

IRIN

TRABAJO OBLIGATORIO 1

ASIER CULEBRAS PORRAS
RODRIGRO BRECHAR ALARCIA

GRUPO: 12

Índice.

Índice	2
Introducción.	3
Desarrollo.....	3
Herramientas necesarias.	4
Esquema del comportamiento.....	5
Descripción:.....	5
Escenario.....	6
Estadísticas.	7
Recorrido del robot.....	7
Sensor de la batería.....	8
Posibles mejoras.	9
Conclusión.....	9

Introducción.

En este primer trabajo obligatorio hemos querido simular el comportamiento de una persona que se va al campo a recoger champiñones. En el campo nos encontramos con árboles, arbustos, rocas, etc. por los cuales no se puede pasar. Además existen zonas montañosas y de alta pendiente por las cuales dicha persona no tendrá la capacidad de ir tan rápido y tendrá que reducir su velocidad de búsqueda. Como es sabido hay setas que no son comestibles y que el recolector es capaz de identificar, cuando se encuentre en este caso intentará alejarse ya que son malas para la salud.

Esta persona nunca ha recolectado en esta zona por tanto debe ir memorizando los lugares por donde puede pasar y por donde no. También guardará las zonas donde ha encontrado setas pues es habitual que vuelvan a crecer en el mismo lugar.

El recolector sabiendo donde empezó a recolectar setas, será capaz de, una vez conseguido un número determinado de setas, regresar a su posición inicial, dónde se encuentra la base de todos los recolectores.

Desarrollo.

El objetivo de este trabajo es implementar una arquitectura basada en el **comportamiento**, y extenderla a una **arquitectura híbrida**.

Para desarrollar dicho trabajo se utilizará el simulador **IRSIM** y los ejemplos de arquitecturas presentes en la página web de la asignatura.

En cuando a la simulación que hemos llevado a cabo, nos basamos en la arquitectura híbrida (comportamiento y conocimiento).

El robot tendrá un comportamiento reactivo, con el objetivo ir hacia las luces amarillas, evitando diversos obstáculos. Hasta que llegue a un punto determinado del mapa, una vez haya recolectado todas las luces amarilla, que con la ayuda de un **comportamiento híbrido** volverá al punto de partida de dicho robot mediante de resolución de mapas.

- Las luces amarillas son el **objetivo** del recolector, es decir, esta simulan los champiñones que tanto desea encontrar nuestro robot.
- Las baldosas negras instaladas en el suelo, van a simular **montañas** por las cuales, el robot va más lento, pues en la vida real, un recolector tarda más en recorrer un terreno en **pendiente** que uno normal.
- En cuanto a las luces rojas, van a simular setas venenosas, pes un buen recolector es capaz de reconocer a una distancia de 1 metro la calidad de la seta y si es posible recolectarla o al contrario, es venenosa. Por ese motivo el robot al encontrarse con una luz roja tiene que alejarse de ella todo lo rápido que pueda, por si le pudiese afectar el veneno. Como las setas venenosas pueden resultar muy perjudiciales, intentar evitarlas será el **comportamiento más importante**, o con mayor **prioridad** que el resto.

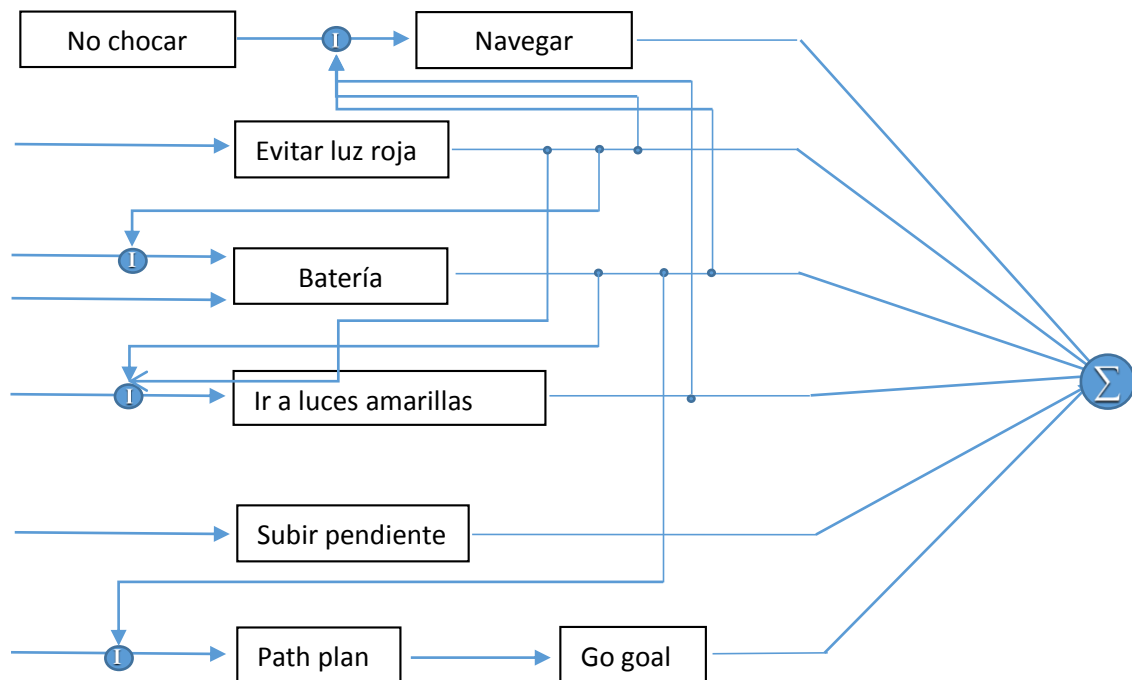
- Como parte final del recolector, supone llevar todas las setas recolectadas a un punto donde se guardaran y almacenarán de forma temporal. Este recorrido, ha de hacerse en el menor tiempo posible, y esto se consigue recorriendo el **camino más corto**, y con **menos obstáculos** entre donde se encuentra al recoger todas las setas, y su almacén.
- Y por último la batería azul de nuestro robot, va a simular las **energías** que le quedan al recolector tiene, pudiéndose comparar con el hambre o sed. Cuanta menos batería tiene, menos energía le quedan y esto supone que el robot tenga sed y hambre; teniendo que ir a reponer fuerzas a la luz azul.

Herramientas necesarias.

Hemos tenido que utilizar los 8 sensores de proximidad del robot para evitar que se chocara con las paredes del mapa, y los muros intermedios.

- **Sensores** de largo alcance de **luces amarillas** con el fin de poder ir a por ellas.
- **Sensores** del mismo alcance para poder evitar las **luces rojas** y no chocar con ellas.
- **Sensor de suelo** para detectar las zonas montañosas que posee el mapa.
- **Sensor de batería** para detectar cuando el robot tiene un nivel de batería crítico.
- **Sensor de luces azules**, de alcance todo el mapa, para que el robot sepa dónde está la fuente de carga de batería en todo momento.
- **Encoder**, así como la **brújula** para poder recorrer el camino hacia la posición de origen.
- Almacenar, mediante el comportamiento de la estructura basada en el conocimiento, donde está la zona de origen del robot para volver a ella, y donde se encuentra respecto a esa zona (zona origen) **en todo momento**.
- Variables auxiliares, para controlar la **prioridad** de la batería frente a los otros comportamientos. Funcionando a modo de inhibidor, cuando la condición de la batería salta, el comportamiento de ir a por luces, o de recorrer la ruta, es interrumpido por el de ir a por la luz azul. Pero sin interrumpir el comportamiento de evitar las luces rojas, por lo tanto tiene este último comportamiento mayor prioridad.
- **Almacenar el recorrido** que ha ido haciendo el robot, para saber por qué zonas puede pasar o no, y cuál es el camino más rápido a su punto final.
- Mantener una trayectoria **rectilínea** y a velocidad **constante** y **continua**, siempre y cuando no se vea interrumpida por ninguno de los 5 procesos: ir hacia luces amarillas y apagarlas, evitar luces rojas, frenarse en las baldosas negras, recorrer su camino de vuelta al origen o ir a cargar su batería a la luz azul.

Esquema del comportamiento.



Descripción:

- 1) No chocar: Este es el comportamiento principal del comportamiento del robot, es decir que es un comportamiento que a hacer siempre, aun teniendo que realizar otro tipo de comportamientos, este siempre lo va a tener en cuenta. Así evitamos que el robot se choque.
- 2) Navegar: Este comportamiento es complementario al anterior, pero con la peculiaridad que es inhibido por todos los demás. Hará que el robot nunca se encuentre parado si no detecta ninguna de las luces, es decir que navegará en línea recta hasta que; detecte una luz amarilla, le afecte una luz roja, o se quede sin batería.
- 3) Evitar luz roja: Como podemos observar evitar las luces rojas es un comportamiento fuerte comparado con los dos anteriores, puesto que el robot no podrá acercarse a una luz roja por mucho que le fuerce ir a por una luz amarilla o ir a cargar la batería. El robot nunca podrá estar muy próximo a una luz roja, pues este tendrá que alejarse olvidado la función que esté realizando en ese momento.
- 4) Batería: Esta acción es una de las principales, pues cuando se activa, es decir el sensor de la batería detecta que no tiene batería suficiente, va a dejar de: perseguir las luces amarillas, seguir su trayectoria rectilínea, o incluso dejará de guardar como ruta oportuna para llegar a su punto final (Path plan). Y va a tener que ir hacia la luz azul y quedarse dando vueltas muy cerca de esta luz hasta que el porcentaje de la batería llegue a 100%.
- 5) Ir a luces amarillas: Esta es el comportamiento principal del robot que como su propio

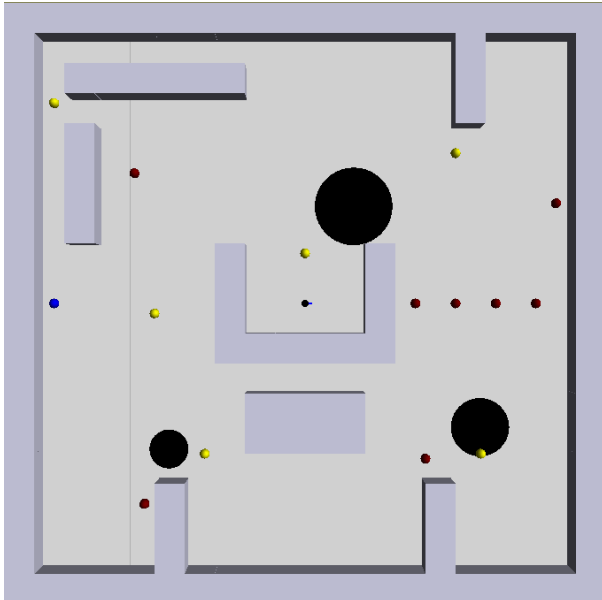
nombre indica, se trata de ir hacia la luces amarillas y una vez ha alcanzado su posición (se encuentra casi a distancia 0) las apaga. Pero como podemos ver, tanto el comportamiento de “Evitar luces rojas” como el de “Batería” son superiores a este, inhibiéndolo fuertemente y consiguiendo que deje de ir hacia las luces amarillas.

6) Subir pendiente: Es un comportamiento complementario, que simula que el robot se encuentra en una montaña y como es normal su velocidad disminuye, esto lo conseguimos mediante los sensores de suelo.

7) Path plan y Go goal: Es el encargado de guardar en su memoria interna todo los puntos del mapa recorridos así como una vez apagadas las 6 luces recorra el camino más corto hacia el punto final que coincide con el punto de inicio, excepto cuando el robot tiene que ir a recargar la batería, que ni almacenará, ni irá al destino establecido.

Escenario.

Este es el escenario inicial de la simulación. El robot empieza en el centro del mapa rodeado de tres paredes laterales y una inferior, haciendo que este tenga que salir por la zona norte, forzado por que se encuentra con una luz amarilla.



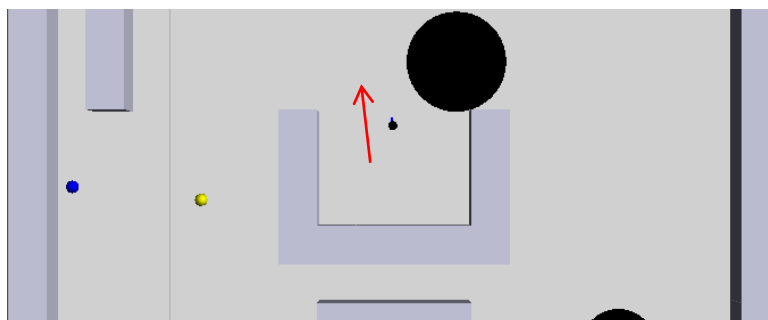
Los círculos en el suelo, también llamados baldosas, son los que utilizamos para simular las montañas, o como mencionábamos anteriormente, ralentizan la velocidad del robot.

La fuente de batería, o luz azul, la encontramos a la izquierda del todo, para hacer que el robot tenga que recorrer grandes distancias para ir a por ella.

Las luces amarillas se encuentran distribuidas por el mapa, para que tenga que recorrer todo el mapa, y así almacene la mayor información posible para después poder regresar al punto de origen.

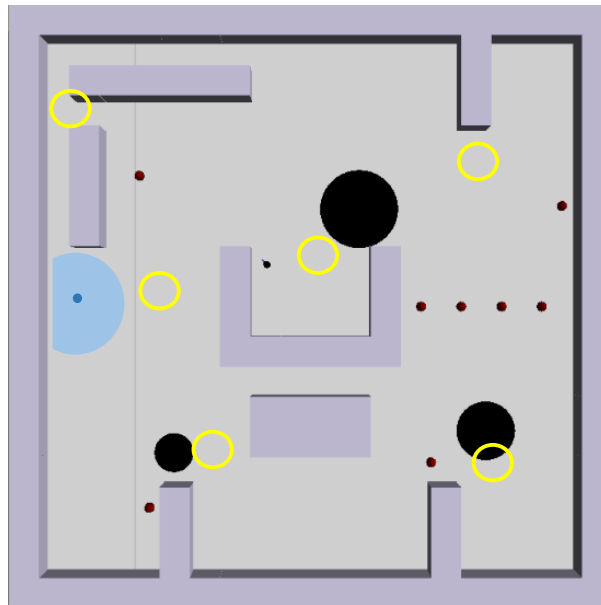
Por último, destacar que imponemos una barrera de luces rojas en la zona este del mapa, para forzar que recoja las luces amarillas pero sin acercarse a las rojas. En esta zona y tras muchos intentos de traspasar la muralla de luces rojas consigue encontrar su punto débil y atravesarla.

A continuación, mostramos como el robot apaga primera luz que se encuentra y es encaminado



hacia el norte (dirección de la flecha roja).

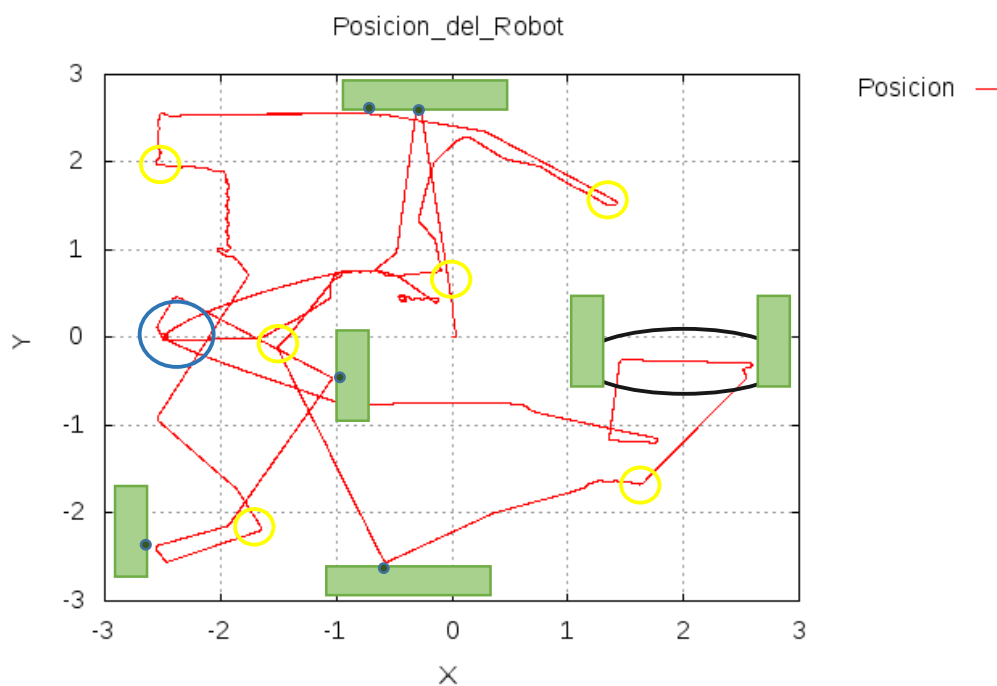
Y por último, como termina el escenario de la simulación: Con todas las luces amarillas apagadas y el robot en el punto de inicio, es decir en el punto correcto donde debe acabar.



Mostrando con círculos amarillos donde se encontraban las luces antes de apagarlas, y con una zona sombreada de cuanta distancia se trata la zona de carga.

Estadísticas.

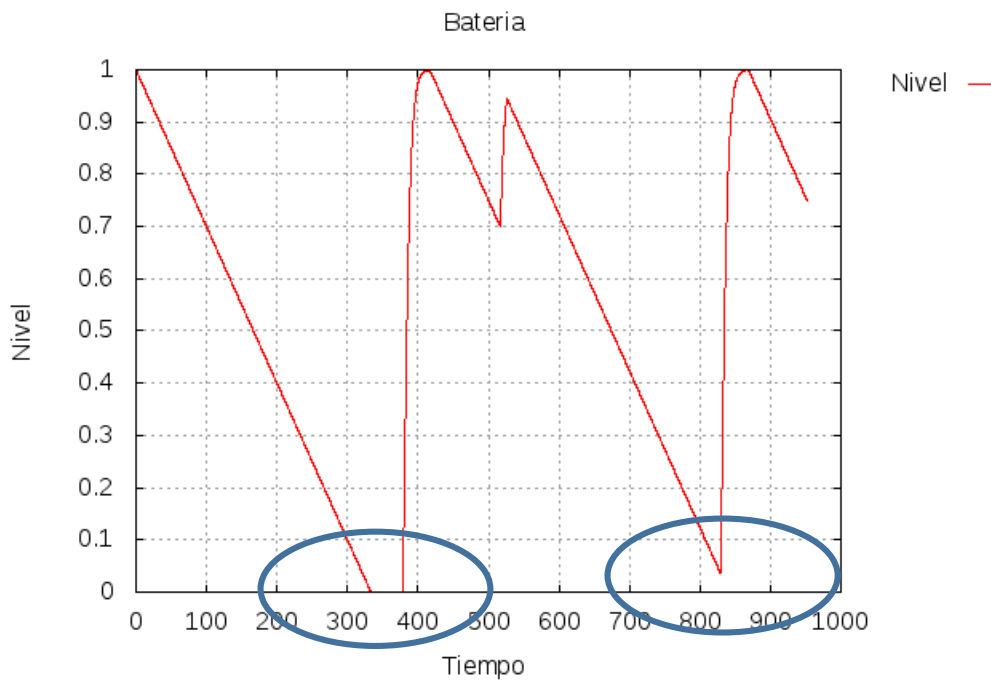
Recorrido del robot.



Como podemos observar, en el comportamiento del robot, vemos que una vez apagadas las luces amarillas cambia su trayectoria, que la barrera de luces rojas (marcada en color negro) no es capaz

de atravesarla a la primera, y la recorrer de lado a lado hasta encontrar su punto débil. También podemos observar como evita chocar con las paredes, marcado con zona verdes los muros y zonas verde más oscuro donde hubiese chocado (solo unos ejemplos no todos). Por último, podemos apreciar como el robot recorre la zona de batería durante más tiempo, pues las zonas cercanas a la batería (marcadas en azul) tienen las líneas rojas, que indican el recorrido del robot, mucho más marcadas, significando eso que ha pasado por esa misma trayectoria más de una vez.

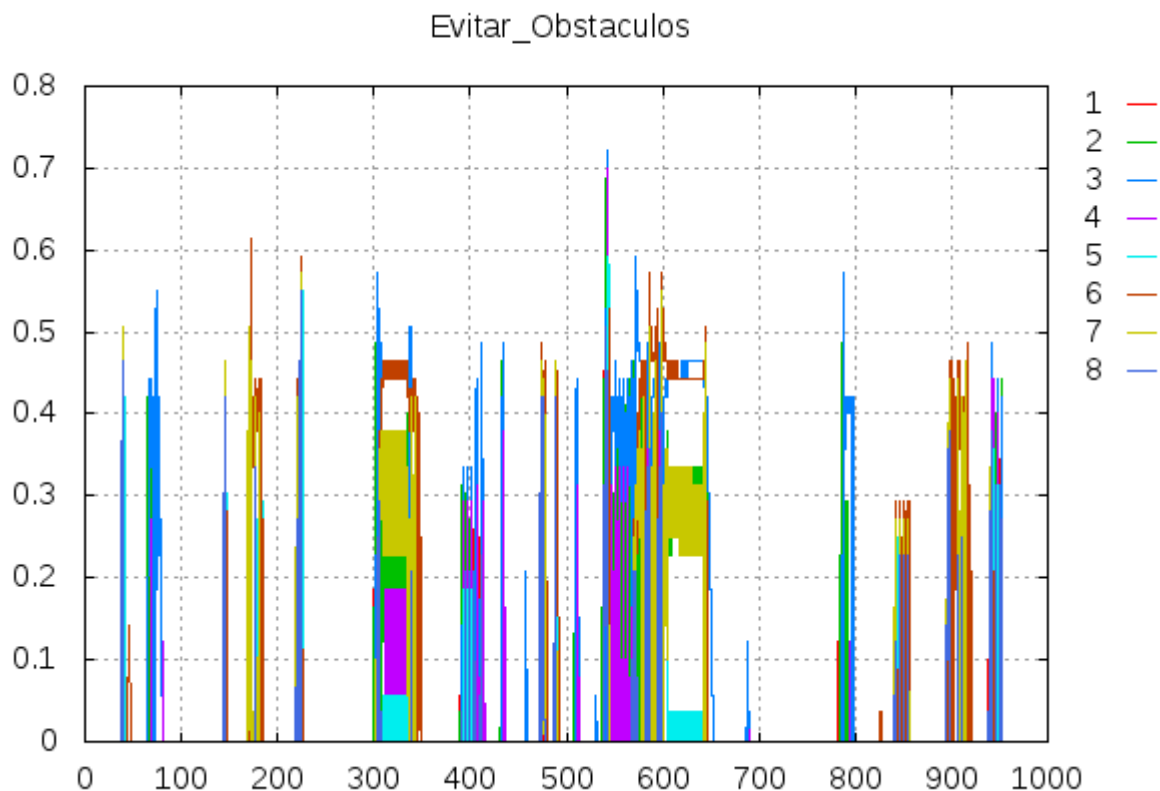
Sensor de la batería.



Como observamos en esta gráfica, podemos ver como la batería se descarga por debajo del umbral permitido dos veces, que es el que hace que se inhiban el resto de comportamientos como esta explicado en el esquema del comportamiento, y una vez se ha acercado a la batería lo suficiente empieza a cargarse hasta que llega al 100% de la carga y el robot continua con su proceso haciendo esto que se vuelva a descargar linealmente la batería.

Un punto que podemos destacar en esta grafica es que, a la altura del tiempo de 510, el robot pasa cerca de la luz azul, porque su recorrido ha surgido así, y este recarga la batería, sin que fuese necesario, y cuando se aleja de la zona de carga, continua su descarga lineal.

Sensores de proximidad



Como podemos observar en esta gráfica ninguno de los sensores de proximidad pasa de 0.8 lo que demuestra gráficamente que en ningún momento el robot llega a chocarse pues implicaría un valor de 1 o muy próximo en algún sensor.

Posibles mejoras.

En esta sección, presentamos una lista de posibles mejoras, o modificaciones que creemos interesantes desarrollar para mejorar el comportamiento del robot y hacer una simulación más parecida a la recolecta de setas, pero que por motivos diversos como puedes ser el tiempo, o la dificultad de llevarlo a cabo en el simulador, o mismamente nuestra limitación como programadores en este lenguaje C++, no hemos sido capaces de llevarlas a cabo, pero que nos hubiese encantado poder desarrollarlas.

- Una posible mejora sería guardar en el mapa las zonas montañosas de forma que cuando vuelva a casa evite esas zonas en los casos que exista una ruta alternativa más rápida.
- Conseguir que el robot almacene también donde se encuentran las luces rojas para no tener que acercarse a ellas.
- Una vez llegado al punto de donde el robot parte vuelva a las luces amarillas y las encienda de nuevo y una vez encendidas las 6 luces, vuelva al inicio y que borre todos los datos del mapa que había obtenido y vuelva a empezar.

- Que repita el comportamiento, de apagar las luces y volverlas a encender, con el fin de conseguir la ruta más rápida para apagar las luces desde el punto de origen.

Conclusión.

Como conclusión de este trabajo, podemos llegar a que fue complicado empezar con la simulación, y que al principio del desarrollo de la práctica, no éramos capaces de conseguir ningún resultado y casi ni de simular los propios ejemplos del programa, haciendo esto que llegáramos a pensar *“En que asignatura nos hemos metido”*, *“Que asignatura más inútil”*, *“vaya pérdida de tiempo”* pero pasadas las horas, frente al programa, y varias tutorías con el diseñador del simulador **Álvaro Gutiérrez Martín**, fuimos capaces de diseñar nuestra propia simulación, y pasar buenos ratos experimentando con el robot. Haciéndole pasar varias pruebas entretenidas y divertidas, con por ejemplo borrando las paredes del mapa, para que se volviera loco fuera del escenario o rodearle de obstáculos para ver si era capaz de superarlos.

Todo esto llevó a que la asignatura nos empezara a gustar, y de verdad encontrarle un uso práctico a la robótica así como a la simulación, incluso influenciando a uno de los compañeros del grupo a animarse y seguir persiguiendo su sueño que es la domótica y las aplicaciones domesticas inteligentes. Y como no, recomendar la asignatura a compañeros de la carrera puesto la satisfacción que conseguimos tras este proyecto.