



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA DE  
VALPARAÍSO

[pucv.cl](http://pucv.cl)

# Robótica e inteligencia artificial

Módulo 2  
Robótica Móvil S12

# ROBÓTICA MÓVIL

## SESIÓN 12

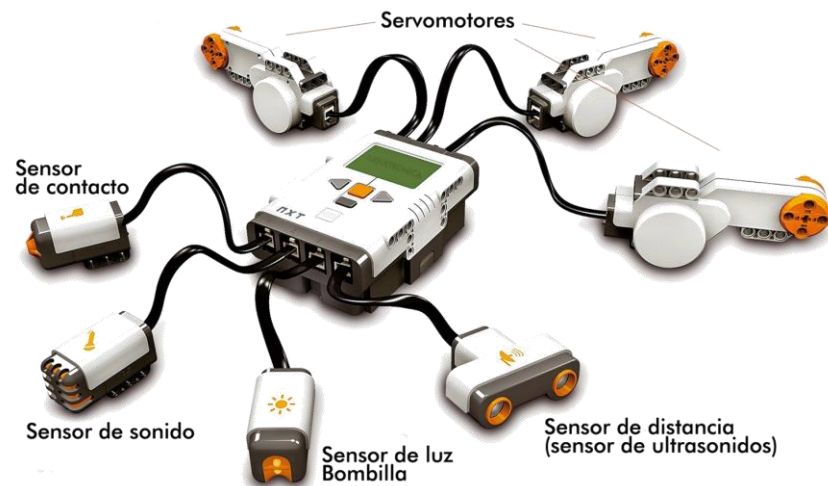
# Robótica Móvil

Para que la robótica pueda cumplir su cometido de sustituir al hombre en labores repetitivas o forzosas, es necesario que la máquina robótica tenga la capacidad de tomar decisiones en casos de corrección o ajuste de su trabajo.

Para realizar esta actividad de decisión la máquina debe ser capaz de percibir los estímulos del entorno que le indica variables de interés.

Por ejemplo :

- Un robot puede detectar una pared, por medio de un sensor de distancia, para decidir detenerse, girar y continuar avanzando.
- Un robot puede detectar un precipicio, para decidir detenerse y no caer.
- Un robot puede detectar variables (temperatura, humedad, pH, calidad de aire, etc.) para luego informarlas a un operador.



## Percepción del entorno

Los robots poli articulados Kuka en algunos casos poseen sensores de proximidad para detectar si un humano está muy cerca del área de trabajo del robot, deteniendo su movimiento para prevenir un accidente.



# Percepción del entorno

Los robots transportadores de Amazon movilizan estanterías con gran cantidad de productos. Realizando un trabajo rápido, fluido y sin retrasos.

Para realizar esta tarea, los robots tienen sensores de líneas magnéticas para seguir las trayectorias necesarias para distribuir los productos.

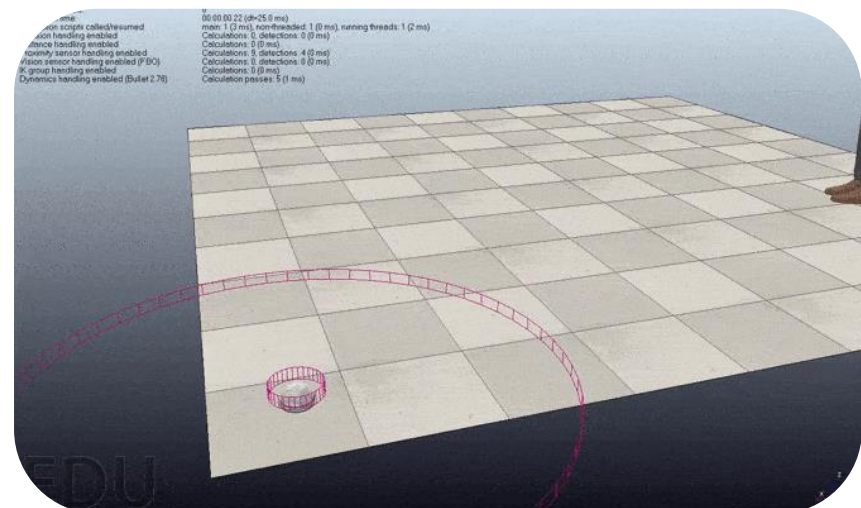
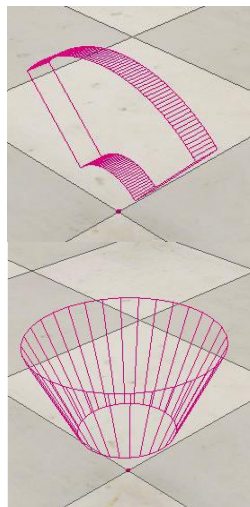
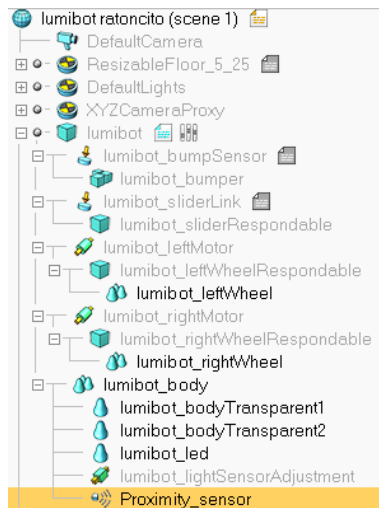
Además, también cuentan con sensores de proximidad para detenerse en caso de que un humano se cruce en su camino.



# Percepción del entorno

CoppeliaSim posee una carta de sensores de proximidad que permiten reconocer el entorno respecto a la proximidad de este al robot. Los sensores de proximidad varían en su forma de detección, encontrando conos, discos, cilindros, línea directa, entre otros. Además es posible modificar los volúmenes y parámetros de detección de estos, de esta manera es posible ajustar este tipo de sensor de detección virtual para que se asemeje a un sensor de detección real.

Luego de insertar un elemento “Sensor” en la estructura del robot, es posible tomar decisiones respecto a estos valores de lectura.





# Percepción del entorno

Al igual que en otros lenguajes de programación, el comando “IF” permite establecer una decisión respecto a parámetros de lectura, para posteriormente ejecutar una acción u otra

```
while true do
lectura()|

    if (distancia<1) then --Si detecta algo

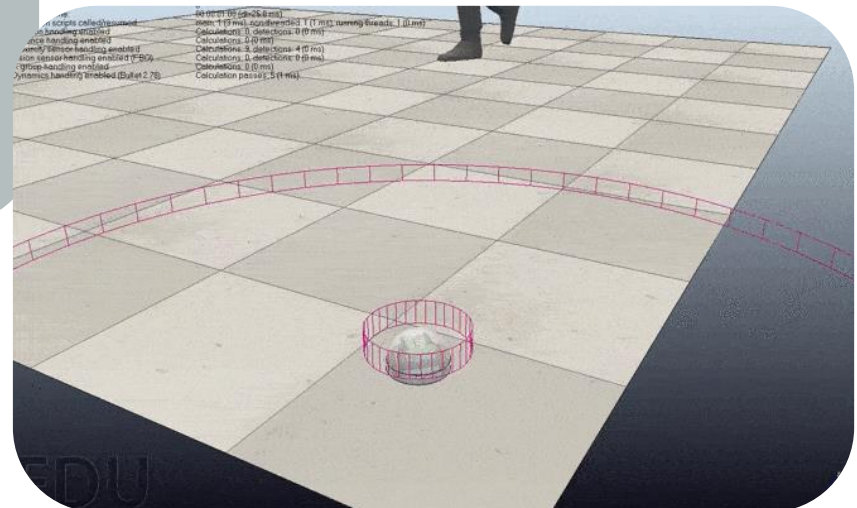
        sim.setJointTargetVelocity(motorizquierdo,0) --Parar
        sim.setJointTargetVelocity(motorderecho, 0)
        sim.wait(1)

    else --No se detecta nada

        sim.setJointTargetVelocity(motorizquierdo,3) --Avanzar
        sim.setJointTargetVelocity(motorderecho, 3)
        sim.wait(1)

    end

end
```



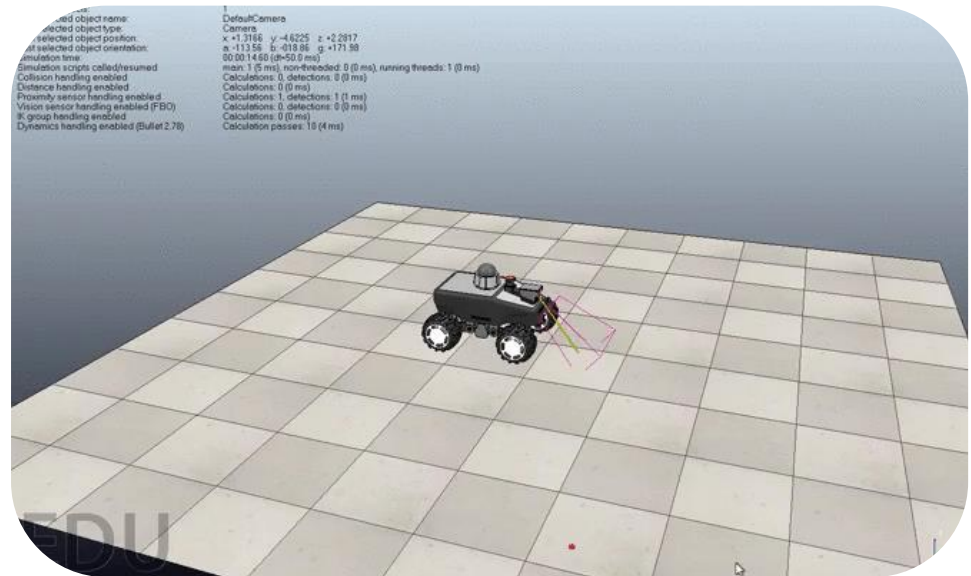
# Actividad de exploración

Con la ayuda de un sensor de distancia que apunta al suelo, es posible detectar cuando hay o no superficie.

Se requiere realizar un código que permita al robot explorar la plataforma sin caer de esta.

La función de lectura utiliza comandos de LUA más elaborados que no han sido vistos aún.

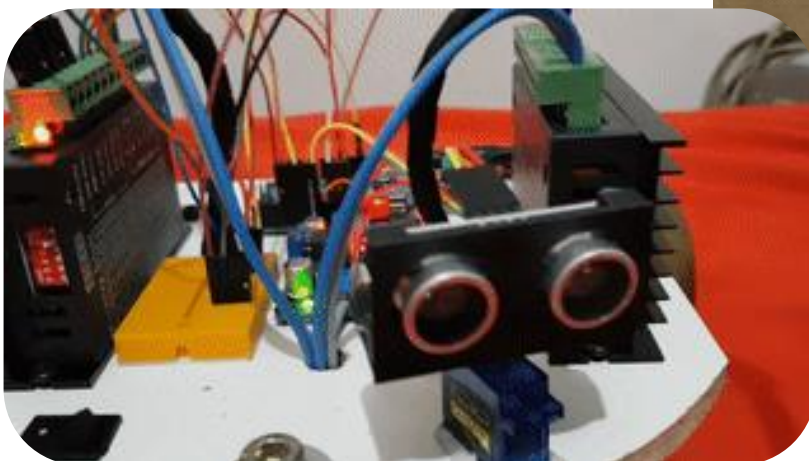
En las siguientes clases se analizará la estructura de esta programación





# Exploración libre

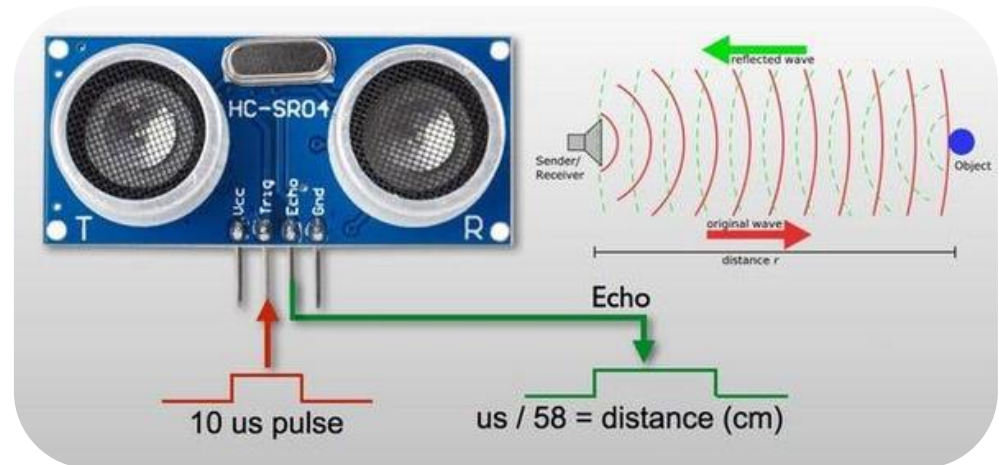
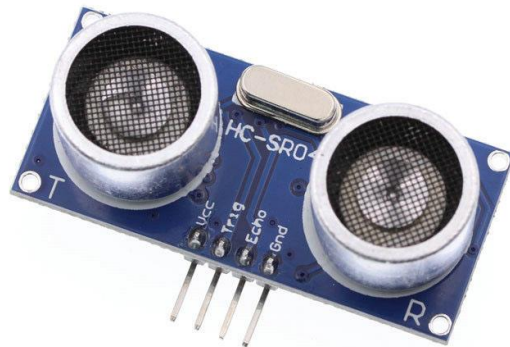
Un robot móvil que se encuentre dotado de sensores puede realizar una exploración libre evadiendo obstáculos y realizando movimientos, quién permita que en un periodo de tiempo el área de operación sea recorrida o explorada.



# Sensores de proximidad

Los sensores de proximidad no solo se utilizan para la evasión de obstáculos, sino también para la detección de objetos a manipular o trabajar. Dependiendo del tipo de objeto se utilizarán distintos tipos de detección.

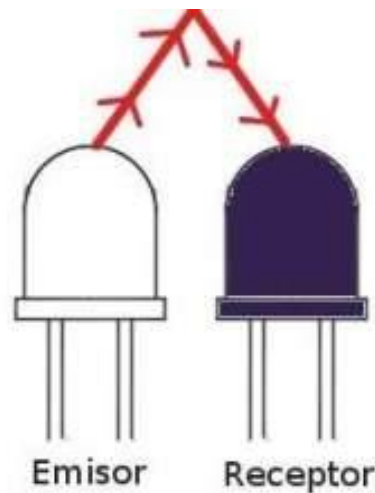
Uno de los sensores más usados en la robótica educativa o de investigación es el sensor de distancia por ultrasonido HCSR04. El cual calcula la distancia a un objeto haciendo uso de una onda ultrasonido emitida y luego recibida (rebote por eco) midiendo el tiempo que tarda el sonido en viajar y volver al sensor.



# Sensores de proximidad

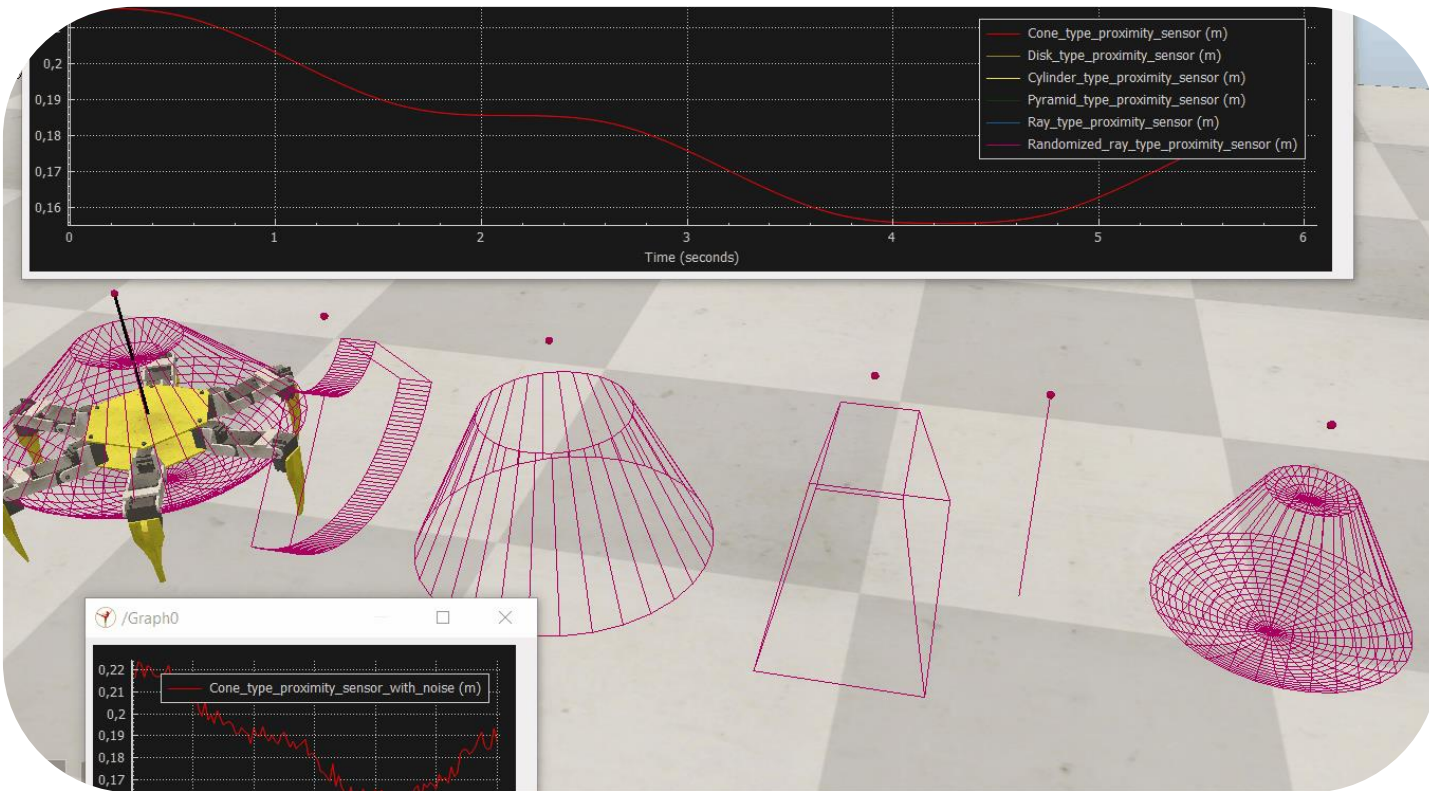
Otro tipo de sensor de proximidad o distancia es a través del principio de reflexión de luz. Comúnmente es utilizado el espectro de luz infrarroja, ya que esa no es visible y por lo tanto posee menos tendencia a que el estímulo sea contaminado con otras fuentes de luz.

Pues el sensor funciona por principios de reflexión de luz es importante identificar que el objeto a detectar debe tener la capacidad de reflejar gran parte de la luz, ya que si la señal de luz recibida posee menos potencia, esta puede no ser detectada.



# Sensores de proximidad

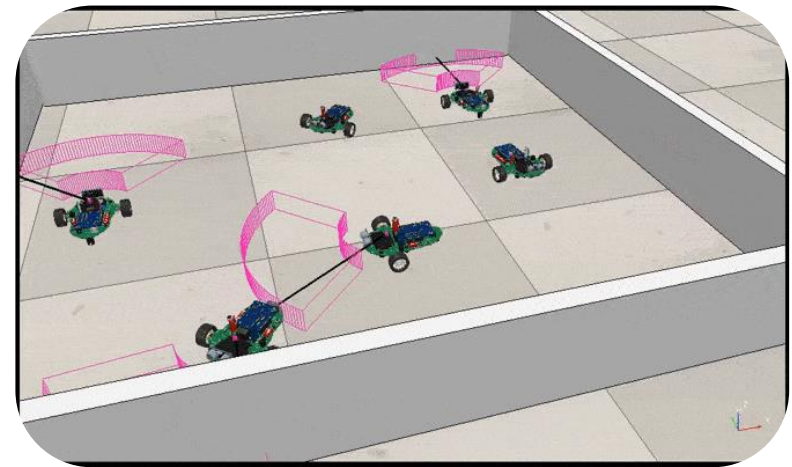
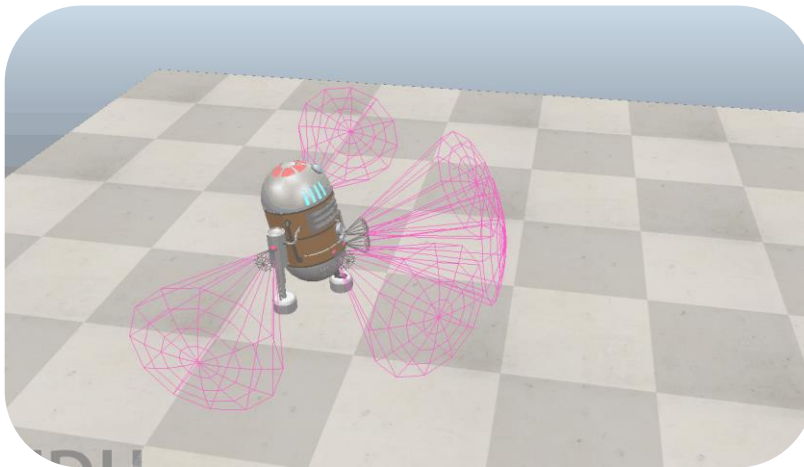
En CoppeliaSim Los distintos tipos de sensores de proximidad tendrán distintos funcionamientos tal como se muestra en la siguiente imagen:



# Sensores de proximidad

En CoppeliaSim Se puede agregar una gran cantidad de sensores de proximidad al objeto robótico de tal manera de preparar a la máquina para un determinado objetivo.

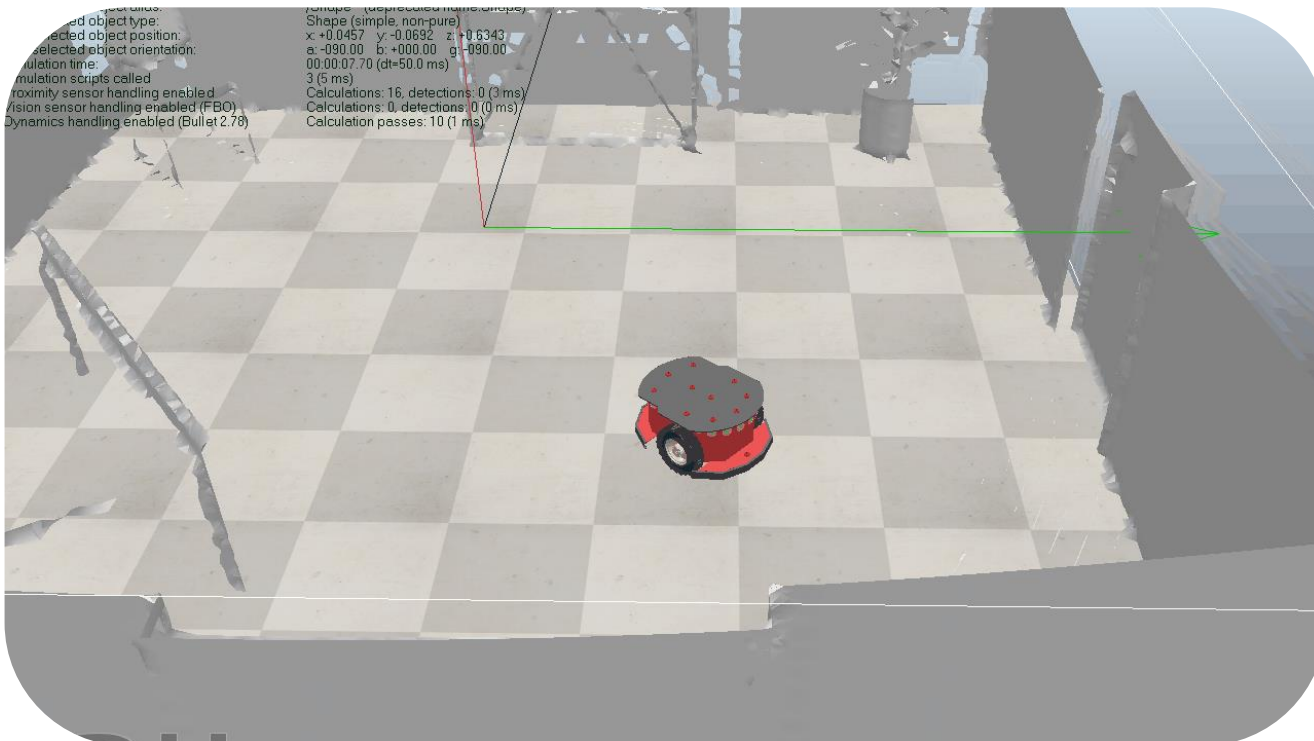
Podemos encontrar distintos tipos de robots de navegación que ya cuentan una gran cantidad de sensores montados, mientras que otros robots Pueden utilizar una menor cantidad de sensores a costa de un sistema de navegación y detección de obstáculos más precario.





# Navegación y evasión de obstáculos

El propósito de algunos robots si basa directamente en la navegación y la detección de obstáculos, generando rutas que impidan el choque con estos. Mientras el robot posea mayor cantidad de sensores, las decisiones programadas en el código deberán contar con mayor cantidad de variables a considerar.





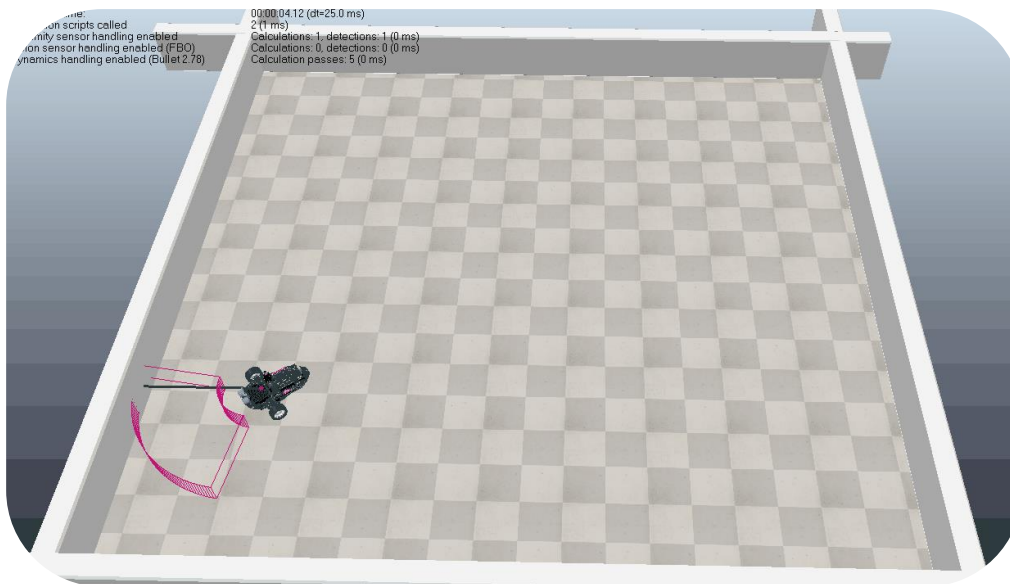
# Actividad de aplicación

El objetivo de esta actividad es implementar un código que haga uso de la movilidad de un robot diferencial de 2 ruedas dotado de un sensor de proximidad que apunta hacia adelante.

Implementar técnicas para que el robot pueda realizar una exploración por el escenario con la menor cantidad de choques posibles.

Necesariamente la exploración debe ser con direcciones al azar, es decir, que la simulación debe Generar una trayectoria diferente cada vez que se inicia una nueva simulación.

Puede utilizar el siguiente comando para generar números aleatorios en LUA:



```
aux=math.random(1,10)
```

# Actividad de aplicación

La programación del bucle puede ser bastante sencilla, pero para lograr el cumplimiento del objetivo se recomienda aplicar técnicas vistas anteriormente, como procesos de aceleración o precisión con valores de potencias de actuadores bajas.



# Clave de la actividad

## Actividad de aplicación

```
while true do

    lectura()

    if (distancia<0.5) then --Caso si se detecta un obstaculo

        sim.setJointTargetVelocity(motorRight,0) --stop
        sim.setJointTargetVelocity(motorLeft,0)
        sim.wait(1);

        aux=math.random(1,10)

        sim.setJointTargetVelocity(motorRight,4*(-1)^aux) --movimiento random
        sim.setJointTargetVelocity(motorLeft,4*(-1)^aux)
        sim.wait(1);

    else --caso donde no se detecta nada

        sim.setJointTargetVelocity(motorRight,-10) --adelante
        sim.setJointTargetVelocity(motorLeft,10)
        sim.wait(0.1);

end
```