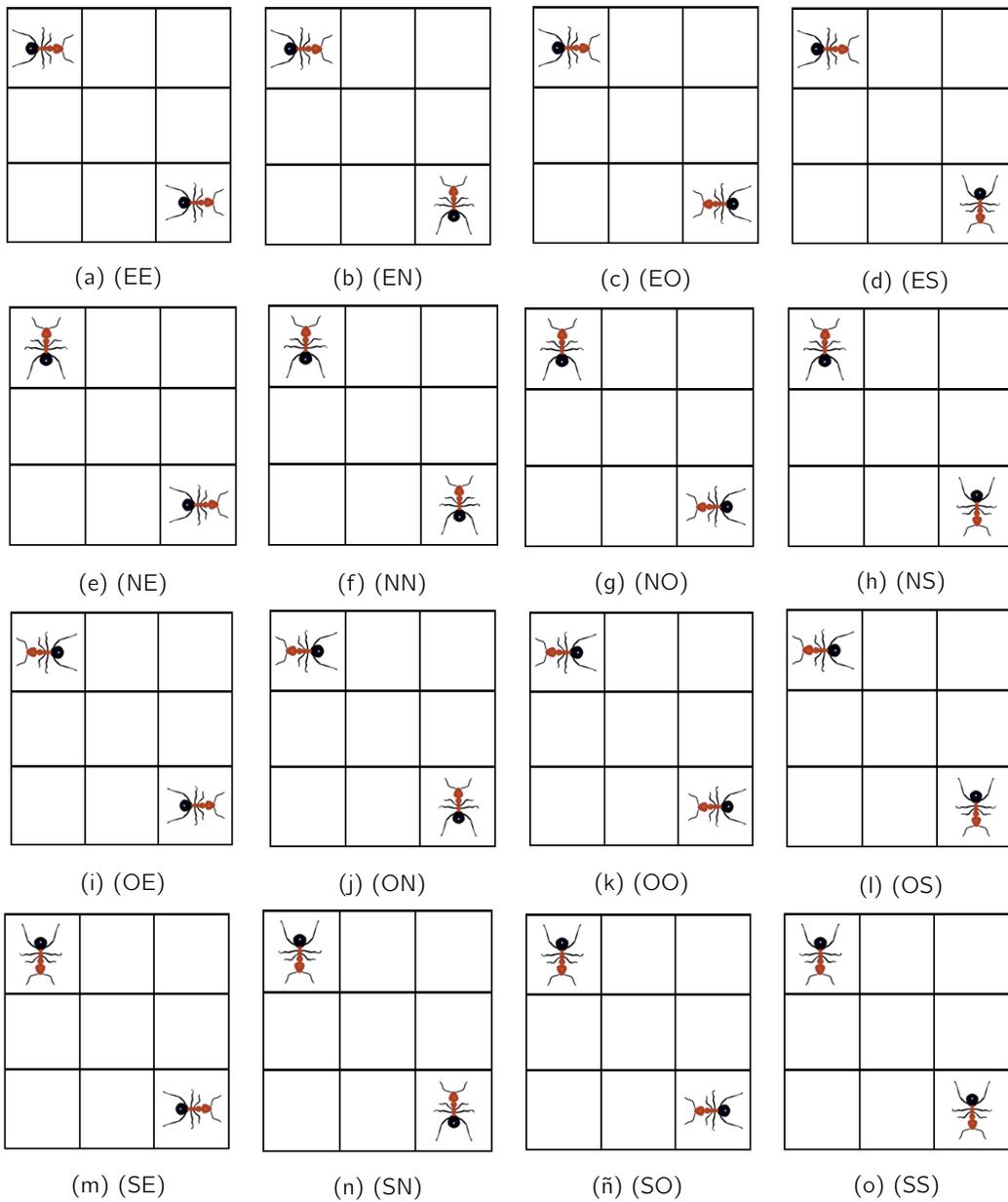


Figura 7.6: Arreglos de la combinación 10000001

7.1.1.6. Configuración 260 (100000100)

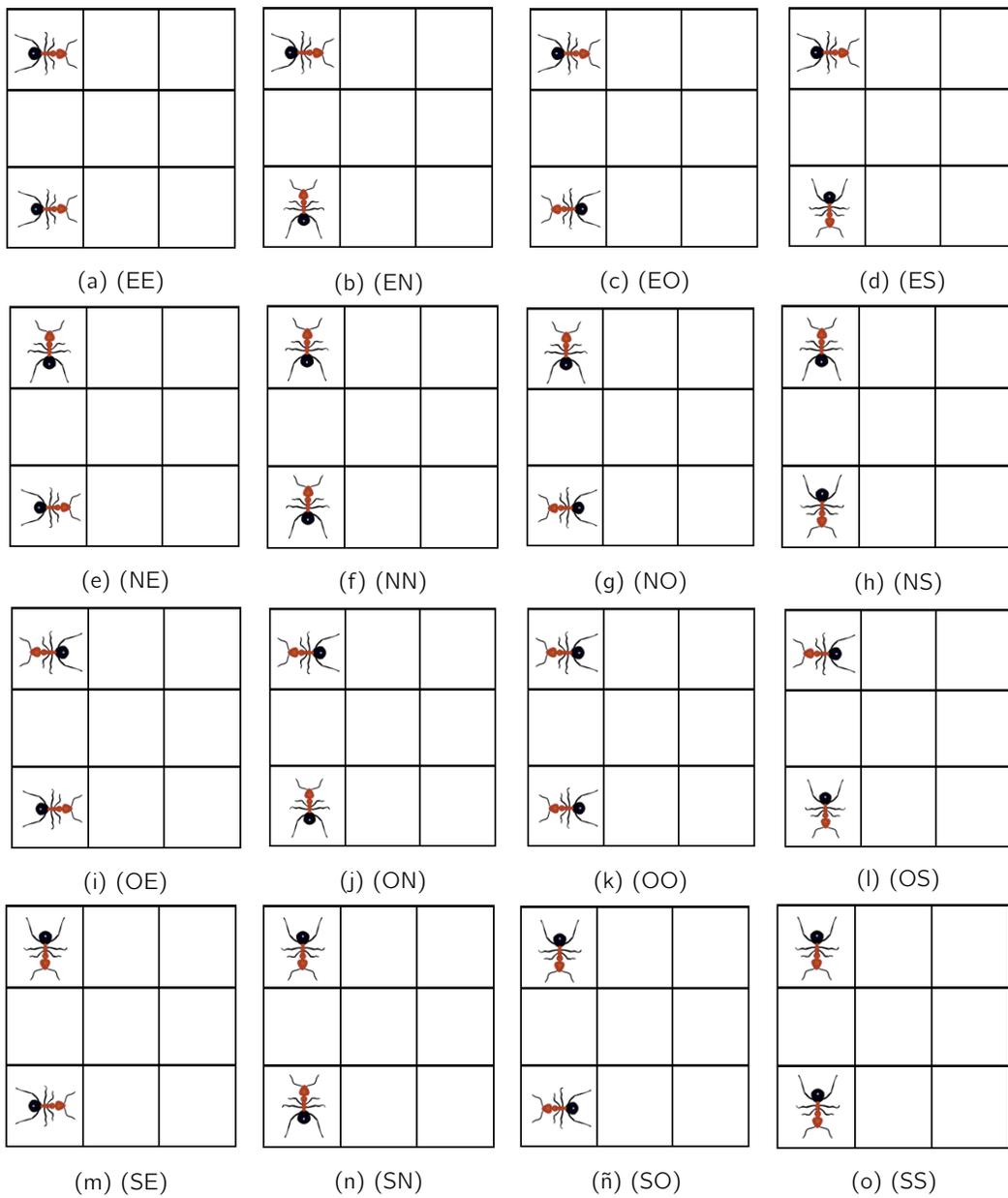


Figura 7.7: Arreglos de la combinación 100000100

7.1.1.7. Configuración 272 (100010000)

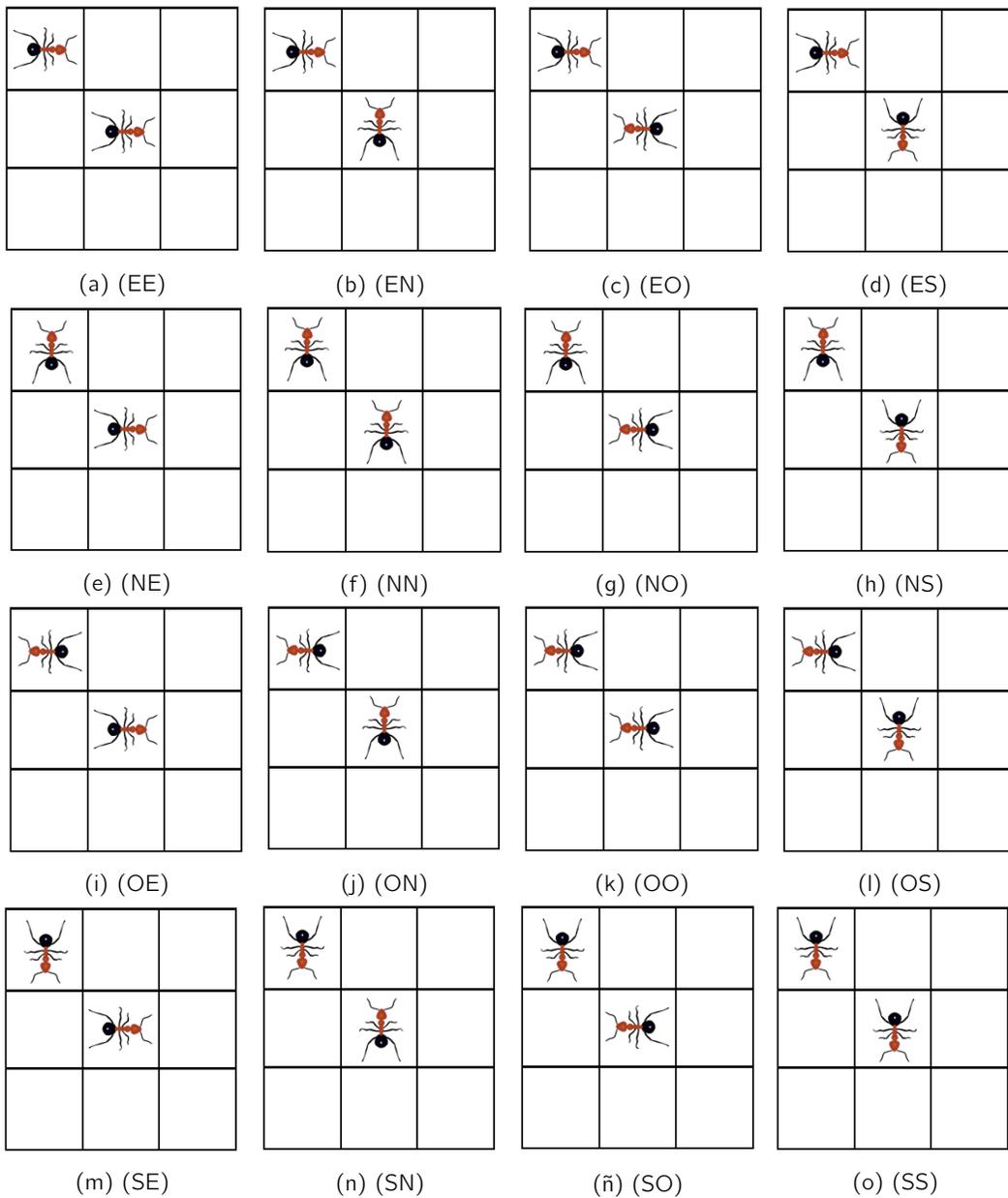


Figura 7.8: Arreglos de la combinación 100010000

7.1.1.8. Configuración 288 (100100000)

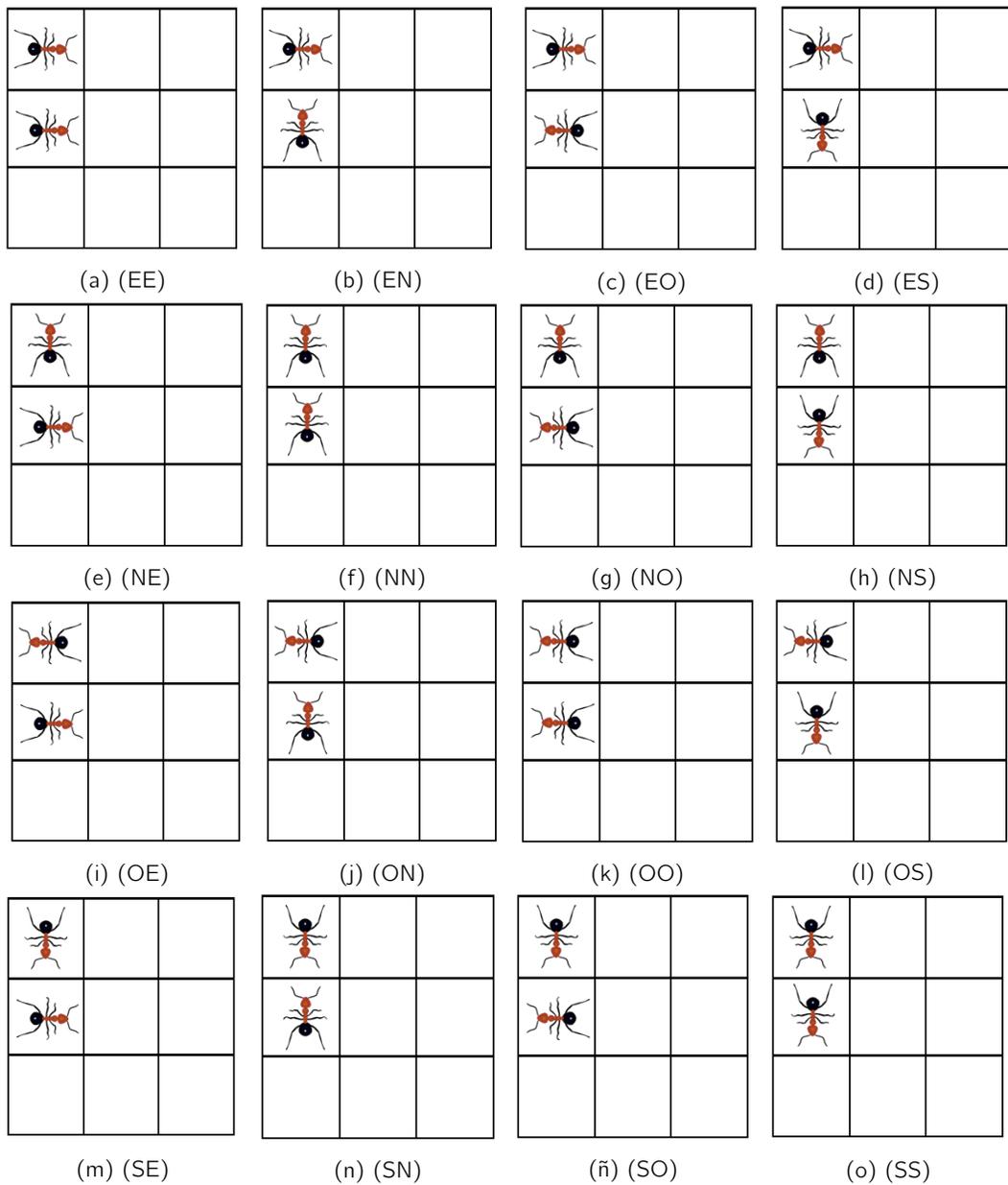


Figura 7.9: Arreglos de la combinación 100100000

7.1.1.9. Configuración 320 (101000000)

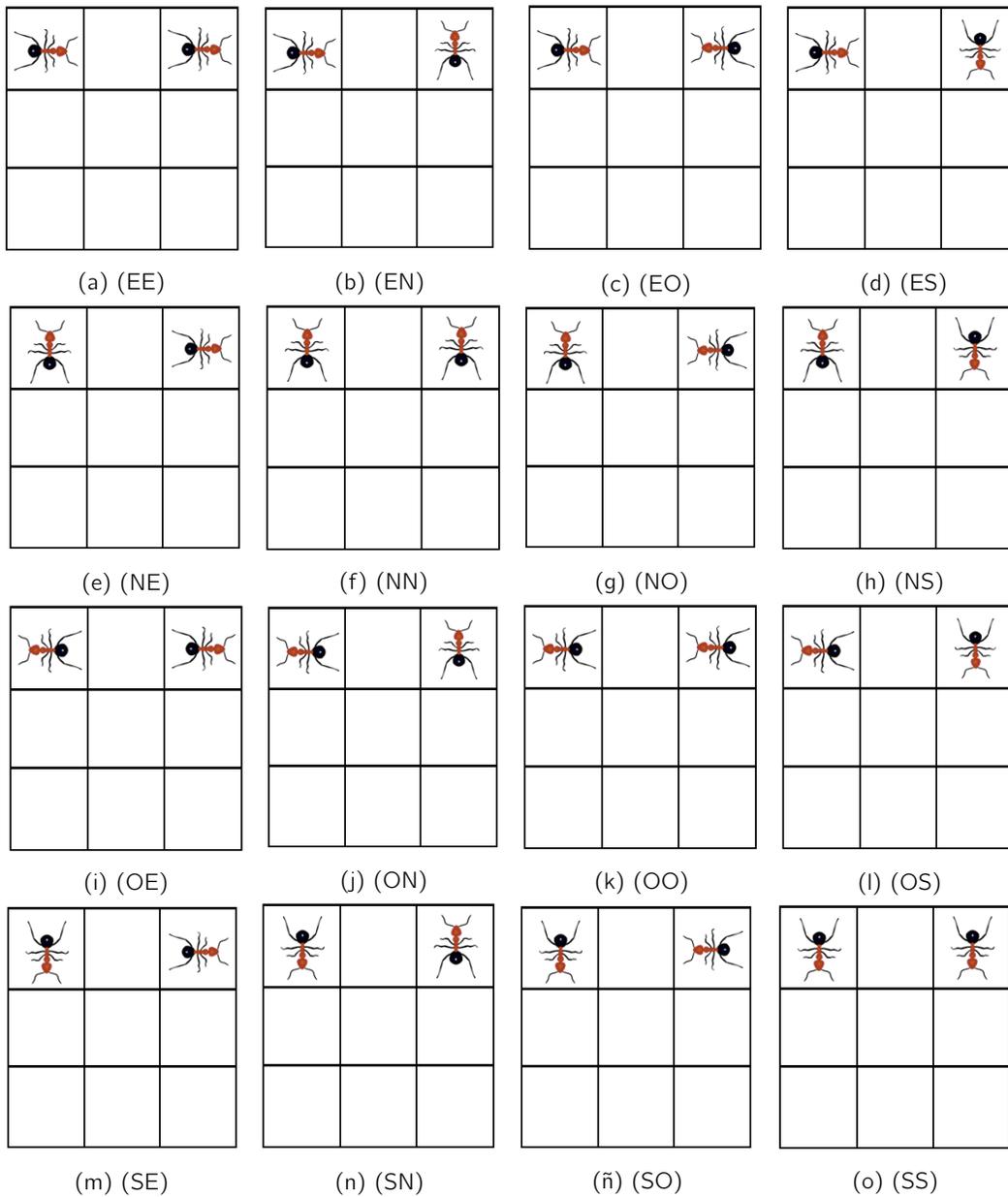


Figura 7.10: Arreglos de la combinación 101000000

7.1.1.10. Configuración 384 (11000000)

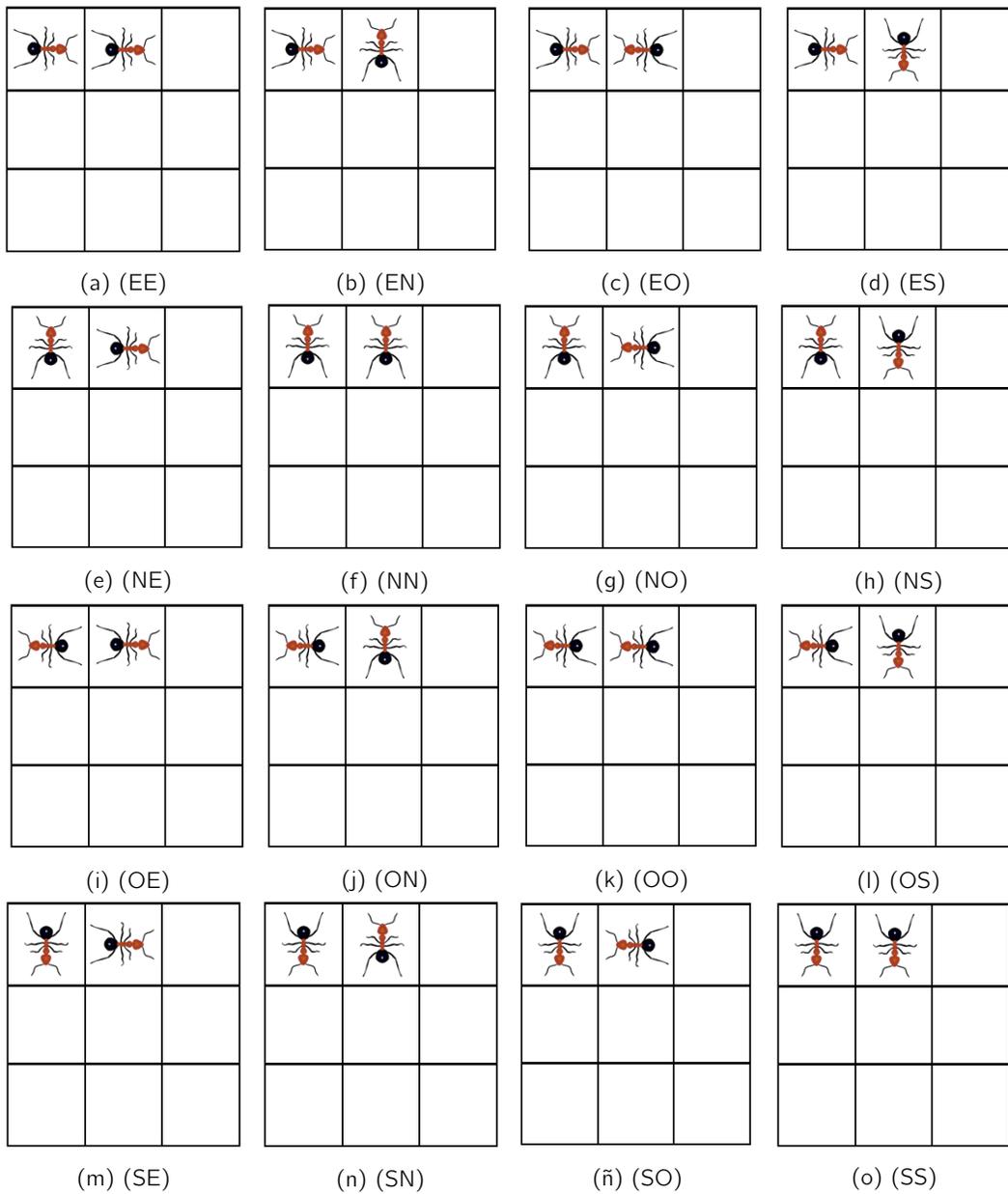


Figura 7.11: Arreglos de la combinación 110000000

7.1.2. Patrones resultantes

7.1.2.1. Patrones del arreglo 033

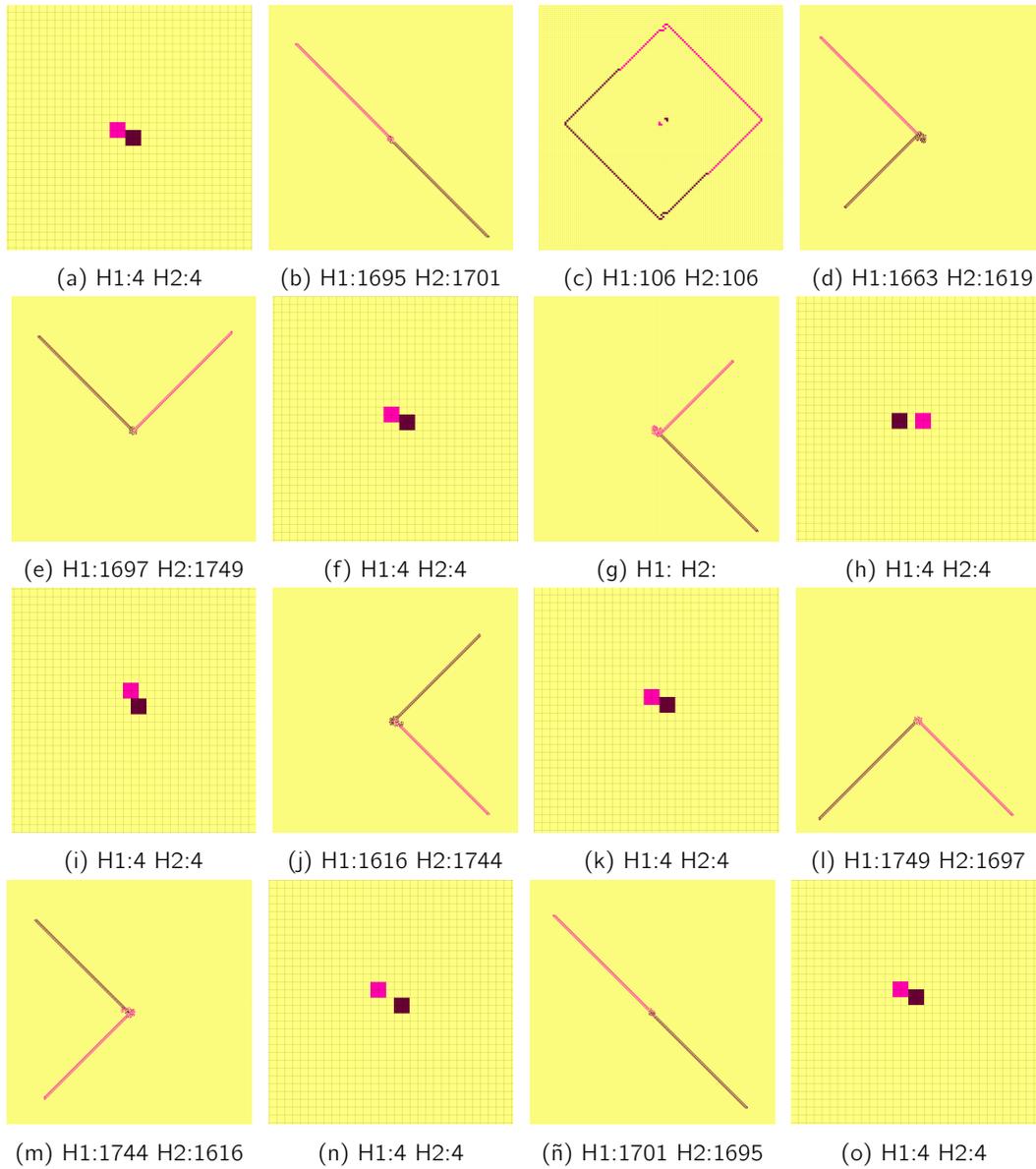


Figura 7.12: Patrones de la combinación 000100001

7.1.2.2. Patrones del arreglo 068

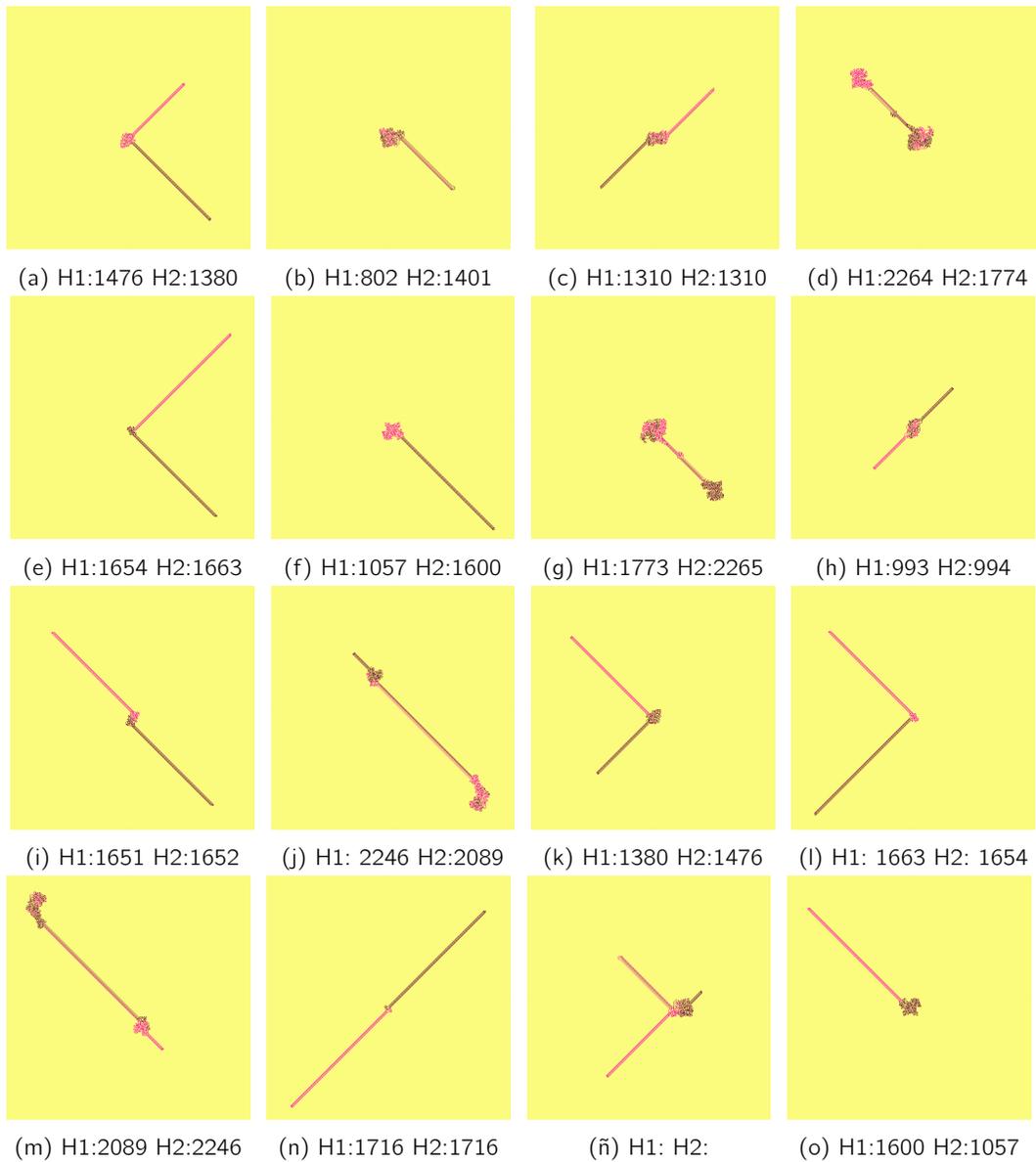


Figura 7.13: Patrones de la combinación 001000100

7.1.2.3. Patrones del arreglo 096

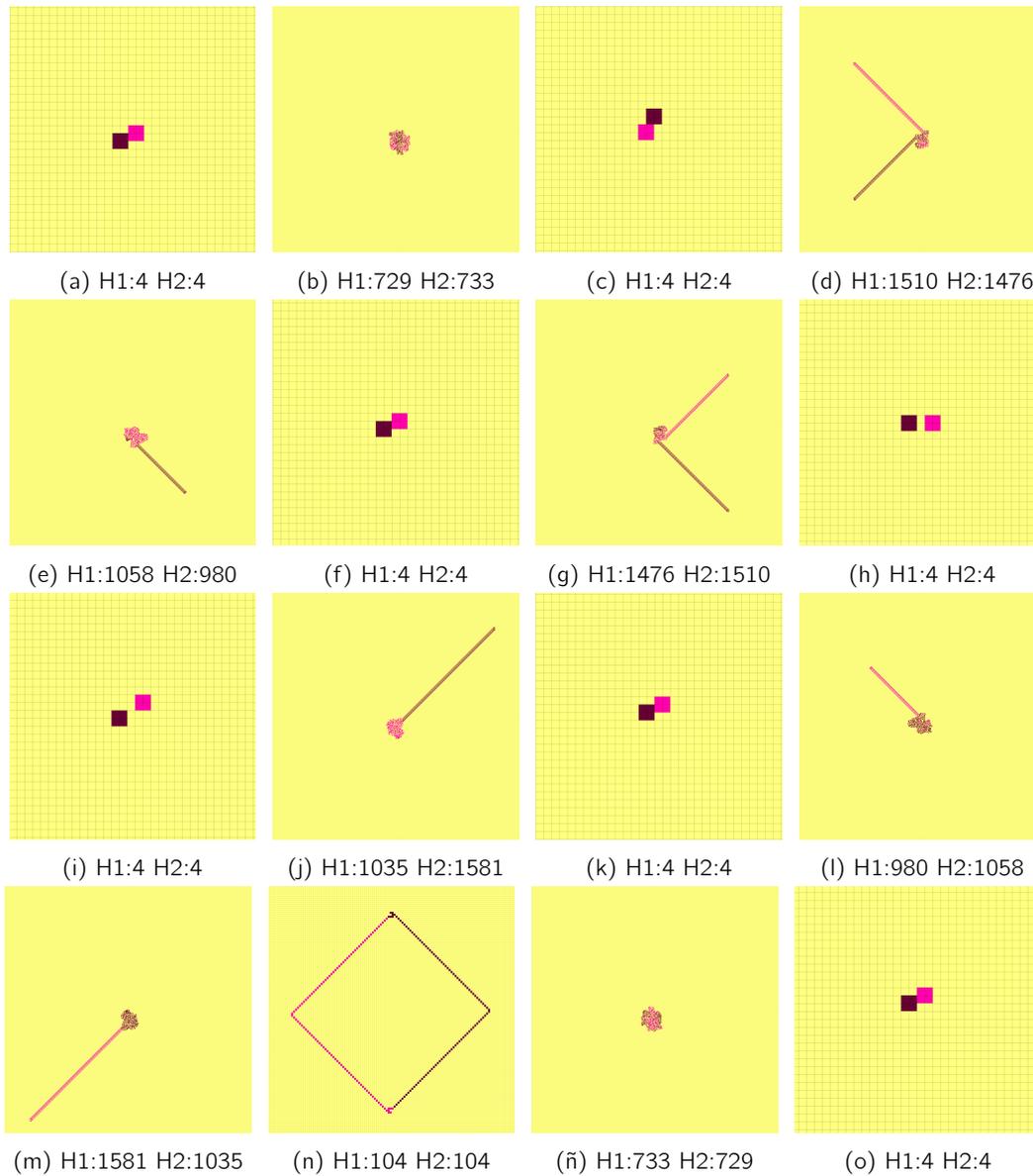


Figura 7.14: Patrones de la combinación 001100000

7.1.2.4. Patrones del arreglo 160

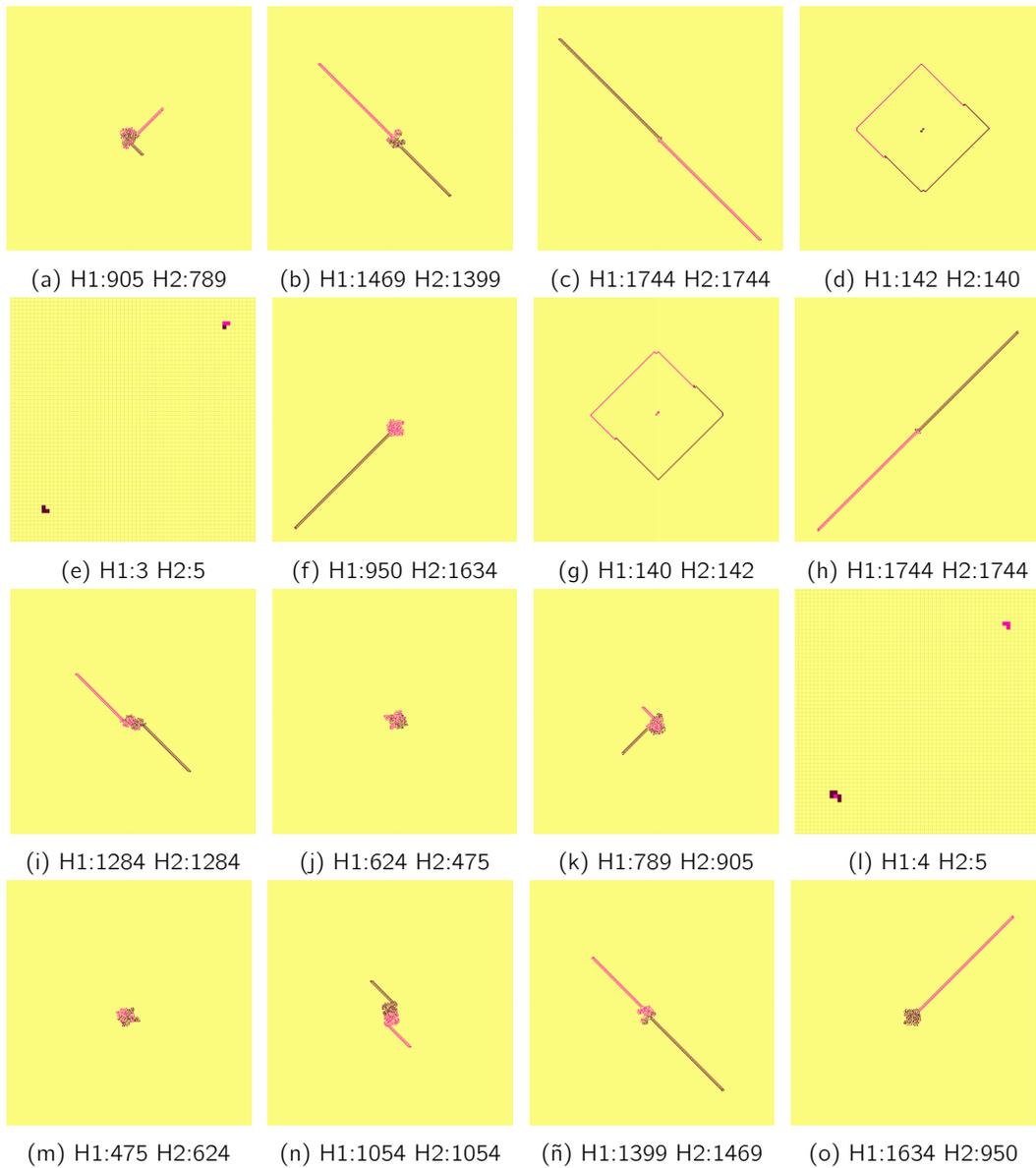


Figura 7.15: Patrones de la combinación 010100000

7.1.2.5. Patrones del arreglo 257



Figura 7.16: Patrones de la combinación 010000001

7.1.2.6. Patrones del arreglo 260

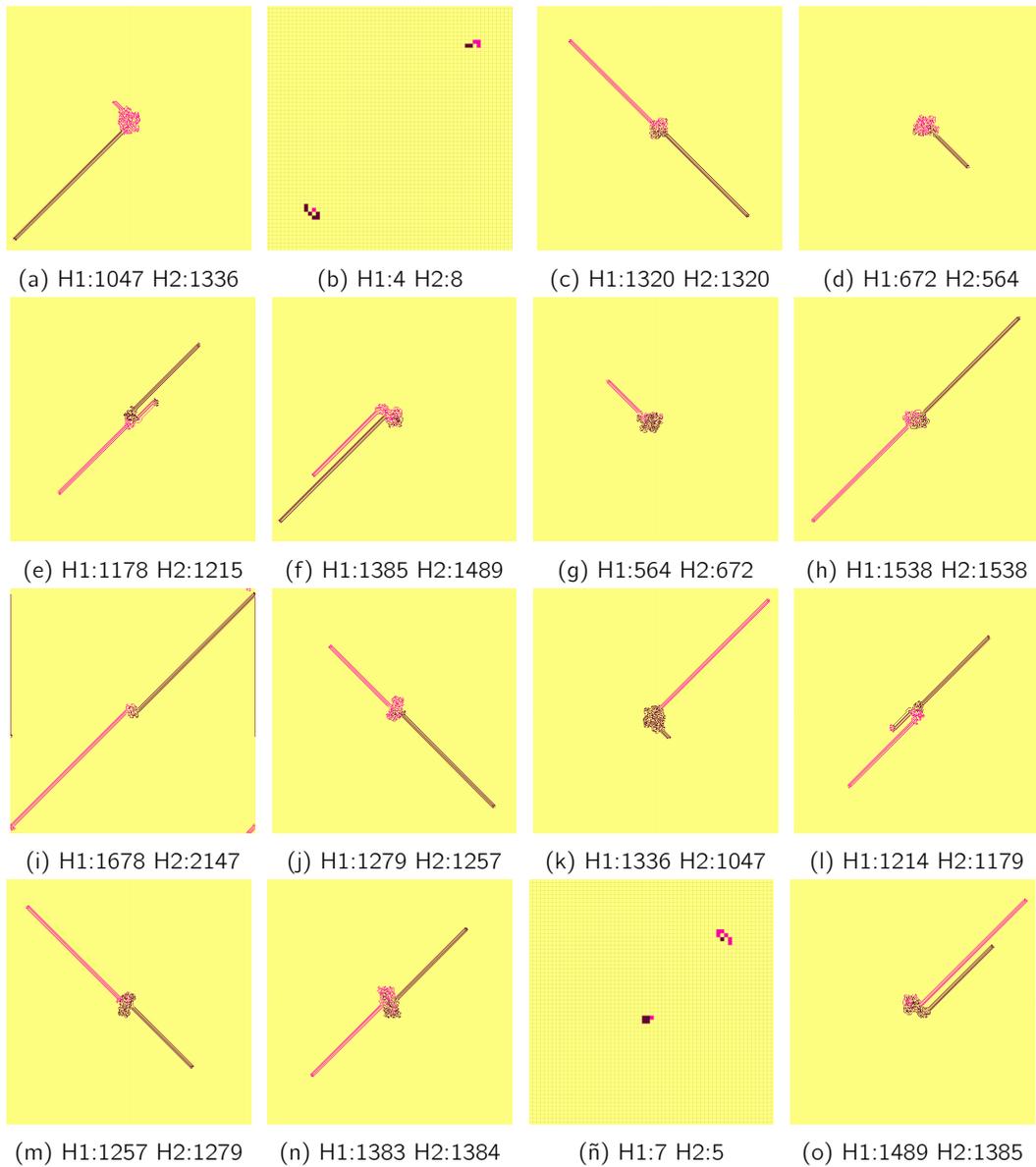


Figura 7.17: Patrones de la combinación 100000100

7.1.2.7. Patrones del arreglo 272

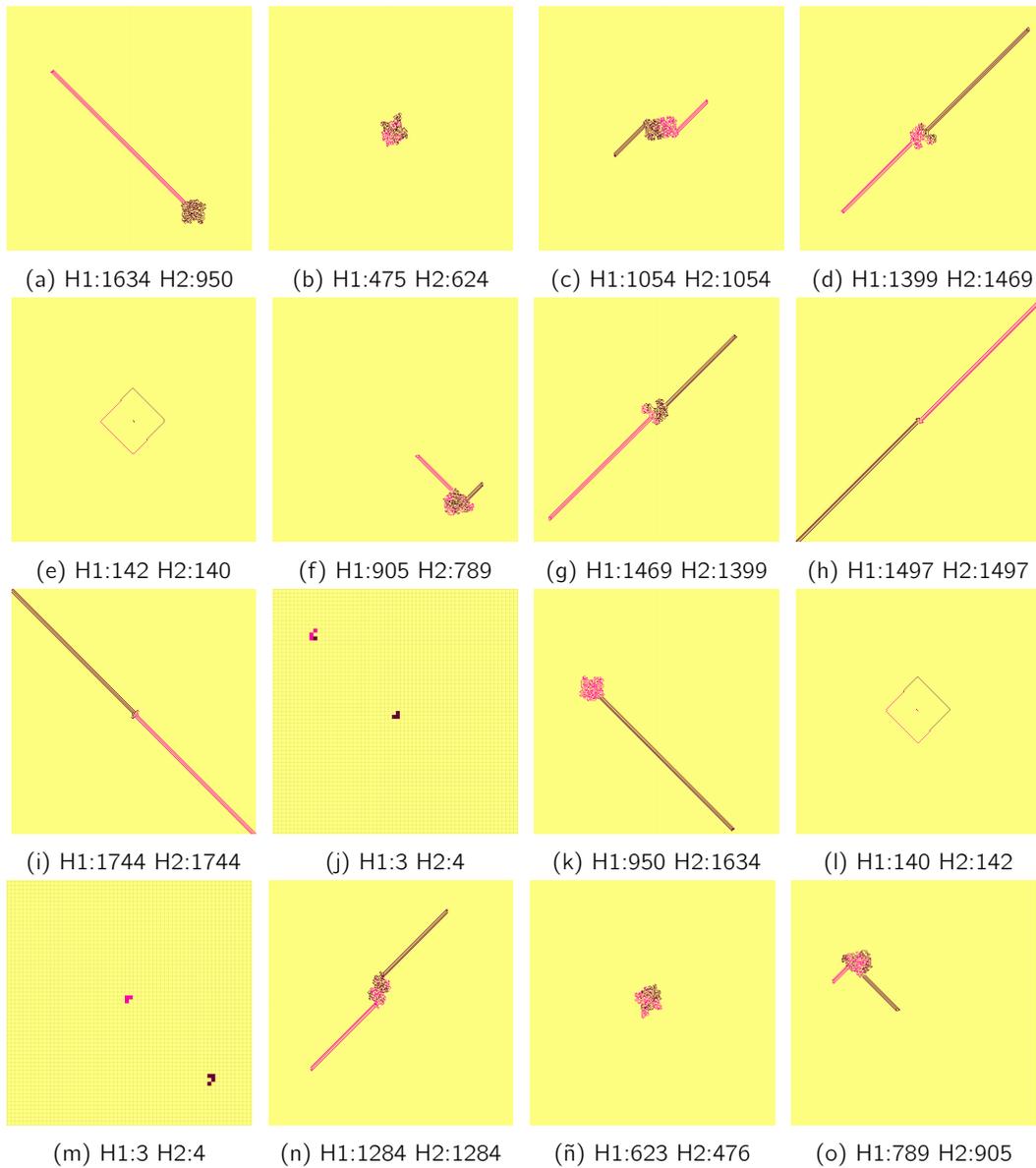


Figura 7.18: Patrones de la combinación 100010000

7.1.2.8. Patrones del arreglo 288

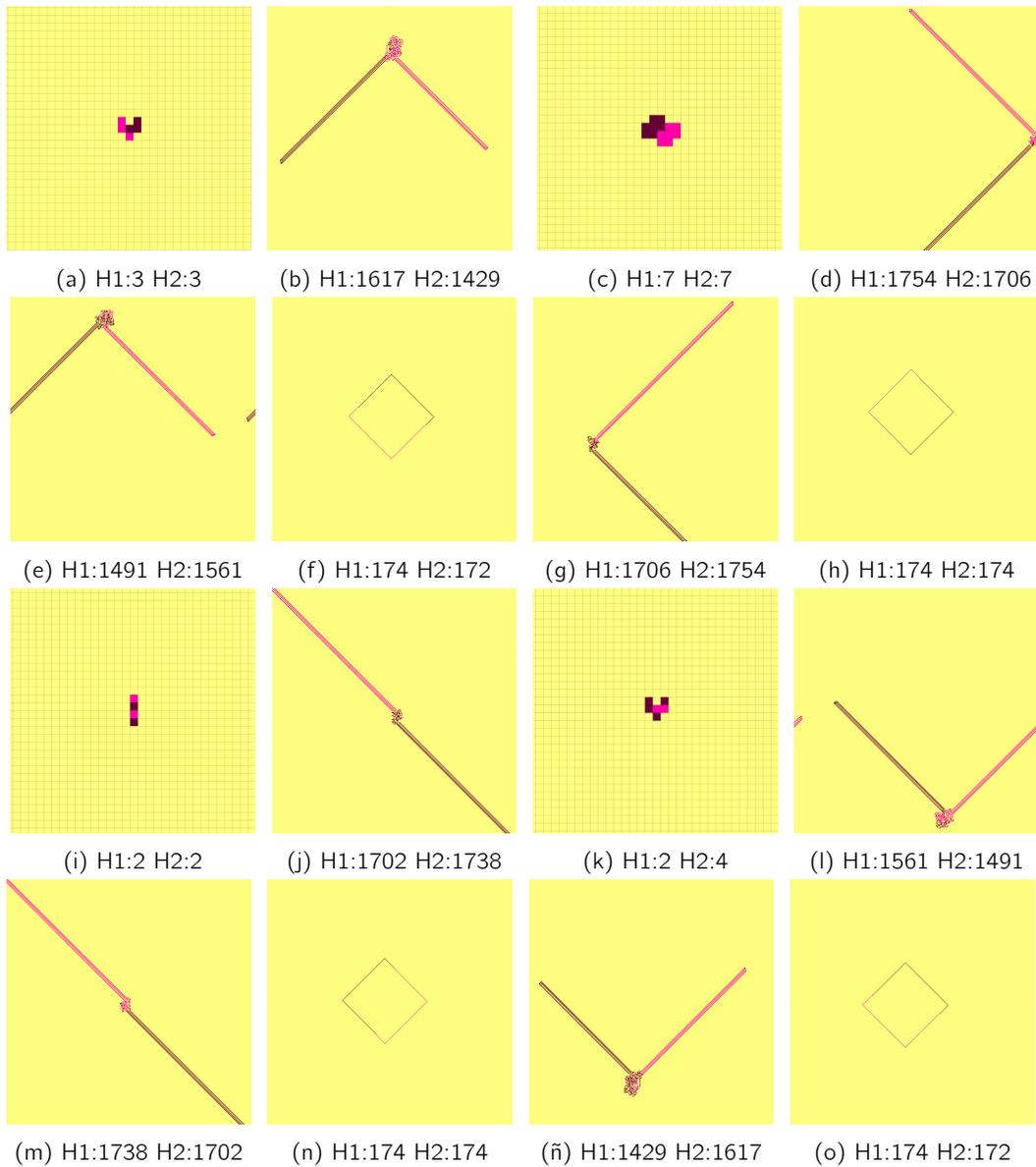


Figura 7.19: Patrones de la combinación 100100000

7.1.2.9. Patrones del arreglo 320

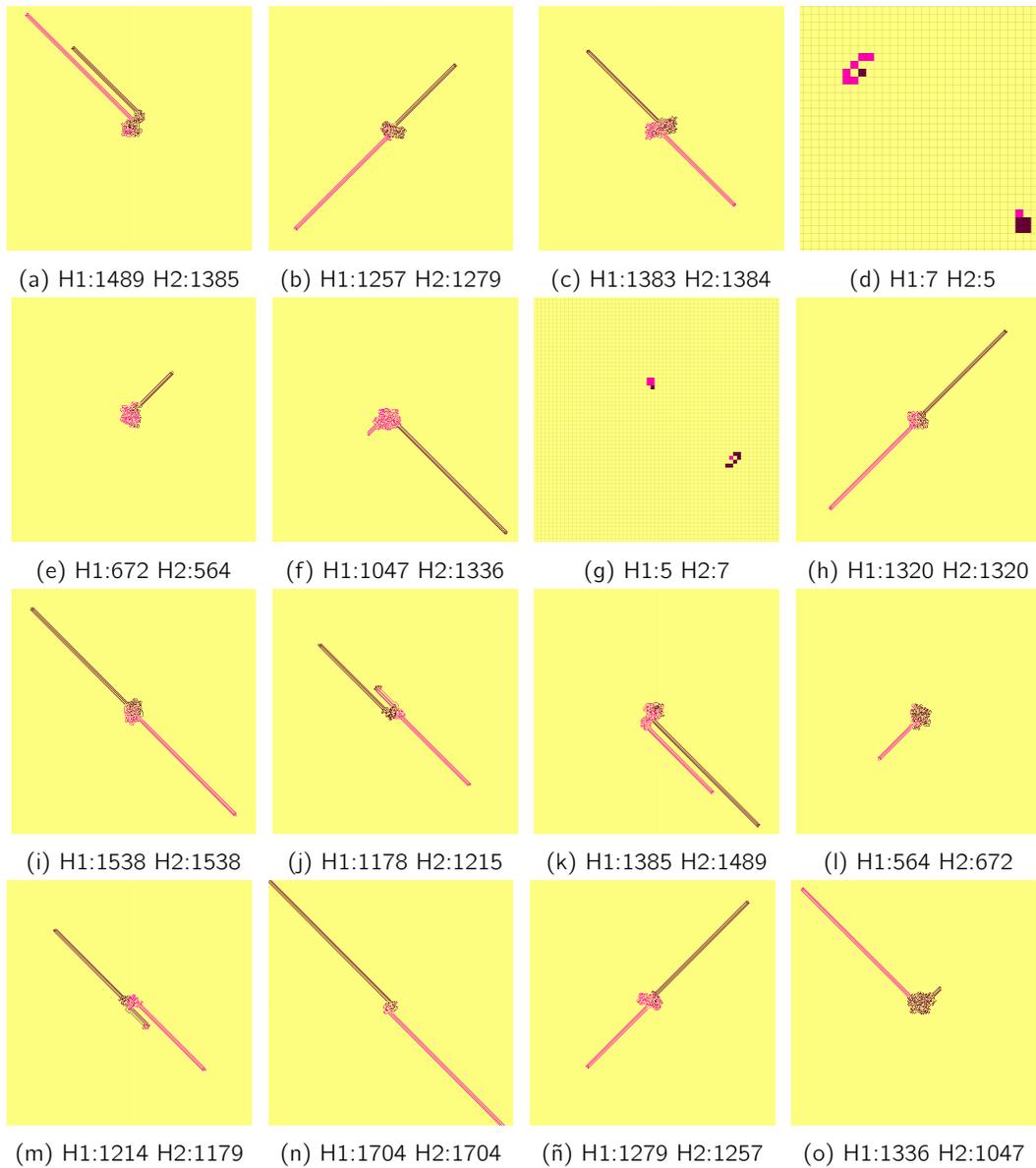


Figura 7.20: Patrones de la combinación 101000000

7.1.2.10. Patrones del arreglo 384

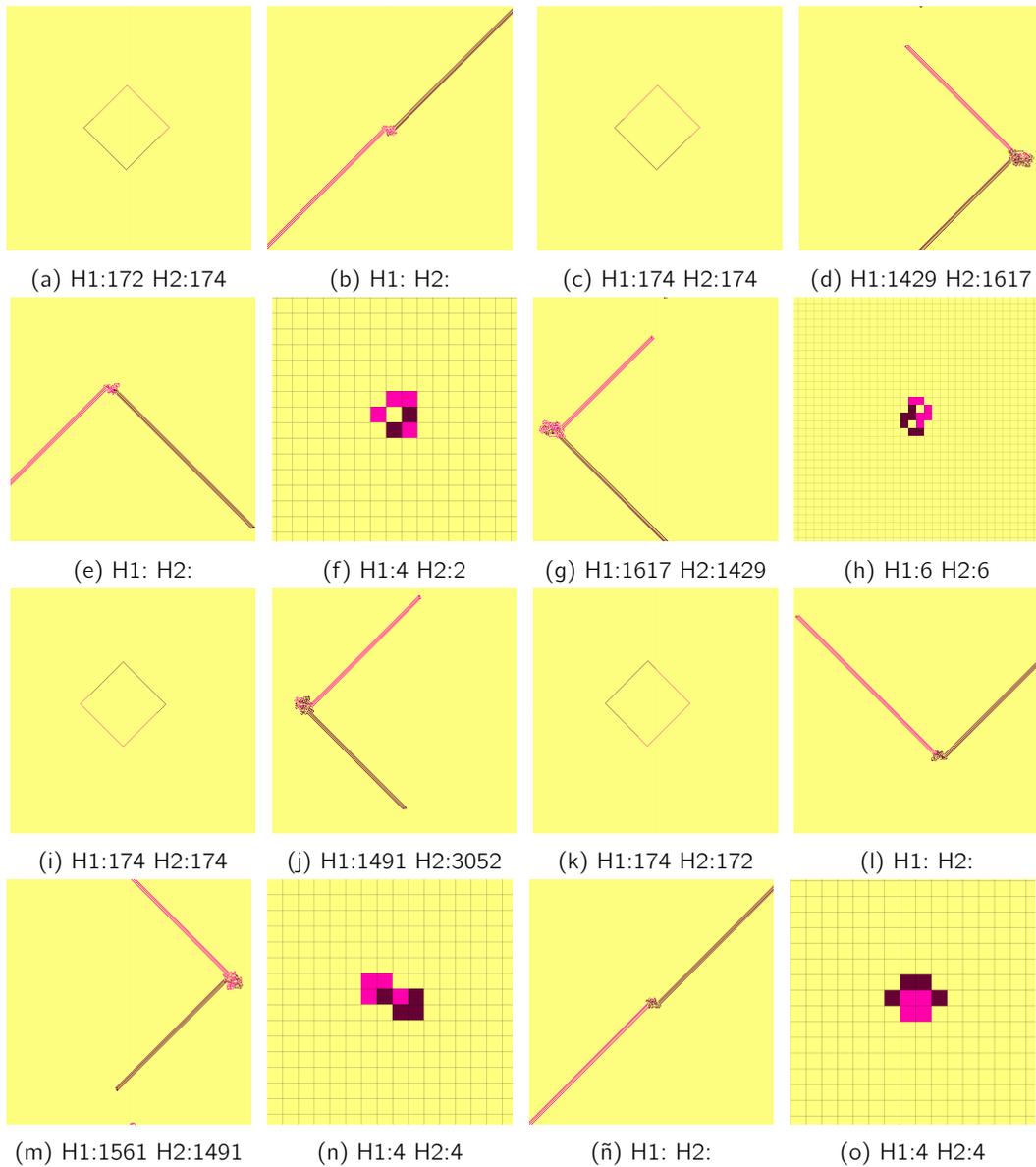


Figura 7.21: Patrones de la combinación 110000000

7.1.3. Matrices de incidencia

En las matrices presentadas a continuación se muestra la incidencia de las pruebas realizadas respecto a la interacción de dos hormigas colocadas en distinta orientación y posición siguiendo la regla clásica de “La hormiga de Langton”.

La relación 1 a 1 propuesta en las siguientes matrices refiere al resultado del arreglo dispuesto con las dos hormigas en la rejilla (fila superior), contra el patrón obtenido (primer columna izquierda).

7.1.3.1. Arreglo 033

$$\begin{pmatrix}
 & 33_{\Rightarrow} & 33_{\rightarrow\uparrow} & 33_{\Leftarrow} & 33_{\rightarrow\downarrow} & 33_{\uparrow\rightarrow} & 33_{\uparrow\uparrow} & 33_{\uparrow\leftarrow} & 33_{\uparrow\downarrow} & 33_{\Leftarrow} & 33_{\leftarrow\uparrow} & 33_{\Leftarrow} & 33_{\leftarrow\downarrow} & 33_{\downarrow\rightarrow} & 33_{\downarrow\uparrow} & 33_{\downarrow\leftarrow} & 33_{\downarrow\downarrow} \\
 P1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 P12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 P13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 P14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 P15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 P16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}$$

7.1.3.2. Arreglo 068

$$\begin{pmatrix}
 & 68_{\Rightarrow} & 68_{\rightarrow\uparrow} & 68_{\Leftarrow} & 68_{\rightarrow\downarrow} & 68_{\uparrow\rightarrow} & 68_{\uparrow\uparrow} & 68_{\uparrow\leftarrow} & 68_{\uparrow\downarrow} & 68_{\Leftarrow} & 68_{\leftarrow\uparrow} & 68_{\Leftarrow} & 68_{\leftarrow\downarrow} & 68_{\downarrow\rightarrow} & 68_{\downarrow\uparrow} & 68_{\downarrow\leftarrow} & 68_{\downarrow\downarrow} \\
 P17 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P18 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P19 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P20 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P21 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P22 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P23 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 P28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 P29 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 P30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 P31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 P32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}$$

7.1.3.3. Arreglo 096

$$\begin{pmatrix}
 & 96_{\Rightarrow} & 96_{\rightarrow\uparrow} & 96_{\Leftarrow} & 96_{\rightarrow\downarrow} & 96_{\uparrow\rightarrow} & 96_{\uparrow\uparrow} & 96_{\uparrow\leftarrow} & 96_{\uparrow\downarrow} & 96_{\Leftarrow} & 96_{\leftarrow\uparrow} & 96_{\Leftarrow} & 96_{\leftarrow\downarrow} & 96_{\downarrow\rightarrow} & 96_{\downarrow\uparrow} & 96_{\downarrow\leftarrow} & 96_{\downarrow\downarrow} \\
 P33 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P34 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P35 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P36 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P37 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P38 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P39 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P40 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P41 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P42 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 P43 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 P44 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 P45 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 P46 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 P47 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 P48 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}$$

7.1. SIMULADOR DE LA REGLA CLÁSICA "LA HORMIGA DE LANGTON "

7.1.3.4. Arreglo 160

	160⇒	160→↑	160⇄	160→↓	160↑→	160⇄↑	160↑←	160↑↓	160⇄	160←↑	160⇄	160←↓	160↓→	160↓↑	160↓←	160⇄↓
P49	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P50	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P51	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P52	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
P53	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P54	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P55	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P56	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P57	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P58	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P59	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P60	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P61	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P62	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

7.1.3.5. Arreglo 257

	257⇒	257→↑	257⇄	257→↓	257↑→	257⇄↑	257↑←	257↑↓	257⇄	257←↑	257⇄	257←↓	257↓→	257↓↑	257↓←	257⇄↓
P65	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P66	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P67	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P68	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
P69	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P70	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P71	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P72	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P73	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P74	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P75	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P60	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P77	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P78	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

7.1.3.6. Arreglo 260

	260⇒	260→↑	260⇄	260→↓	260↑→	260⇄↑	260↑←	260↑↓	260⇄	260←↑	260⇄	260←↓	260↓→	260↓↑	260↓←	260⇄↓
P81	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P82	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P83	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P84	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
P85	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P86	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P87	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P88	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P89	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P90	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P91	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P92	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P93	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P94	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

7.1.3.7. Arreglo 272

	272_{\rightarrow}	$272_{\rightarrow\uparrow}$	272_{\rightleftharpoons}	$272_{\rightarrow\downarrow}$	$272_{\uparrow\rightarrow}$	$272_{\uparrow\uparrow}$	$272_{\uparrow\leftarrow}$	$272_{\uparrow\downarrow}$	272_{\rightleftharpoons}	$272_{\leftarrow\uparrow}$	272_{\leftarrow}	$272_{\leftarrow\downarrow}$	$272_{\downarrow\rightarrow}$	$272_{\downarrow\uparrow}$	$272_{\downarrow\leftarrow}$	$272_{\downarrow\downarrow}$
P97	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P98	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P99	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P100	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
P101	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P102	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P103	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P104	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P105	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P106	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P107	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P108	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P109	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P110	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

7.1.3.8. Arreglo 288

	288_{\rightarrow}	$288_{\rightarrow\uparrow}$	288_{\rightleftharpoons}	$288_{\rightarrow\downarrow}$	$288_{\uparrow\rightarrow}$	$288_{\uparrow\uparrow}$	$288_{\uparrow\leftarrow}$	$288_{\uparrow\downarrow}$	288_{\rightleftharpoons}	$288_{\leftarrow\uparrow}$	288_{\leftarrow}	$288_{\leftarrow\downarrow}$	$288_{\downarrow\rightarrow}$	$288_{\downarrow\uparrow}$	$288_{\downarrow\leftarrow}$	$288_{\downarrow\downarrow}$
P113	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P114	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P115	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P116	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
P117	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P118	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P119	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P120	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P121	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P122	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P123	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P124	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P125	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P126	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P127	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P128	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

7.1.3.9. Arreglo 320

	320_{\rightarrow}	$320_{\rightarrow\uparrow}$	320_{\rightleftharpoons}	$320_{\rightarrow\downarrow}$	$320_{\uparrow\rightarrow}$	$320_{\uparrow\uparrow}$	$320_{\uparrow\leftarrow}$	$320_{\uparrow\downarrow}$	320_{\rightleftharpoons}	$320_{\leftarrow\uparrow}$	320_{\leftarrow}	$320_{\leftarrow\downarrow}$	$320_{\downarrow\rightarrow}$	$320_{\downarrow\uparrow}$	$320_{\downarrow\leftarrow}$	$320_{\downarrow\downarrow}$
P129	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P130	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P131	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P132	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
P133	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P134	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P135	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P136	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P137	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P138	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P139	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P140	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P141	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P142	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P143	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P144	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

7.1. SIMULADOR DE LA REGLA CLÁSICA "LA HORMIGA DE LANGTON "

7.1.3.10. Arreglo 384

	384 \Rightarrow	384 $\rightarrow\uparrow$	384 \rightleftharpoons	384 $\rightarrow\downarrow$	384 $\uparrow\rightarrow$	384 $\uparrow\uparrow$	384 $\uparrow\leftarrow$	384 $\uparrow\downarrow$	384 \Leftarrow	384 $\leftarrow\uparrow$	384 \Leftarrow	384 $\leftarrow\downarrow$	384 $\downarrow\rightarrow$	384 $\downarrow\uparrow$	384 $\downarrow\leftarrow$	384 $\downarrow\downarrow$
P145	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P146	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P147	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P148	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
P149	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P150	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
P151	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P152	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P153	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P154	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P155	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P156	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P157	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P158	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P159	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

7.1.4. Periodos de patrones oscilatorios

Osciladores de período 28		
Patrón	Celdas visitadas	
	Mínimo	Máximo
288 \rightarrow	0	8
288 \leftarrow	0	8
384 $\uparrow\uparrow$	0	8
384 $\downarrow\downarrow$	0	8

Osciladores de período 24		
Patrón	Celdas visitadas	
	Mínimo	Máximo
33 $\uparrow\downarrow$	0	14
33 \leftrightarrow	0	14
96 $\uparrow\downarrow$	0	14
288 \leftrightarrow	0	14
384 $\uparrow\downarrow$	0	14

Osciladores de período 16		
Patrón	Celdas visitadas	
	Mínimo	Máximo
96 \leftrightarrow	0	8
288 \leftrightarrow	0	8
384 $\downarrow\uparrow$	0	8

Osciladores de período 88		
Patrón	Celdas visitadas	
	Mínimo	Máximo
96 \leftrightarrow	0	26

Osciladores de período 44		
Patrón	Celdas visitadas	
	Mínimo	Máximo
96 \rightarrow	0	12
96 \leftarrow	0	12

Osciladores de período 40		
Patrón	Celdas visitadas	
	Mínimo	Máximo
33 $\uparrow\uparrow$	0	10
33 $\downarrow\downarrow$	0	10

Osciladores de período 432		
Patrón	Celdas visitadas	
	Mínimo	Máximo
96 $\uparrow\uparrow$	0	32
96 $\downarrow\downarrow$	0	32

Osciladores de período 284		
Patrón	Celdas visitadas	
	Mínimo	Máximo
33 \rightarrow	0	26
33 \leftarrow	0	26

Osciladores de período 96		
Patrón	Celdas visitadas	
	Mínimo	Máximo
33 \uparrow	0	26

7.2. Prototipo de robot: antBot

7.2.1. Esquema de pruebas

Con el objetivo de encontrar el prototipo más adecuado, aquel que se desempeñase de la mejor manera y se adaptase a las necesidades del presente trabajo terminal, se realizó una serie de pruebas que abarcó principalmente los componentes del robot así como la superficie sobre la que se desplazaba. A continuación se presenta un resumen.

7.2.1.1. Prueba 1

Objetivo: Encontrar el par de motores de vibración que permitan el óptimo desplazamiento del robot.

El movimiento de la “hormiga” es uno de los aspectos más importantes a considerar por lo que se llevaron a cabo pruebas con modelos diferentes de motores de vibración.

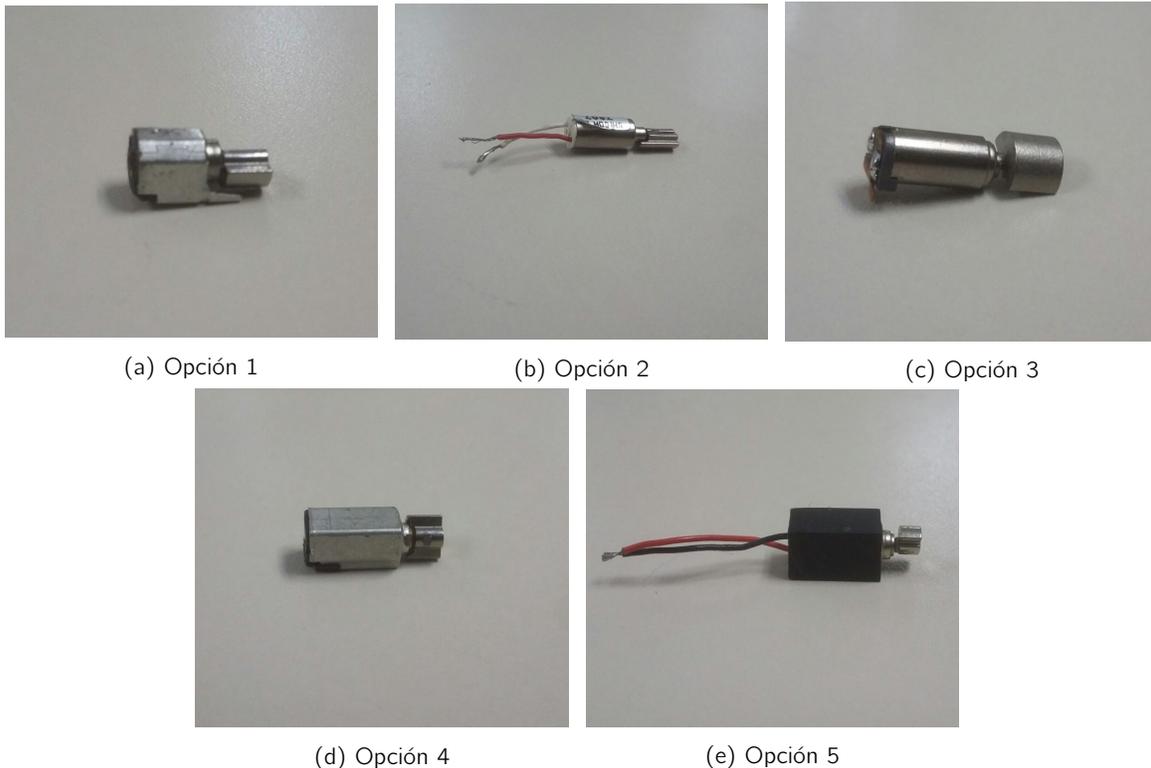


Figura 7.22: Opciones de motores de vibración.

Conclusión: La opción de motor número 1 fue la que permitió un mejor movimiento debido tanto a su tamaño como al número de revoluciones por minuto que realiza.

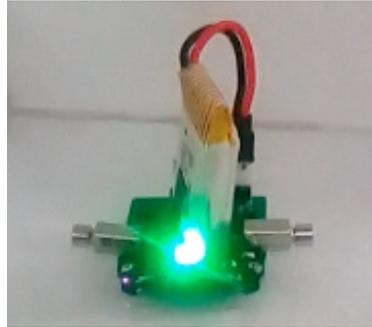
7.2.1.2. Prueba 2

Objetivo: Posicionar la batería en el robot de tal forma que no desvíe la trayectoria de este.

El robot requiere una alimentación de 3.3 V por lo que se seleccionó una batería LiPo debido a su tamaño y peso. La correcta ubicación de dicho componente es de suma importancia para el conseguir un desplazamiento rectilíneo.



(a) Batería vertical centrada



(b) Batería vertical cargada



(c) Batería inclinada verticalmente



(d) Batería horizontal

Figura 7.23: Diferente posicionamiento de la batería.

Conclusión: La posición (c) fue la mejor en cuanto a relación peso/espacio pues permitió un avance uniforme con la menor desviación.

7.2.1.3. Prueba 3

Objetivo: Definir una superficie de desplazamiento de acuerdo al comportamiento de los robots en ella.

Se seleccionó una serie de materiales con superficie lisa sobre los cuales la "hormiga" pudiese moverse sin obstáculos entre los cuales destacaron:

- Acrílico
- Vidrio

Conclusión: Se notó un mejor desplazamiento en el vidrio debido a su nula porosidad.

7.2.2. Prototipo final antBot

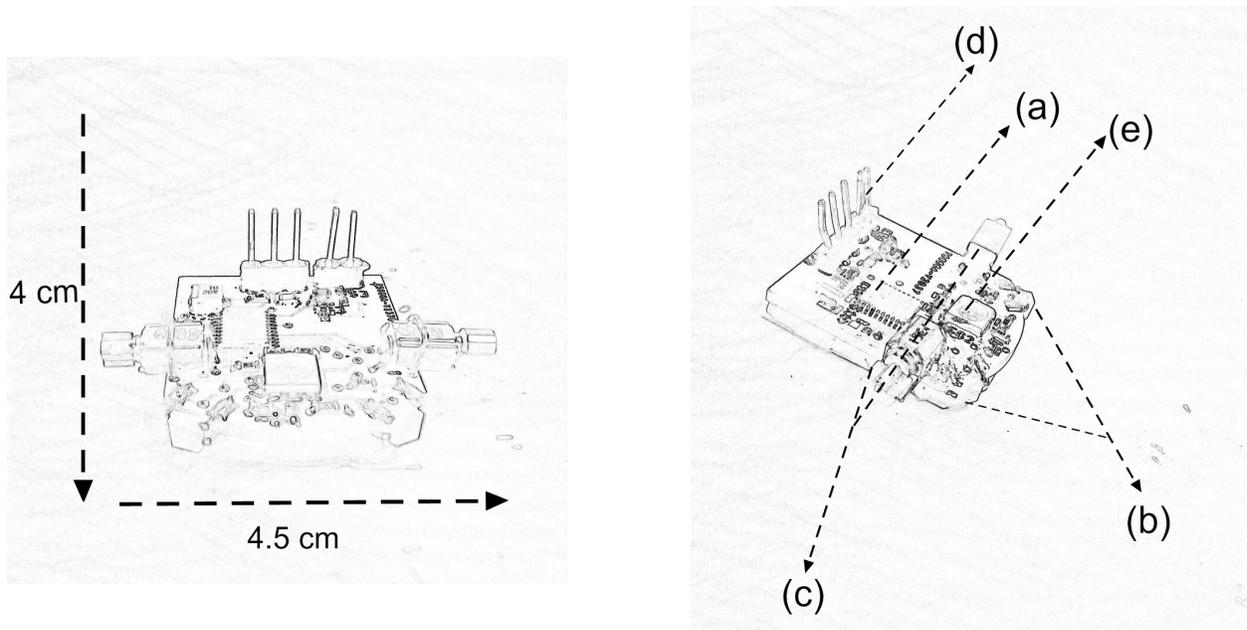


Figura 7.24: Especificaciones de un antBot

El prototipo final cuenta con las siguientes características:

- (a) Microcontrolador MSP430g255
- (b) Dos sensores QRE1113
- (c) Dos motores de vibración de dimensiones 1cm X 0.5cm X 0.5 cm
- (d) Batería de 3.3V
- (e) Un led RGB

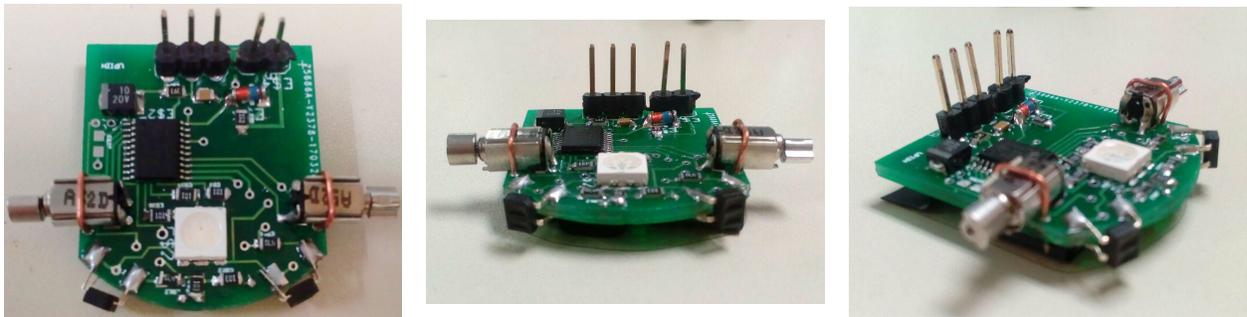


Figura 7.25: Perspectivas de un antBot

Posteriormente se ensambló un número de elementos para la realización de pruebas a mayor escala.

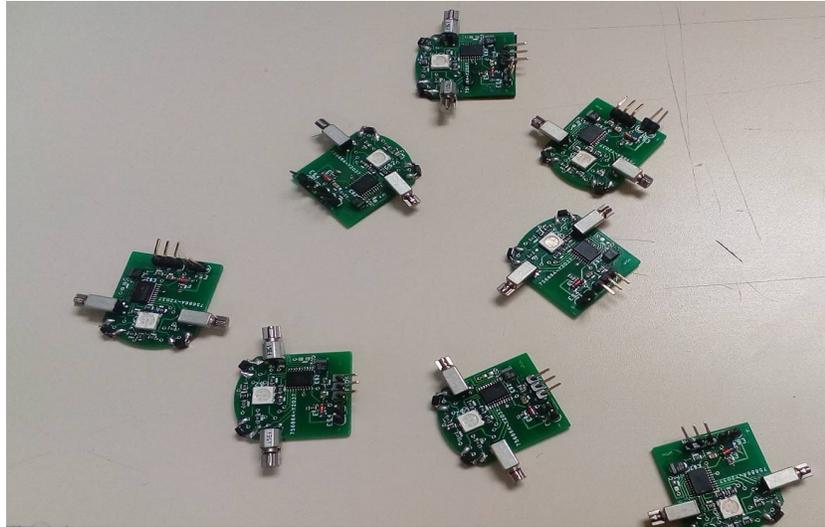


Figura 7.26: Primer avance de la colonia

Las pruebas se realizaron tomando como espacio de prueba una rejilla con celdas de dimensiones correspondientes a la medida de cada robot (4.5 X 4.5 cm).

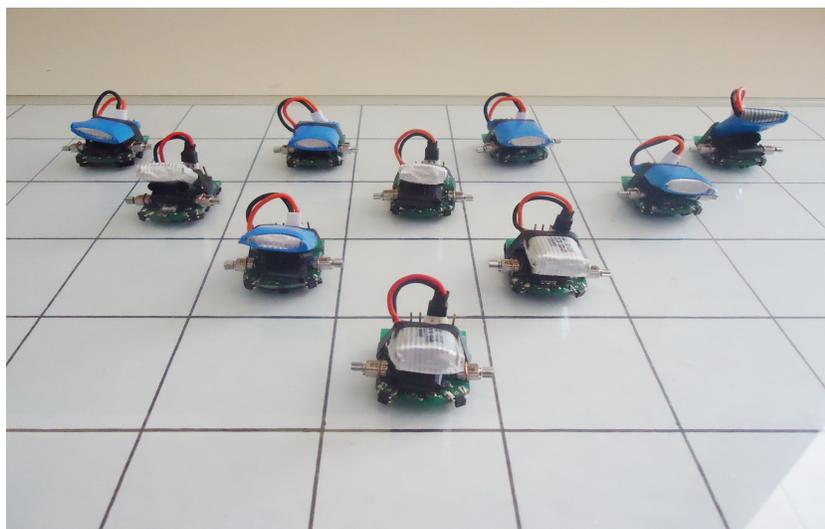


Figura 7.27: Colonia de antBots

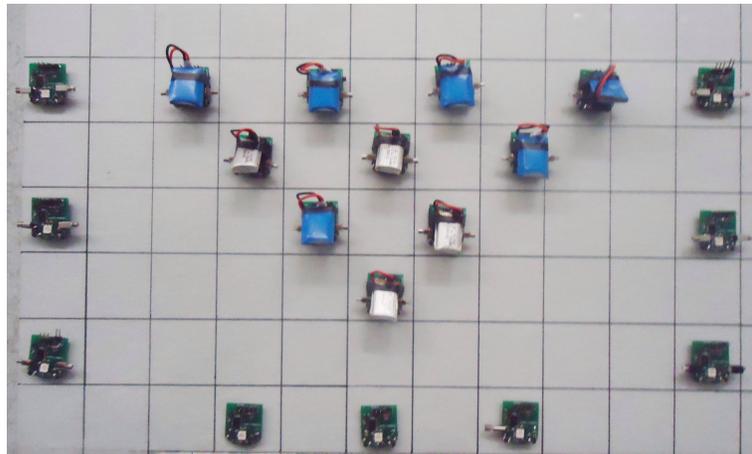


Figura 7.28: Colonia de antBots

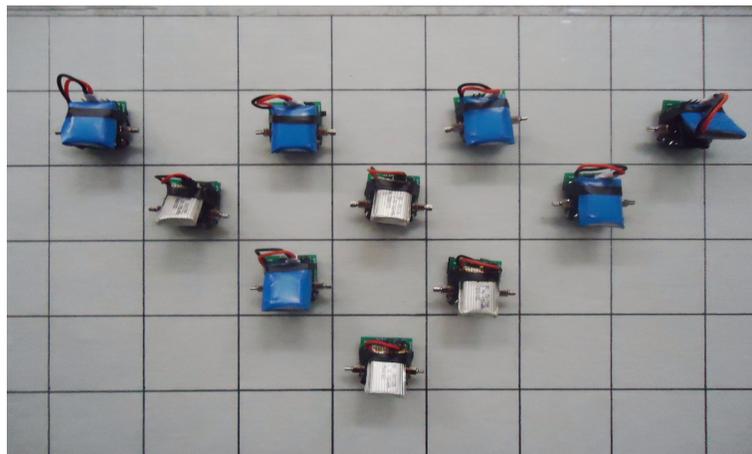


Figura 7.29: Colonia de antBots

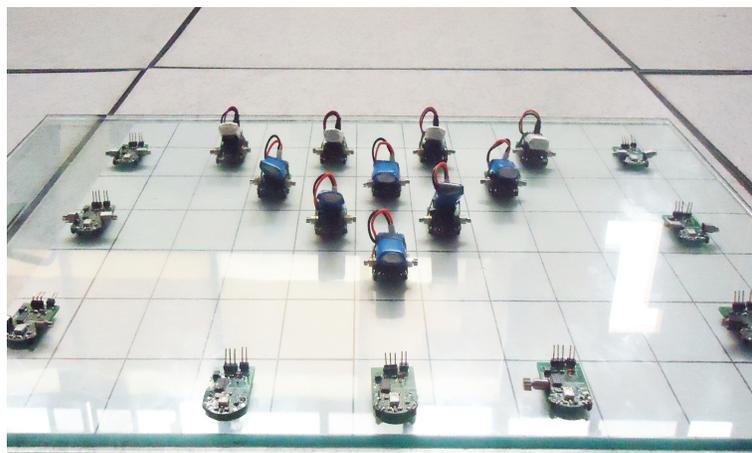


Figura 7.30: Colonia de antBots

7.2.3. Prototipo de carcasa para un antBot

La carcasa propuesta para el antBot está diseñada para el correcto funcionamiento del robot, no obstruyendo sus movimientos.

Debido al tamaño del prototipo la técnica seleccionada para la realización de la carcasa fue la de corte láser en acrílico.

Características:

- Medidas: 4 X 4.4 X 2.3 cm
- Peso: 5.8 g
- Material: Acrílico



Figura 7.31: Perspectivas de una carcasa para un antBot

7.2.4. Análisis de la dinámica de la colonia de robots

7.2.4.1. Análisis de movimientos

Con el propósito de proyectar a gran escala la regla de Langton adaptada y programada en cada robot, se llevó a cabo un análisis de los movimientos que emergen del mismo.

7.2.4.1.1. Cálculo probabilidad de giros.

Para obtener el cálculo de la probabilidad de giros fue considerado un grupo de diez robots. La siguiente tabla muestra el número de giros de noventa grados hacia la derecha y hacia la izquierda que el robot dio de un total de 100 intentos.

antBot	Giros de 90		Giros de 45	
	D	I	D	I
1	66	60	34	40
2	60	58	40	42
3	68	56	32	44
4	66	62	34	38
5	64	64	36	36
6	58	60	42	40
7	62	60	38	40
8	58	62	42	38
9	66	58	34	42
10	56	60	44	40

Total de antBots	Promedio Giros 90		Promedio Giros 45	
	D	I	D	I
10	62.4	60	37.6	40

De acuerdo a los resultados, la distribución normal implementada en el software es de 62/38.

7.2.4.2. Análisis de trayectorias

En las siguientes imágenes se ilustran los posibles movimientos de un robot trasladados al simulador

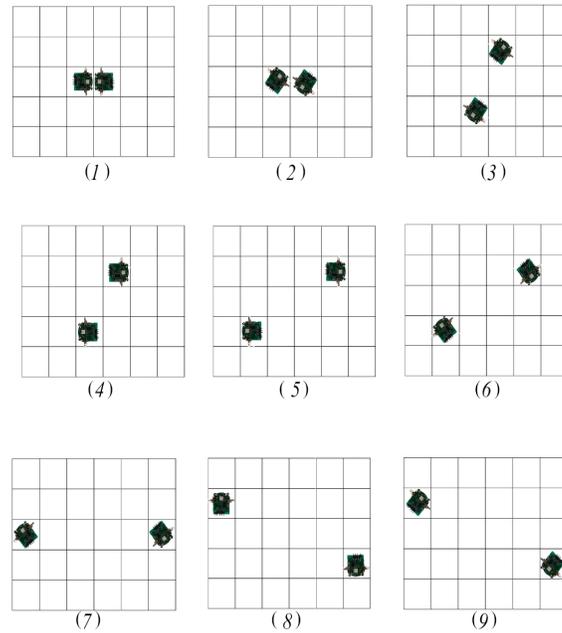


Figura 7.32: Trayectoria de dos robots cuando inician uno frente a otro

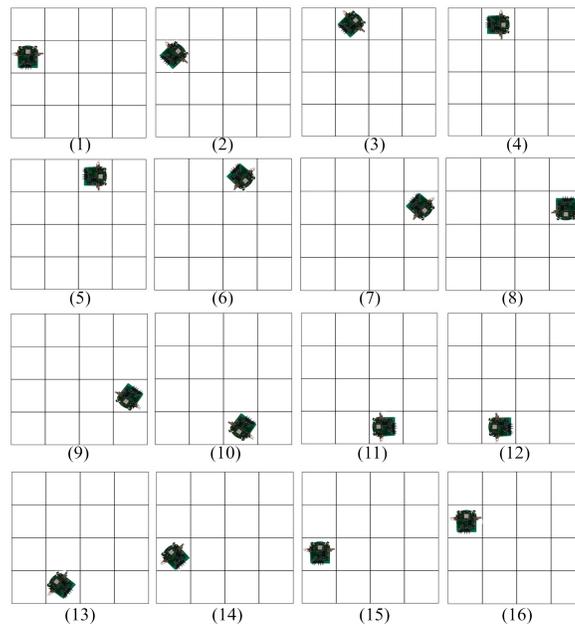


Figura 7.33: Trayectoria de un robot girando 45 grados a la derecha

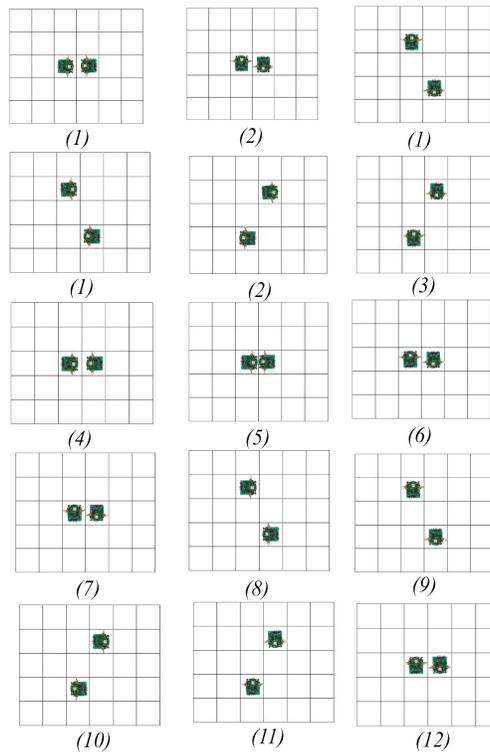


Figura 7.34: Trayectoria de dos robots cuando inician uno frente a otro

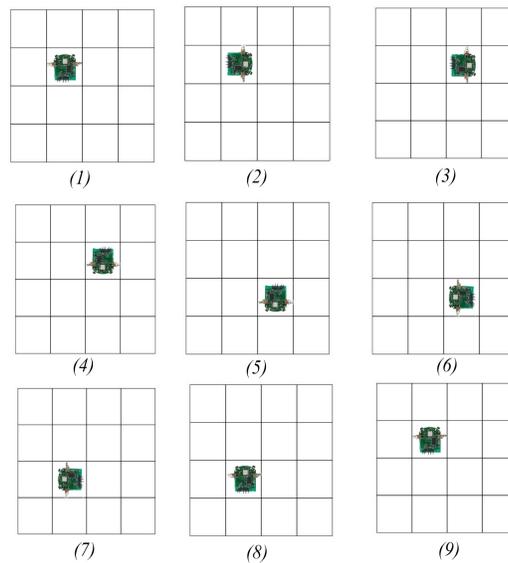


Figura 7.35: Trayectoria de un robot girando 90 grados a la derecha

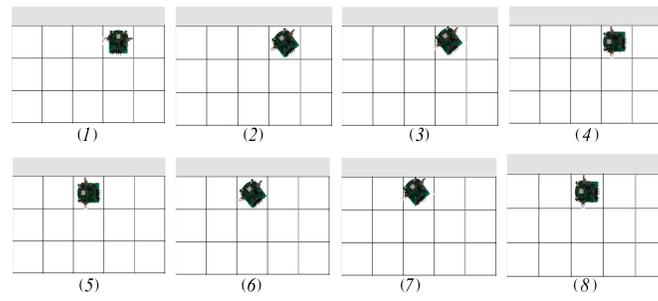


Figura 7.36: Trayectoria de un robot cuando llega a la frontera y gira 45 grados

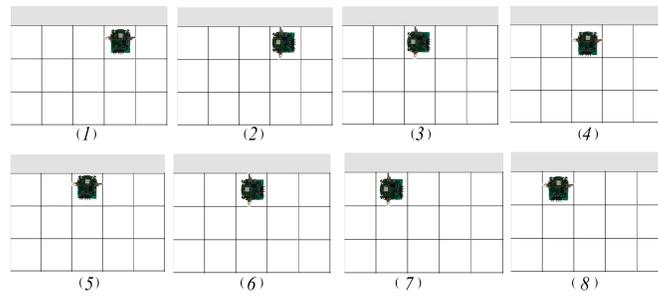


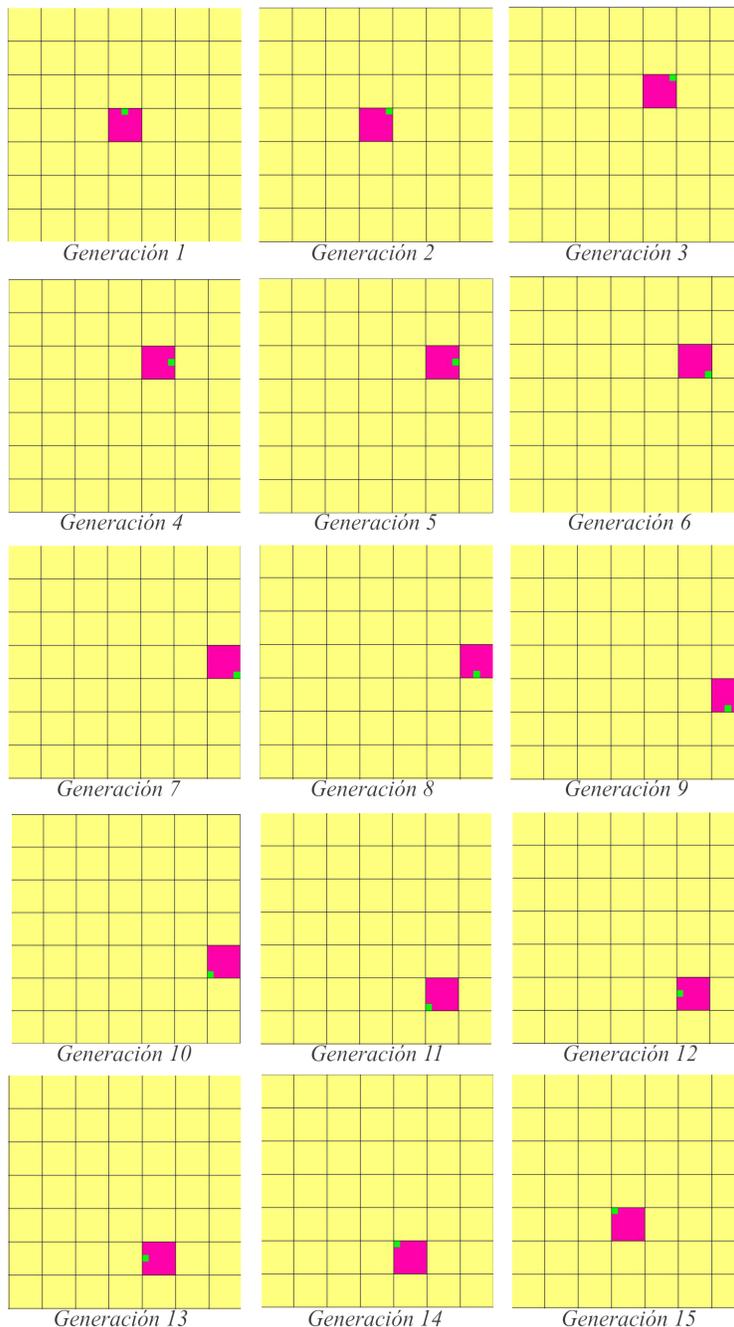
Figura 7.37: Trayectoria de un robot cuando llega a la frontera y gira 90 grados

7.3. Simulador de la regla adaptada de "La hormiga de Langton "

7.3.1. Giros válidos del antBot

Con el objetivo de simular la regla de Langton implementada en la colonia de robots, se llevó a cabo un conjunto de pruebas exhaustivas con cada uno de los elementos, dichos resultados permitieron aplicar los cambios pertinentes a la regla original de acuerdo a la física que se observó en la colonia de 20 robots. Las trayectorias mostraron cuatro posibles giros. A continuación se ilustra cada uno de ellos.

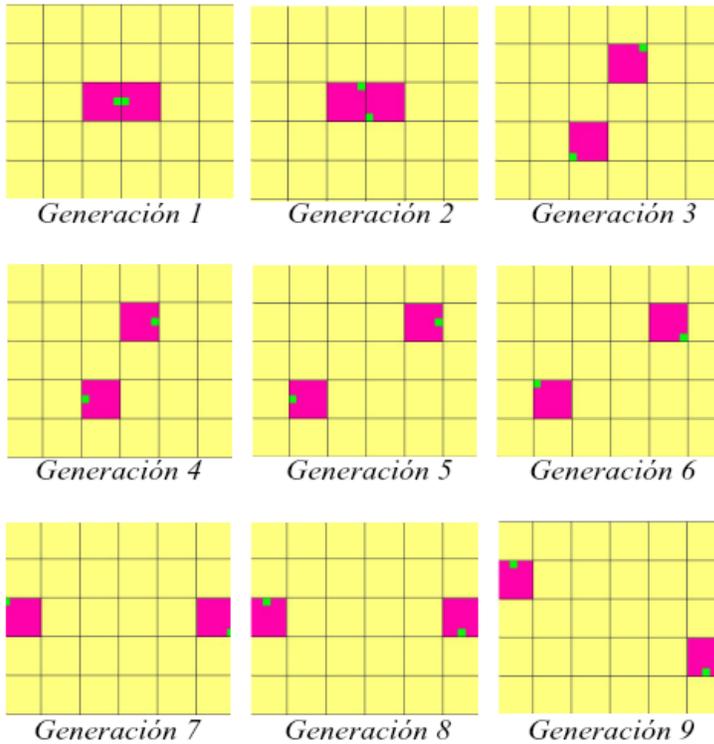
7.3.1.1. Giro de 45° a la Derecha



Dado que el robot posee un par de sensores infrarrojos al frente de su estructura que le facilita detectar a corta distancia algún obstáculo que impida su desplazamiento rectilíneo, éste verifica cada cierto tiempo si frente a él hay otro robot, en caso de no hallar obstáculo alguno el robot tiene la capacidad de girar cuarenta y cinco grados a la derecha y avanzar al siguiente espacio.

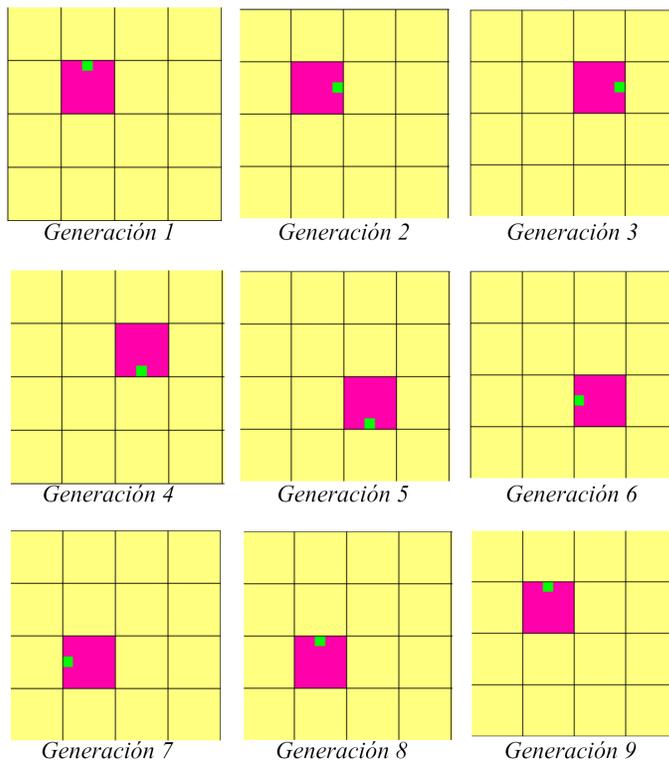
La secuencia de imágenes de la izquierda muestran el movimiento que el robot realiza generación a generación cuando no encuentra algún obstáculo. El punto verde indica el frente del robot. Para la generación uno, éste verifica la presencia de otro elemento, como no detecta algo, para la segunda generación el frente se encuentra ubicado en la esquina superior derecha para que en la generación número tres el robot avance una unidad.

7.3.1.2. Giro de 45° a la Izquierda



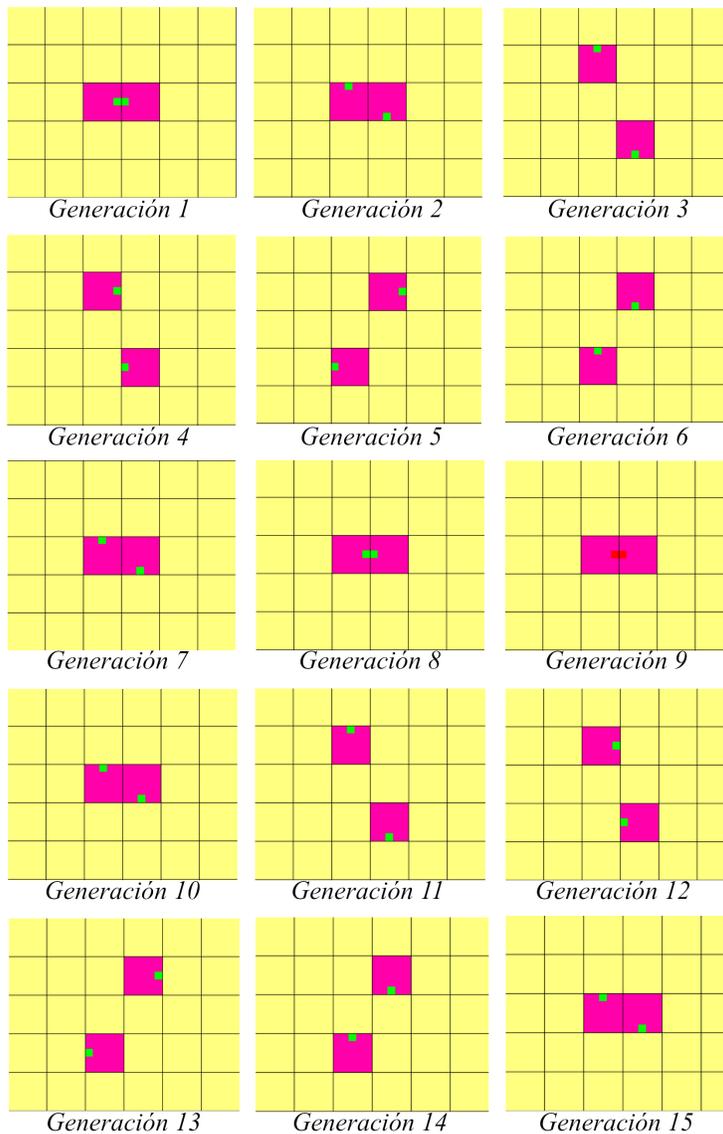
En el caso de que el sensor frontal del robot detecte que hay otro elemento frente a él, entonces éste deberá girar cuarenta y cinco grados a la izquierda y avanzar al siguiente espacio. La secuencia de imágenes de la izquierda muestra a dos robots que inician posicionados uno frente al otro, durante la generación número uno cada uno de ellos detecta que hay otro elemento enfrente, para la generación número dos cada uno realiza un giro de 45 grados hacia la izquierda en su misma ubicación, para que en la generación número tres los robots se muevan al siguiente espacio. El proceso se vuelve a repetir generación tras generación verificando si hay algún robot frente a ellos. Como éste no es el caso, entonces giran 45 grados a la derecha en las próximas seis generaciones.

7.3.1.3. Giro de 90° a la Derecha



El robot posee la capacidad de girar 90 grados a la derecha si el sensor no detecta algún elemento frente a él. A la izquierda observamos un robot inicia viendo hacia el norte, al no haber nada frente al mismo, hace un giro de noventa grados hacia la derecha para la generación número dos y para la generación tres avanza al siguiente espacio repitiendo el proceso durante las siguientes seis generaciones.

7.3.1.4. Giro de 90° a la Izquierda



Si un robot detecta la presencia de otro elemento frente a él, tiene la capacidad de realizar un giro de noventa grados en sentido opuesto a las manecillas del reloj. A la izquierda se puede observar un par de robots posicionados uno frente al otro, así que durante la generación uno cada elemento detectará un obstáculo enfrente obligándolo para la generación dos realizar un giro de noventa grados dentro de su espacio para que en la tercera generación cada uno de ellos avance a la siguiente unidad, repitiendo el proceso hasta la doceava generación. En la generación nueve, se observa que los frentes de cada robot se encuentran pintados de color rojo, éste mecanismo lo denominamos colisión, en párrafos posteriores se explica a detalle.

7.3.2. Casos a considerar por el antBot

7.3.2.1. Choque en la frontera

En el caso de que el robot se encuentre posicionado de frente a una frontera, los sensores frontales advertirán que hay un obstáculo, así que de acuerdo a la regla original de Langton, el robot se verá en la necesidad de girar ya sea noventa o cuarenta y cinco grados hacia la izquierda, avanzar a la siguiente unidad y repetir el proceso. A continuación se muestra una secuencia de movimientos con giros de cuarenta y cinco y giros de noventa grados cuando el robot enfrenta un choque con la frontera

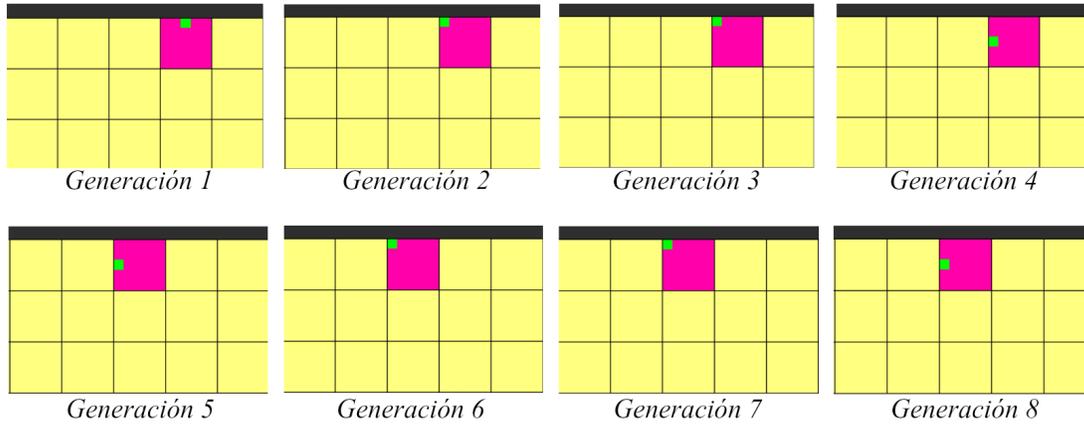


Figura 7.38: Giros de 45 grados

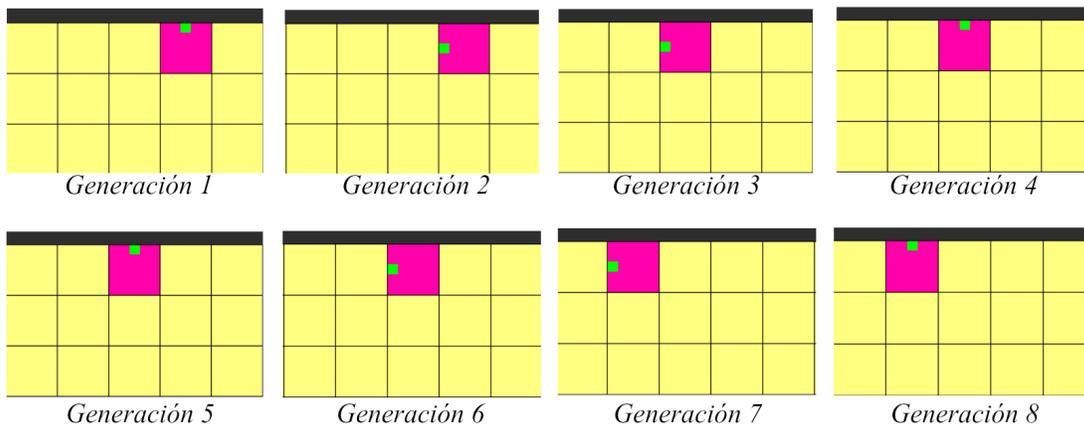


Figura 7.39: Giros de 90 grados

7.3.2.2. Colision de antBots

Es probable que frente al robot no exista obstáculo alguno, de modo que siguiendo la regla de Langton, éste elemento tiene que girar hacia la derecha, pero qué sucede cuando una vez hecho el giro, dos robots "chocan" y se impiden mutuamente avanzar a la siguiente unidad. Éste mecanismo recibe el nombre de colisión y la forma en que los robots físicos se encargan de resolverlo se traduce a lo siguiente:

El tiempo que le tomaría avanzar al siguiente espacio lo emplea en "empujar" al robot que tiene enfrente, haciendo que cada uno de ellos no se mueva de posición. Una vez que éste tiempo pasa y vuelve a verificar si hay un robot frente a él, ésta vez lo detectará y por ende hará un giro hacia el lado izquierdo.

A continuación se observa cómo es simulada la colisión para cuarenta y cinco y noventa grados. Sabemos que existe una colisión cuando en lugar de avanzar a la siguiente casilla, el frente del robot se pinta en color rojo y simplemente deja pasar esa generación en representación del choque. Para la siguiente generación, ambos detectan el obstáculo y realizan el giro pertinente.

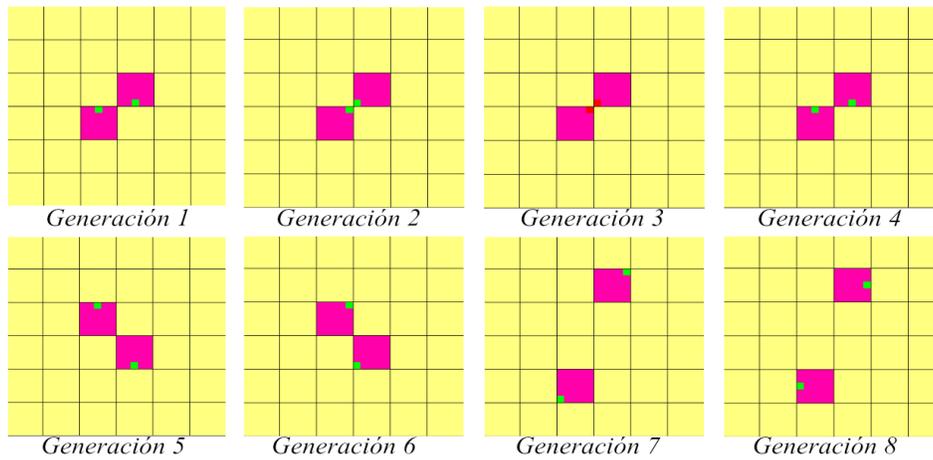


Figura 7.40: Giros de 45 grados

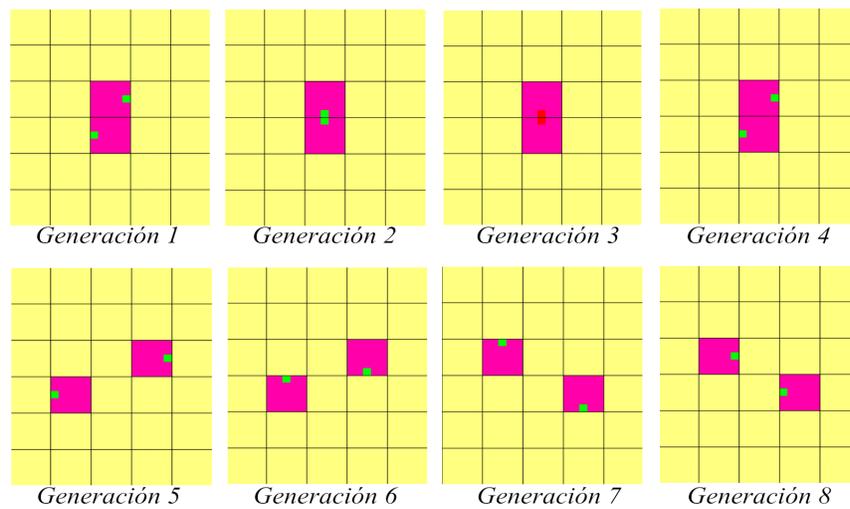


Figura 7.41: Giros de 90 grados

7.3.3. Comportamiento colectivo

En esta sección se presentan ejemplos del resultado de las simulaciones realizadas en el software desarrollado, considerando concentraciones de elementos en distintos puntos del plano, así como densidades de variación variable.

Densidad: 80 %	Densidad: 80 %	Densidad: 60 %
Densidad de concentración: 100 %	Densidad de concentración: 100 %	Densidad de concentración: 100 %
Número de generaciones: 200,000	Número de generaciones: 200,000	Número de generaciones: 200,000

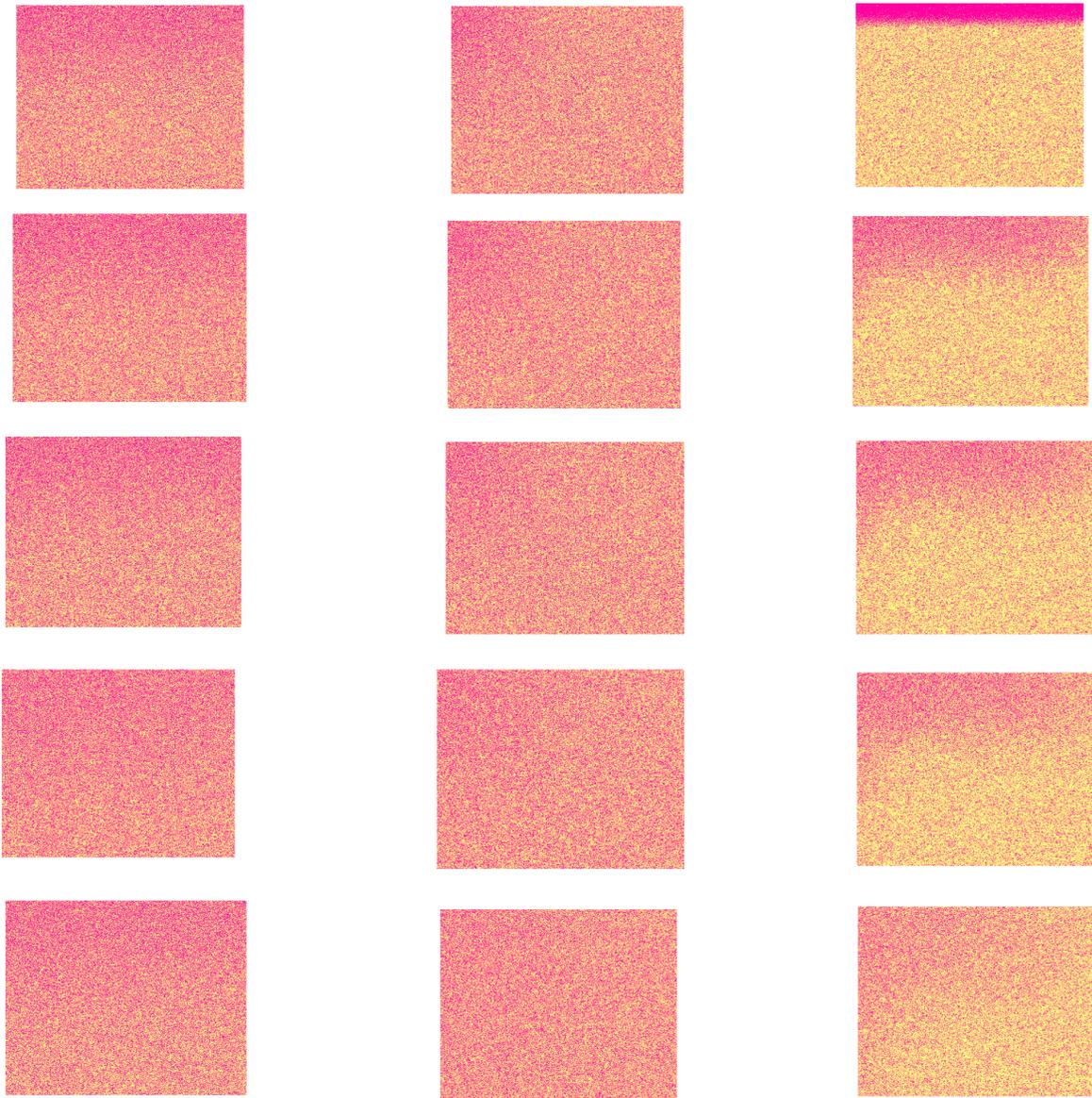


Figura 7.42: Simulaciones

7.3. SIMULADOR DE LA REGLA ADAPTADA DE "LA HORMIGA DE LANGTON "

Densidad: 15 %
Densidad de concentración: 100 %
Número de generaciones: 200,000

Densidad: 15 %
Densidad de concentración: 100 %
Número de generaciones: 200,000

Densidad: 20 %
Densidad de concentración: 100 %
Número de generaciones: 200,000

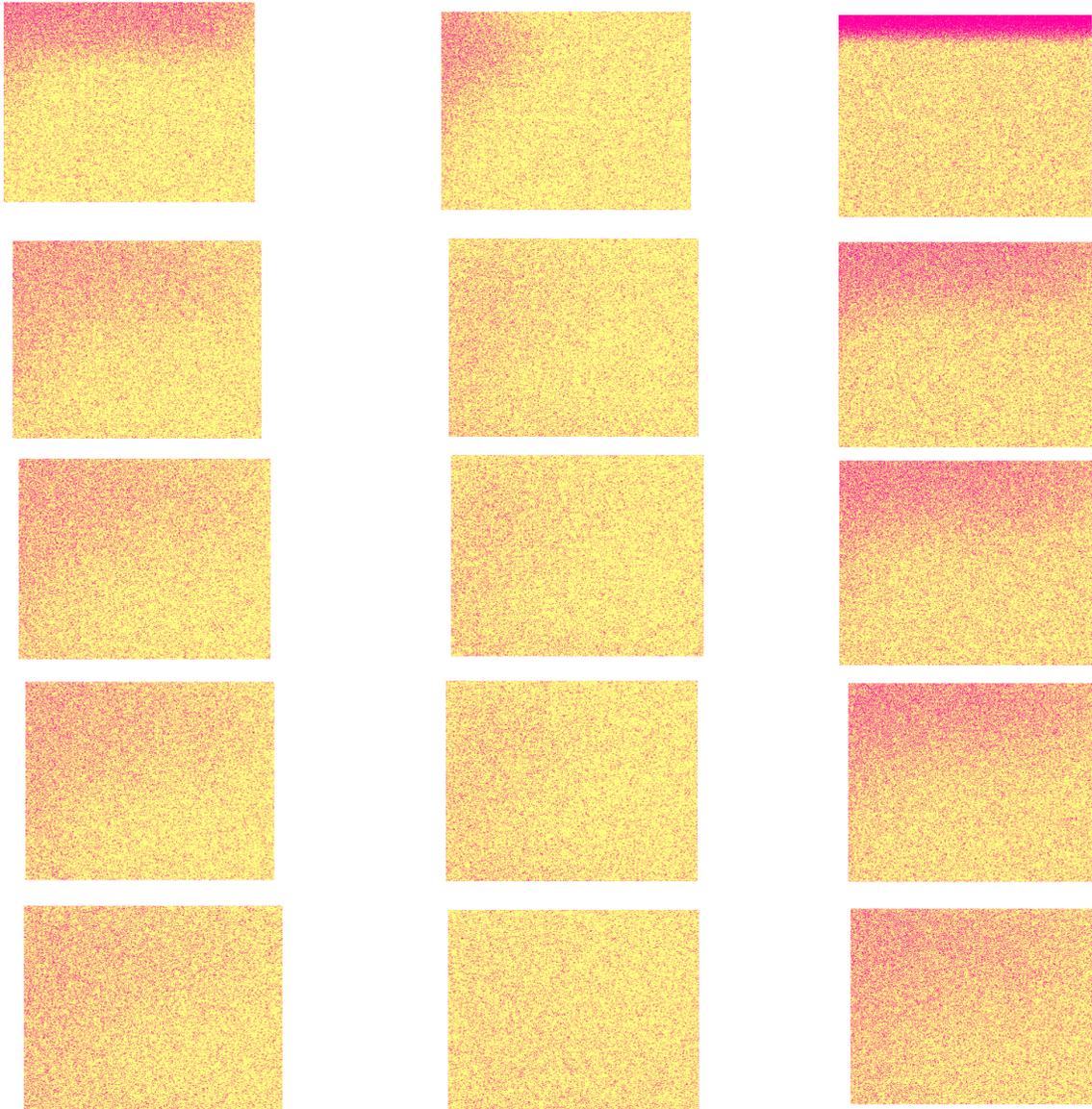


Figura 7.43: Simulaciones

Densidad: 25 %
Densidad de concentración: 100 %
Número de generaciones: 200,000

Densidad: 5 %
Densidad de concentración: 100 %
Número de generaciones: 200,000

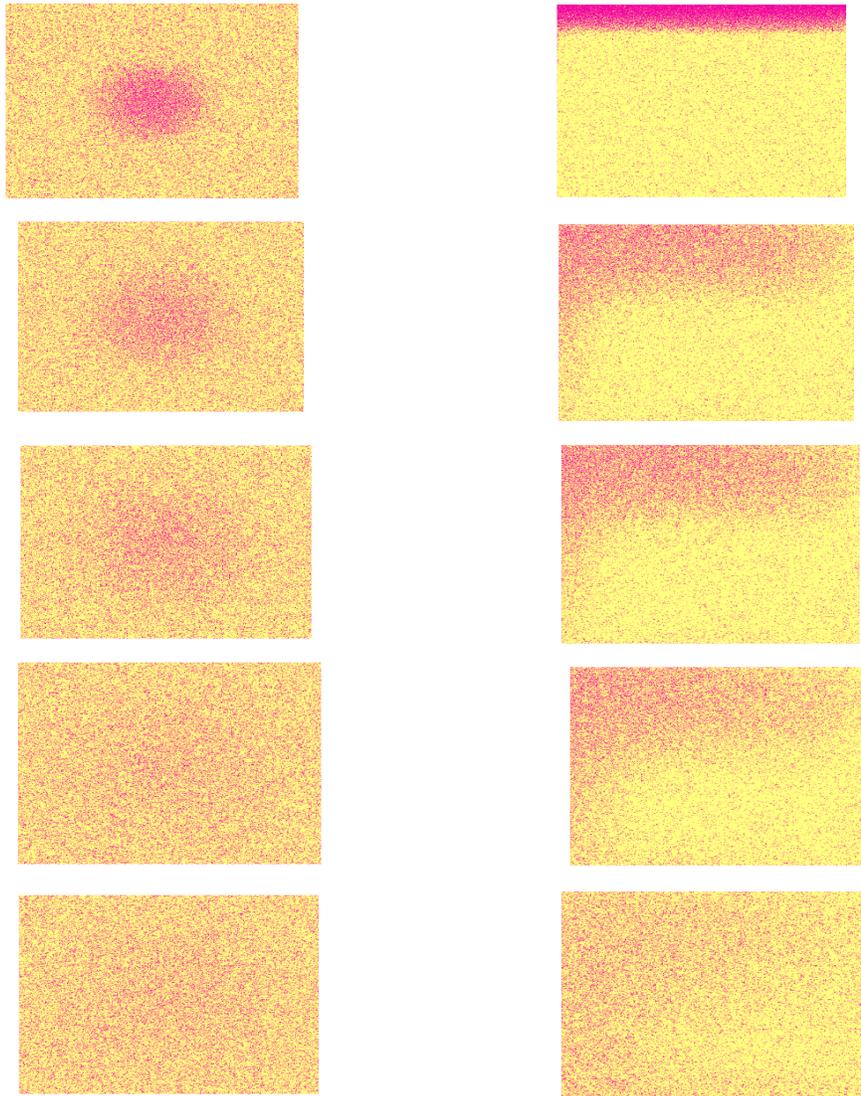


Figura 7.44: Giros de 45 grados

Se puede entonces concluir que es posible conseguir un comportamiento de tipo colectivo al implementar el autómata celular "La hormiga de Langton", considerando ciertas adaptaciones para su traslado a robots.

Conclusiones

Los autómatas celulares son una herramienta de gran ayuda al describir un sistema complejo ya que su simplicidad permite el fácil análisis del comportamiento resultante de dichos sistemas dinámicos.

El principal objetivo del presente trabajo terminal fue el de realizar un análisis sistemático del autómata celular “La hormiga de Langton”, el cual nos permitiría modelar una adaptación a la regla clásica, que llevada a la colonia de robots detonaría un comportamiento de tipo colectivo.

Al momento de iniciar la investigación, la información referente al autómata celular “La hormiga de Langton” era realmente limitada e implicó la recabación de datos desde fuentes en otro idioma.

Durante el desarrollo del trabajo se obvio el hecho de requerir habilidades más allá de la programación pues implicó el adquirir conocimientos más a detalle en ciencias de la computación así como en el área de electrónica.

En el proceso de elaboración del prototipo del robot “antBot” se presentaron ciertos problemas que se puntualizan a continuación:

- La falta de información en el área nos llevo a invertir una gran cantidad de tiempo en adentrarnos en el marco teórico del trabajo así como verificando los resultados de las pruebas realizadas.
- El tamaño de los robots fue una propiedad crítica durante el desarrollo del prototipo pues los componentes fueron seleccionados de acuerdo a dicha característica.
- Debido al tamaño del prototipo la fijación de los componentes resultó complicada pues la precisión necesaria era difícil de conseguir.
- El movimiento vibratorio de la “hormiga” derivó en el desajuste de los propios motores por lo que su calibración no fue trivial y lograr su correcto funcionamiento requirió un gran esfuerzo.
- La obtención de la regla adaptada para su implementación en el enjambre conllevó un trabajo arduo de observación y un largo proceso de prueba y error.

La pertinente solución de los obstáculos antes mencionados permitió la correcta traslación del modelo del autómata celular “La hormiga de Langton” a una colonia de robots los cuales demostraron un comportamiento

colectivo no trivial.

Como resultado final del proyecto de titulación se obtuvo una serie de pruebas de la dinámica del autómata celular “La hormiga de Langton” provenientes de una investigación sistemática. Se ensambló una colonia de robots de bajo costo y se concluyó la presencia de comportamiento colectivo no trivial en la adaptación de la regla clásica implementada en la colonia de antBots.

Ciertamente una aplicación directa de la investigación realizada requeriría de una inversión mayor de tiempo pero las bases asentadas en cuanto a información de este autómata celular y sus propiedades de comportamiento colectivo permitirían en el futuro su implementación en casos prácticos.

CAPÍTULO 9

Bibliografía

Bibliografía

- [1] S. Wolfram, *"A new kind of science"*, Champaign, Ill.: Wolfram Media, 2002.
- [2] C. Langton, *"Studying artificial life with cellular automata"*, *Physica D: Nonlinear Phenomena*, vol. 22, no. 1-3, pp. 120-149, 1986.
- [3] D. Reyes Gómez, *"Descripción y Aplicaciones de los Autómatas Celulares"*, 1st ed. Puebla, México., 2011, pp. 3-4. [En línea]. Disponible en: http://delta.cs.cinvestav.mx/~mcintosh/cellularautomata/Summer_Research_files/Arti_Ver_Inv_2011_DARG.pdf
- [4] S. Wolfram, *"Theory and applications of cellular automata"*. Singapore: World Scientific, 1986.
- [5] G. Juárez Martínez, *"A Note on Elementary Cellular Automata Classification"*, *Journal of Cellular Automata*, vol. 8, pp. 3-4, 2013.
- [6] Martin Gardner, *"The fantastic combinations of John Conway's new solitaire game life"*. *Scientific American*, vol. 223, pp. 120–123, 1970.
- [7] Christopher Swann, *"Cardumen de caballas"*, Vista al mar, 2013. [En línea]. Disponible en: <https://www.vistaalmar.es/images/ampliadas/cardumen-caballas.jpg>
- [8] Stephanie Gannon, *"The Black Sun, Denmark"*, IMGUR, 2015. [En línea]. Disponible en: <http://imgur.com/gallery/w9ZYR>
- [9] Latty T, *"Hormigas argentinas"*, tanyalatty, 2014. [En línea]. Disponible en: <https://static1.squarespace.com/static/53c5d670e4b015d0827e391f/t/53d255c9e4b02f566edf4a68/1406293456305/3+nest+best.png?format=500w>
- [10] A. Gajardo, A. Moreira and E. Goles, *"Complexity of Langton's ant"*, *Discrete Applied Mathematics*, vol. 117, no. 1-3, pp. 41-50, 2002.
- [11] A. Gajardo, E. Goles and A. Moreira, *"Generalized Langton's Ant: Dynamical Behavior and Complexity"*, *STACS 2001*, pp. 259-270, 2001.
- [12] M. Rubenstein, Ch. Ahler and R. Nagpal, *"Kilobot: A Low Cost Scalable Robot System for Collective Behaviors"*, *Proceedings ICRA IEEE International Conference on Robotics and , 2012*, vol. 5, pp. 4-5.
- [13] G. Juárez Martínez, *"Introducción a la simulación de procesos con autómatas celulares"*, 1st ed. México, 2006, pp. 7-8. [En línea]. Disponible en: <https://2006.igem.org/wiki/images/1/1b/IntoCA.pdf>

- [14] G. Juárez Martínez and L. Oliva Moreno, "*Comportamiento colectivo no trivial en sistemas complejos con mini-robots*", 1st ed. México: UWE, 2015, pp. 12-14. [En línea]. Disponible en:<http://eprints.uwe.ac.uk/26735/6/collectiveBehaviour.pdf>
- [15] S. Wolfram, "*Universality and complexity in cellular automata*", Physica D: Nonlinear Phenomena, vol. 10, no. 1-2, pp. 1-35, 1984.
- [16] S. Shanthi, P. Rao, M. Latha and E. Rajan, "*Cellular Automata and Their Realizations*", 2012 International Conference on Computing Sciences, 2012.
- [17] C. Langton, "*Self-reproduction in cellular automata*", Physica D: Nonlinear Phenomena, vol. 10, no. 1-2, pp. 135-144, 1984.
- [18] S. Wolfram, "*Cellular automata and complexity*", 1st ed. Boulder, Co: Westview-Press, 2002.
- [19] J. Davies, "*MSP430 microcontroller basics*", 1st ed. Amsterdam: Elsevier Newnes, 2011.
- [20] M. Dorigo, "*Ant colony optimization and swarm intelligence*", 1st ed. Berlin: Springer, 2006.
- [21] J. Holland and E. Torres Alexander, "*El orden oculto*", 1st ed. Mexico: Fondo de Cultura Económica, 2004.
- [22] R. Lewin, "*Complexity*", 1st ed. Chicago: The University of Chicago, 1999.
- [23] J. Schiff, "*Cellular automata*", 1st ed. Hoboken, New Jersey: Wiley-Interscience, 2008.
- [24] R. Turner and L. Killian, "*Collective behavior*", 1st ed. Englewood Cliffs, N.J.: Prentice-Hall, 1987.
- [25] G. Sirakoulis and A. Adamatzky, "*Robots and Lattice Automata*", Springer, 2015.
- [26] Mitchell M., "*Complexity: A Guided Tour*", Oxford University Press, 2009.
- [27] K. Mainzer, and L. Chua, "*The Universe as Automaton*", Springer, 2012.
- [28] H. Chate, and P. Manneville, "*Evidence of collective behavior in cellular automata*", Europhysics Letters, vol. 14, pp. 409-413, 1991.
- [29] G. Juárez Martínez, A. Adamatzky and H. V. McIntosh, "*Complete Characterization of Structure of Rule 54*", Complex Systems, vol. 23, 2014.
- [30] J. C. Seck-Tuoh-Mora, J. Medina-Marina, N. o Hernandez-Romero, G. Juárez Martínez and I. Barragan-Vite, "*Welch sets for random generation and representation of reversible one-dimensional cellular automata*", Information sciences, vol. 382-383, pp.81-95, 2017.
- [31] A. Adamatzky, G. Juárez Martínez and J. C. Seck Tuoh Mora, "*Phenomenology of reaction-diffusion binary-state cellular automata*", Information sciences, vol. 16, pp. 2985-3005, 2006.
- [32] R. Pressman and B. Maxim, "*Software engineering*", New York: McGraw-Hill Education, 2015.
- [33] Ian Sommerville, "*Ingeniería de Software*", Pearson, 2011.
- [34] A. Fernandez and D. Dang, "*Getting started with the MSP430 Launchpad*", UK: Newnes, 2013.
- [35] M. Gomez Gom, "*Electronica general*", Alfaomega, 2007.
- [36] D. Harresm, "*MSP430-based Robot Applications*", Burlington: Elsevier Sciencel, 2013.
- [37] D. Harresm, "*MSP430-based Robot Applications*", Burlington: Elsevier Sciencel, 2013.
- [38] H. V. McIntosh, "*One Dimensional Cellular Automata*", Luniver Press, 2009.

CAPÍTULO 10

Glosario de términos

En esta sección se definen algunos términos necesarios para comprender el presente trabajo temrinal.

AC : Es la abreviación de autómata celular.

GOL : Es la abreviación del autómata celular más famoso “El juego de la vida” (Game of Life).

N : El punto cardinal norte.

S : El punto cardinal sur.

E : El punto cardinal este.

O : El punto cardinal oeste.

CCS : Code Composer Studio.

Qt : Framework multiplataforma orientado a objetos ampliamente usado para desarrollar programas (software) que utilicen interfaz gráfica de usuario

RGB : Red, Green, Blue, se refiere a un sistema de representación de colores.

Batería LiPo : Batería de polímero de litio, son batería recargables compuestas generalmente de varias células secundarias idénticas en paralelo para aumentar la capacidad de la corriente de descarga.

Swarm : Palabra en inglés que significa enjambre.

Trivial :Que no sobresale de lo ordinario y común.

Autómata celular : Modelo matemático de un sistema dinámico.

Vida artificial : Disciplina que estudia la vida, tratando de recrear o imitar fenómenos biológicos mediante el empleo de técnicas de computación.

Sistema complejo : Un sistema complejo está compuesto por varias partes interconectadas cuyos vínculos crean información adicional no visible antes por el observador.

Comportamiento complejo : Emerge espontáneamente a partir de principios sencillos.

Rejilla : Tejido en forma de red.

Bioquímica : Ciencia que estudia la estructura química y las funciones de los seres vivos.

Virtual : Que tiene existencia aparente y no real.

Vants : Expresión acuñada por C. Langton que significa hormigas virtuales (virtual ants).

Máquina de Turing : Modelo matemático que consiste en un autómata capaz de implementar cualquier problema matemático expresado por medio de un algoritmo.

Imprevisible : Que no se puede prever.

Optimizar : Buscar la mejor manera de realizar una actividad.

Determinista : Completamente predictivo, no hay parte de azar.

Problema NP : Es el conjunto de problemas que pueden ser resueltos en tiempo polinómico por una máquina de Turing no determinista.

Problema P completo : Que se encuentra dentro de los NP y puede ser resuelto mediante paralelización.

Grafo : Diagrama que representa mediante puntos y líneas las relaciones entre pares de elementos y que se usa para resolver problemas lógicos, topológicos y de cálculo combinatorio.

Algoritmo : Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.