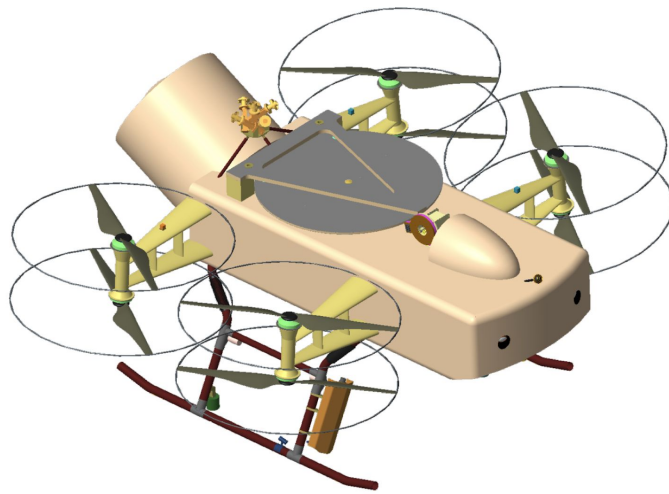


# PRÁCTICA 2

---

## Misión DragonFly



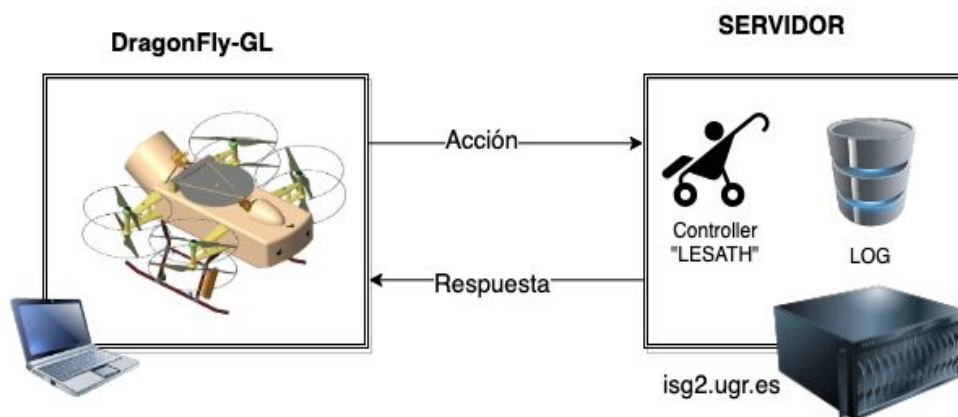
### **Grupo L**

Ana Rodríguez  
Juan Francisco Díaz  
Alicia Rodríguez  
Valentine Seguinou  
Alberto Rodríguez

El objetivo de la práctica 2 era el diseño e implementación de un drone autónomo que se desplaza por un mundo virtual en tres dimensiones.

Para alcanzar la solución al problema propuesto en esta práctica, hemos desarrollado un agente (**DragonFly-GL**) de la plataforma **Magentix2** que se comunica con otro agente controlador (**Lesath**), alojado en un servidor y encargado de proporcionarnos y controlar el entorno.

A través de nuestro agente percibimos la información del entorno, proporcionada por el controlador, mediante una serie de sensores.



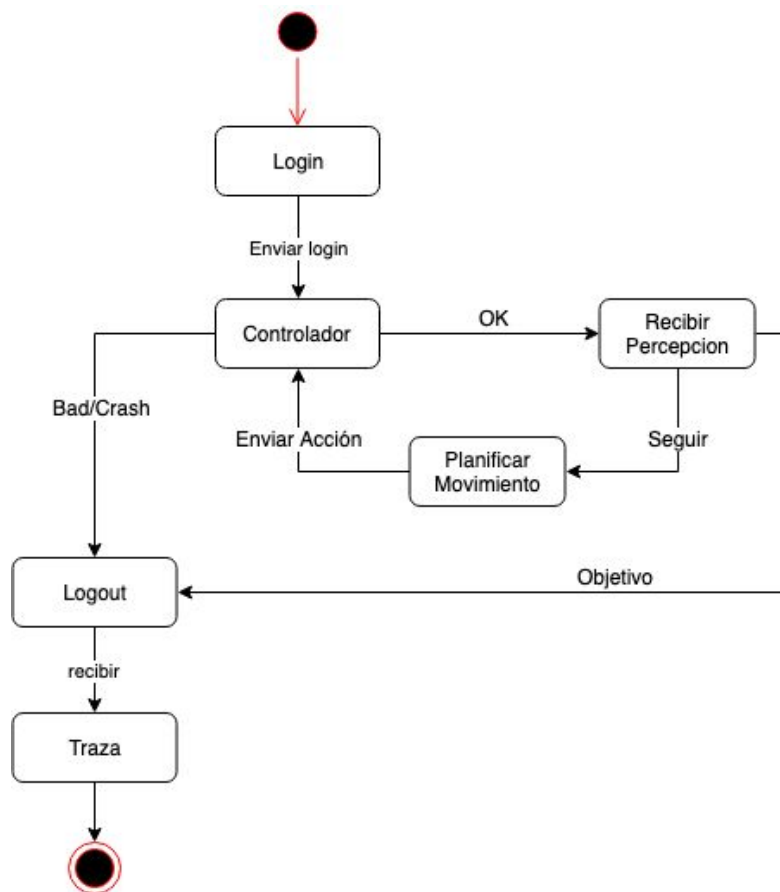
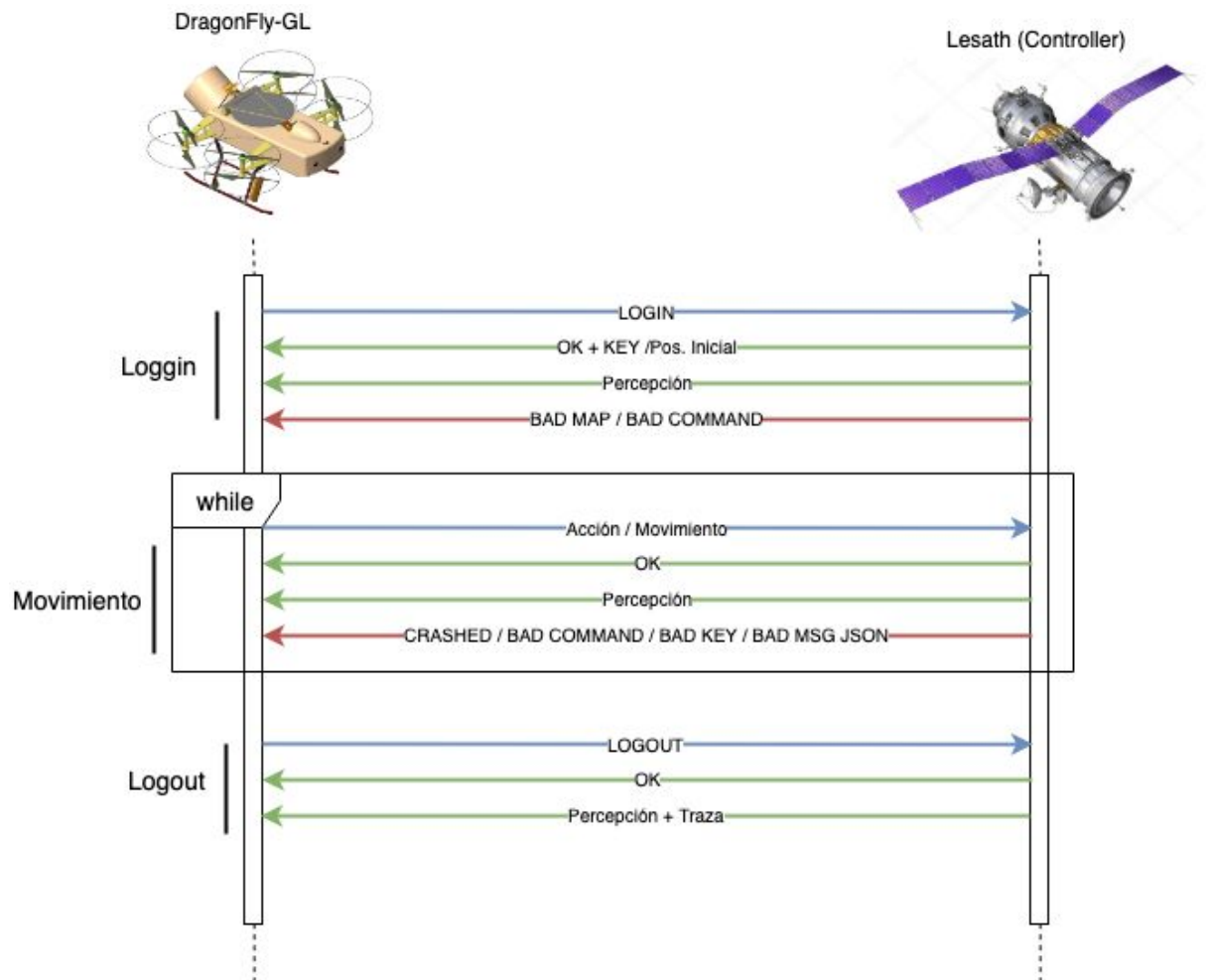
Para el modelado hemos diseñado un único agente capaz de procesar y tratar la comunicación con el agente controlador.

Inicialmente nuestro agente envía una petición al controlador para loguearse, a lo que este responde OK en caso de ser correcta la acción y realiza el login. Tras esto informa de las dimensiones del mundo en el que se ha logueado y de la altura mínima y máxima a las que el drone puede volar. En este mensaje, a su vez, el controlador manda una **key** para mantener la comunicación durante todo el ciclo de ejecución del agente.

```
{"result": "ok",  
  "in-reply-to": "login",  
  "key": "xxxxxxxx",  
  "dimx": 51, "dimy": 51,  
  "min": 5, "max": 180}
```

Una vez ha comenzado el diálogo entre nuestro agente y el controlador, éste último manda la percepción del entorno según los sensores que tengamos activado. En nuestro caso hacemos uso de **radar**, **elevation**, **fuel** y **gonio**.

La secuencia de mensajes en la comunicación sería la siguiente:



En estos diagramas podemos observar un modelo basado en un único agente, la carga de trabajo es mayor pero nos ahorramos el problema de sincronización en el caso de haber tenido más agentes.

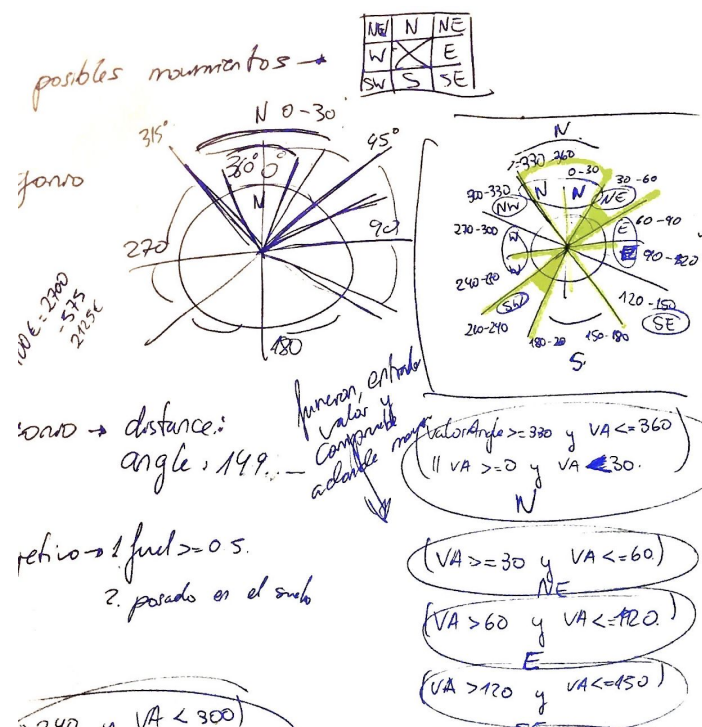
En el diagrama de secuencia podemos observar la comunicación entre nuestro agente DragonFly-GL y el controlador Lesath para las acciones de login, realizar un movimiento y logout.

Para el diagrama de actividad se describen las acciones del agente para llegar a cada estado.

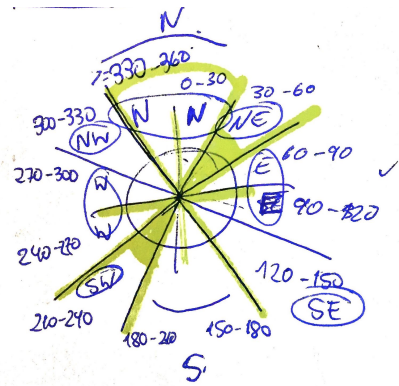
## HEURÍSTICA PRINCIPAL

Para la heurística principal hemos decidido hacer uso del sensor "Goniómetro" que nos indicará, a través del agente controlador, el ángulo hacia donde se encuentra el objetivo. Haciendo uso de esta medida, junto con la distancia, decidimos el movimiento que debe realizar nuestro agente.

Para el desarrollo partimos de una circunferencia completa de  $360^\circ$  (circunferencia goniométrica) en cuatro cuadrantes, haciendo referencia el inicio en  $0/360^\circ$  al Norte,  $90^\circ$  al Este,  $180^\circ$  al Sur y  $270^\circ$  al Oeste. A su vez, hemos dividido estos 4 cuadrantes en 3, haciendo divisiones de  $30$  en  $30^\circ$  para decidir los movimientos intermedios (NE, SE, SW y NW). Esto se hizo en una de las primeras reuniones y se puede ver documentado el boceto inicial en la siguiente imagen:



Esta heurística nos ha permitido resolver los 3 primeros mapas. Tras la motivación de haber resuelto los primeros mapas, decidimos pensar la estrategia de repostaje del drone que nos permitió resolver los mapas 4-5. Como mejora, para resolver los siguientes mapas, realizamos las comprobaciones necesarias para que el drone no chocará con las montañas del mapa. Decidimos activar entonces los sensores **elevation** y **radar** para obtener las alturas relativas y absolutas y así, aparte del movimiento que podemos realizar en “horizontal”, calculamos si es posible el movimiento teniendo en cuenta la altura, tanto al subir como al bajar. Tan importante es este movimiento como los anteriores, ya que aparte de poder esquivar los obstáculos en altura, nos permite subir y bajar para poder repostar, como veremos ver a continuación. Para los mapas 8-9-10 necesitamos hacer que el agente tuviera memoria, lo que nos ha resultado finalmente algo más complicado de pensar como estrategia y que por errores técnicos no hemos sido capaces de conseguir.



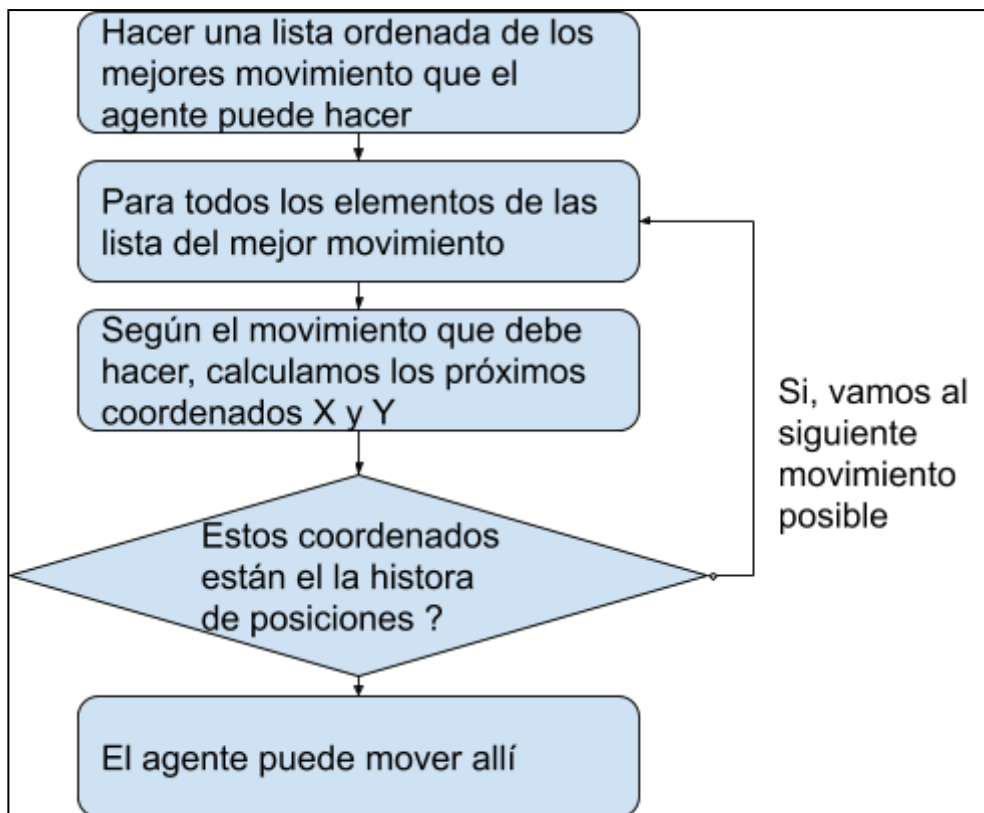
## ESTRATEGIA REFUEL

Para la estrategia de repostaje nuestro agente principal tiene que estar posado en el suelo. Hemos decidido que en el momento que tiene menos de un 10% de batería baje a repostar.

Una vez decide donde ir, el drone comprueba si tiene fuel suficiente, haciendo uso de una función que hemos diseñado para comprobar si necesita repostar. Esto lo hace calculando la diferencia entre la altura actual del dron y la altura de la posición donde quiere moverse y multiplicando el gasto por el número de pasos que tendría que dar. Si el resultado de esta operación está por debajo del umbral que hemos establecido, realizamos la llamada a esta función

## MEMORIA DEL AGENTE

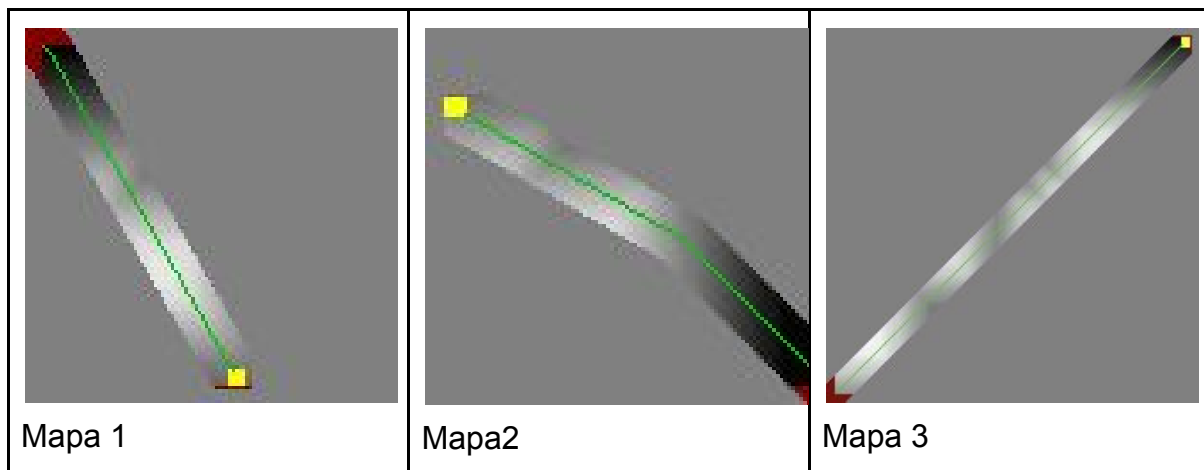
El agente necesita tener memoria para así no quedar bloqueado ya que por ejemplo el agente puede ir delante y detrás y repetir este indefinidamente si el mejor movimiento por el ángulo es este. Entonces debíamos implementar una función para que el agente recordará dónde fue y que pudiera elegir el mejor segundo movimiento a hacer. A continuación, mostramos la secuencia de acciones:

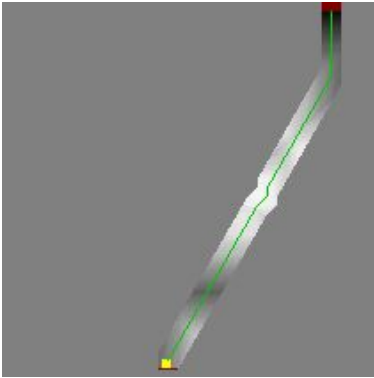
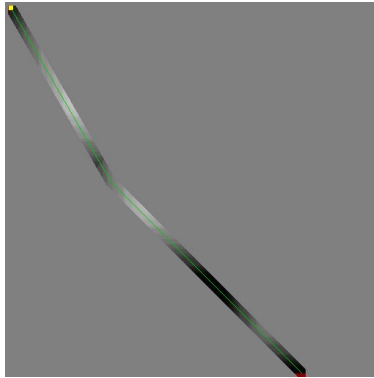
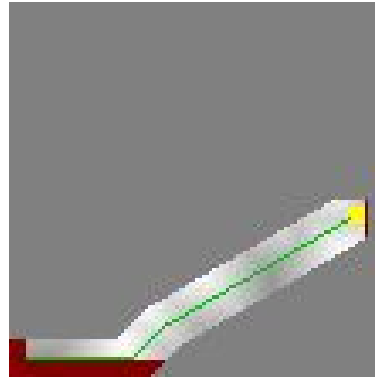
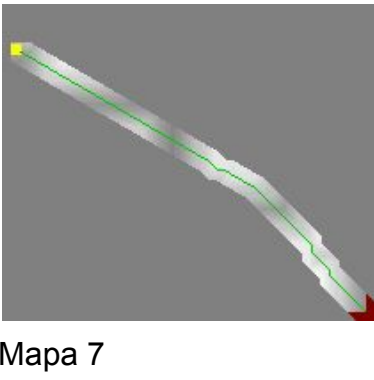
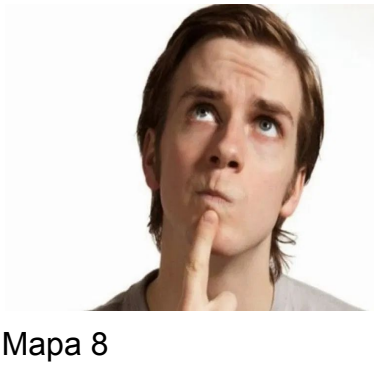



En nuestro código este está implementado pero como hemos comentado anteriormente, no hemos alcanzado la solución esperada debido a errores en el código que no hemos conseguido solventar.

## RESULTADOS

En las siguientes imágenes se puede ver el recorrido que ha realizado el drone en cada uno de los mapas, alcanzando el objetivo de cada uno de ellos.



		
Mapa 4	Mapa 5	Mapa 6
		
Mapa 7	Mapa 8	Mapa 9

Situación de nuestro agente en el mapa 10 -> [Enlace video](#).

## CONCLUSIÓN

En esta práctica hemos podido poner en práctica una metodología ágil de desarrollo como es SCRUM, enfrentándonos a una situación “más real” de lo que nos espera en un futuro cercano cuando salgamos al mundo laboral.

Sacamos como conclusión, que en un proyecto hay que hacer una muy buena planificación y una estimación lo más real posible en base a las necesidades del cliente (profesor) y al potencial o velocidad de trabajo del equipo, ya que como hemos podido comprobar de primera mano, alguna historia se nos ha ido de las manos ya que supusimos que el desarrollo de algún aspecto del problema iba a llevarnos menos tiempo y por lo tanto estaba mal planificada. Esto más que un problema lo ganamos en experiencia para la siguiente práctica poder realizar una planificación más ajustada a la realidad.