



# UNIVERSIDAD DE GRANADA

## Práctica 2

Desarrollo Basado en Agentes  
Grado en Ingeniería Informática  
Curso 2020/2021

Jose Armando Albarado Mamani  
Miguel Ángel Molina Jordán  
Francisco José Molina Sánchez  
Pedro Serrano Pérez

Escuela Técnica Superior de  
Ingenierías Informática y de Telecomunicación

# Índice

<b>1. Sensores utilizados</b>	<b>1</b>
<b>2. Diagramas comunes a todos los agentes</b>	<b>1</b>
2.1. Diagrama de clases . . . . .	1
2.2. Diagrama de secuencia . . . . .	2
2.3. Diagrama de actividad . . . . .	2
<b>3. Diseño del agente básico</b>	<b>3</b>
3.1. Descripción del agente básico . . . . .	3
3.2. Diagrama de actividad de la toma de decisión agente básico . . . . .	4
<b>4. Diseño del agente con memoria</b>	<b>4</b>
4.1. Descripción del agente con memoria . . . . .	4
4.2. Diagrama de actividad de la toma de decisión agente con memoria . . . . .	5
<b>5. Diseño del agente con doble guiado</b>	<b>6</b>
5.1. Descripción del agente con doble guiado . . . . .	6
5.2. Diagrama de actividad de la toma de decisión agente con guiado de doble mano	7

## 1. Sensores utilizados

- **Alive:** es imprescindible para detectar si el drone ha muerto. Si es el caso, tenemos que realizar un *logout* inmediatamente y salir del servidor.
- **Compass:** este sensor es muy importante, ya que nos indica nuestro ángulo con respecto al norte y con esto podemos saber hacia donde está mirando nuestro drone.
- **Angular:** nos indica el ángulo que forman el drone y el objetivo con respecto al vector norte. En conjunción con la medida que nos proporciona el sensor *Compass*, podemos orientarnos siempre hacia el objetivo.
- **Distance:** nos indica la distancia a la que se encuentra el objetivo respecto del drone.
- **Energy:** es el sensor que nos indica la energía de la que dispone el drone en cada momento. Podríamos prescindir de él, ya que el drone internamente puede calcular la energía que utiliza (y por tanto, la energía restante en su batería) pero hemos considerado que tenerlo es rentable, ya leer este sensor que gasta muy poca energía y a cambio significa considerablemente los cálculos que el drone debe hacer.
- **GPS:** este sensor nos indica las coordenadas del drone. Con esta información podemos almacenar en memoria las casillas (coordenadas  $x,y$ ) por las que ya hemos pasado. Además, con la coordenada  $z$  podemos saber la altura del drone en cada momento.
- **Visual:** nos informa sobre la altura del terreno que hay alrededor del drone. Con estos datos podemos ver si los obstáculos que tenemos alrededor pueden ser sobrevolados o han de ser rodeados.
- **Thermal:** este sensor es necesario para decidir si vamos a rodear un objeto por la izquierda o por la derecha, ya que iremos por la dirección en la que el valor de la lectura de este sensor sea menor.

## 2. Diagramas comunes a todos los agentes

Estos diagramas son comunes en los tres agentes.

### 2.1. Diagrama de clases

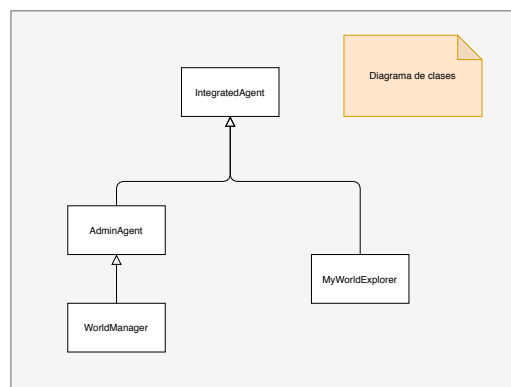


Figura 1: Diagrama de clases.

- **MyWorlExplorer:** Es nuestro agente que interactuará con el mundo.

- **WorldManager:** Es el agente que se encuentra en el mundo con el que establecemos comunicación.

## 2.2. Diagrama de secuencia

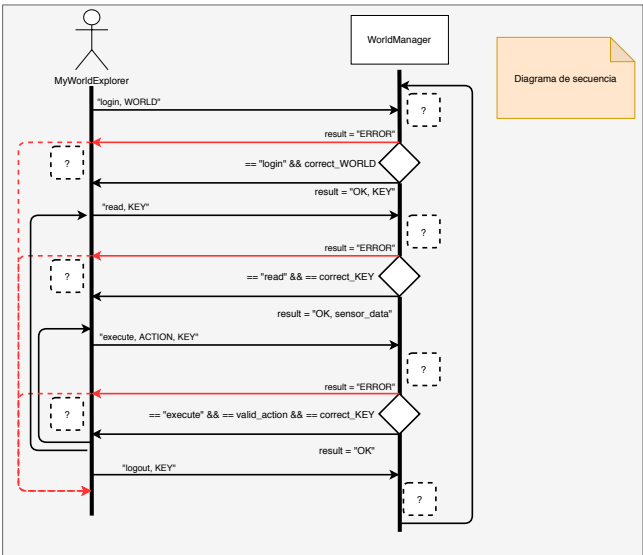


Figura 2: Diagrama de secuencia.

En la figura 2 podemos ver diagrama de secuencias, en este diagrama se expresa el flujo de comunicación entre nuestro agente **MyWorldExplorer** con el agente del mundo **WorldManager**. Se inicia la conversación con el agente del mundo mediante un *login*, una vez realizado el *login* recibimos una *key*, con esta *key* nuestro agente podrá realizar todas las acciones que considere oportunas para rescatar al objetivo. Para finalizar la comunicación se envía un *logout* con la *key* correspondiente.

### 2.3. Diagrama de actividad

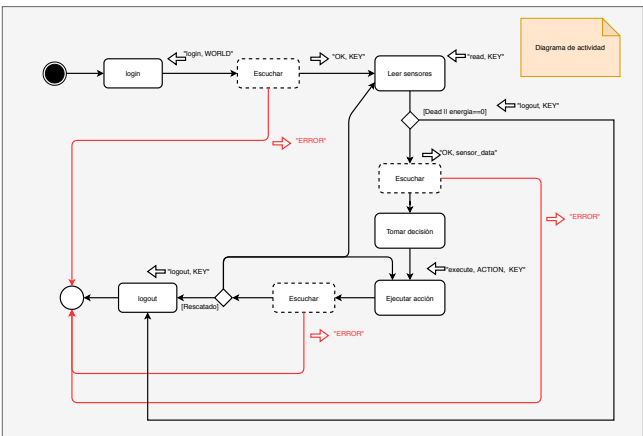


Figura 3: Diagrama de actividad común a los tres agentes.

Este diagrama de actividad representa el flujo de las actividades que se realizan desde que el agente manda la acción *login* hasta que termina la ejecución al mandar la acción *logout*. En el diagrama se puede observar las acciones comunes a los agentes:

- Realizar login.
- Obtener los datos de los sensores.
- Comprobar si el agente ha muerto o no tiene energía. Si esto ocurre se envía la acción *logout*.
- Decidir y ejecutar las acciones necesarias para rescatar al objetivo. Una vez se rescata el objetivo se envía la acción *logout*.

La actividad “Tomar decisión” dependerá de cada agente. Esta actividad va a definir cómo va a actuar al agente en función de la información que tiene del mundo.

### 3. Diseño del agente básico

#### 3.1. Descripción del agente básico

Lo primero que hace nuestro agente es comprobar que la distancia con el objetivo no es cero, ya que si la distancia fuese cero, esto significaría que estamos justo encima del objetivo, por lo que lo único que deberíamos hacer es bajar. No es necesario que comprobemos la energía en este punto, ya que para recargar también tendríamos que bajar. Por tanto, cuando estamos justo encima del objetivo descenderemos directamente a recoger a Ludwig.

Tras esto, comprobamos el nivel de energía del drone, si es muy bajo procederemos a recargar y si no seguiremos con las comprobaciones. Para recargar, simplemente bajamos hasta el suelo y recargamos el drone. El nivel que hemos considerado crítico para la recarga de la energía es un poco superior a 255 (concretamente 270), ya que es la cantidad de energía que se necesita leer los sensores y luego bajar desde el punto más alto de cualquier mapa ( $z=255$ ) hasta el punto más bajo ( $z=0$ ). Dicha bajada se realiza directamente, es decir, sin leer los sensores hasta que se llega a la altura deseada, ya que podemos calcular el número de veces que tendremos que bajar con solo una lectura del sensor de *visual* y *gps*, restando la coordenada  $z$  que nos indica el *gps* a la altura del terreno que hay justo debajo de nuestro drone (esto nos lo indica el sensor *visual*).

Para continuar, comprobamos el *visual* y el *angular* para ver si hay un obstáculo en la casilla en la que queremos ir. Si hay un obstáculo, comprobaremos si podemos sobrevolarlo para poder seguir avanzando o si por el contrario el sobrepasa la altura máxima y tenemos que rodearlo.

Finalmente, si no hay obstáculo que nos impida el movimiento, orientaremos el drone hacia la dirección en la que el valor del *angular* y el valor del *compass* sean más parecidos y avanzaremos.

De este agente se van a conservar algunas funcionalidades en los agentes más avanzados:

- comprobar la distancia con respecto al objetivo y rescatarlo si esta es 0.
- comprobar la energía restante del drone y recargar si esta es inferior al umbral.

### 3.2. Diagrama de actividad de la toma de decisión agente básico

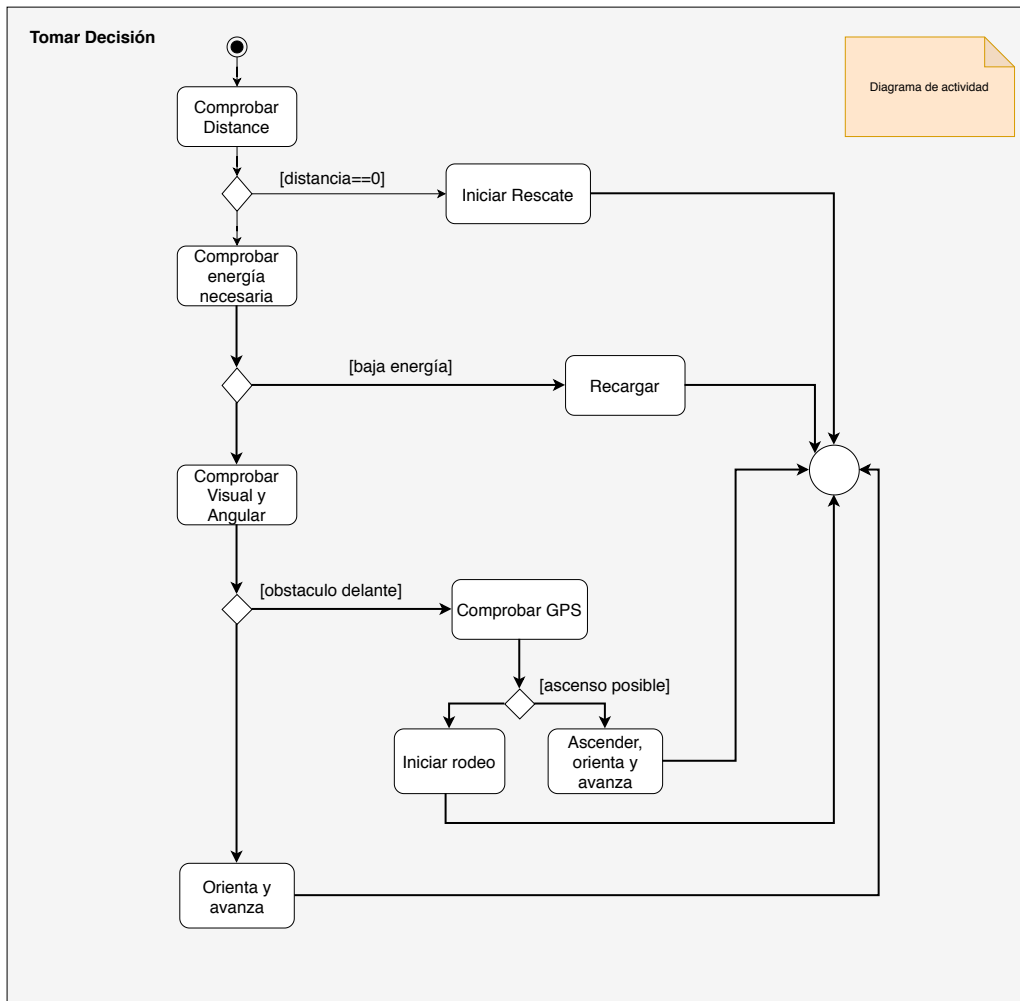


Figura 4: Diagrama de actividad de la toma de decisión del agente básico.

## 4. Diseño del agente con memoria

### 4.1. Descripción del agente con memoria

Este agente usará memoria para evitar pasar por casillas que ya ha recorrido. Para modelar la memoria utilizamos una matriz cuyo ancho y alto coinciden con el ancho y el alto del mapa, datos que se nos proporcionan al hacer el *login*. Se inicializan todas las posiciones de la memoria a 0 al empezar, debido a que no se ha recorrido ninguna casilla.

Lo primero que se realiza es guardar la posición actual en la que se encuentra el drone. Una vez se haya guardado la posición actual del drone se procede a comprobar si ha alcanzado el objetivo como lo hace el agente básico. De igual forma se realiza la comprobación de la energía.

Entonces procede a comprobar el sensor *visual* y *angular* para seleccionar la casilla deseada, es decir, la posible posición a la que el drone se desplazaría en el siguiente paso. Pero antes de confirmar que la casilla deseada será el siguiente paso del drone, comprueba en su memoria si esa casilla ya ha sido visitada por el drone:

- Si no ha visitado la casilla, el drone actúa como el agente básico; se orienta y avanza.

- Si la casilla no ha sido visitada por el drone se comprueba si hay un obstáculo delante:
  - En caso de que no haya obstáculo, el drone se orienta y avanza hacia el objetivo.
  - En caso de que haya obstáculo se debe comprobar si este es alcanzable:
    - Si es alcanzable el drone actúa como el agente básico, es decir, se orienta, asciende y avanza a la casilla.
    - Si no es alcanzable, se debe encontrar la casilla por la cual empezar a rodear. Esta casilla no debe haber sido visitada por el drone, por lo que se busca la siguiente casilla no visitada por la cual comenzar a rodear.

Finalmente, cuando la distancia al objetivo es 0, al igual que el agente básico, el drone desciende para rescatar al objetivo.

## 4.2. Diagrama de actividad de la toma de decisión agente con memoria

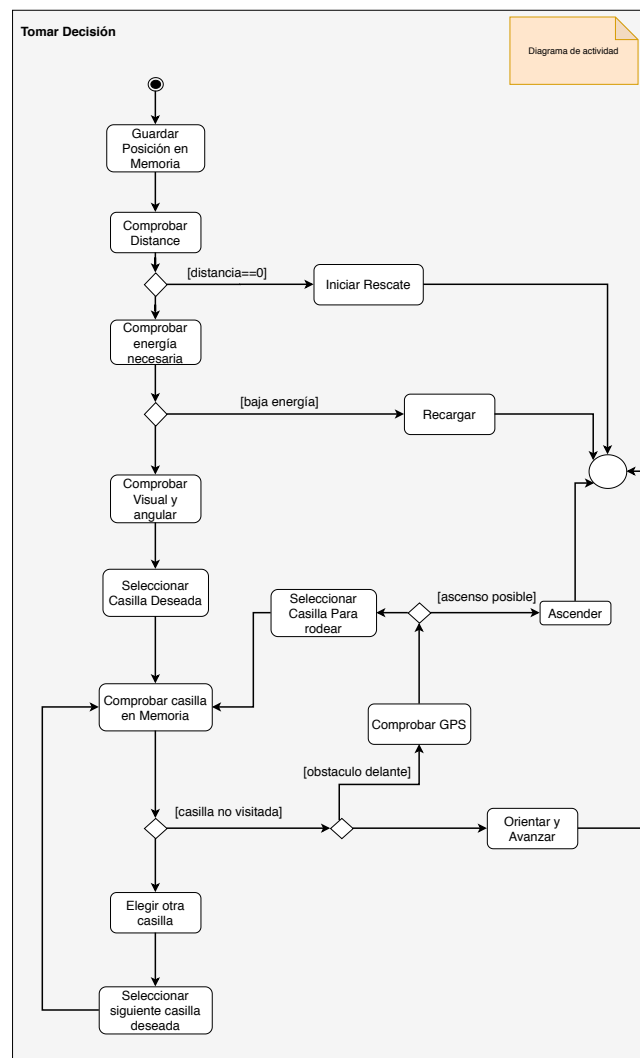


Figura 5: Diagrama de actividad del agente con memoria.

## 5. Diseño del agente con doble guiado

### 5.1. Descripción del agente con doble guiado

Este agente, además de utilizar memoria como el anterior, hará uso de dos tipos de guiado: guiado básico utilizando el sensor *angular* (como los agentes anteriores) y el guiado de doble mano para rodear un obstáculo que no pueda sobrevolar (es decir, un obstáculo inalcanzable). El agente deberá utilizar uno de los dos guiados en función de lo que más le convenga según los valores que lea de los sensores y de su memoria. En este agente también se va a hacer uso de una matriz que actúe como memoria, para evitar volver por zonas que ya ha recorrido y así evitar posibles bucles.

Lo primero que hace el drone es marcar su posición en su memoria. Luego comprueba si ya está sobre el objetivo y el nivel de energía que tiene, por si fuera necesario recargar. Este comportamiento es el mismo que el del agente anterior.

Realizados los pasos anteriores el agente debe encontrar la mejor casilla a la que ir (sea alcanzable o no), utilizando los sensores *Compass* y *Angular*. Cuando obtenemos esta casilla tenemos dos posibilidades:

- Si la casilla que ha encontrado es alcanzable y el drone no está retrocediendo o no hay ningún obstáculo alrededor del drone, este puede desplazarse a la casilla. Para evitar que el drone retroceda, comprobamos si las cuatro casillas más cercanas a la deseada se han visitado, proporcionando al drone una “visión en cono”, debido a que el drone solo puede avanzar a una de las 3 casillas que tiene frente a él (en el guiado *angular*).
- Si no se cumplen las restricciones anteriores debemos rodear. El agente puede rodear por la mano izquierda o por la derecha. Para esto se comparará el valor del sensor *Thermal* del extremo izquierdo de la “visión en cono” del drone con el del extremo derecho. Si este valor es menor o igual en el lado izquierdo, hará uso del guiado de mano derecha (gira a la izquierda, dejando el obstáculo a la derecha); si es menor en el lado derecho, usará el guiado de mano izquierda (gira a la derecha, dejando el obstáculo a la izquierda). Una vez ha empezado a rodear utilizando uno de los dos tipos de rodeos, el drone podrá cambiar al otro tipo de rodeo si el valor del *Thermal* actual empeora un número  $k$  de veces seguidas con respecto al valor del *Thermal* almacenado al iniciar el rodeo. En este caso se cambiará el tipo de rodeo y el drone se comprometerá a seguirlo; es decir, solo podrá cambiarlo una vez, ya que de esta forma el drone no podrá cambiar de mano continuamente, evitando realizar bucles.

Mientras el agente está rodeando un obstáculo es posible que el guiado *angular* sugiera volver a una casilla cercana a alguna visitada anteriormente, algo que podemos evitar, como hemos mencionado antes, haciendo uso de una “visión en cono”. En este caso la casilla deseada no es un obstáculo inalcanzable, por lo que debemos encontrar la siguiente casilla no alcanzable más cercana comenzando a buscar por la derecha o por la izquierda en función de la mano que se está siguiendo en el rodeo.

Finalmente, cuando la distancia entre el drone y el objetivo sea 0, el drone desciende para rescatar al objetivo, al igual sucede con los agentes anteriormente mencionados.



## 5.2. Diagrama de actividad de la toma de decisión agente con guiado de doble mano

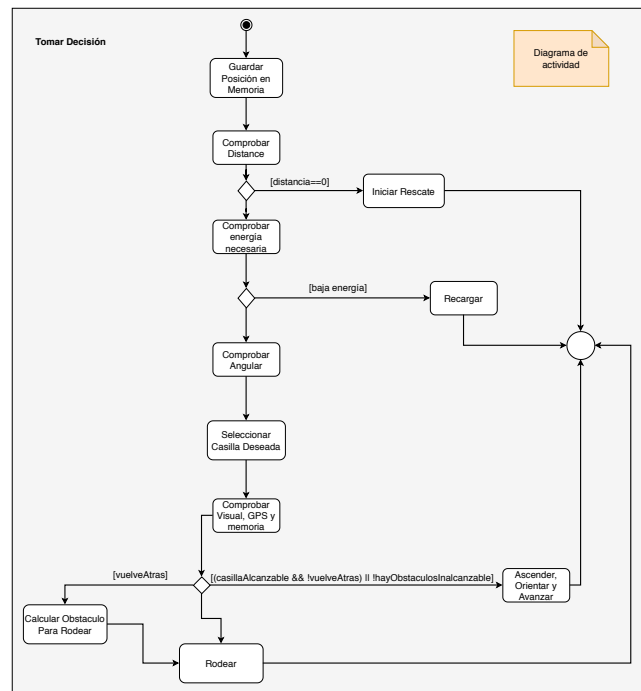


Figura 6: Diagrama de actividad de la toma de decisión del agente con doble guiado.

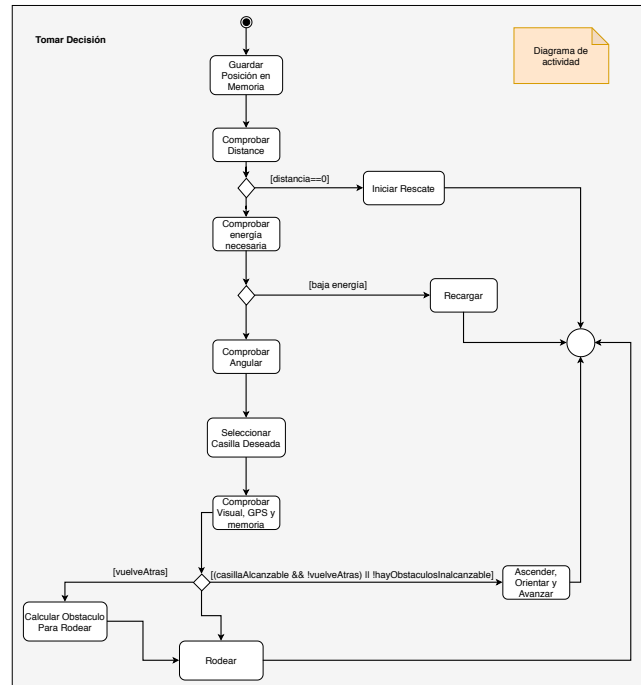


Figura 7: Diagrama de actividad de la toma de decisión del agente con doble guiado.