

Programación

Introducción a la programación
(Robomind)

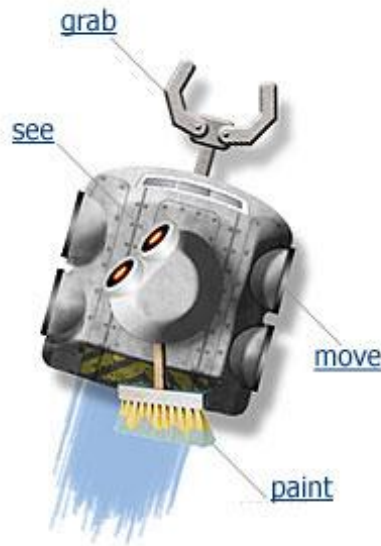
Programación

Robomind

Robomind

Introducción

- Robomind es un software educativo que permite programar un robot para que ejecute tareas en su entorno.



<https://www.robomind.net/es/>

Robomind

Sitio oficial

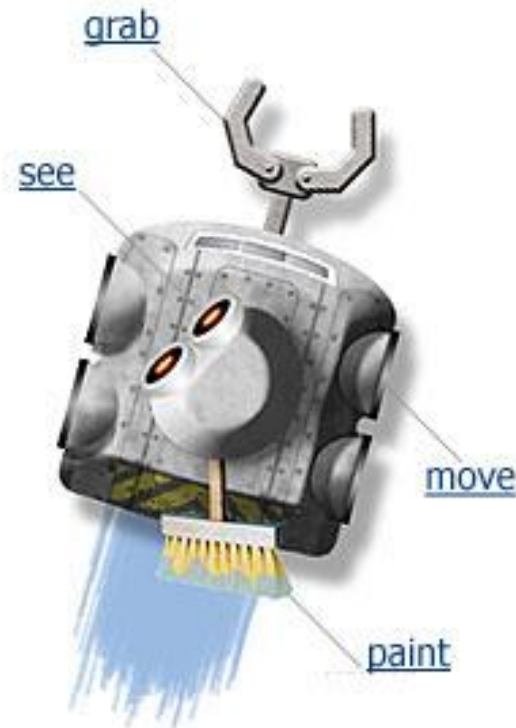
Página web de Robomind:

- <https://www.robomind.net/>
- <https://www.robomind.net/es/>
- <https://www.robomind.net/es/docOverview.htm>
- <https://www.robomind.net/en/download.html>
- <https://www.robomind.net/en/challenges.html>
- <https://www.robomind.net/es/docExamples.htm>

Robomind

Índice

- Comentarios
- Movimiento del robot.
- Pintar.
- Sensores.
- Acciones con las balizas.
- Condicionales.
- Bucles.
- Romper un bucle (truncar)
- Detener el programa (fin)
- Expresiones lógicas.
- Procedimientos.



<https://www.robomind.net/es/docOverview.htm>

Robomind

Comentarios

- Comentarios:
 - Todo texto que aparezca después del simbolo '#' no se interpreta como una instrucción.

Esto es un comentario

- Los comentarios se utilizan para hacer anotaciones sobre partes del script para recordar como funciona.

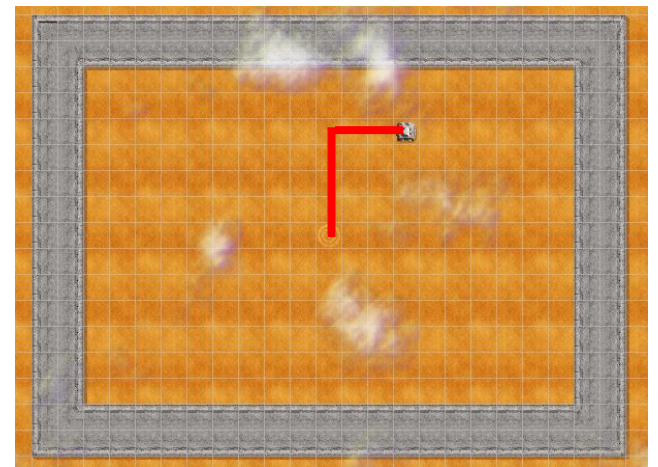
<https://www.robomind.net/es/docProgrammingStructures.htm#commentaar>

Robomind

Movimiento del robot

- Movimiento del robot:
 - adelante(1)
 - atras(1)
 - izquierda()
 - derecha()

adelante(4)
derecha()
adelante(3)



Programación

Ejercicios

Abrir el mapa *openArea.map* y hacer los siguientes ejercicios:

Ejercicio 1:

Mover el robot cinco casillas hacia adelante.

Ejercicio 2:

Mover el robot cinco casillas hacia adelante, girar a la derecha y mover el robot cinco posiciones hacia adelante.

Ejercicio 3:

Mover el robot cinco casillas hacia adelante, girar a la izquierda y mover el robot cinco posiciones hacia adelante.

Ejercicio 4:

Mover el robot cinco casillas hacia adelante y cinco casillas hacia atrás.

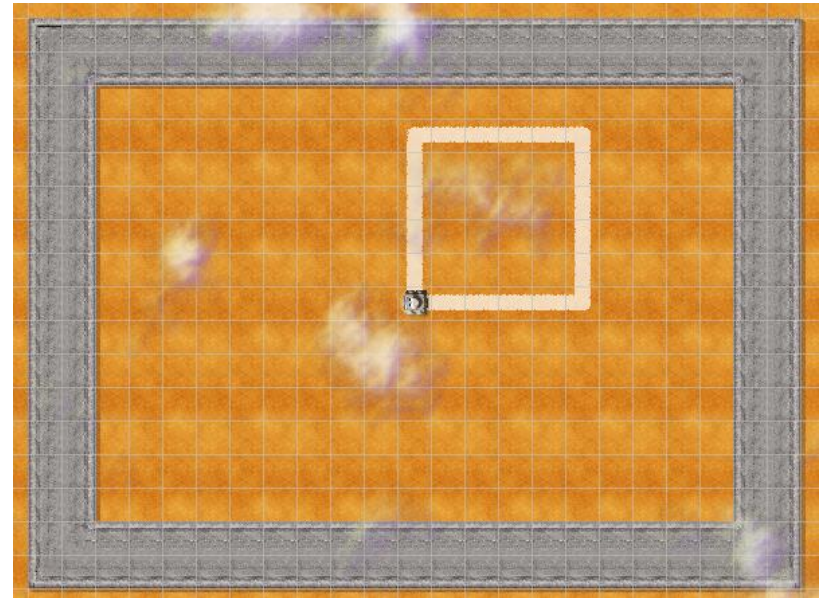
Robomind

Movimiento del robot

- Movimiento del robot:

- norte(1)
- sur(1)
- este(1)
- oeste(1)

```
pintarBlanco()  
norte(5)  
este(5)  
sur(5)  
oeste(5)  
detenerPintar()
```



<https://www.robomind.net/es/docBasicInstructions.htm>

Programación

Ejercicios

Abrir el mapa *openArea.map* y hacer los siguientes ejercicios:

Ejercicio 1:

Mover el robot cinco casillas hacia el norte.

Ejercicio 2:

Mover el robot cinco casillas hacia el norte y cinco casillas hacia el este.

Ejercicio 3:

Mover el robot cinco casillas hacia el norte y cinco casillas hacia el oeste.

Ejercicio 4:

Mover el robot cinco casillas hacia el norte y cinco casillas hacia el sur.

Robomind

Pintar

- Pintar:
 - pintarBlanco()
 - pintarNegro()
 - detenerPintar()

```
pintarBlanco()  
adelante(5)  
detenerPintar()
```



Programación

Ejercicios

Abrir el mapa *openArea.map* y hacer los siguientes ejercicios:

Ejercicio 1:

Escribir un programa que el robot dibuje tus iniciales.

Ejercicio 2:

Escribir un programa que haga que el robot dibuje un cuadrado de color blanco.

Ejercicio 3:

Escribir un programa que haga que el robot dibuje un cuadrado de color negro.

Ejercicio 4:

Escribir un programa que haga que el robot dibuje un cuadrado de color blanco y al lado otro de color negro.

Robomind

Sensores

- Sensores del robot:
 - frenteEsClaro()
 - frenteEsBlanco()
 - frenteEsNegro()
 - frenteEsObstaculo()
 - frenteEsBaliza()



<https://www.robomind.net/es/docBasicInstructions.htm>

Robomind

Sensores

- Sensores del robot:
 - derechaEsClaro()
 - derechaEsBlanco()
 - derechaEsNegro()
 - derechaEsObstaculo()
 - derechaEsBaliza()



<https://www.robomind.net/es/docBasicInstructions.htm>

Robomind

Sensores

- Sensores del robot:
 - izquierdaEsClaro()
 - izquierdaEsBlanco()
 - izquierdaEsNegro()
 - izquierdaEsObstaculo()
 - izquierdaEsBaliza()



<https://www.robomind.net/es/docBasicInstructions.htm>

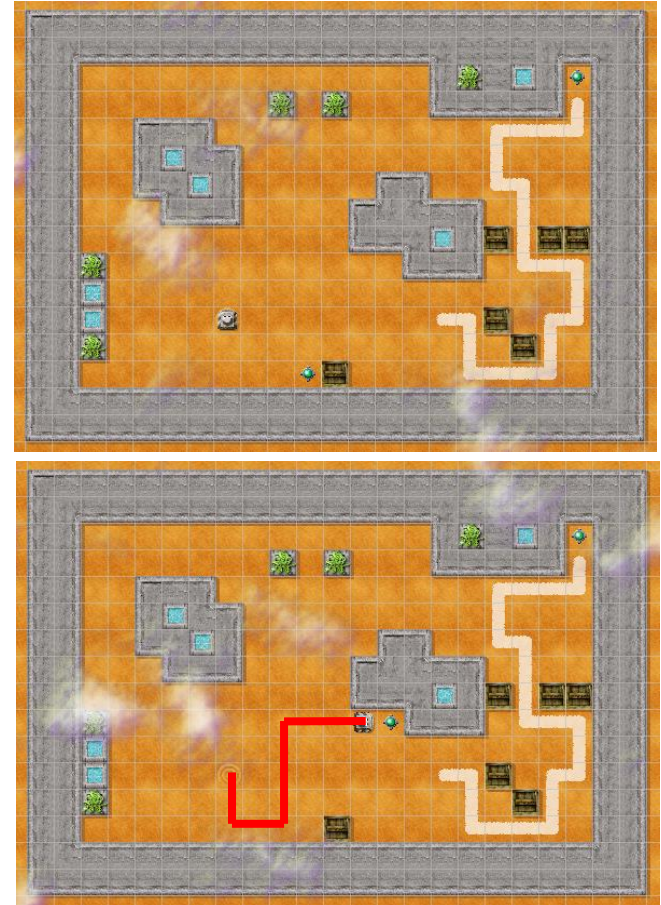
Robomind

Acciones con las balizas

- Acciones del robot:

- tomar()
- poner()
- come()

atras(2)
derecha()
adelante(2)
tomar()
izquierda()
adelante(4)
derecha()
adelante(3)
poner()



Robomind

Condicional simple (I)

```
#Se mueve hacia adelante  
#si la casilla que hay delante no es un obstáculo  
si (frenteEsClaro()){  
    adelante(1)  
}
```



<https://www.robomind.net/es/docProgrammingStructures.htm#als>

Robomind

Condicional simple (II)

```
# Se mueve hacia adelante  
# si la casilla que hay delante no es un obstáculo,  
# sino se mueve hacia atrás  
si(frenteEsClaro()){  
    adelante(1)  
}  
otro{  
    atras(1)  
}
```



<https://www.robomind.net/es/docProgrammingStructures.htm#als>

Robomind

Condicional múltiple (I)

```
si(frenteEsBaliza()){  
    tomar()  
}  
otro si(frenteEsObstaculo()){  
    derecha()  
}  
otro si(frenteEsBlanco ()){  
    izquierda(1)  
}  
otro{  
    adelante(1)  
}
```



<https://www.robomind.net/es/docProgrammingStructures.htm#als>

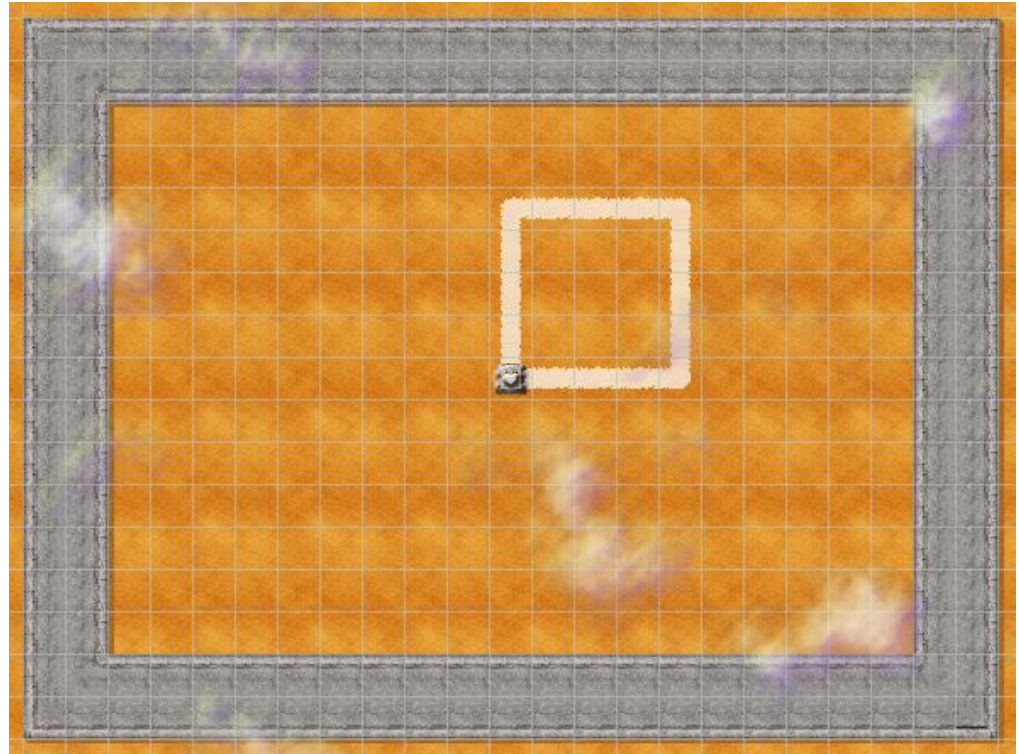
Robomind

Bucle repetir (I)

#El bucle se ejecuta un determinado número de veces

```
repetir(5){  
    adelante(1)  
}
```

```
pintarBlanco()  
repetir(4){  
    adelante(4)  
    derecha()  
}  
detenerPintar()
```



<https://www.robomind.net/es/docProgrammingStructures.htm#herhalingen>

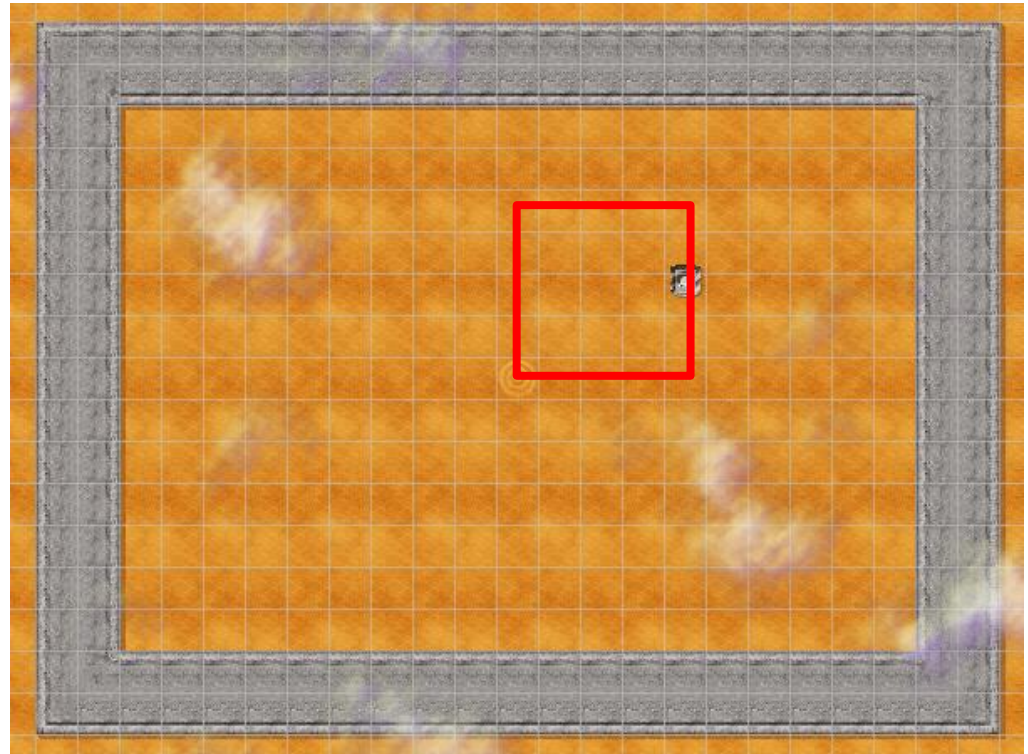
Robomind

Bucle repetir (II)

#El bucle se ejecuta indefinidamente

```
repetir(){  
    adelante(1)  
}
```

```
repetir(){  
    adelante(4)  
    derecha()  
}
```



<https://www.robomind.net/es/docProgrammingStructures.htm#herhalingen>

Robomind

Bucle repetir (III)

```
#El bucle se ejecuta mientras se cumple una condición  
repetirMientras(frenteEsClaro()){  
    adelante(1)  
}
```



Programación

Ejercicios

Ejercicio 1:

Abrir el mapa *openArea.map* y escribir un programa que permita al robot recorrer el perímetro de manera indefinida.

Ejercicio 2:

Abrir el mapa *openArea.map* y escribir un programa que permita al robot recorrer el perímetro dibujando una línea blanca.

Ejercicio 3:

Abrir el mapa *openArea.map* y escribir un programa que permita al robot recorrer el perímetro dibujando una línea blanca en las vueltas impares y negra en las pares.


Robomind

Romper un bucle

#Detener el programa
truncar

Ejemplo:

```
repetir(){  
    adelante(1)  
    si (frenteEsBaliza()){  
        coger()  
        truncar  
    }  
}
```



La instrucción **truncar** permite salir de un bucle cuando se cumple una determinada condición

<https://www.robomind.net/es/docProgrammingStructures.htm#herhalingen>

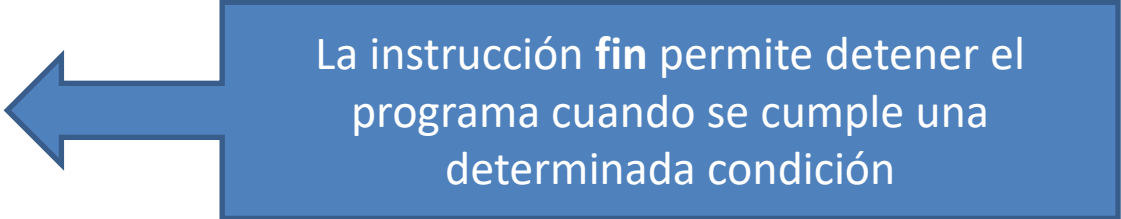
Robomind

Detener el programa

#Esta instrucción hace que todo el programa se detenga
fin

Ejemplo:

```
repetir(){  
    adelante(1)  
    si (frenteEsBaliza()){  
        coger()  
        fin  
    }  
}
```



La instrucción **fin** permite detener el programa cuando se cumple una determinada condición

<https://www.robomind.net/es/docProgrammingStructures.htm#einde>

Robomind

Seguir línea

```
derecha()
adelante(7)
repetir(){
    si(frenteEsBlanco()){
        adelante(1)
    }
    otro si(derechaEsBlanco()){
        derecha()
        adelante(1)
    }
    otro si(izquierdaEsBlanco()){
        izquierda()
        adelante(1)
    }
    otro{
        fin
    }
}
```



<https://www.robomind.net/es/docProgrammingStructures.htm#als>

Programación

Ejercicios

Ejercicio 1:

Escribir un programa que permita al robot seguir una línea blanca y coger la baliza.

Ejercicio 2:

Escribir un programa que permita al robot seguir una línea negra y coger la baliza.

Ejercicio 3:

Escribir un programa que permita al robot seguir una línea blanca, coger la baliza y volver siguiendo una línea negra.

Robomind

Lanzar moneda

```
#Lanzar moneda  
lanzarMoneda()
```

Ejemplo:

```
repetir(){  
    adelante(1)  
    si (lanzarMoneda()){  
        derecha()  
    }  
    otro{  
        izquierda()  
    }  
}
```



<https://www.robomind.net/es/docBasicInstructions.htm>

Programación

Ejercicios

Ejercicio 1:

Abrir el mapa *openArea.map* y escribir un programa que mueva el robot cinco casillas a la derecha o a la izquierda en función del resultado de lanzarMoneda.

Ejercicio 2:

Abrir el mapa *openArea.map* y escribir un programa que dibuje un cuadrado blanco o negro en función del resultado de lanzarMoneda.

Ejercicio 3:

Abrir el mapa *openArea.map* y escribir un programa que permita al robot moverse aleatoriamente por el mapa de manera indefinida.

Robomind

Expresiones lógicas

- Operadores lógicos:

- no (~)
- y (&)
- o (|)

```
repetirMientras (no frenteEsObstaculo()){  
    adelante(1)  
}
```



- Ejemplos:

- no frenteEsObstaculo()
- frenteEsObstaculo() y derechaEsObstaculo()
- izquierdaEsClaro() o derechaEsClaro()
- frenteEsObstaculo y (izquierdaEsClaro() o derechaEsClaro())

<https://www.robomind.net/es/docProgrammingStructures.htm#logic>

Robomind

Procedimientos

Llamada al procedimiento 'rectangulo'

```
rectangulo(3, 2)
```

```
adelante(3)
```

```
rectangulo(2, 3)
```

Procedimiento para dibujar un rectángulo

```
procedimiento rectangulo(anchura, altura)
```

```
{
```

```
    pintarBlanco()
```

```
    repetir(2)
```

```
{
```

```
        adelante(altura)
```

```
        derecha()
```

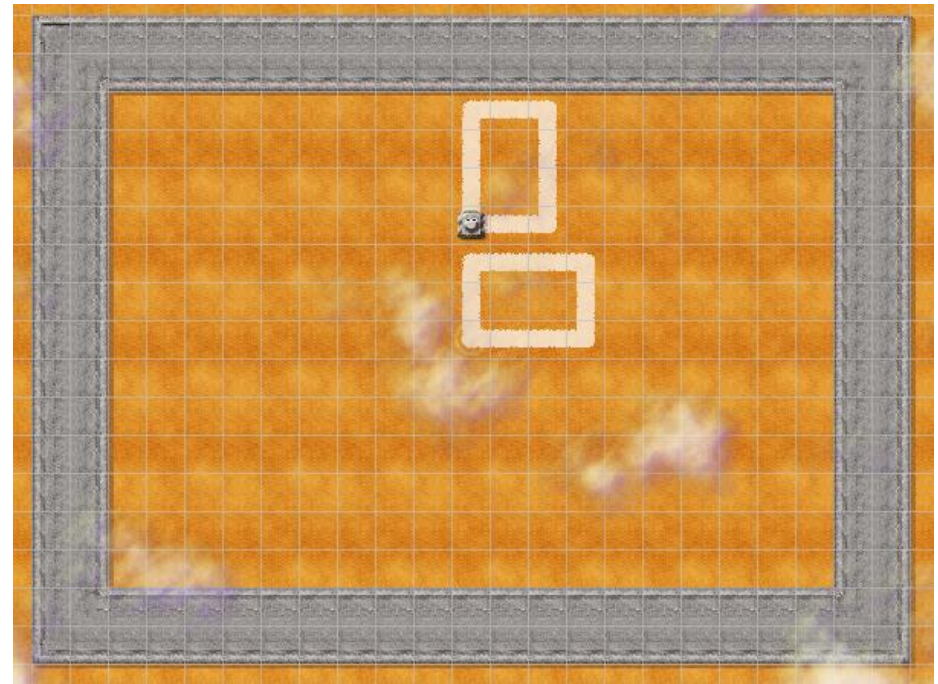
```
        adelante(anchura)
```

```
        derecha()
```

```
}
```

```
    detenerPintar()
```

```
}
```



<https://www.robomind.net/es/docProgrammingStructures.htm#procs>

Robomind

Equivalencia bucles repetir y repetirMientras

```
repetir(){  
    si (frenteEsObstaculo())  
        truncar  
}  
otro {  
    adelante(1)  
}  
}
```

```
repetirMientras(no frenteEsObstaculo())  
    adelante(1)  
}
```



Robomind

Fin

Programación

Introducción a la programación
(Robomindn)