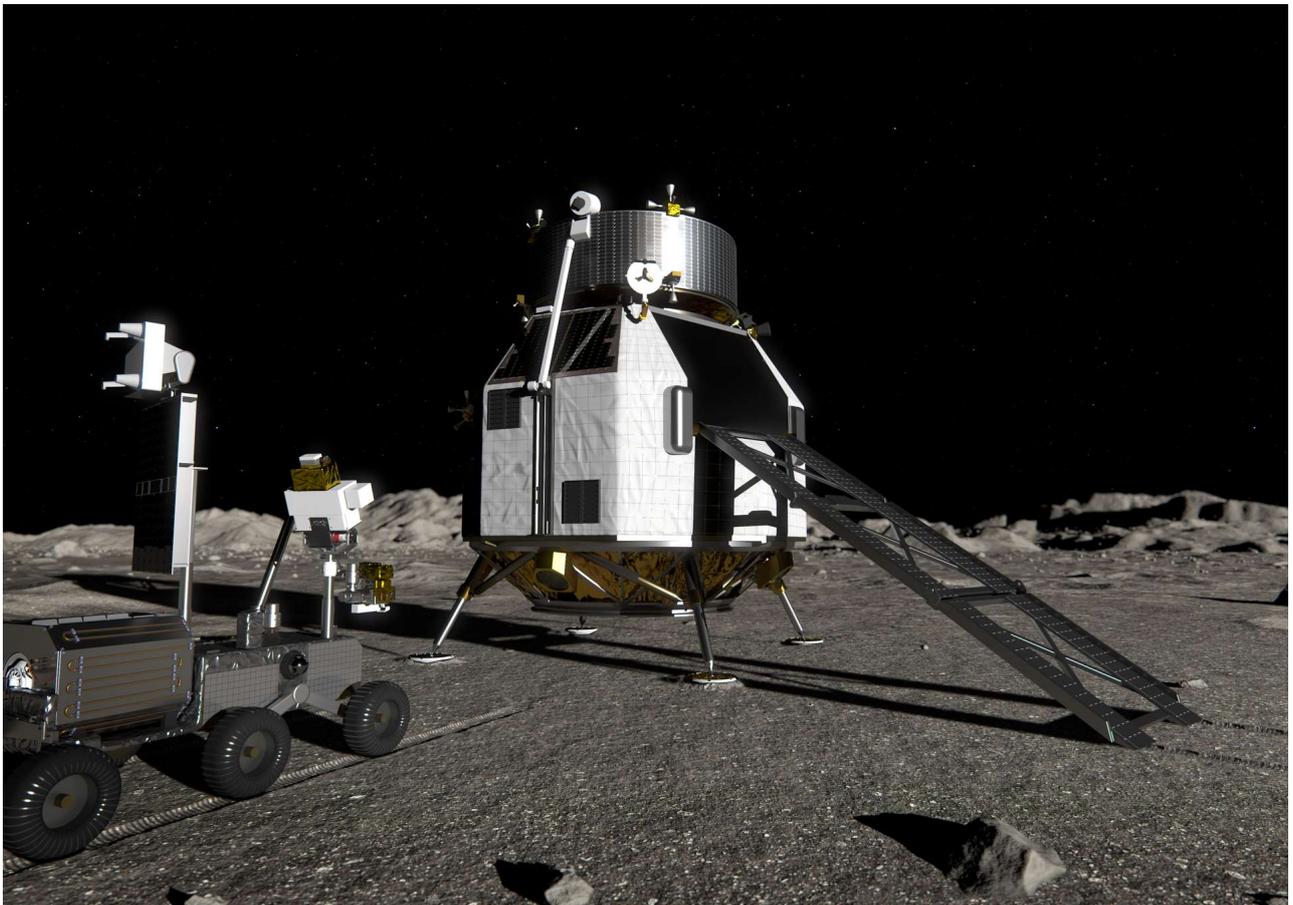

III Feria de las Ingenierías. Abril 2021

Taller: Programa un robot en la Luna

Por **Elena Álvarez Castro**

ESERO Spain Education Technician.

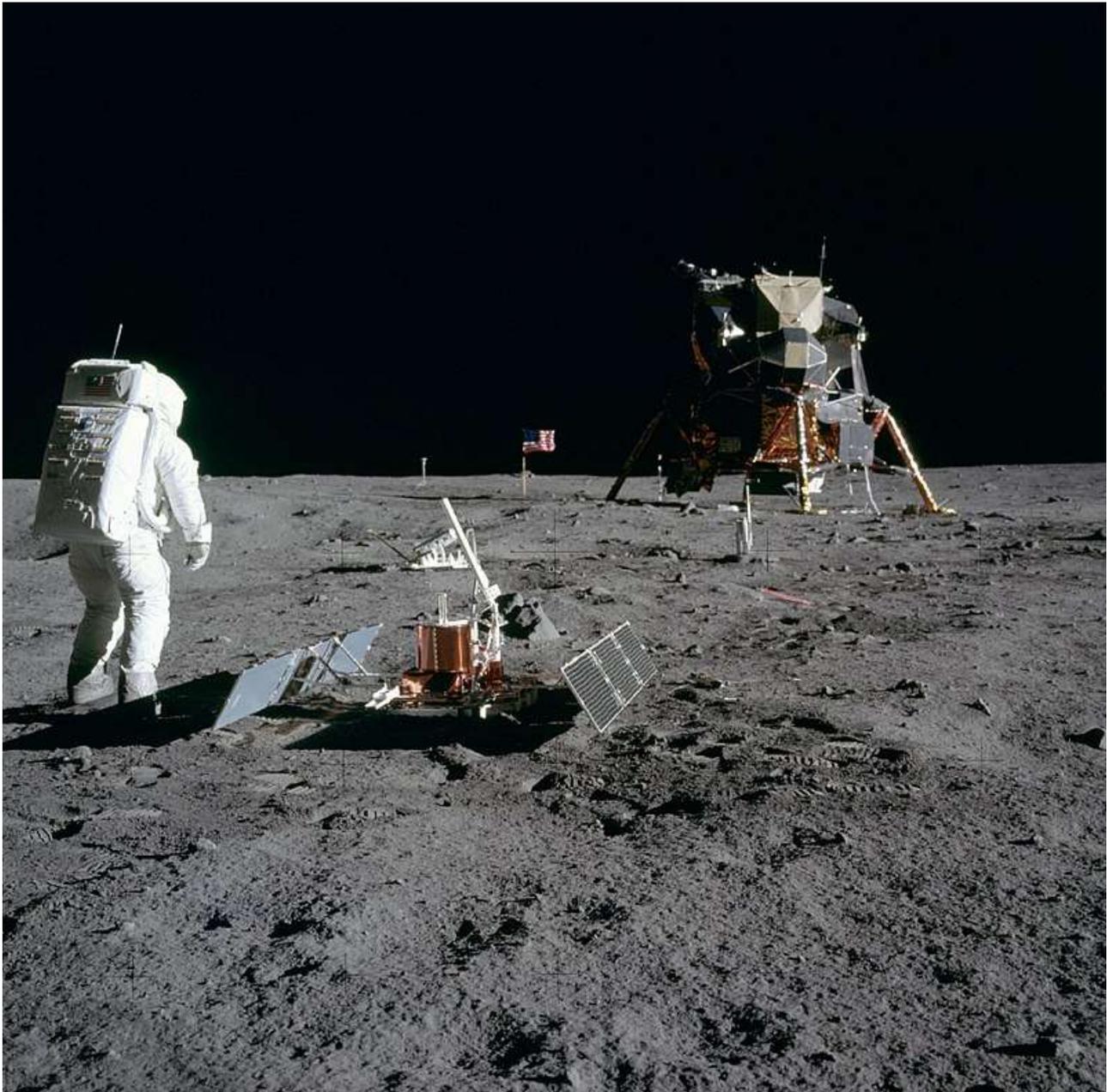


Heracles on the Moon:

https://www.esa.int/ESA_Multimedia/Images/2019/06/Heracles_on_the_Moon#.YHsashbSB98.link

Contexto

La primera vez que la humanidad llegó a la Luna fue en 1969, en la misión Apollo 11 de la NASA. Desde entonces, sólo 12 astronautas han pisado la superficie Lunar. El regreso a la Luna es uno de los próximos pasos de la exploración espacial, que nos permitirá saber más sobre el universo.

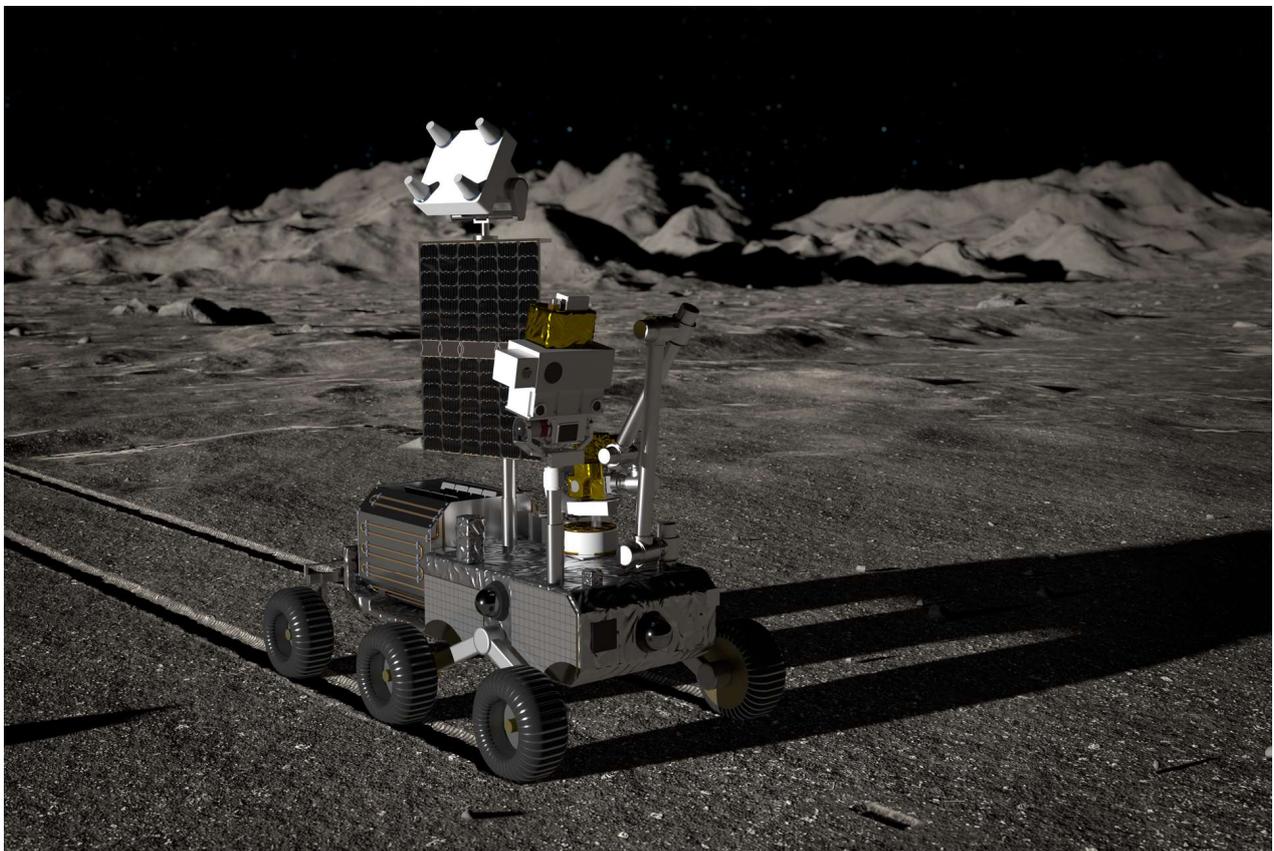


De NASA Neil A. Armstrong - Great Images in NASA Description, Dominio público,
<https://commons.wikimedia.org/w/index.php?curid=6449728>

El objetivo de las próximas misiones a la Luna es realizar experimentos y tomar muestras de su superficie. Esta misión resulta muy arriesgada para cualquier astronauta, pero hoy en día contamos con robots teledirigidos que pueden realizar estas tareas por nosotros.

La primera misión planeada para aterrizar en la Luna y traer muestras a la Tierra es Heracles, una misión conjunta entre la Agencia Espacial Europea y las agencias espaciales de Canadá y Japón. En esta misión, se controlará de forma remota un robot todoterreno, que recorrerá la superficie Lunar para recoger diferentes muestras.

Programar estos robots permite explorar un espacio desconocido sin poner en riesgo la vida de los astronautas. Los todoterrenos espaciales también usan diversos sensores para escanear y cartografiar el terreno circundante de manera que puedan moverse de forma autónoma.



Heracles Rover:

https://www.esa.int/ESA_Multimedia/Images/2019/06/Heracles_rover#.YHsbMfBH0bA.link

En este taller, vamos a programar un Rover que recorre la superficie Lunar en busca de muestras. Utilizaremos un simulador, en el que nuestro Rover podrá moverse por una superficie Lunar simulada y recoger muestras gracias al sensor de color.

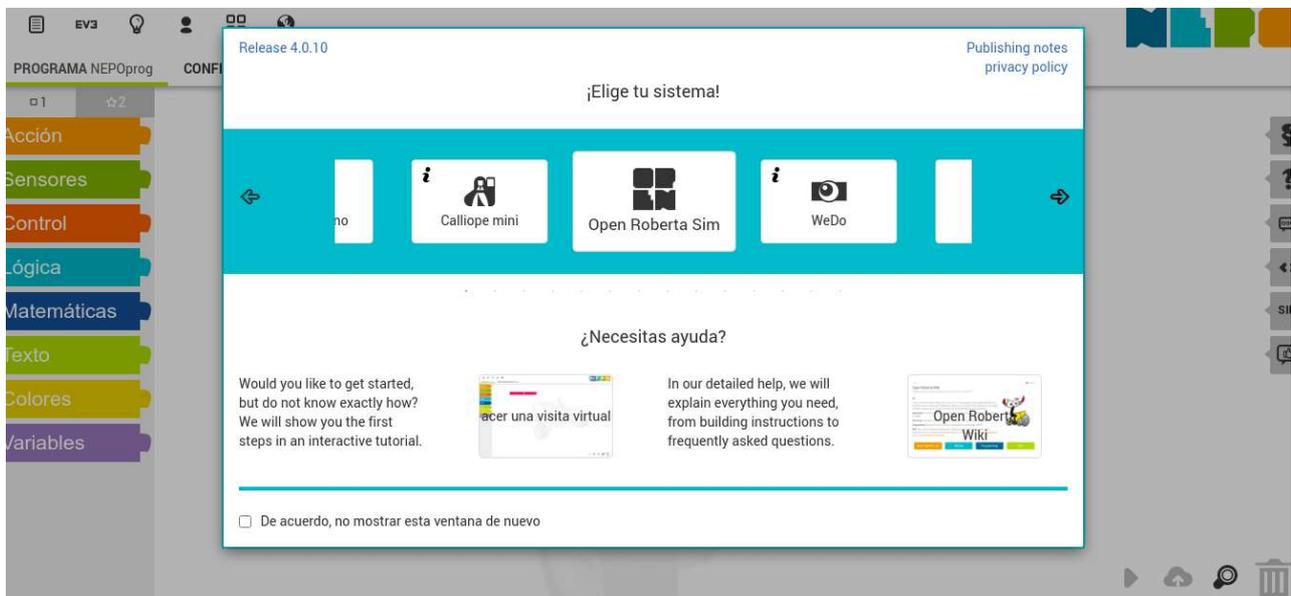
¿Cómo funciona nuestro simulador?

El simulador que vamos a utilizar es Open Roberta Lab. Es un simulador que se puede utilizar de forma gratuita desde el navegador, sin necesidad de instalar nada en el ordenador.

Para entrar, sólo tenemos que ir a este enlace: <https://lab.open-roberta.org/>

¡Ve ahora al navegador y visita Open Roberta Lab!

Al entrar nos pregunta cuál es nuestro sistema, tenemos que seleccionar **OPEN ROBERTA SIM**

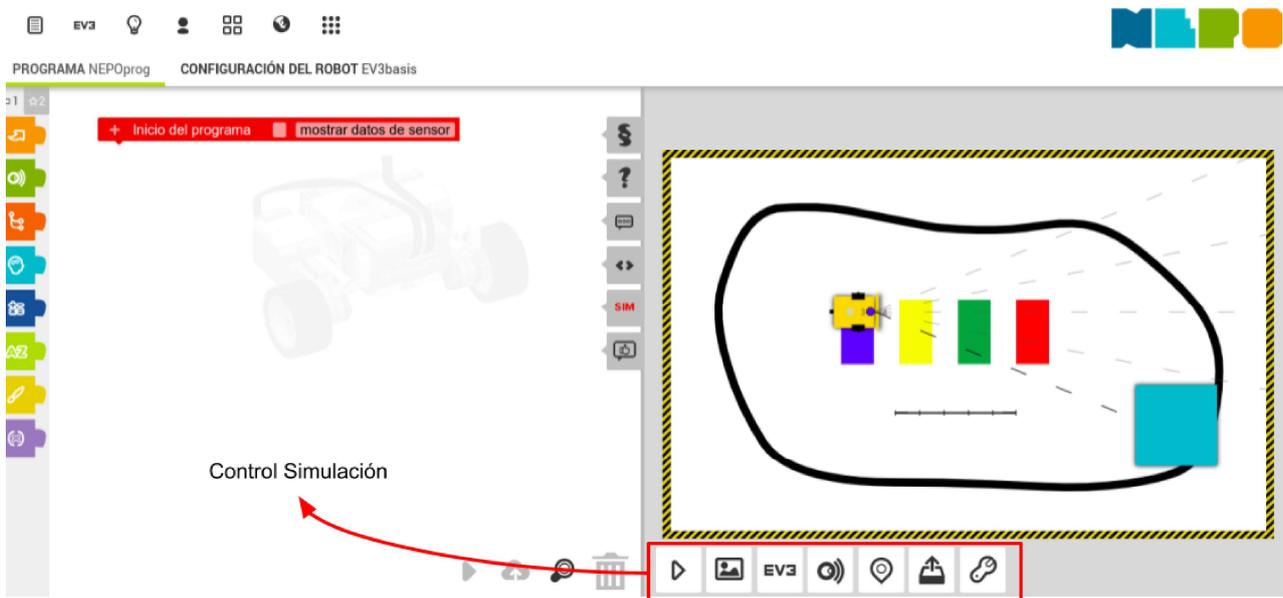


Una vez que seleccionamos nuestro sistema, entramos a la página principal del simulador. En las imágenes que hay en la página siguiente, aparecen indicadas las partes que vamos a utilizar de este simulador:

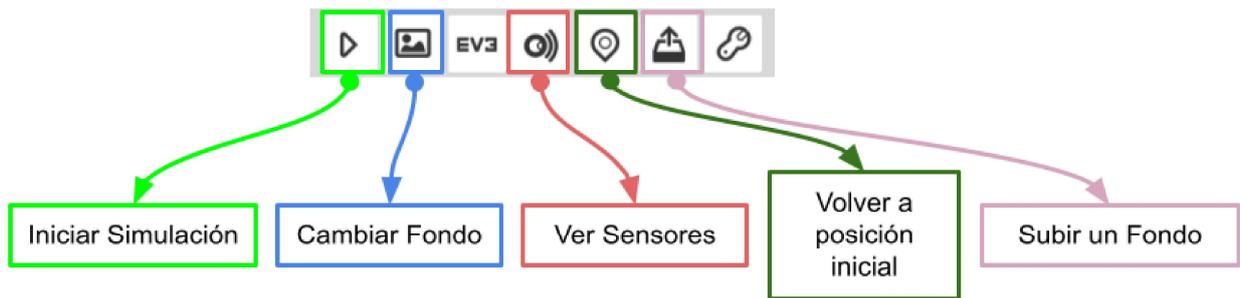
- Bloques de programación
- Zona de programación
- Simulador
- Botones de control de simulación



Para abrir el simulador, tenemos que hacer click en la pestaña **SIM** de la derecha.



En la ventana de simulación podemos ver nuestro robot con una imagen de fondo y varias opciones de control en la parte de abajo. Aquí aparecen indicadas las que vamos a utilizar:



- **Iniciar Simulación:** este botón nos permite empezar con la simulación.
- **Cambiar fondo:** haciendo clic en este botón, puedes cambiar de fondo de simulación y elegir entre uno de los que hay por defecto en el simulador.
- **Ver sensores:** cuando hacemos clic aquí, aparece una nueva ventana con el valor de nuestros sensores en cada momento. Nosotros sólo vamos a utilizar el sensor de color para nuestra misión.
- **Volver a la posición inicial:** esta opción es muy útil, nos permite volver al estado inicial. La utilizaremos para volver a empezar de nuevo en nuestras pruebas.
- **Subir un fondo:** con este botón podemos incluir un fondo de simulación nuevo y personalizado y es lo que nos va a permitir añadir nuestro fondo de la superficie de la luna, dónde tiene lugar nuestra misión.

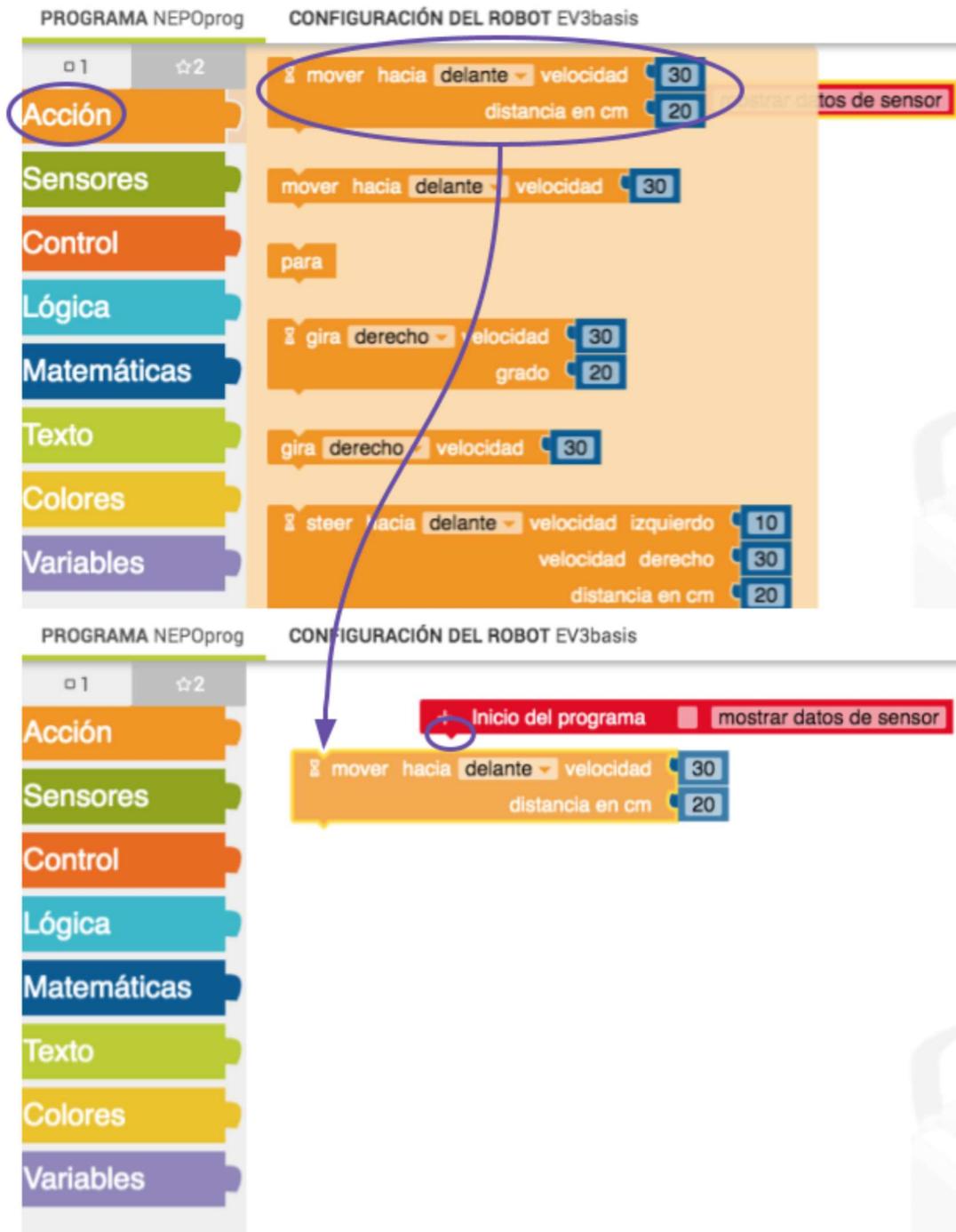
Para comenzar y saber cómo funciona nuestro entorno de programación, vamos a realizar una Misión inicial.

Misión Inicial. Nuestro Simulador

La misión inicial es un entrenamiento para conocer mejor cómo programar nuestro robot dentro de nuestro simulador. Sigue los siguientes pasos para completar la misión.

Paso 1: Añade el bloque de código ‘**Mover hacia delante con una velocidad y una distancia**’ a la zona de programación. Este bloque lo encontramos en la pestaña de Acción, de color naranja.

Tal y como aparece en la siguiente imagen, haz clic en la pestaña de Acción y selecciona el bloque. Debes mantener pulsado el ratón encima del bloque para poder arrastrarlo al entorno de programación y encajarlo debajo del bloque rojo de inicio de programa.

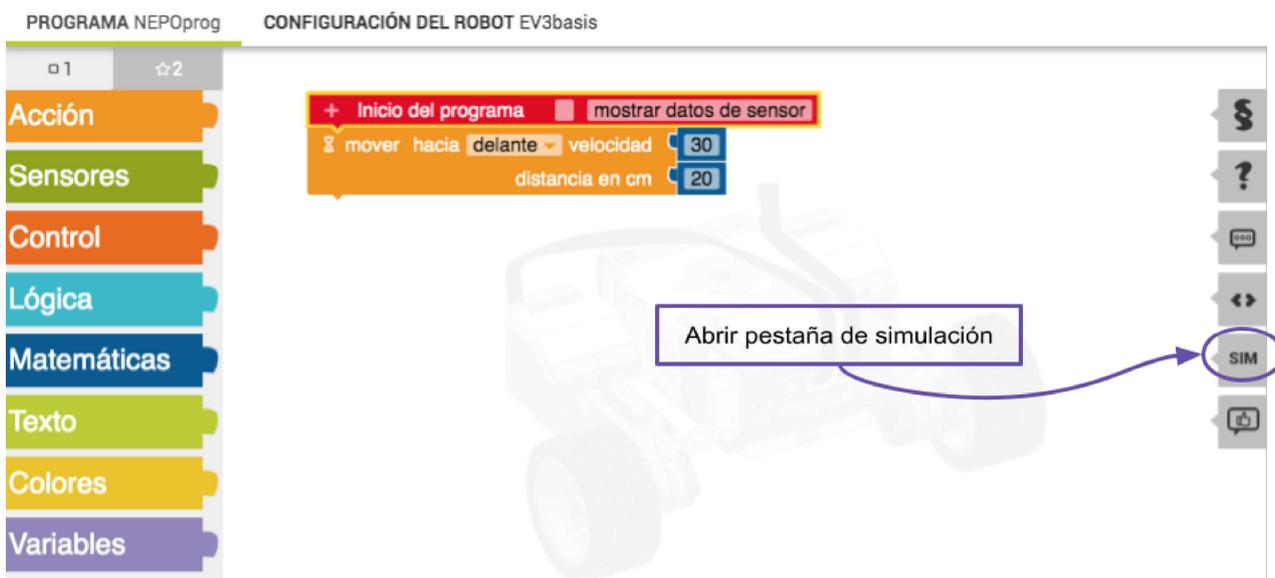


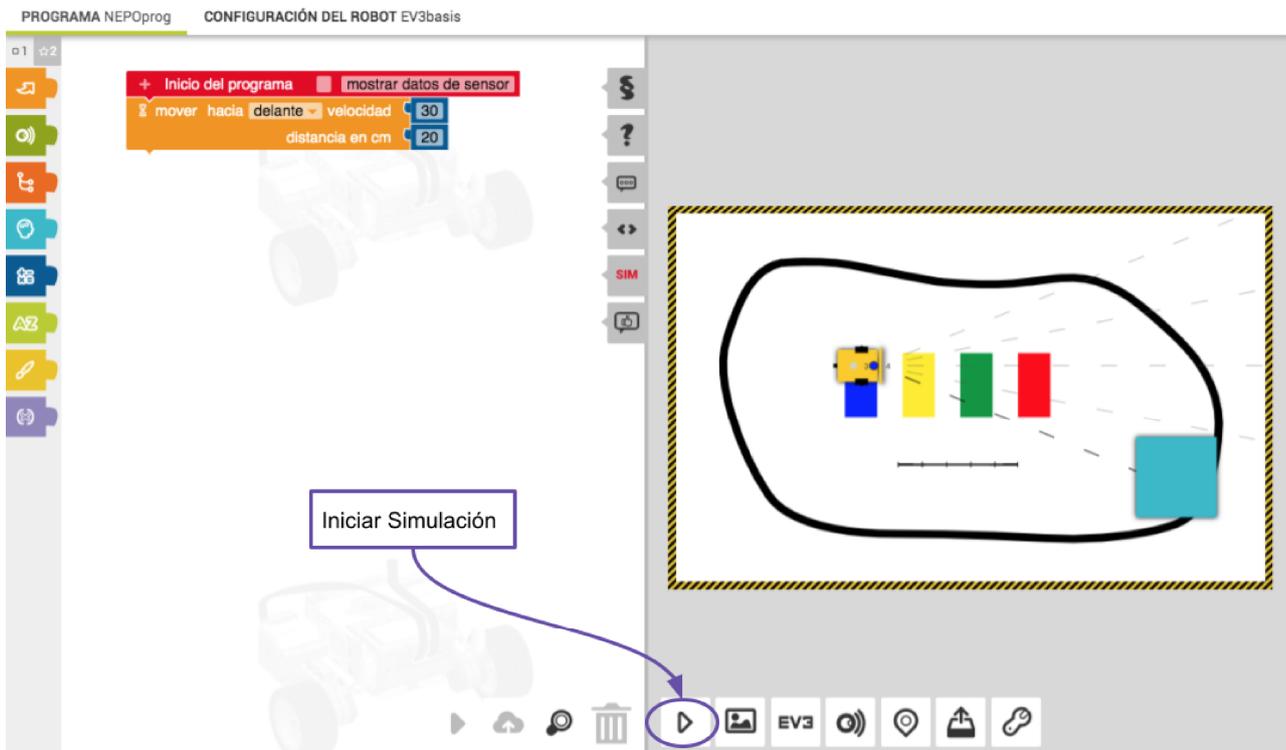
Una vez encajes el nuevo bloque debajo del bloque de inicio de programa, tu entorno de programación quedará como el de la siguiente imagen:





Paso 2: Abrir la pestaña de simulación y hacer clic en el botón **Iniciar Simulación**. La simulación comenzará en el primer fondo por defecto.





¿Puedes ver a tu robot moverse? ¿Va muy lento?

Esta instrucción le está diciendo al robot que se mueva hacia delante, con una velocidad de 30 % a una distancia de 20 cm. Si cambiamos los números de velocidad y distancia podemos controlar estas variables.

Modifica el programa para que el robot se mueva a una velocidad del 50 %, una distancia de 100 cm. ¿Se mueve más rápido? ¿Qué pasa si cambias la velocidad al 100 %?

Ahora vamos a cambiar un poco el código para comprobar que nuestro sensor de color funciona correctamente.

¿Qué es un sensor de color? Es un aparato con el que nuestro robot puede saber de qué color es el suelo.

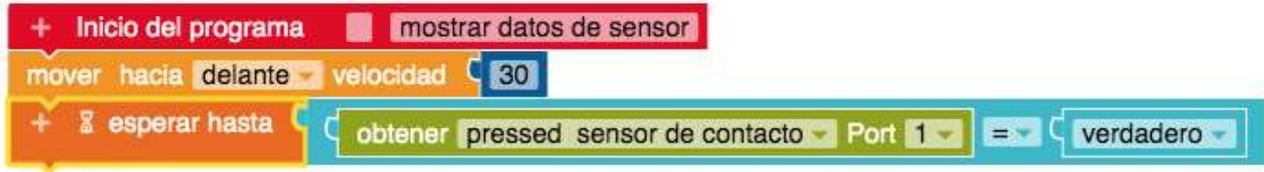
Paso 3: Cambia el bloque anterior por el de simplemente **‘mover hacia delante velocidad’**. Busca en el apartado de control el bloque **‘esperar hasta’** y añádelo a la zona de programación.

PROGRAMA NEPOprog CONFIGURACIÓN DEL ROBOT EV3basis

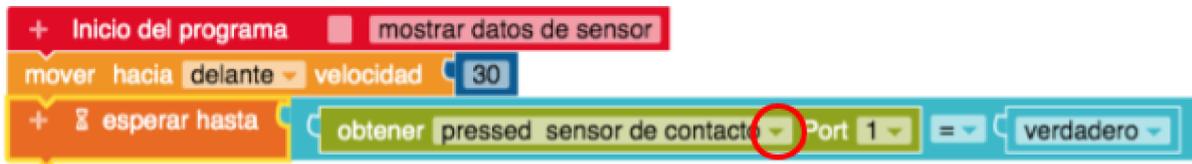
PROGRAMA NEPOprog CONFIGURACIÓN DEL ROBOT EV3basis



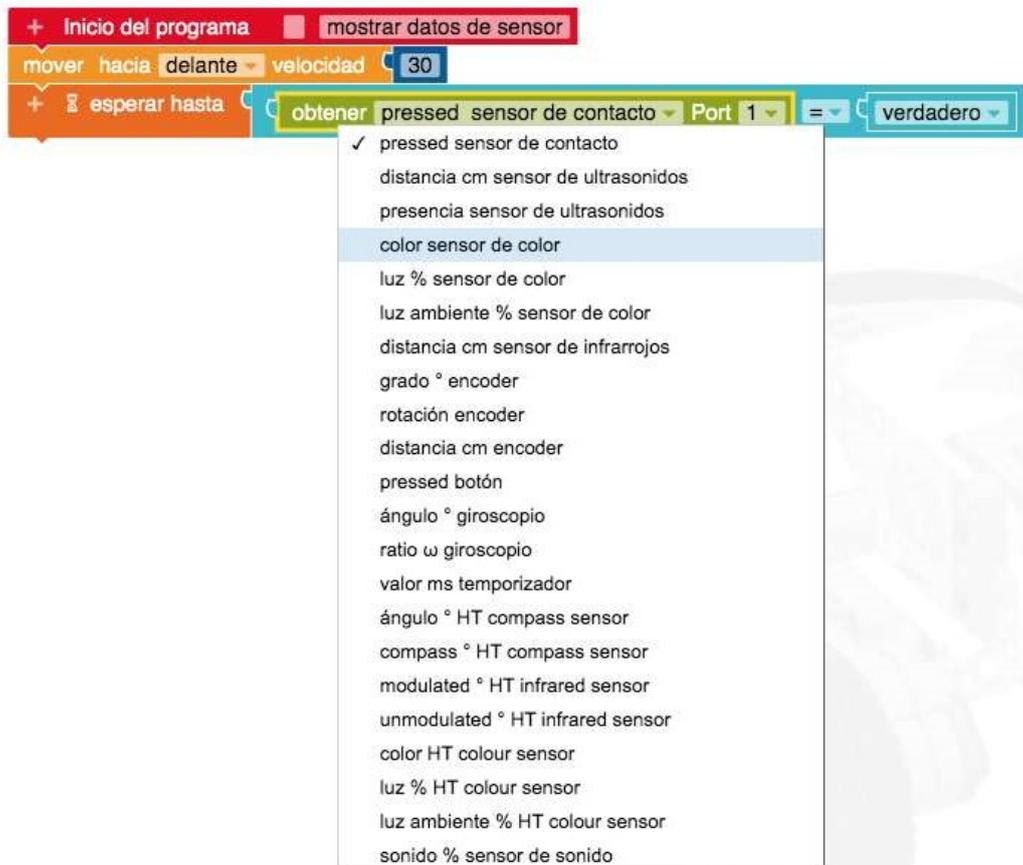
Después de añadir estos dos bloques, en el entorno de programación debes tener el sistema de bloques siguiente:



Paso 4: Vamos a cambiar el bloque verde (dentro del bloque ‘esperar hasta’), para obtener lo que nuestro robot ve desde el sensor de color. Para esto, hacemos clic en la flecha pequeña, como se ve en la imagen:



Se abrirá un menú, tenemos que elegir ‘color sensor de color’

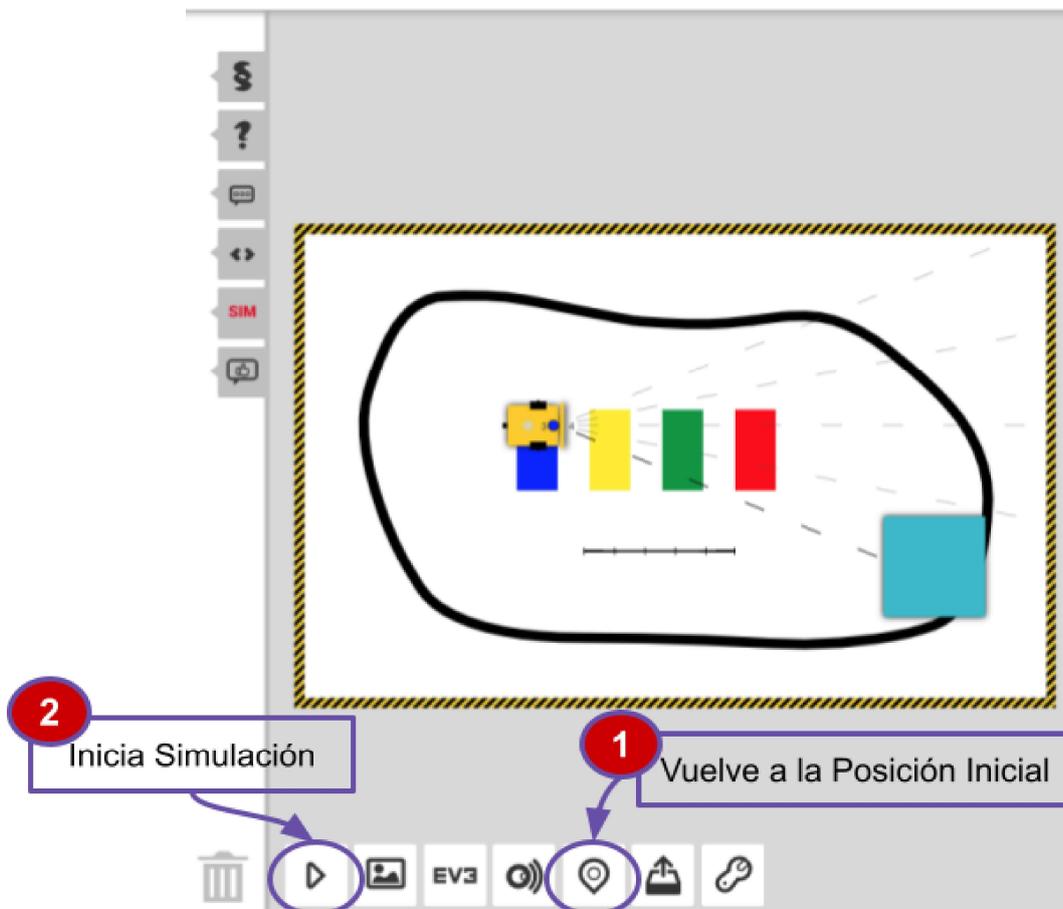


Al seleccionar el sensor de color, nuestro bloque quedará así:

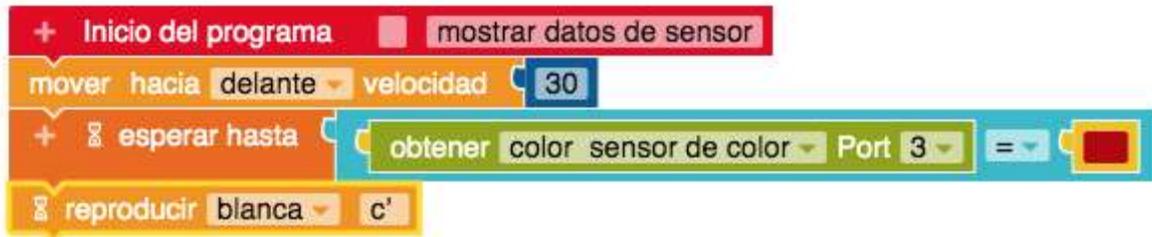


Si iniciamos nuestra simulación, nuestro robot se moverá hacia delante hasta que esté encima del bloque rojo.

!!! Antes de iniciar la simulación, recuerda pulsar el botón de volver a la posición inicial !!!



Paso 5: ¿Quieres incluir una alarma?, añade el bloque ‘reproducir blanca c’ (en el apartado de Acción) y vuelve a iniciar la simulación, ¿qué escuchas? Prueba a cambiar las opciones de este último bloque hasta encontrar un sonido que te guste.



¡Ya podemos comenzar a realizar nuestra Misión en la Luna!

Misión 01. ¡Tenemos que volver a la base!

En esta misión vamos a regresar desde nuestra posición inicial hasta nuestra base lunar. Lo primero que tenemos que hacer es añadir nuestro fondo de simulación, donde veremos nuestro robot en la luna y nuestra base lunar.

Paso 1: Ve a este enlace y descarga las imágenes:

https://drive.google.com/drive/folders/1hlnH2QLkIJkmqYN0P7X-8BtuJTyl_q1A?usp=sharing

Cuando entres en el enlace te aparecerá esta carpeta:



Al entrar en la carpeta imágenes (haz clic encima de la carpeta), te aparecerán las tres imágenes de las tres misiones listas para descargar:

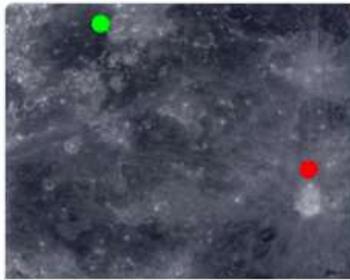


FerialIngenierias2021_MoonRover > Imagenes

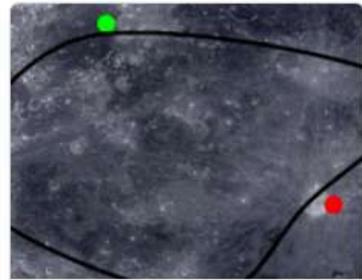
Archivos



 Mision01.jpeg



 Mision02.jpeg



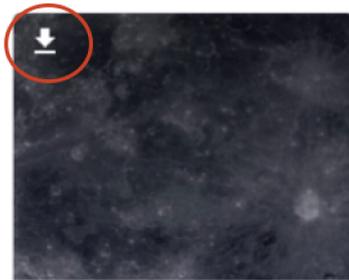
 Mision03.jpeg

Lleva el ratón hacia el pico izquierdo de las imágenes y haz clic sobre el icono que aparece para descargarlas:

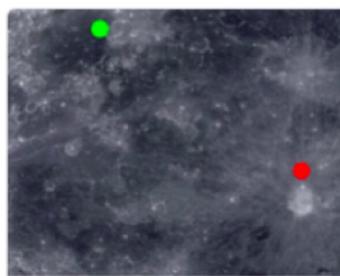


FerialIngenierias2021_MoonRover > Imagenes

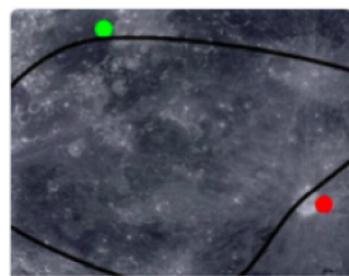
Archivos



 Mision01.jpeg



 Mision02.jpeg

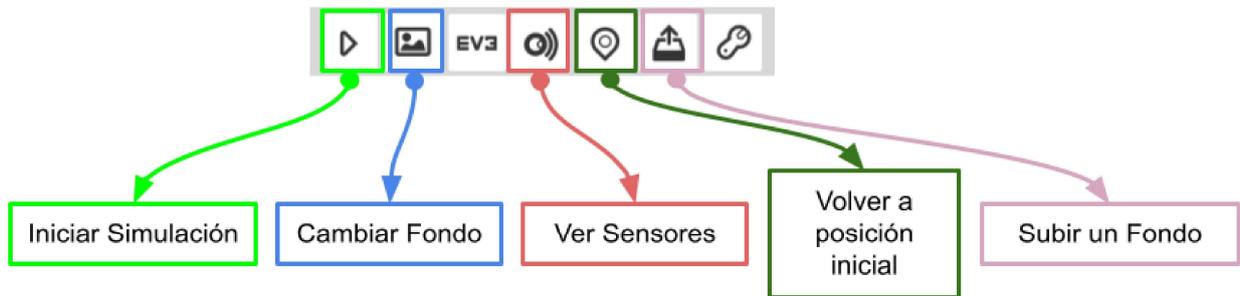


 Mision03.jpeg

¡¡Es importante que recuerdes dónde guardas las imágenes!! Puedes guardar las imágenes en una carpeta en el **ESCRITORIO**. Si al descargar las imágenes, el navegador no te da opción para elegir en qué carpeta guardarlas, las encontrarás en la carpeta **DESCARGAS**.



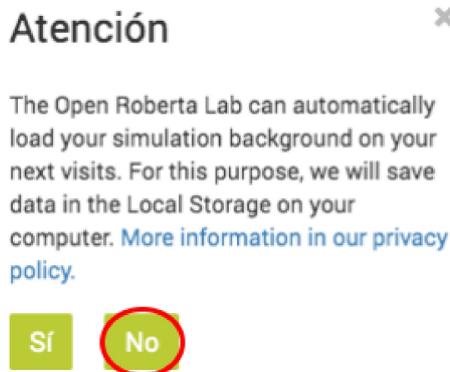
Una vez hayas guardado las imágenes, abre la ventana de simulación de Open Roberta (la pestaña SIM del lateral derecho) y haz clic en el botón de subir un fondo, que está en la sexta posición empezando por la izquierda:



Al darle a este botón, se abrirá una ventana, dónde hay que buscar nuestras imágenes dónde las hemos guardado. Busca Mision01.jpeg y selecciona abrir:



Pulsa **NO** en la ventana que te aparece:



Ahora ya tenemos nuestro fondo, en el que nuestro robot se ha convertido en un Rover Lunar. Debes ver a tu robot en la superficie de la luna.

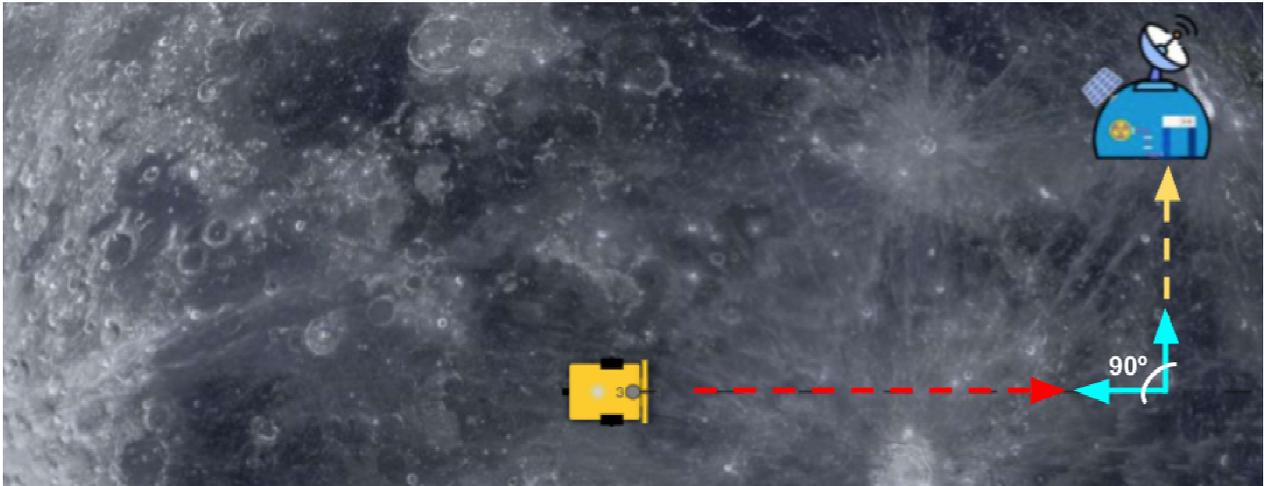
La base lunar, es la casita que puedes ver arriba a la derecha. Nuestra base lunar está equipada con una antena para las comunicaciones, placas solares para tener energía y un pequeño tubo para la gestión de residuos.



Para volver a la base desde nuestra posición inicial, un ejemplo de trayectoria a seguir es:

1. Seguir hacia delante (hacia la derecha, tal y como está colocado nuestro robot) hasta ponernos en línea con la base lunar
2. Girar a la izquierda
3. Seguir recto hacia arriba hasta llegar

De esta forma, necesitamos tres instrucciones, una para cada punto de la trayectoria. En la siguiente imagen puedes ver la trayectoria dibujada. Nuestro robot realizará una camino en L, dónde el ángulo de giro es de 90° .



Para mover nuestro Rover hacia adelante tenemos que incluir el bloque de programación **mover**, que utilizamos en la misión inicial, en el que se establece una velocidad y una distancia. Este bloque lo encontramos entre los bloques de Acción.



Cambiaremos la velocidad y la distancia para llegar a ponernos en línea con la base lunar y poder girar 90°.

Para realizar el giro, utilizamos este bloque de programación, que también se encuentra en los bloques de Acción. Tenemos que cambiar la velocidad y los grados para que se ajusten a nuestra misión.



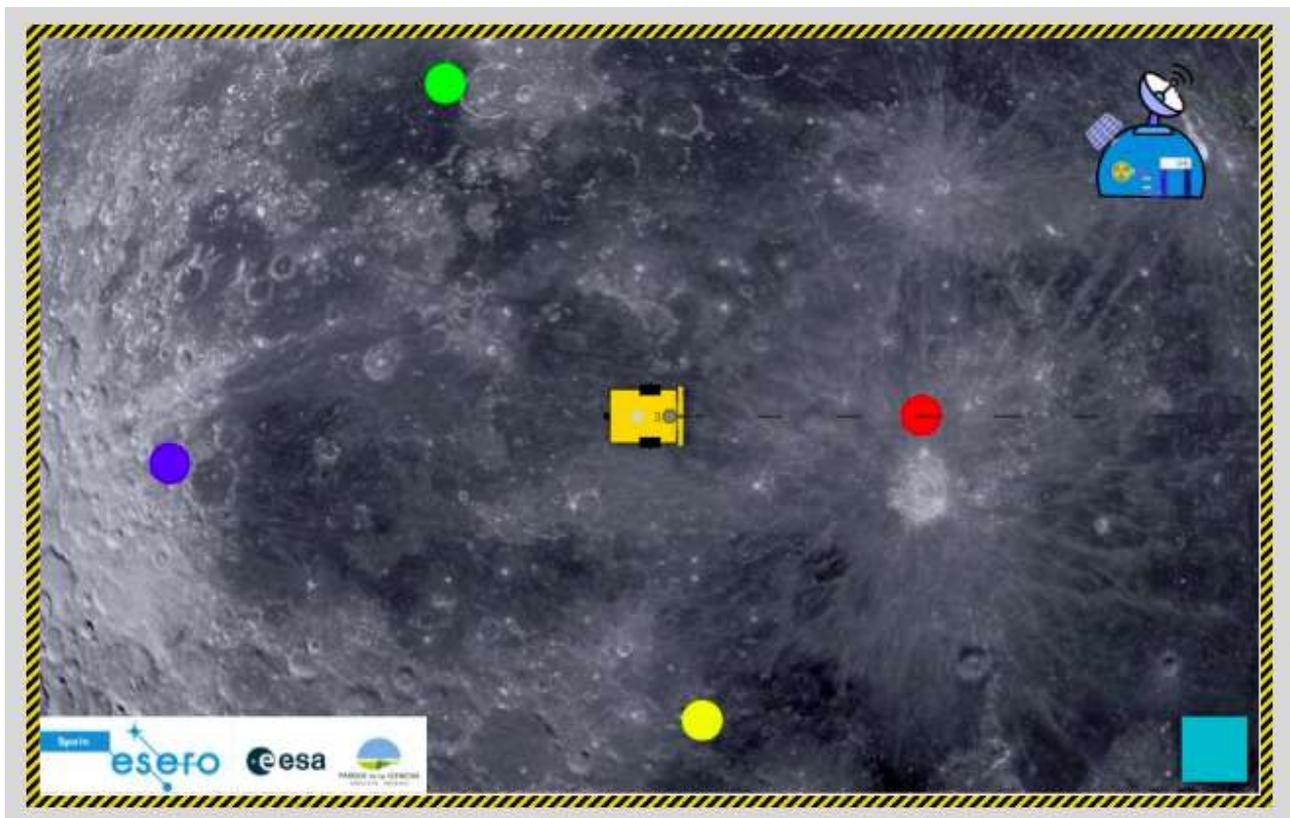
Este es un ejemplo de programación para llegar a la base lunar, hay muchas soluciones y puedes volver a la base de diferentes formas. En este caso, al igual que en la misión inicial, se ha añadido una alarma cuando llegamos a la base, con el bloque **'reproducir'**.

```

+ Inicio del programa
  mostrar datos de sensor
  mover hacia delante velocidad 70
  distancia en cm 130
  gira izquierdo velocidad 50
  grado 90
  mover hacia delante velocidad 70
  distancia en cm 50
  reproducir blanca c'
  
```

Misión 02. Busquemos muestras de la superficie Lunar

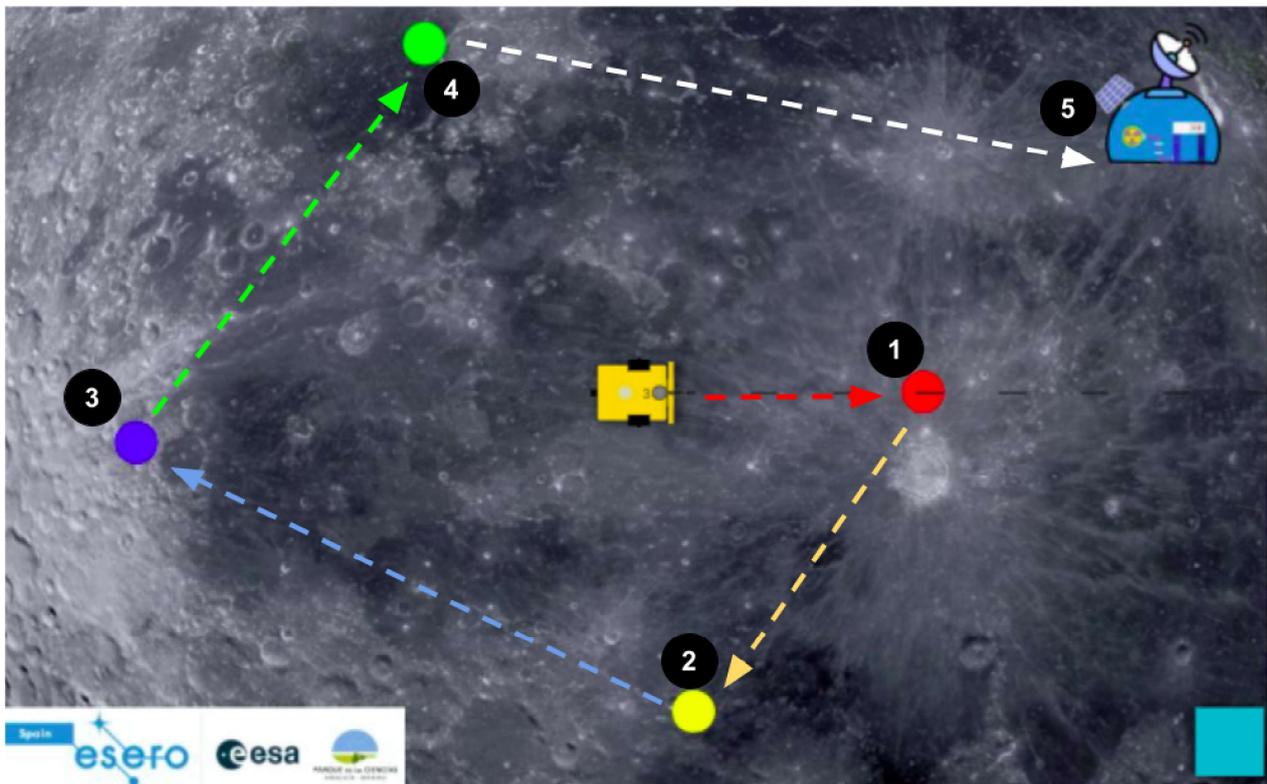
En esta misión programamos nuestro Rover para que pueda tomar todas las muestras de la superficie lunar, utilizando el sensor de color. Para realizar esta misión, debes cargar la imagen Mision02.jpg, en el simulador al igual que la de la primera misión. En esta imagen aparecen las diferentes muestras con varios colores.



Antes de empezar a programar es importante **plantear una trayectoria a seguir**. En este ejemplo, vamos a programar nuestro Rover para que recoja las muestras según el siguiente orden:

1. Recoger la muestra roja
2. Recoger la muestra amarilla
3. Recoger la muestra azul
4. Recoger la muestra verde
5. Volver a la base Lunar

Según esta secuencia, nuestro robot seguirá el camino que marcan las flechas en la imagen:



Para realizar el paso 1 y llegar hasta la muestra roja, vamos a utilizar el sensor de color como condición para que nuestro Rover se detenga. Nuestro Rover, tal y como está colocado debe **seguir hacia delante hasta que el sensor de color sea rojo**. Esta programación, ya la hemos realizado en la misión inicial y es necesario un bloque de Acción y un bloque Control:

```

+ Inicio del programa
  mostrar datos de sensor
  mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [rojo]
  
```

Ahora, presiona el botón de **volver a la posición inicial** y después el botón de **Play** para comprobar que el Rover llega a la muestra roja y se para. Una vez en la muestra roja, nuestro Rover debe girar un cierto ángulo para llegar a la muestra amarilla.

Añade un bloque de giro y ve probando hasta encontrar el ángulo en el que nuestro Rover se queda preparado para llegar a la muestra amarilla en línea recta.

```

+ Inicio del programa
  mostrar datos de sensor
  mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [rojo]
  gira derecho velocidad 50
  grado 120
  
```

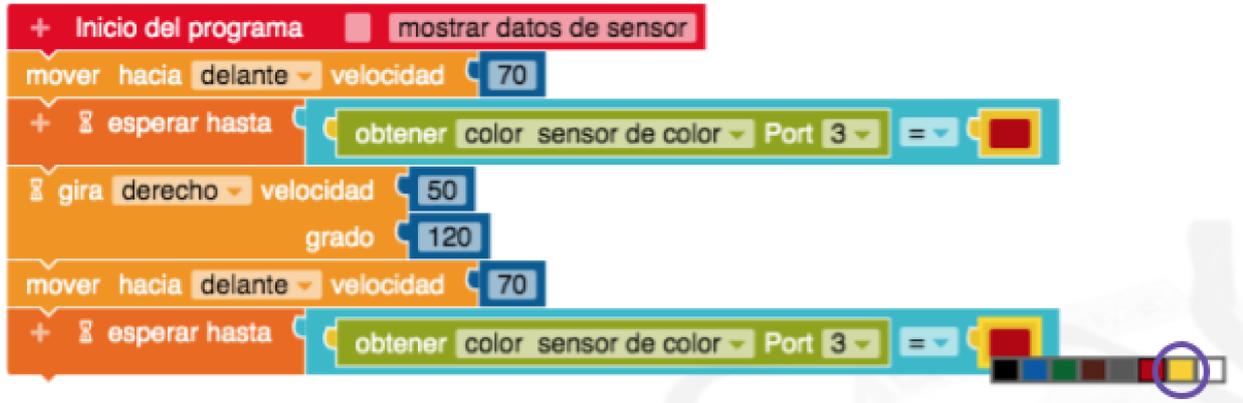
Cuando nuestro Rover esté orientado, tiene que seguir hacia delante **hasta que encuentre la muestra amarilla**. Para esto, volvemos a repetir la secuencia anterior de bloques, pero cambiando el color que detecta el sensor por **amarillo**. Si haces clic en el color, te aparecerán varios para elegir.

```

+ Inicio del programa
  mostrar datos de sensor
  mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [rojo]
  gira derecho velocidad 50
  grado 120
  mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [rojo]
  
```

```

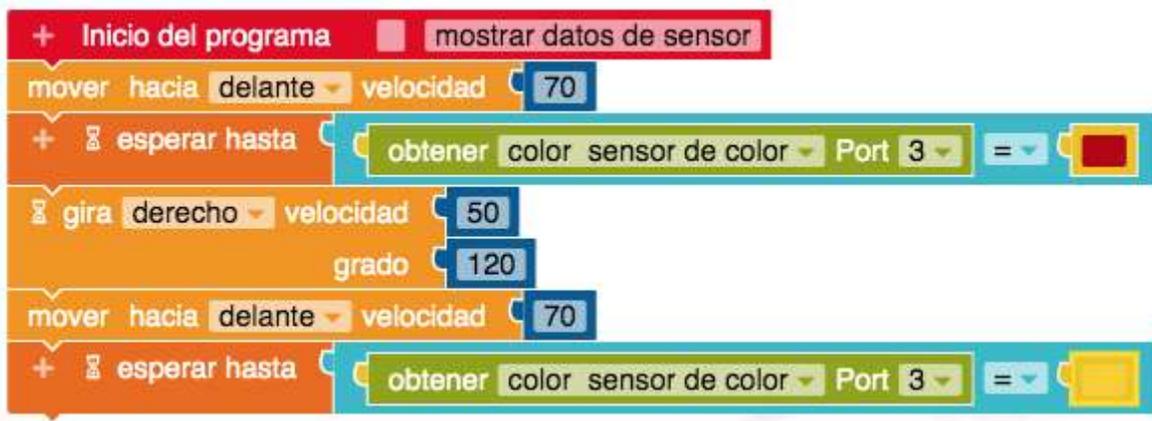
+ Inicio del programa  mostrar datos de sensor
mover hacia delante velocidad 70
+ esperar hasta obtener color sensor de color Port 3 = [rojo]
gira derecho velocidad 50 grado 120
mover hacia delante velocidad 70
+ esperar hasta obtener color sensor de color Port 3 = [rojo]
    
```



Una vez elegido el color amarillo, las instrucciones son que el Rover siga hacia delante hasta que el sensor de color detecte color amarillo.

```

+ Inicio del programa  mostrar datos de sensor
mover hacia delante velocidad 70
+ esperar hasta obtener color sensor de color Port 3 = [amarillo]
gira derecho velocidad 50 grado 120
mover hacia delante velocidad 70
+ esperar hasta obtener color sensor de color Port 3 = [amarillo]
    
```



!!!Vuelve a la posición inicial y dale al Play para comprobar que tu programación funciona!!!

Ahora, al igual que antes hay que realizar un giro para poner a nuestro Rover mirando hacia la muestra azul. Prueba varios grados hasta encontrar uno que se ajuste. Después añade los bloque de código para que **siga hacia delante hasta que encuentre la muestra azul**.

```

+ Inicio del programa
  mostrar datos de sensor
  mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [rojo]
  gira derecho velocidad 50
    grado 120
  mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [amarillo]
  gira derecho velocidad 30
    grado 82
  mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [azul]
  
```

Continúa con la programación para encontrar la muestra verde. Recuerda los pasos a seguir para programar al Rover.

1. Encuentra el ángulo de giro
2. Sigue hacia delante hasta que el sensor de color detecte la muestra
3. Vuelve a la posición inicial y dale al Play para comprobar si la programación funciona

Una vez que el Rover llegue a la muestra verde, tenemos que volver a la base Lunar. En este caso giramos un ángulo adecuado y añadimos el bloque de código que nos permite establecer una determinada distancia.

```

gira izquierda velocidad 30
  distancia en cm 20
  
```

Cambiamos la distancia y la velocidad para llegar a nuestra base Lunar.

!!! Vuelve a la posición inicial y dale al play para comprobar que tu programación funciona!!!

Cuando la programación esté completa y funcione correctamente, puedes añadir una alarma cuando llegues a la base. Este es un ejemplo de cómo quedaría nuestra programación completa.

```

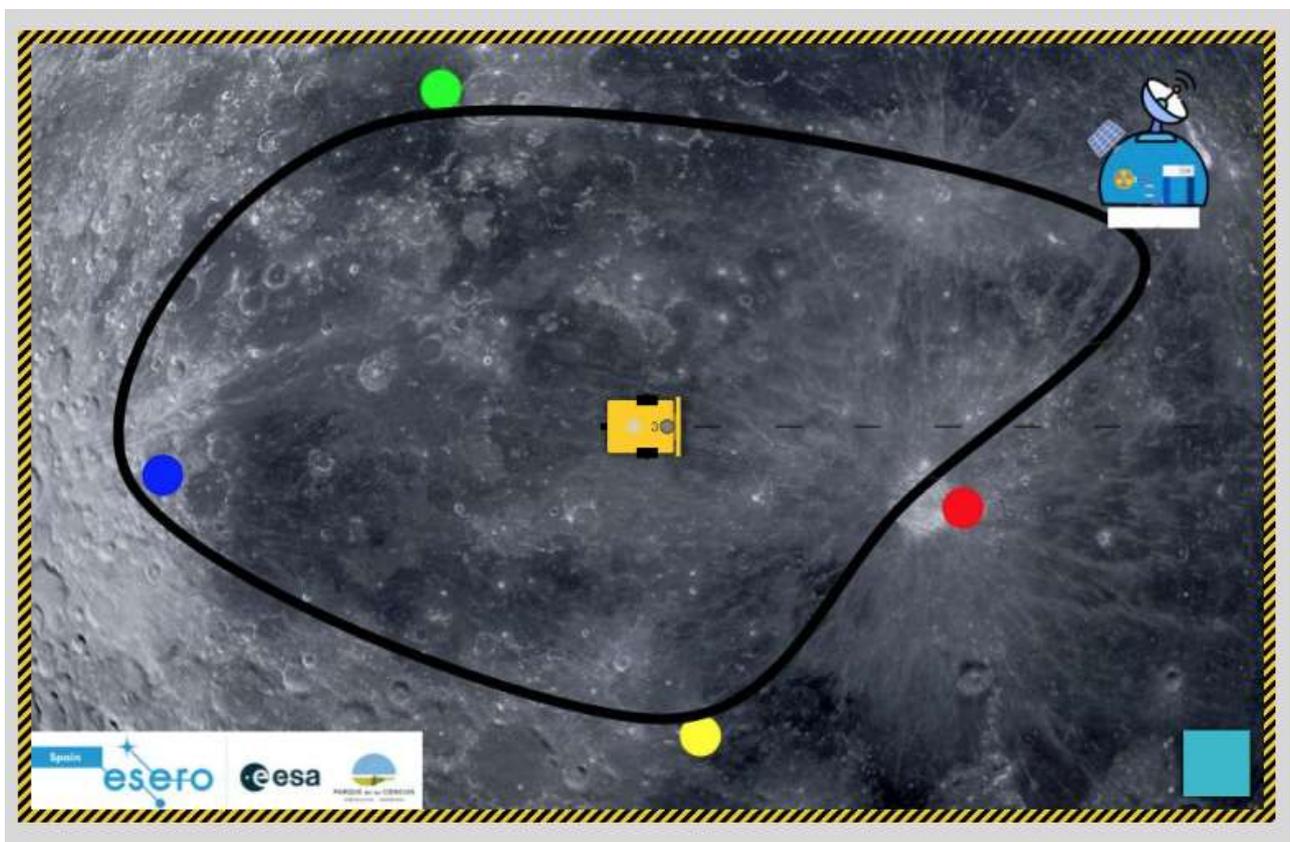
+ Inicio del programa
  mostrar datos de sensor
  mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [Red]
  para
    gira derecho velocidad 50
    grado 120
    mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [Yellow]
  para
    gira derecho velocidad 30
    grado 82
    mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [Blue]
    gira derecho velocidad 30
    grado 100
    mover hacia delante velocidad 70
  + esperar hasta
    obtener color sensor de color Port 3 = [Green]
    gira derecho velocidad 30
    grado 60
    mover hacia delante velocidad 70
    distancia en cm 150
  reproducir negra e"
  
```



Ampliación: ¿Te atreves a completar una nueva misión?

Misión 03. ¡Hagamos una carrera en la Luna!

En esta misión, nuestro Rover pasará por todas las muestras, siguiendo una línea negra. Al añadir la imagen Mision03.jpg al entorno de simulación, podrás ver la línea negra.



¿Cómo funciona un siguelineas?

El principio básico de un robot que sigue una línea negra es que siempre esté comprobando que está en la línea negra.

Siguiendo este concepto, podemos llegar a una programación básica, en la que el Rover sigue adelante cuando detecta el color negro y gira (para encontrar el color negro) cuando no lo detecta.

Es necesario utilizar los bloques de control:

- **Repetir indefinidamente:** este bloque permite que nuestro Rover siempre esté haciendo lo mismo (comprobar que está en la línea negra), es lo que en programación se llama bucle. Nuestro Rover estará en bucle y no podrá salir, repetirá siempre lo mismo.
- **Si/hacer/Sino:** en este bloque, ponemos una condición y damos una instrucción en el caso de que se cumpla y una instrucción en el caso de que no se cumpla: **Si el robot está en la línea negra, entonces sigue adelante, si no está en la línea negra, giro a la izquierda para encontrar la línea.**

El bloque **Si/hacer/Sino**, se coloca dentro del bloque **Repetir indefinidamente**, de esta forma, nuestro Rover siempre está comprobando si está en la línea negra.



Pista: para establecer la condición de obtener el color del sensor de color y saber si es igual a negro, hay que utilizar un bloque de Lógica, un bloque de Sensores y un bloque de Colores.

Prueba esta programación. Coloca el robot directamente encima de la línea antes de comenzar la programación.

¿Cómo funciona? ¿Es capaz el Rover de seguir la línea sin problemas? ¿cuándo se desvía? ¿Cómo podemos mejorar esta programación?

Esta programación no funciona del todo bien, el robot se va de la línea si hay curvas marcadas. Para mejorar esto, la clave está en los **giros**. Los bloques de giro que hemos añadido hasta ahora, no indican la rueda que gira, es decir, giran las dos ruedas delanteras a la vez.

Si queremos hacer el giro más suave, para que nuestro Rover no se desvíe, tenemos que utilizar este bloque de Acción:



En este bloque de programación, podemos establecer la velocidad de la rueda izquierda y de la rueda derecha.

¿Qué pasará cuándo la rueda izquierda gira más rápido que la derecha? ¿Y cuándo la derecha gira más rápido que la izquierda? Intenta mejorar la programación utilizando este bloque de programación en las dos condiciones del bloque **Si/hacer/Sino**.

Referencias

Guía didáctica Misión en la Luna:

<http://esero.es/wp-content/uploads/2019/08/Mision-en-la-Luna-062019.pdf>

