

## Sensores numéricos

### Ejercicio 1) El pozo de las ánimas

El pozo de las ánimas es una formación geológica que se encuentra en Malargüe, Mendoza, y que consiste, como su nombre lo indica, en un gigantesco pozo. Como turistas curiosos que somos, queremos ir hasta el pozo a sacar una fotografía, pero hay que tener mucha precisión, muy lejos y la fotografía no saldrá bien, un paso de más y podemos caer al pozo y morir solo por sacar una selfie.

Así que en lugar de ir nosotros, decidimos programar un robot que vaya hasta el lugar y saque la foto (por supuesto, tampoco queremos que el robot se caiga al pozo), por lo que queremos programarlo. Las primitivas que este robot entiende son:

<b>Avanzar un metro al pozo</b> Hace que el robot avance hacia el pozo una distancia de un metro.	<b>Tomar fotografía</b> Hace que el robot tome una fotografía.
<b>cantidad de metros hasta el pozo</b> Sensor numérico que describe la cantidad de metros que hay desde donde está el robot hasta el pozo de las ánimas (es decir, hasta el punto donde comienza el pozo) .	

El robot debe posicionarse justo un metro antes de donde comienza el pozo, tomar la fotografía. Se pide entonces que escriba el procedimiento **Tomar fotografía al pozo de las ánimas** que haga precisamente lo que su nombre indica.

### Ejercicio 2) La peregrinación a Luján

La peregrinación a Luján es una tradicional celebración religiosa en donde diversas congregaciones católicas salen caminando desde Liniers hasta Luján, en un trayecto que implica múltiples kilómetros. En el trayecto hay varias paradas en donde los feligreses pueden detenerse e hidratarse o alimentarse.

Nuestra misión es programar el primer robot que realice la peregrinación a Luján. Nuestro robot no va a hidratarse, pero sí va a tener que realizar recambios de aceite y neumáticos en cada parada.

Para ello contamos con un robot con las siguientes primitivas:

<b>Andar durante un kilómetro</b> Hace que el robot avance por el camino durante un trayecto de un kilómetro.	<b>Cambiar neumáticos y aceite</b> Hace que el robot cambie sus neumáticos y aceite. El robot debe estar en una parada para poder realizar el cambio.
<b>cantidad de kilómetros hasta llegar a la próxima parada</b> Sensor numérico que describe la cantidad de kilómetros que debe recorrer el robot para llegar a la siguiente parada.	<b>cantidad de paradas en el recorrido</b> Sensor numérico que describe la cantidad de paradas que hay en el trayecto para llegar a Luján.

Lo que se pide es que realice un procedimiento **Llevar robot hasta Luján** que haga que el robot realice el trayecto hasta Lujan, cambiando el aceite en cada parada, salvo en la última, en donde no hay que hacer nada.

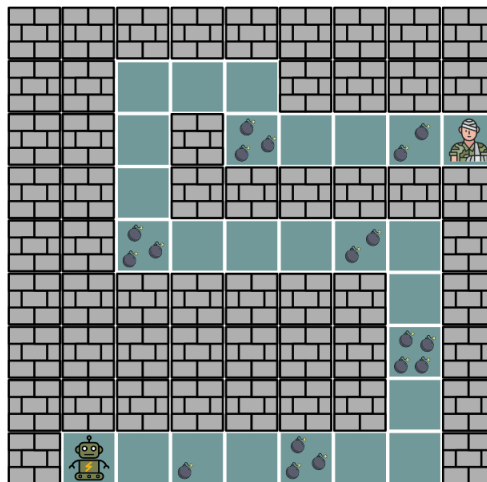
### Ejercicio 3) Rescatando al soldado Brian

Brian se encuentra perdido en el frente de batalla. Los alemanes lo tienen escondido en los complejos túneles del castillo Wolfenstein, donde han dejado varias minas que deben ser desactivadas previo a poder llegar a Brian. Al menos esa es la trama de la nueva superproducción de Jollywood, el lugar donde se hacen las mejores películas clase B del mundo.

Su objetivo, programar al actor robótico “Tomás Janks”, para que realice las peligrosas escenas de acción, que implican recorrer el castillo hasta llegar donde se encuentra Brian. En su camino, el robot deberá desactivar las minas dejadas por los alemanes. El robot cuenta con las siguientes primitivas:

<b>Dar un paso en el camino</b> Mueve el robot una única ubicación en el laberinto. El robot no debe haber llegado a dónde está Brian.	<b>Desactivar mina</b> Hace que el robot desactive una única mina de la ubicación actual.
<b>cantidad de minas en la ubicación</b> Sensor numérico que describe cuántas minas hay en la ubicación donde se encuentra el robot.	<b>¿está Brian?</b> Indica sí Brian está en la ubicación actual.

El escenario consiste en un laberinto, del cual no se sabe a priori su forma, dividido en varias ubicaciones. En cada ubicación puede haber minas (ninguna, una, dos, o muchas más, tampoco lo sabemos a priori). Al final del laberinto, en la última ubicación, está Brian, y es la ubicación en donde debe finalizar el robot. Notar que puede haber minas en donde se encuentra Brian, que también deben ser desactivadas, y también en la ubicación en donde comienza el robot. Para darnos una idea del escenario nos brindan el siguiente dibujo de ejemplo.



Se pide entonces escriba un programa que lleve al Robot hasta la ubicación de Brian, desactivando todas las minas que encuentre en el camino.

## Ejercicio 4) Regreso a la isla del sol

Todo está listo para llegar a la paradisíaca Isla del Sol nuevamente. Pero esta vez queremos poder llevar la cena lista, por lo que queremos pescar a los peces del camino (Los peces, no a los tiburones). Además, contamos esta vez con un nuevo conjunto de primitivas,

En esta actividad el escenario puede variar, tanto en el lugar de donde arranca el bote, como el lugar donde se encuentra la isla, como el camino a seguir para llegar, como los lugares donde están los peces, y la cantidad de peces en cada lugar (puede no haber peces o puede haber varios en una misma ubicación). A diferencia de la última vez, tampoco sabemos exactamente a cuántas ubicaciones de distancia está la isla, aunque sabemos que siempre hay un camino a la misma.

Pueden verse dos ejemplos de escenario a continuación, en el primero, el bote aún no ha arrancado el recorrido, mientras que en el segundo ya ha dado 3 pasos por el camino sin tiburones.



El bote es nuestro autómata, y entiende las siguientes primitivas:

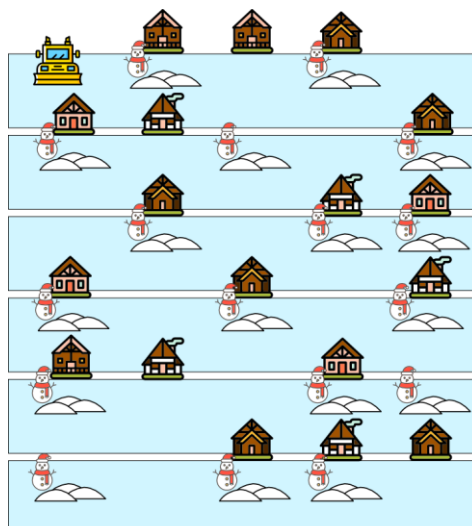
<b>Avanzar en el camino</b> Hace que el bote avance una ubicación por el camino sin tiburones. Falla sí el bote ya está en la Isla del Sol.	<b>Pescar</b> Hace que el bote pesque un pez de la ubicación donde se encuentra. Falla sí no hay un pez en la ubicación del bote o sí todos los peces de la ubicación ya fueron pescados.
<b>cantidad de ubicaciones hasta la isla</b> Sensor numérico que describe la cantidad de pasos que hay desde donde comienza el bote hasta la Isla del Sol, sin incluir la ubicación de esta última.	<b>¿hay pez?</b> Indica sí hay un pez en la ubicación en donde se encuentra el bote que aún no se haya pescado.
<b>Desembarcar</b> Desembarca en la isla. El bote debe estar en la misma ubicación que la isla.	

Se busca un procedimiento “Ir a la isla del sol” que lleve el bote a la isla del sol, pescando todos los peces que encuentre en el camino, y desembarque allí.

Ejercicio 5) Me llama usted, entonces voy...

Vuelve el invierno y Don Barredora se pone otra vez en acción. Pero la ciudad cambia constantemente, y ahora las casas están por todos lados.

En este problema sigue habiendo filas de calles, pero no sabemos ni cuántas ni el largo de las mismas. Tampoco tenemos información sobre dónde están las casas en dichas calles, que pueden estar en cualquier lugar. No solo eso, sino que algunas de ellas ya fueron limpiadas el día anterior. y no necesariamente tienen nieve. También hay nieve en lugares donde no hay casas, y Don Barredora no trabajará de más, dejando esa nieve donde está.



Para manipular a Don Barredora están las siguientes primitivas:

<b>Mover barredora arriba</b> Mueve la barredora una ubicación hacia arriba. Falla sí no hay más ubicaciones hacia arriba.	<b>Mover barredora abajo</b> Mueve la barredora una ubicación hacia abajo. Falla sí no hay más ubicaciones hacia abajo.
<b>Mover barredora izquierda</b> Mueve la barredora una ubicación hacia la izquierda. Falla sí no hay más ubicaciones hacia la izquierda.	<b>Mover barredora derecha</b> Mueve la barredora una ubicación hacia la derecha. Falla sí no hay más ubicaciones hacia la derecha.
<b>Quitar nieve</b> Quita la nieve de la ubicación actual. Falla sí no hay nieve en la ubicación actual.	<b>¿hay casa?</b> Indica sí hay una casa en la ubicación de Don Barredora.
<b>¿hay nieve?</b> Indica sí hay nieve en la ubicación de Don Barredora	<b>¿se puede mover a la derecha?</b> Indica sí Don Barredora tiene lugar para moverse a la derecha.
<b>¿se puede mover a la izquierda?</b> Indica sí Don Barredora tiene lugar para moverse a la izquierda.	<b>cantidad de calles de la ciudad</b> Sensor numérico que indica cuántas calles (filas) hay en la ciudad.

Se le pide que escriba un procedimiento **Quitar nieve de la ciudad** que quite toda la nieve de las casas que aún tengan nieve.

## Ejercicio 6) El bombero valiente

Hay un edificio en llamas, y un bebé en el último piso. Pero no hay que tener miedo, pues Bob el bombero valiente está listo para la acción.

En esta actividad Bob será nuestro autómatas, el escenario representará las escaleras del edificio. Bob comienza en la punta más abajo y a la izquierda del edificio, y el bebe se encuentra en la punta más arriba. Nuestro trabajo será ayudar a Bob a rescatar al bebé. Eso sí, hay fuego en el camino, y deberemos asegurarnos de ir apagando todo el fuego que podamos. Para ello, vamos a contar con baldes de agua o extintores en diversas ubicaciones.

El dibujo a la derecha representa el edificio y sus escaleras, la forma del escenario puede variar en el ancho de los pisos, así como en la ubicación de los fuegos, extintores y baldes. Por otro lado, la altura entre un piso y otro siempre es de dos ubicaciones, y siempre hay ocho pisos en el edificio. Por ello, se espera realice una solución general, para cualquier distribución de incendios, y no particular para la que se muestra en el dibujo de ejemplo.



Su trabajo será entonces escribir el procedimiento **Salvar al bebé apagando incendios** que haga que Bob llegue hasta el bebe, y lo salve, a la vez que apague todos los incendios posibles en el camino.

El dibujo a la derecha representa el edificio y sus escaleras, la forma del escenario no varía, pero la ubicación de los fuegos, extintores y baldes sí. Por ello, se espera realice una solución general, para cualquier distribución de incendios, y no particular para la que se muestra en el dibujo de ejemplo.

Vamos a contar con las siguientes primitivas.

Algunas aclaraciones adicionales:

- Puede haber fuego a apagar donde arranca el bombero, pero no dónde está el bebé.
- No puede haber matafuegos y balde de agua en la misma ubicación, nunca.

<b>Mover arriba</b> Hace que Bob se mueva hacia arriba una ubicación. Debe haber una ubicación hacia arriba.	<b>Mover abajo</b> Hace que Bob se mueva hacia abajo una ubicación. Debe haber una ubicación hacia abajo.
<b>Mover a la izquierda</b> Hace que Bob se mueva hacia la izquierda una ubicación. Debe haber una ubicación hacia la izquierda.	<b>Mover a la derecha</b> Hace que Bob se mueva hacia la derecha una ubicación. Debe haber una ubicación hacia la derecha.
<b>¿hay fuego?</b> Indica sí hay un fuego en la ubicación donde está Bob.	<b>¿hay extintor?</b> Indica sí hay un extintor en la ubicación donde está Bob.
<b>¿hay balde de agua?</b> Indica sí hay un balde de agua en la ubicación donde está Bob.	<b>Apagar incendio</b> Apaga el incendio de la ubicación donde está Bob. Falla si no hay fuego en la ubicación donde está Bob, o si hay fuego, pero no hay un balde de agua o extintor.
<b>Rescatar bebé</b> Hace que Bob rescate al bebé. Falla si Bob no está sobre el bebé.	<b>ancho del piso</b> Sensor numérico que describe el ancho del piso actual en el que se encuentra Bob.