

# Implementación de un agente basado en objetivos mediante estrategias de búsqueda y cálculo situacional

Gómez, José Ignacio  
Hechim, Pablo Nicolás  
Paciuk, Sebastián Federico

*Alumnos ISI de 5to año  
UTN – Fac. Regional Santa Fe*

## Abstract

*La inteligencia artificial es una rama de las ciencias de la computación que estudia cómo agentes inteligentes se desenvuelven en un ambiente determinado para lograr conseguir un objetivo específico.*

*El presente trabajo se basa en una consigna de la cátedra Inteligencia Artificial, que consiste en el diseño e implementación de un agente inteligente para el control de un drone que tiene por objetivo encontrar la posición exacta de los victimarios en un sector específico de la ciudad de Santa Fe. Para alcanzarlo, el drone puede utilizar técnicas de cálculo situacional y búsqueda - informadas, en el caso de A\*, o no informadas en el caso de amplitud, profundidad y costo uniforme -.*

*Se estudian las diferencias más importantes entre estos algoritmos de resolución de problemas, al mismo tiempo que se diseña y construye un agente que brinde respuesta a la situación planteada.*

## Palabras Clave

Inteligencia Artificial, agente inteligente, búsqueda, cálculo situacional, ambiente, percepciones, acciones.

## 1. Introducción

Un agente es un sistema que está situado en algún ambiente, y que es capaz de realizar acciones autónomas en ese ambiente, para cumplir sus objetivos de diseño [1]. Para que un agente sea inteligente debe ser proactivo, reactivo y tener habilidad social [2]. Aplica técnicas de Inteligencia Artificial (IA) para definir su comportamiento y establecer las acciones que puede llevar a cabo, de acuerdo a la circunstancia en la que se encuentra.

Se considera como ambiente a un sector de la ciudad de Santa Fe donde el agente drone percibe señales emitidas por pequeños dispositivos que son disparados sobre

personas por un sistema de alertas en un ambiente nano-radio. Estas alertas son accionadas como consecuencia de un hecho ilícito y ponen al agente en búsqueda del criminal que ha cometido el hecho, debiendo éste distinguir entre víctimas y victimarios para así encontrar al victimario en cuestión. Para ello, contará con una cámara que le permitirá identificar si una persona que tiene frente a él es víctima o victimario.

El agente se mueve en tres niveles de altitud: alto, medio y bajo. En el nivel alto y medio, el drone se mueve por encima de cualquier edificio, mientras que en el nivel bajo deberá moverse por encima de las calles, yendo de una esquina a la otra. Podrá conocer su ubicación (latitud, longitud) y altura por medio de un GPS.

Además, sólo podrá conocer la ubicación exacta de una persona cuando se encuentre en el nivel bajo, ya que de otro modo distingue sólo la potencia de señales provenientes de una determinada zona (cuyo alcance se incrementa en el nivel alto). Para percibir las señales, el drone utiliza un tercer sensor, que funciona a través de una antena que percibe las señales de los nano-radios.

Para los niveles alto y medio, el drone sólo podrá utilizar una estrategia de búsqueda, mientras que para el nivel bajo puede escoger entre búsqueda o cálculo situacional. La diferencia entre estas dos estrategias radica en que el cálculo situacional utiliza un agente basado en conocimiento, que actúa según ciertas reglas que le permiten razonar, y de esa forma reaccionar a sus percepciones hasta poder

llegar a su objetivo. Por otro lado, búsqueda es un método exploratorio, donde el agente puede realizar una determinada cantidad de acciones, y en base a las distintas secuencias de acciones elige aquella que le permite llegar a su objetivo; el criterio de elección depende de cuál algoritmo se utilice (amplitud, profundidad, costo uniforme o A\*).

## 2. Elementos del Trabajo y Metodología

### 2.1 Técnicas

#### 2.1.1 Búsqueda

El agente percibirá el entorno a través de su cámara, GPS y antena, y decidirá qué acciones pertinentes debe llevar a cabo para distinguir y encontrar a un victimario.

El ambiente del agente será lo que el dron percibe y a quien responde, es decir, el ambiente le proveerá señales y será el dron quien le devuelva una acción, para seleccionar la acción, el agente utilizará una estrategia de búsqueda que le permite explorar los diferentes estados alcanzados al ejecutar acciones. Luego, si el agente encuentra un camino desde el estado inicial al estado objetivo, entonces la solución consistirá en seleccionar la primera acción de este camino. Se ilustra lo expresado en la Figura 1.

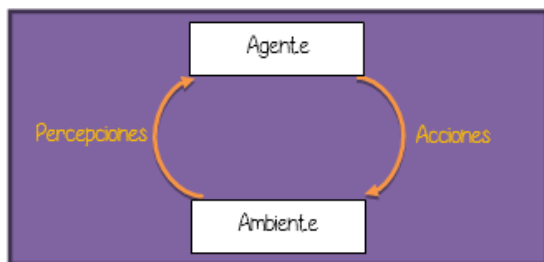


Figura 1 - Interacción Agente – Ambiente

Se necesita que el agente realice un proceso de búsqueda cada vez que perciba señales del ambiente. Este proceso se representa en la Figura 2.

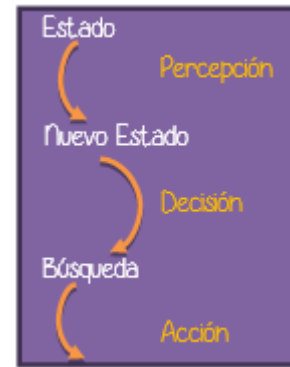


Figura 2 - Proceso de decisión de acción

#### 2.1.2 Cálculo situacional

El agente percibirá el entorno a través de su cámara y GPS, y decidirá qué acciones pertinentes debe llevar a cabo para distinguir y encontrar un victimario. Notar que la percepción de antena que posee en la técnica de búsqueda no se utiliza, debido a que para cálculo situacional el dron se mueve únicamente en el nivel bajo.

El ambiente del agente será lo que el dron percibe. Incorporará conocimiento a partir de sus percepciones, y ejecutará una acción racional a partir de su base de conocimiento que modificará el ambiente. Se necesita que realice un proceso de cálculo situacional cada vez que perciba señales del ambiente. El comportamiento se ilustra en la Figura 3.

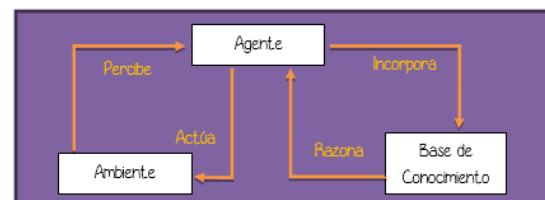


Figura 3 - Interacción Agente - Ambiente

## 2.2 Marco de implementación

Como marco de implementación para la resolución del problema, se utilizó lenguaje de programación JAVA SE 1.6, utilizando la herramienta Eclipse Galileo como entorno de desarrollo. Se utilizó un framework que permite realizar el modelo conceptual de un agente y, a partir de éste, generar las clases Java que representan el ambiente, el agente, y toda clase involucrada en su interacción (percepciones, acciones, entre otros). Dicho framework recibe el nombre de FAIA [3].

Por otro lado, se utiliza la herramienta IDEM-IA [4], que permite la representación gráfica del modelo agente-ambiente; además, al estar integrado a FAIA, permite generar las clases del modelo directamente del diagrama.

Para la etapa de cálculo situacional se emplea un lenguaje de programación lógica: SWI-Prolog; mediante el mismo se representa la base de conocimiento del agente, y el conjunto de reglas que le permiten resolver el problema de buscar un victimario en el nivel bajo.

## 2.3 Modelado del agente y su ambiente

En IDEM-IA se modela el estado del agente, las operaciones que este puede realizar, el estado del ambiente, las percepciones del agente y la función objetivo definida. La figura 4 muestra el modelo conceptual del agente definido para resolver el problema.

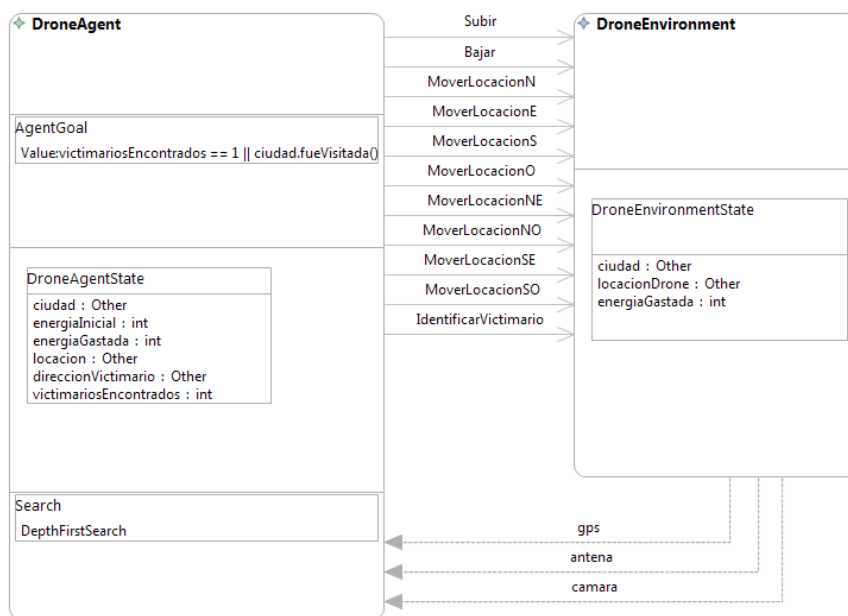


Figura 4 – Diagrama IDEM-IA para el agente cuando éste utiliza búsqueda

Como se observa, se define **DroneAgent** y **DroneEnvironment** como el agente y su ambiente respectivamente.

Puede observarse que el dron deberá finalizar su búsqueda cuando haya encontrado al victimario o bien cuando haya recorrido toda la “ciudad” (que representa el mapa con los lugares que puede recorrer) y no lo haya encontrado. Esta segunda

situación se introduce como consecuencia del uso de una estrategia de búsqueda para resolver el problema, ya que inicialmente todo lo que el dron percibe son señales de distintas intensidades según la zona; en otras palabras, en la etapa de búsqueda no sabrá cuál es el lugar donde se halla el victimario, y aunque busque no podrá encontrarlo.

Al utilizar cálculo situacional, esta situación no resulta necesaria ya que el dron contará con reglas que le permiten llegar al objetivo aún cuando no conoce el lugar preciso en que se encuentra el victimario. Recordemos, además, que en el cálculo situacional se utilizará sólo en el nivel bajo, por lo cual el dron ya habrá percibido las señales con la antena, y no es necesario percibir otra vez.

### 2.3.1 Estado del agente

*ciudad*: Estructura de datos que representa el mapa donde el dron puede moverse, y

sólo posee los datos que éste conoce.

*energíaInicial*:

Representa las unidades de energía inicial que dispone el dron.

*energíaGastada*:

Representa el consumo total acumulado de energía que realiza el dron al ejecutar acciones.

*locación*: Representa la posición actual del dron dentro del mapa (ciudad).

*direcciónVictimario*:

Cuando el dron se encuentra en el nivel

bajo, indica si hay un victimario en alguna de las posiciones cardinales respecto a la posición actual del dron. En otros niveles existe, pero carece de valor utilizarlo.

*victimariosEncontrados*: Indica la cantidad de victimarios encontrados por el dron.

### 2.3.2 Estado del ambiente

*ciudad*: Estructura de datos que representa el mapa donde el dron puede moverse. Contempla las intensidades, cantidades y ubicaciones de víctimas y victimarios.

*locaciónDrone*: Indica la ubicación actual del dron.

*energíaGastada*: Indica la energía gastada hasta el momento por el dron.

### 2.3.3 Percepciones del dron

*GPS*: Permite al dron saber cuál es su ubicación actual.

*Antena*: Permite al dron identificar unívocamente las señales de los nano – radios. Dichas señales son percibidas con distinta intensidad según el nivel de altura en el que se encuentre el dron.

*Cámara*: Permite al dron identificar a un victimario en el nivel de altitud “bajo”. Le muestra en línea recta todas las calles desde la posición actual hasta el límite del cuadrante en el cual se encuentra, pudiendo así decidir en qué dirección moverse para encontrar un victimario.

### 2.3.4 Operadores del dron

*Subir*: Acción que permite al dron desplazarse desde el nivel de altitud “bajo” a “medio” y desde “medio” a “alto”.

*Bajar*: Acción que permite al dron desplazarse desde el nivel de altitud “alto” a “medio” y desde “medio” a “bajo”.

*MoverLocación<DirecciónCardinal>*:

Acción que permite al dron desplazarse hacia un cuadrante, subcuadrante o esquina adyacente.

*Identificar*: Acción que permite al dron diferenciar víctimas de victimarios cuando se encuentra en una esquina.

### 2.3.5 Prueba de meta - Estrategias de Búsqueda

Se considera que se alcanzó el objetivo si se encuentra un victimario o se recorre toda la ciudad (o todas las esquinas que tienen señal):

```
victimariosEncontrados == 1 ||  
ciudad.fueVisitada() == TRUE
```

Vale la pena destacar que la condición en que se considera la ciudad visitada, es introducida para dos casos: el primero es el caso en que el victimario haya salido del rango alcanzable por el dron y, por lo tanto,

no exista victimario en las zonas que éste vigila. El segundo de los casos y tal vez más importante, es que en la etapa de búsqueda en la que el dron “piensa” cuál es la mejor acción a utilizar a continuación, el dron puede (y muy seguramente) que no conozca la posición del victimario, por lo cual en esta etapa no podrá encontrarlo.

### 2.3.6 Heurística

La heurística se utiliza en las estrategias de búsqueda A\*, y aquí se plantea como el cociente entre la energía disponible del dron y la intensidad percibida en su locación actual. Si el dron se encuentra en el nivel de altitud “bajo” y no hay victimario en ninguna de las direcciones posibles a tomar en línea recta, se sumará un “plus condicional” que actuará como penalización en la relación definida. En la Figura 5 se expresa la heurística definida.

$$h(n) = \frac{\text{energía disponible}}{1 + \text{intensidad percibida}} + \text{plus condicional}$$

Figura 5 - Función Heurística para Búsqueda A\*.

### 2.3.7 Definición de reglas diagnósticas

Estas reglas se utilizan en cálculo situacional y permiten determinar propiedades del ambiente a través de las percepciones. Se definieron tres tipos de reglas diagnósticas, una que determina si hay un victimario en la esquina actual (a través de la Cámara), la otra si hay un victimario en alguna dirección cardinal (a través de la Cámara) y por último la posición actual del dron (a través del GPS).

Reglas expresadas en SWI-Prolog:

1. `hayVictimario(Esq,S):-  
    perception(_,_,_,_,_,_,_,1,  
        Xpos,Ypos),esquina(Esq,Xpos,  
        ,Ypos),actualSituation(S).`
2. `direccionVictimario(N,NE,E,SE,Su,  
    ,SO,O,NO,PosAct,S):-  
    perception(N,NE,E,SE,Su,SO,O,  
        NO,PosAct,_,_),actualSituation(S).`

3. posicion(Xpos,Ypos,S):-  
 perception(\_,\_,\_,\_,\_,\_,\_,\_,\_,Xpos,  
 s,Ypos),actualSituation(S).

### 2.3.8 Valoración de acciones

En cálculo situacional se valoran las acciones, y aquí se establecieron cuatro valoraciones:

1. Excelente: Se considera así a la acción que permite que *DroneAgent* identifique al victimario. Esto se da cuando se encuentra en una esquina con victimario.

2. Muy buena: Se considera así a la acción que permite que *DroneAgent* se desplace en una dirección en la que hay un victimario.

3. Buena: Se considera así a la acción que permite que *DroneAgent* se desplace hacia una esquina con personas pero sin victimario en esa dirección.

4. Regular: Se considera así a la acción que permite que *DroneAgent* se desplace hacia una esquina vacía y sin victimario en esa dirección.

### 2.3.9 Axiomas de posibilidad

Estos axiomas se utilizan en cálculo situacional y definen cuándo se puede ejecutar una acción.

Se tienen dos tipos de axiomas:

1. identificarVictimario: si el dron se encuentra en una esquina donde hay un victimario, es posible llevar a cabo esta acción.

2. moverLocacion: si el dron quiere moverse en una dirección en la que hay una esquina adyacente y cuenta con la energía necesaria que insume este movimiento, es posible llevar a cabo esta acción.

### 2.3.10 Axiomas de estado sucesor

Estos axiomas se utilizan en cálculo situacional y permiten agregar nuevos conocimientos a partir de las acciones que ejecuta el dron.

Se tienen los siguientes axiomas:

1. Para mantener de situación en situación el valor de la variable “hayVictimario” en una esquina.

2. Para aumentar de situación en situación la variable “cantidadVictimarios” si se identificó un victimario en la última acción.

3. Para mantener de situación en situación la variable “cantidadVictimarios” si no se identificó un victimario en la última acción.

4. Para cambiar de posición si el dron se mueve.

5. Para mantener la posición si el dron no se mueve.

### 2.3.11 Prueba de meta - Estrategia de Cálculo Situacional

Se considera que se alcanzó el objetivo si se encuentra un victimario:

goalReached(S):-  
 cantidadVictimarios(N,S),  $N > 0$ .

### 2.4 Interfaz de usuario

Se presenta la interfaz inicial de usuario y se muestra en la Figura 6.

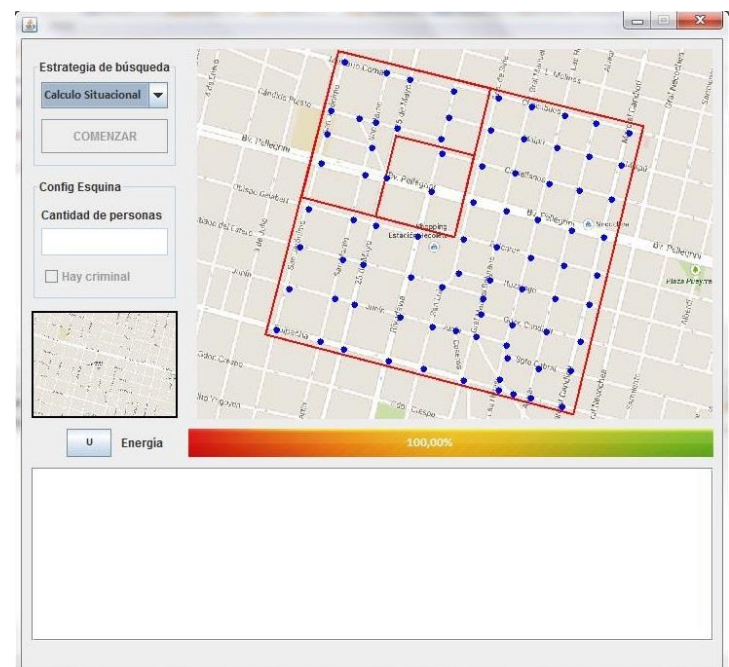


Figura 6 - Interfaz de usuario

El usuario puede elegir la estrategia de búsqueda que quiera utilizar: cálculo situacional, A\*, profundidad o anchura.



Cada esquina es configurable, puede elegirse la cantidad de personas que estén en ella y determinar si habrá o no un victimario en esa esquina. También se exhibe una barra que representa la cantidad de energía disponible del dron.

Dentro del escenario hay un gran cuadrado de color rojo que indica los límites dentro de los cuales podrá moverse el dron. Para indicar el nivel de altitud “alto” se divide el cuadrado más grande en cuatro zonas. Luego para el nivel de altitud “medio” se divide cada zona en cuatro generando un total de dieciséis zonas. Y por último, el nivel de altitud “bajo”.

A continuación, en la Figura 7 se muestra un ejemplo de un escenario inicializado.

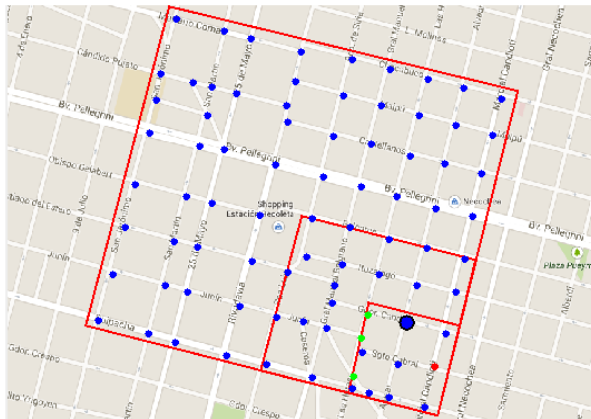


Figura 7 - Ejemplo de un escenario inicializado.

Referencias del escenario:

*La posición inicial del dron:* se representa con un círculo de mayor tamaño respecto al resto de las esquinas.

*Esquinas sin señal:* se representan con puntos de color azul.

*Esquinas con señal:* se representan con puntos de color verde.

*Esquinas con señal y victimarios:* se representan con puntos de color rojo.

*Los límites del subcuadrante:* se representan con cuadrados de color rojo y son las zonas donde podrá moverse el dron.

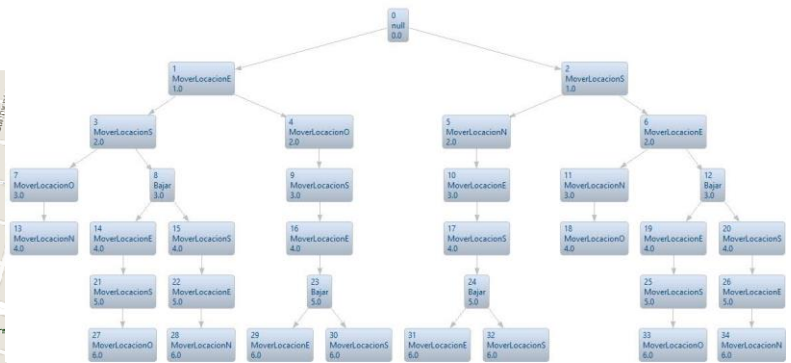
### 3. RESULTADOS

#### 3.1 Búsqueda

Logramos que con todas las estrategias de búsqueda utilizadas, estas son: profundidad, amplitud, costo uniforme y A\*, el dron llegue a su objetivo.

En la figura 8 puede observarse el árbol de búsqueda inicial generado por la estrategia de búsqueda en amplitud para una determinada situación. Puede verse cómo los operadores se ejecutan de izquierda a derecha en el orden en que son seteado en el agente:

*MoverLocacionN, MoverLocacionE, MoverLocacionS, MoverLocacionO, Bajar, Subir.* Vemos, además, la estructura especial del árbol dado que los nodos se expanden en orden horizontal nivel



a nivel.

Figura 8 – Árbol de búsqueda generado con una estrategia en amplitud.

Por otra parte, ese mismo escenario generará un árbol inicial como el de la figura 9 al utilizar una estrategia de búsqueda en profundidad. Vemos que es más bien un árbol que se expande verticalmente, siempre analizando primero el nodo más profundo que queda por expandir.

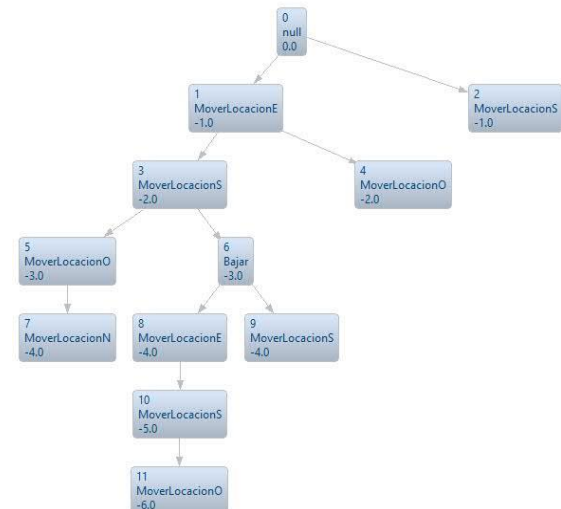


Figura 9 – Árbol de búsqueda generado con una estrategia en profundidad.

Con las estrategias de costo uniforme y de A\*, la búsqueda se restringe; en el caso de costo, sólo expandirá aquellos nodos que queden por expandir cuyo costo acumulado sea menor que el menor costo hallado hasta el momento. En el caso de A\*, además del costo considera un valor heurístico que estima el costo que falta para llegar al objetivo.

### 3.2 Cálculo Situacional

El escenario de prueba utilizado para la estrategia de cálculo situacional corresponde al de la Figura 7.

A continuación, se presentan las acciones ejecutadas por el dron en su proceso para encontrar al victimario:

1. MoverLocacionO.
2. MoverLocacionS.
3. MoverLocacionS.
4. MoverLocacionE.
5. MoverLocacionE.
6. IdentificarVictimario.

Siguiendo las acciones ejecutadas se puede apreciar cómo el dron responde a la evaluación de acciones propuesta.

Inicialmente *puede* ir hacia el Este, Sur y Oeste. Según el orden de las reglas de estado sucesor, debería ir al Este en primera instancia. Esto le permitiría llegar más rápido al objetivo, ya que si primero se mueve al Este y luego al Sur, estará en la misma esquina que el victimario.

Sin embargo, elige ir en un principio hacia el Oeste ya que en esa esquina hay *señal* y esta acción está considerada como una *buena* acción, y mejor que moverse hacia una esquina sin personas que está considerada una acción *regular*. Aquí podemos observar como las reglas de su conocimiento le indican que “es mejor” ir hacia una esquina con señal que sin señal (principalmente por el costo de movimiento, y porque se supone que estará más cerca de hallar el victimario).

En la situación siguiente elige ir al Sur por la misma razón, es decir, en esa dirección hay una esquina con personas; la otra opción es ir al Este pero en dicha esquina no hay personas.

Luego va hacia el Sur de nuevo; ¿por qué ejecuta esta acción si ir a una esquina sin señal es “peor” que ir a una que sí tiene, como la esquina al Norte de su posición actual? Pues la respuesta es sencilla; el dron tiene una regla que le indica que no puede volver a pasar por una esquina si ya ha sido *visitada*. Por este motivo, su único camino es hacia el Sur. Aquí debe elegir entre las próximas dos acciones que *puede* realizar: ir al Este e ir al Sur. Uno supondría que como la esquina hacia el Este no posee señal y la esquina hacia el Sur sí, el dron elegiría ir hacia el Sur. Pero lo que el dron decide en base a su conocimiento es ir hacia el Este. ¿Por qué? Una vez más, las reglas le indican que esa es la mejor acción. Como el dron ha percibido a través de la cámara que hacia el Este encontrará un victimario en alguna de las esquinas de esa dirección, decide aventurarse en ese camino. Es decir, moverse en la dirección en la que se detectó que hay un victimario es una acción *muy buena*, y mejor que ir al Sur hacia una esquina con señal que es una acción *buena*. Así es como primero se mueve al Este y luego, por el mismo motivo, al Este nuevamente. En esta esquina, percibe que en su posición actual hay un victimario. Luego, decide *identificar cuál de ellas es el victimario*, dado que esta acción es considerada como una acción *excelente*. Esta acción se ejecutará siempre que el dron sepa que está en una esquina en que hay un victimario y aún no haya alcanzado su objetivo.

En la implementación de este tipo de estrategia se obtuvieron mejores resultados que con búsqueda, ya que el proceso de unificación de predicados es más veloz que armar árboles sucesivos para tomar una decisión.

#### 4. DISCUSIÓN

Teniendo en cuenta las características del mapa en el nivel bajo, en que existen muchas conexiones entre las esquinas dentro de una determinada zona, encontramos que resulta más conveniente utilizar cálculo situacional en ese nivel, dado que el dron se maneja según ciertas reglas que le permiten inferir en qué lugar se encuentra (y en qué lugar no) el victimario. Esto permite llegar rápido a una respuesta; en contraposición, búsqueda genera un árbol de búsqueda muy grande ya que explora muchas posibilidades al explorar los diferentes caminos y combinaciones de movimientos entre las esquinas. En otras palabras, ambos permiten encontrar la solución a un mismo problema, pero cálculo situacional permite hacerlo en forma más eficiente (computacionalmente hablando). Por otro lado, búsqueda resulta muy bueno para utilizar en los niveles medio y alto, ya que en estos niveles el dron debe desplazarse entre las diferentes zonas del mapa, explorando y percibiendo en qué zonas existen señales y cuál es su intensidad. Es decir, en aquellos niveles en que resulta necesario y beneficioso explorar, búsqueda resulta más útil que cálculo situacional.

#### 5. CONCLUSIÓN

El uso de IDEMIA y FAIA, tanto para el modelado como para la implementación, ha facilitado el desarrollo permitiendo al equipo de trabajo centrarse en los aspectos relevantes del diseño del comportamiento del agente.

En tanto, el dron logra alcanzar el objetivo para los escenarios propuestos con las técnicas de búsqueda y cálculo situacional. En cuanto a las implementaciones de ambas, vemos que la estrategia de búsqueda forma un árbol en cada situación y cada nodo se fija cual es la mejor opción para ir a cualquier nodo adyacente. En cambio en cálculo situacional, el agente percibe, decide y acciona según lo que crea más conveniente, almacenando la información de

la situación actual en una base de conocimiento.

En otras palabras, con búsqueda con cada paso que el dron da, debe recalcular de cero cuál es el mejor camino a la solución; con cálculo situacional, en cambio, el dron puede ir almacenando cierta información que le resulta útil para llegar más rápido al objetivo (es decir, puede ir basándose en conocimiento adquirido).

Otra diferencia importante que encontramos, es que en búsqueda el dron explora; ejecuta todos los operadores que puede y decide cual es la secuencia de acciones que lo lleva a su objetivo, y ejecuta la primera de ellas; en el caso del cálculo situacional, el dron elige la próxima acción a ejecutar basado en la situación actual, los conocimientos previos, sus reglas de operación y un ranking de las posibles acciones a ejecutar. Teniendo en cuenta esto, y el tiempo de implementación de la solución en cada uno de los casos, decimos que nos resultó más conveniente usar cálculo situacional.

#### 6. REFERENCIAS

- [1] Russell, S. & Norvig, P. (2010). Artificial Intelligence: A Modern Approach. Prentice-Hall, 3rd edition.
- [2] Nilsson, N. J. (1998) Artificial Intelligence: A New Synthesis. Morgan Kaufmann ed.
- [3] Roa, J., Gutiérrez, M. y Stegmayer, G. (2009) FAIA: Framework para la enseñanza de agentes en IA. IE Comunicaciones: Revista Iberoamericana de informática educativa. No 7/8, ISSN 1699-4574, Pp. 43-56.
- [4] Santana, W., Roa, J., Gutiérrez, M., Stegmayer, G. (2012). IDEM-IA: Un entorno de desarrollo integrado para el modelado de agentes inteligentes. IE Comunicaciones: Revista Iberoamericana de informática educativa. Nro 13, ISSN 1699-4574, Pp. 43-56.

#### DATOS DE CONTACTO

Gómez, José Ignacio – [gomez.ignacio31@gmail.com](mailto:gomez.ignacio31@gmail.com)  
Hechim, Pablo Nicolás – [nicolashechim@gmail.com](mailto:nicolashechim@gmail.com)  
Paciuk, Sebastián Federico – [sebapaciuk@gmail.com](mailto:sebapaciuk@gmail.com)  
Facultad Regional Santa Fe - Universidad Tecnológica Nacional  
Lavalse 610 (3000) Santa Fe, Santa Fe, Argentina