



SweeperBot

RECOLECCIÓN Y RECICLAJE DE RESIDUOS

PROJECT SPRINT #5.

FECHA: 25 Mayo 2021

Noel Caparrós Burgos
Jordi Adria Campoy Callen
Carlos López Arroyo

Tabla de Contenidos

Descripción del proyecto	1
Componentes Electrónicos	2
Esquema Hardware	3
Arquitectura Software	4
Reconocimiento de forma	5
Cinemática inversa	6
Aportaciones	7
Componentes extra y piezas 3D	7
Brazo mecánico	8
Carrocería	10
Estrategia de Simulación	14
Cinemática inversa	15
Primera simulación	15
Segunda simulación	16
Movimiento del vehículo	17
Primera simulación	17
Segunda simulación	18
Reconocimiento de objetos	19
SweeperBot	

Primera simulación	19
Simulación en CoppeliaSim	20
Texturas	20
Consideraciones	22
Riesgos previstos y plan de contingencia	23

SweeperBot

SweeperBot

RECOLECCIÓN Y RECICLAJE DE RESIDUOS

Descripción del proyecto

Este documento contiene la información necesaria para presentar SweeperBot, un robot autónomo diseñado para ayudar en el ámbito medioambiental, y que es el producto final de una idea que garantiza una mayor sostenibilidad del entorno que nos rodea.

Para llevar a cabo su cometido, el robot se contempla como un vehículo autónomo, de manera que pueda moverse libremente y sin ningún tipo de supervisión para la búsqueda de residuos. Para la recolección de estos, el robot dispone de un brazo mecánico con pinza, esta le permitirá agarrar los desechos y soltarlos en unos compartimentos a modo de reciclaje.

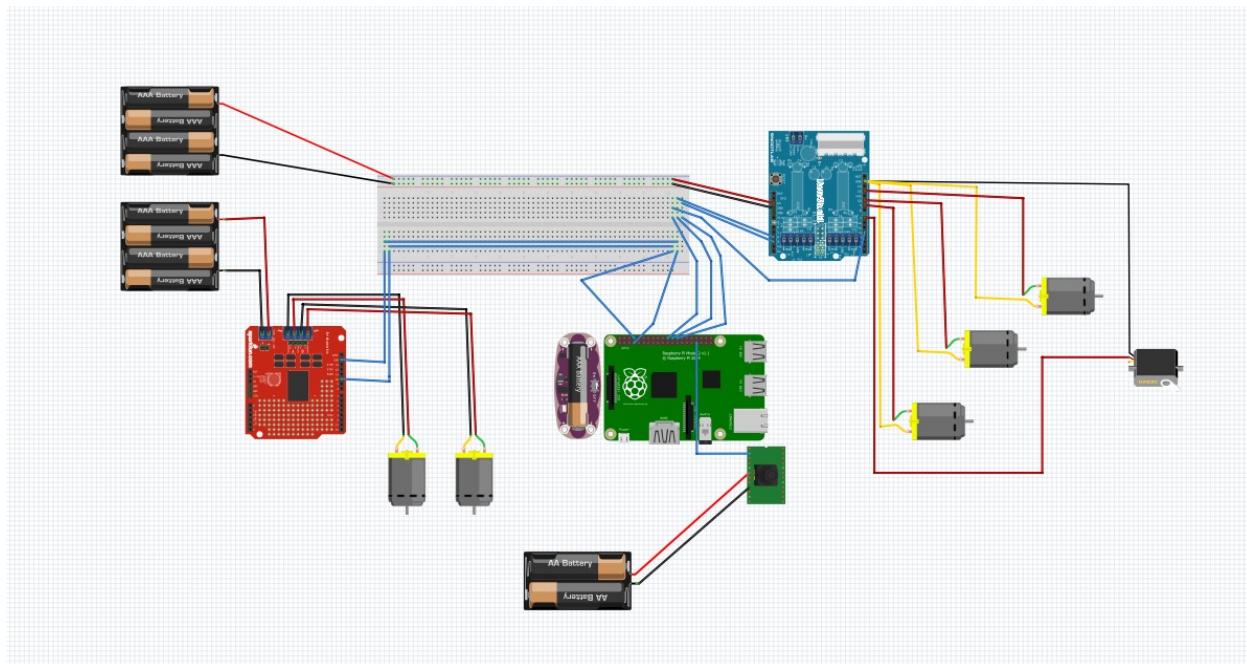
A lo largo del presente documento se definen los componentes electrónicos y estructurales del robot, se muestran diagramas de la arquitectura hardware y software de su diseño, y se contemplan cuales son los riesgos del proyecto y su respectivo plan de contingencia.

Componentes Electrónicos

A continuación se listan los componentes electrónicos:

- *Raspberry Pi 3 B+*
- *Cámara Raspberry*
- *Servomotor*
- *12 Pilas 1.5 V*
- *Base de pilas*
- *Controlador de motores (L298N)*
- *3 Motores high torque*
- *Placa i2c*
- *Powerbank 5V*
- *Chasis del Robot*

Esquema Hardware



Movimiento del vehículo:

Se han conectado los dos motores de las ruedas al controlador L298 junto con la corriente de las 4 pilas de 1,5 Voltios, este controlador actúa de intermediario entre la Raspberry y los movimientos del vehículo. El controlador se ha conectado también a la Raspberry Pi.

Cinemática inversa del brazo:

Los 3 motores utilizados para mover el brazo en el plano de 3 ejes, y el servomotor que controla la apertura de la pinza se han conectado a la placa i2c, esta facilita la comunicación entre la raspberry y los componentes. La placa i2c está conectada a 4 pilas de 1,5 Voltios y a la Raspberry Pi.

Reconocimiento de objetos:

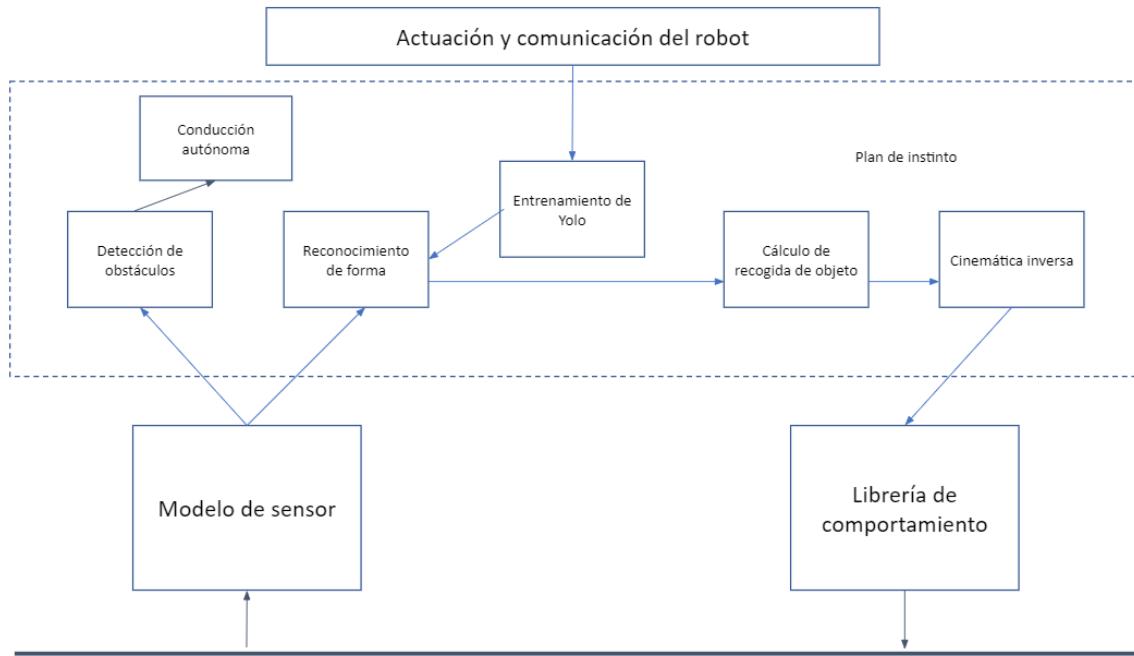
La cámara es alimentada por dos pilas de 1,5 Voltios y controlada directamente por la Raspberry.

Arquitectura Software

Las funcionalidades principales a desarrollar son las siguientes:

- *Conducción autónoma*
- *Detección de objetos*
- *Reconocimiento de forma mediante red neuronal (Yolo)*
- *Cálculo de recogida de objeto*
- *Cinemática inversa del brazo*

La arquitectura propia de nuestro sweeper-bot se puede ver en el siguiente esquema:



Reconocimiento de forma

Se ha utilizado el lenguaje Python para realizar el reconocimiento e identificación de formas, y se ha importado la librería “opencv” para obtener imágenes de la cámara. Y la librería “imageai” que utiliza “tensorflow”, una librería orientada a construir y entrenar redes neuronales, para utilizar un modelo de reconocimiento de objetos (Yolo).

En cuanto a los datos resultantes, se obtiene una lista con los objetos, la probabilidad de acierto en la predicción, y cuales son las coordenadas del objeto en la imagen.

```
[{'name': 'person', 'percentage_probability': 94.99484896659851, 'box_points': [0, 267, 502, 732]}, {'name': 'apple', 'percentage_probability': 91.81467890739441, 'box_points': [328, 235, 464, 393]}]
[{'name': 'person', 'percentage_probability': 99.44558143615723, 'box_points': [1, 246, 491, 711]}, {'name': 'apple', 'percentage_probability': 98.51739406585693, 'box_points': [311, 202, 466, 365]}]
[{'name': 'person', 'percentage_probability': 99.52298998832703, 'box_points': [0, 204, 495, 718]}, {'name': 'apple', 'percentage_probability': 90.68878889083862, 'box_points': [316, 178, 461, 345]}]
```

Cinemática inversa

El brazo de SweeperBot consta de un manipulador antropomórfico de tres ejes rotativos. Por lo tanto, su matriz Denavit-Hartenberg es la siguiente:

	θ	d	a	α
1	θ_1	L1	0	90°
2	θ_2	0	L2	0
3	θ_3	0	L3	0

Para llevar a cabo este proceso, se ha elaborado un fichero en lenguaje python. Mediante los importes de numpy y sysmpy, se ha podido transformar la matriz D-H para la cinemática del brazo encargado de la recolección de residuos. En la figura siguiente, se puede ver el resultado:

$$\begin{bmatrix} \cos(\theta_2 + \theta_3) \cos(\theta_1) & -\sin(\theta_2 + \theta_3) \cos(\theta_1) & 1.0 \sin(\theta_1) & (d_2 \cos(\theta_2) + d_3 \cos(\theta_2 + \theta_3)) \cos(\theta_1) \\ \sin(\theta_1) \cos(\theta_2 + \theta_3) & -\sin(\theta_2 + \theta_3) \sin(\theta_1) & -1.0 \cos(\theta_1) & (d_2 \cos(\theta_2) + d_3 \cos(\theta_2 + \theta_3)) \sin(\theta_1) \\ 1.0 \sin(\theta_2 + \theta_3) & 1.0 \cos(\theta_2 + \theta_3) & 0 & 1.0d_1 + 1.0d_2 \sin(\theta_2) + 1.0d_3 \sin(\theta_2 + \theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Las ecuaciones se encuentran en la cuarta columna y podemos obtener los ángulos pertinentes mediante su resolución.

Las distancias del brazo son las siguientes:

- $d_1=3\text{cm}$
- $d_2=14\text{cm}$
- $d_3=14\text{cm}$

Aportaciones

Cada año que pasa, el planeta se va deteriorando por una serie de factores incontables, como la falta de mantenimiento, la ineficiencia de los sistemas productivos, actos incívicos, obsolescencia programada... Evitar esta degradación es uno de los problemas principales a los que se enfrentará la especie humana los próximos años.

La explícita retirada de residuos es una actividad con un impulso muy pobre por parte de grandes organizaciones y entidades gubernamentales. Aunque parte de las empresas con mayor capacidad productiva gestionan parte de los residuos que generan, esta acción asemejará insignificante comparada a la ola de desechos que las próximas generaciones tendrá que batallar.

Para ayudar en esta tediosa tarea se ha presentado una solución, un robot que desempeñe el papel de “barrendero”, retirando los residuos del área pública y clasificándolos según su tipo. Mejorar el estado de la vía pública se considera la finalidad principal de este proyecto, aunque de esta derivan una serie de objetivos.

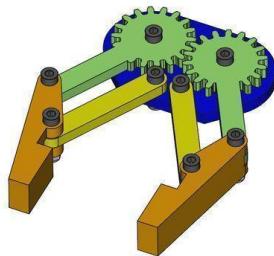
- *Concienciar a la población, haciendo visual y plausible el problema mediante la exposición del robot en las calles.*
- *Aumentar indirectamente el turismo, junto a otros posibles beneficios para la zona, gracias a la mejora de la vía pública.*
- *Reducir el deterioro, contaminación y daños en el entorno.*
- *Decrementar el coste de mantenimiento de la vía pública.*
- *Favorecer a la fauna, flora y habitantes del lugar, reduciendo el impacto residual.*

Componentes extra y piezas 3D

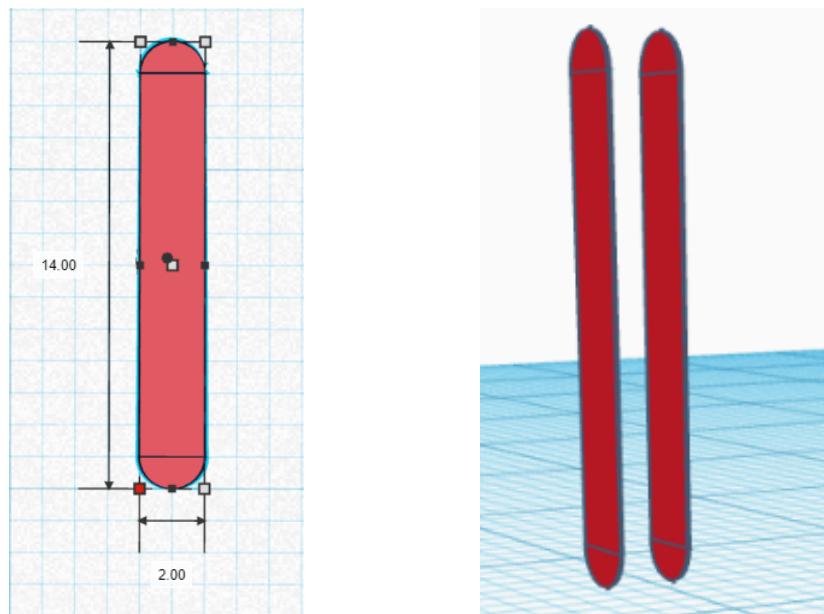
A continuación se listan los componentes no electrónicos del robot, se concretan sus funcionalidades, y se muestran las dimensiones de las piezas que los componen.

Brazo mecánico

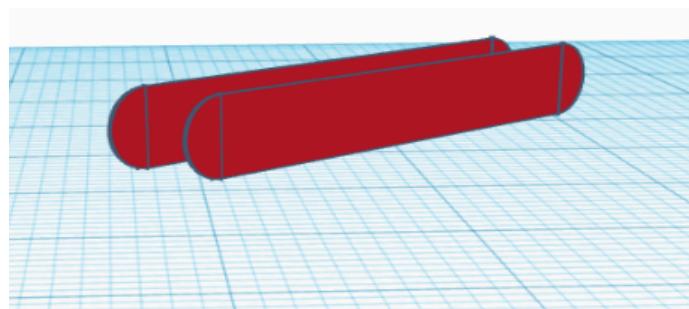
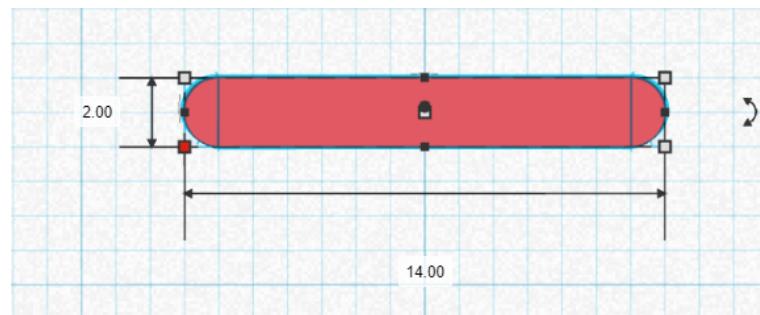
- **Pinza del Robot :** La pinza del robot es un conjunto de piezas impresas en 3D que son movidas por un motor con el fin de agarrar objetos.



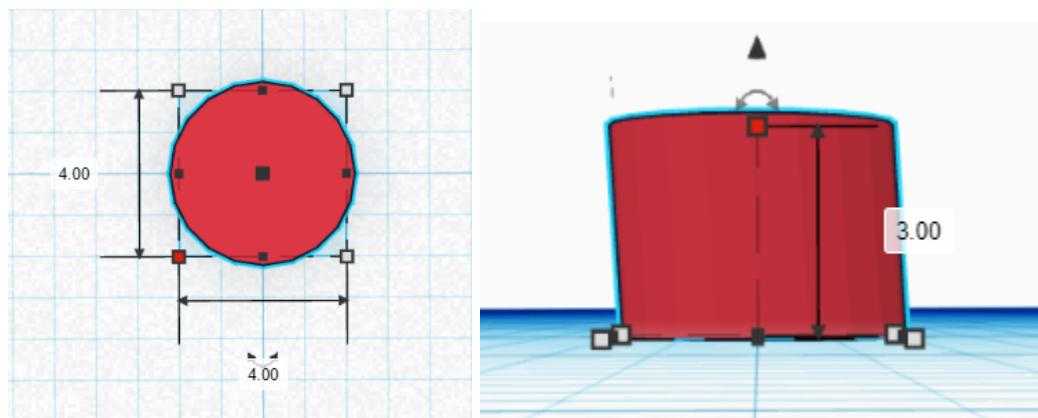
- **Primera articulación (brazo):** La primera articulación del robot es un conjunto de piezas impresas en 3D que realizan un desplazamiento vertical y conectan la base con la segunda articulación.



- **Segunda articulación (brazo):** La segunda articulación del robot es un conjunto de piezas impresas en 3D que realiza un desplazamiento vertical, y conecta la primera articulación con la pinza.



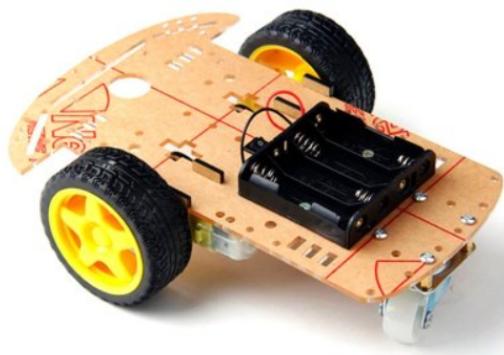
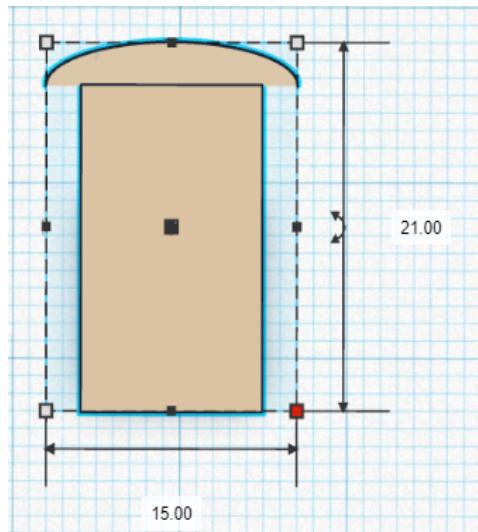
- **Base del brazo:** La base del brazo es una pieza 3D colocada en la parte superior del vehículo, esta base conecta la primera articulación con la carrocería del vehículo y realiza un movimiento de giro horizontal.



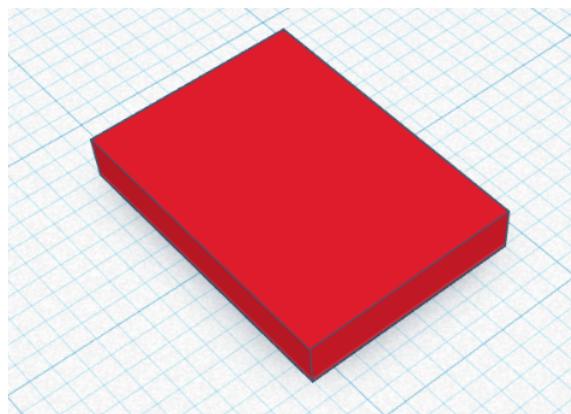
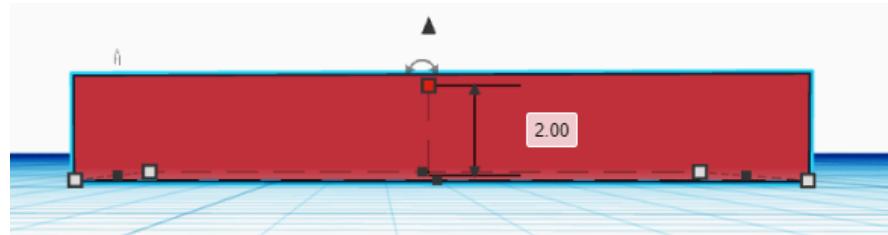
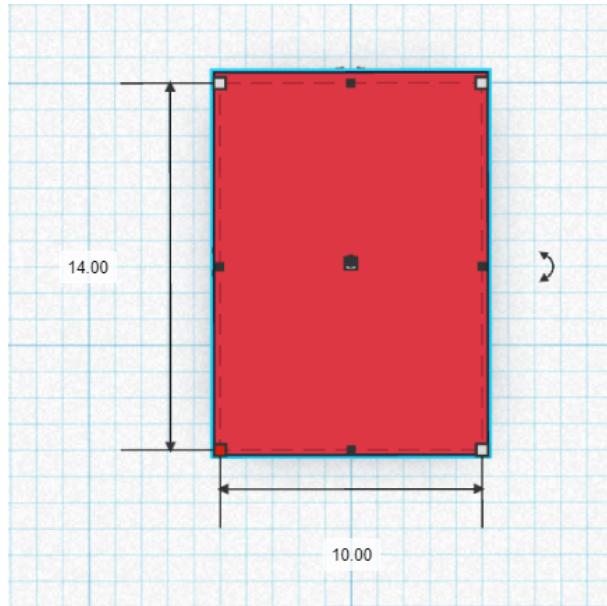
Carrocería

La carrocería es una serie de piezas 3D que conforman la estructura del robot, éstas son el soporte de los componentes electrónicos y del brazo mecánico. Además, contiene diferentes compartimentos donde el robot deposita los residuos encontrados.

- **Base de la carrocería:** La base de la carrocería es la pieza 3D que sostiene la totalidad del vehículo. Se ha diseñado en función de las características que ofrece la página web en la que se va a comprar la pieza.

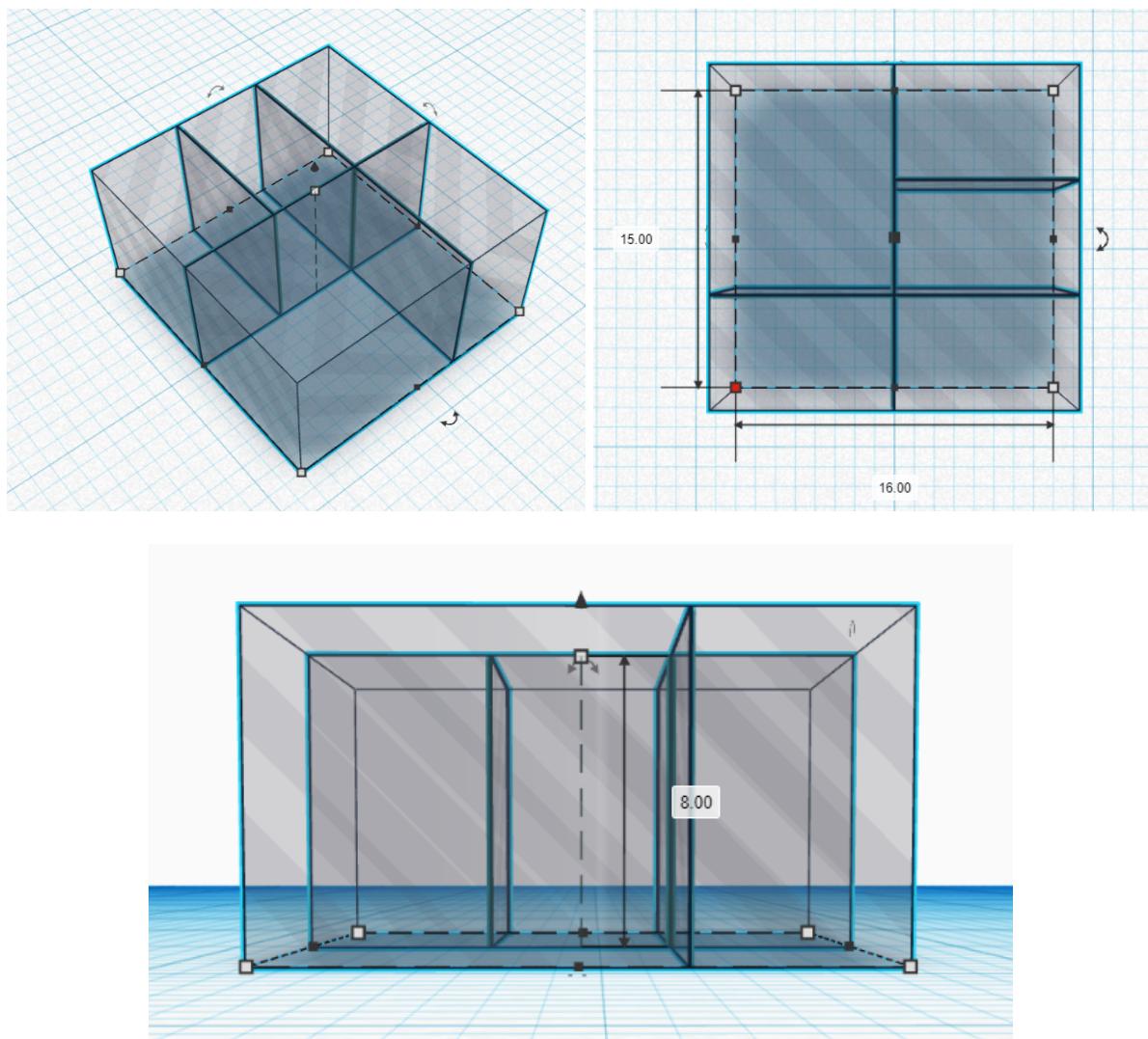


- **Soporte para los compartimentos:** El soporte es una pieza 3D que eleva a la estructura de compartimentos para que no estén en contacto con las ruedas.

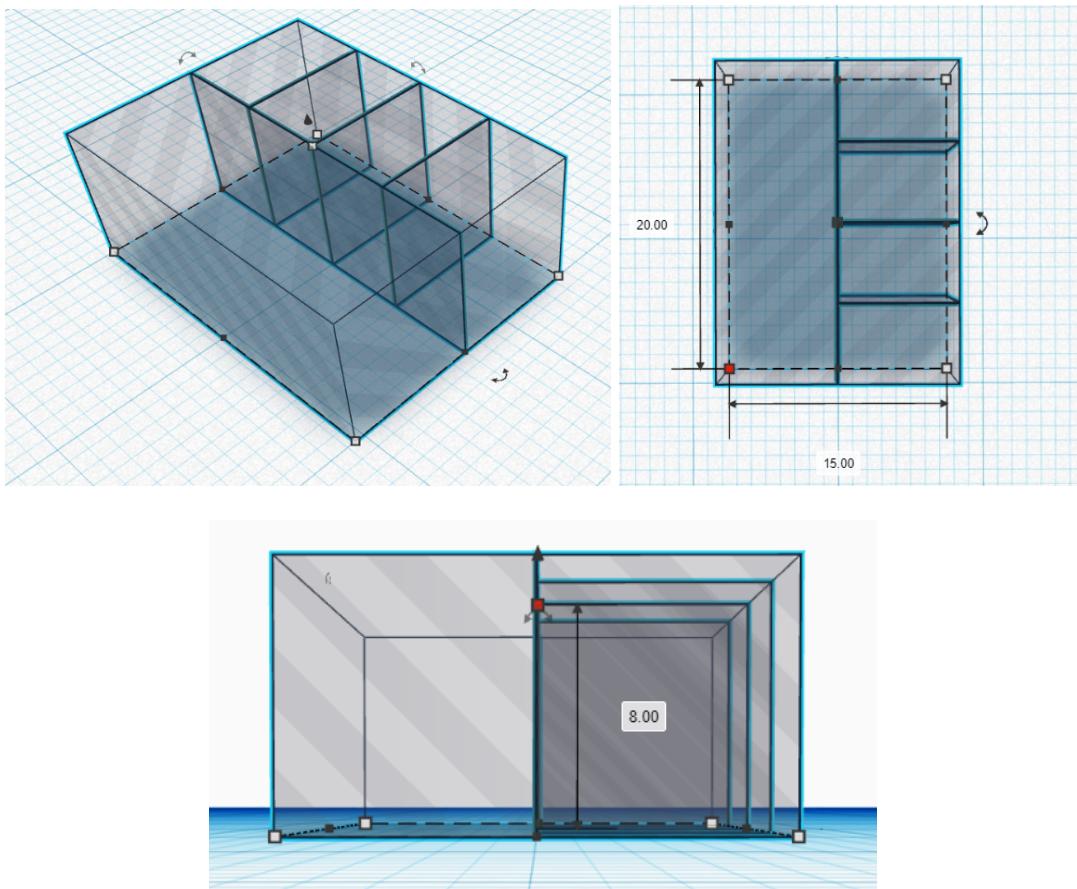


- **Compartimentos:** Los compartimentos son una serie de piezas 3D que guardan los objetos recogidos por el robot. Los cuatro compartimentos pequeños corresponden a los residuos que el robot encuentra mientras que el compartimento grande sirve para proteger a los componentes electrónicos. Se han diseñado dos modelos de la estructura y ambos son compatibles con el resto de piezas.

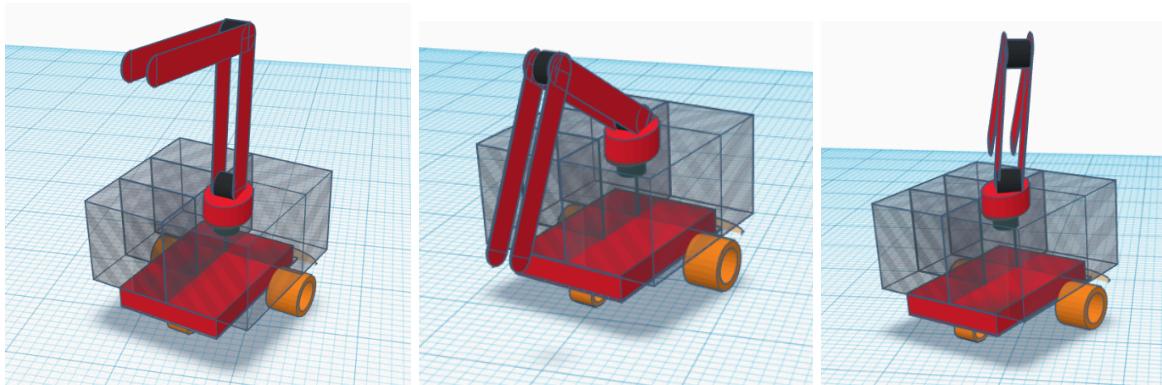
Versión 1:



Versión 2:



A continuación se muestran los diseños del robot tras el montaje.



Estrategia de Simulación

Para obtener un escenario virtual sobre las acciones que puede llevar a cabo SweeperBot, se ha decidido utilizar el simulador CoppeliaSim Edu.

En este, se logra recrear un espacio con una serie de diferentes objetos, que se interponen en el camino del robot. Éste, pudiendo detectarlos, recogerlos y almacenarlos en los compartimentos correspondientes, simula su correspondiente actuación.

Algunos de los objetos pueden ser los siguientes:

- *Envases de plástico*
- *Botellas de vidrio*
- *Pelotas*

Los módulos simulados son los siguientes:

- *Cinemática inversa del brazo*
- *Movimiento del vehículo*
- *Reconocimiento de objetos*

Cinemática inversa

Primera simulación

La cinemática inversa se ha probado introduciendo la posición objetivo a partir de unas coordenadas x, y, z, y observando la respuesta del robot para moverse hacia esta posición.



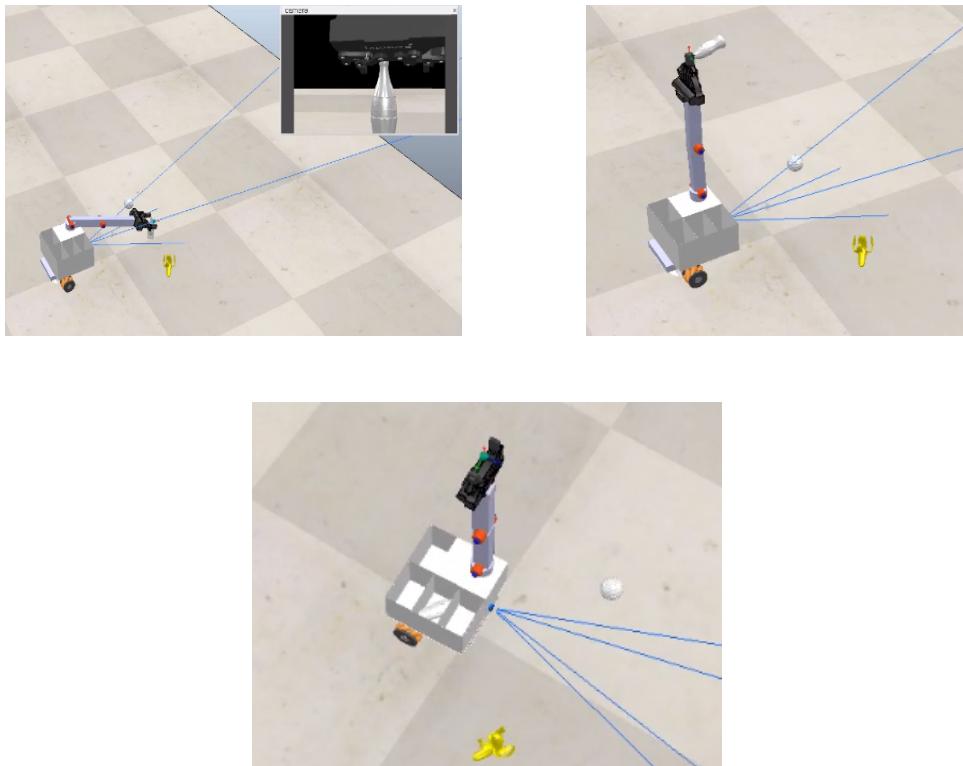
Durante la primera simulación, se ha decidido colocar un objeto en un rango efectivo del área de trabajo del robot, además se le han indicado unas coordenadas cercanas a la posición de dicho objeto. Con esto, se comprueba el correcto funcionamiento de la cinemática inversa del brazo, que se mueve directamente a dónde se había planteado.

Como se puede comprobar, se ha importado la carrocería del vehículo.

Segunda simulación

En esta segunda parte del proceso, se ha decidido llevar a cabo una unión entre los diferentes módulos y algoritmos que forman parte del funcionamiento del robot.

Por lo que a la cinemática inversa se refiere, se ha decidido modificar la estrategia contemplada para llevar al brazo a la posición deseada. Mientras que en la primera simulación se utiliza la matriz y la función nsolve, ahora se ha modificado para resolverlo de manera geométrica. El motivo de este cambio, se debe a la aparición de fallos de tolerancia a la hora de ejecutar la función nsolve y encontrar los ángulos a mover.

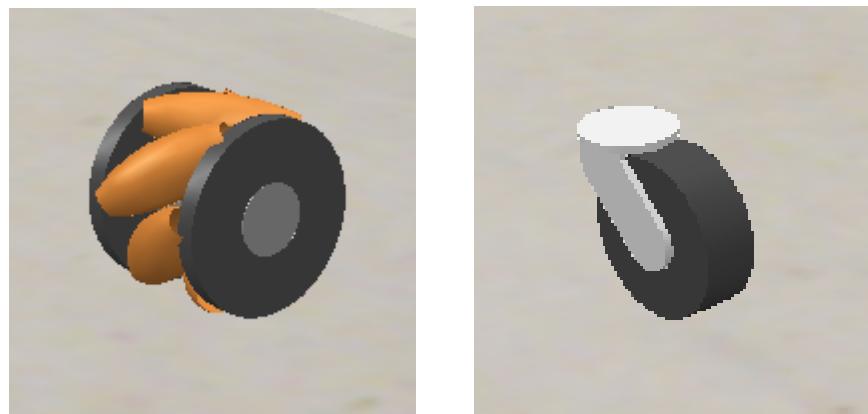


Finalmente, se puede observar como el brazo de tres grados de libertad consigue agarrar la botella, previamente identificada por visión de computador, y guardarla en el compartimiento adecuado.

Movimiento del vehículo

Primera simulación

Para probar el desplazamiento del vehículo se han utilizado componentes de locomoción y propulsión del propio simulador de Coppelia. Estos incluyen dos ruedas impulsoras y un caster como estabilizador de la estructura.

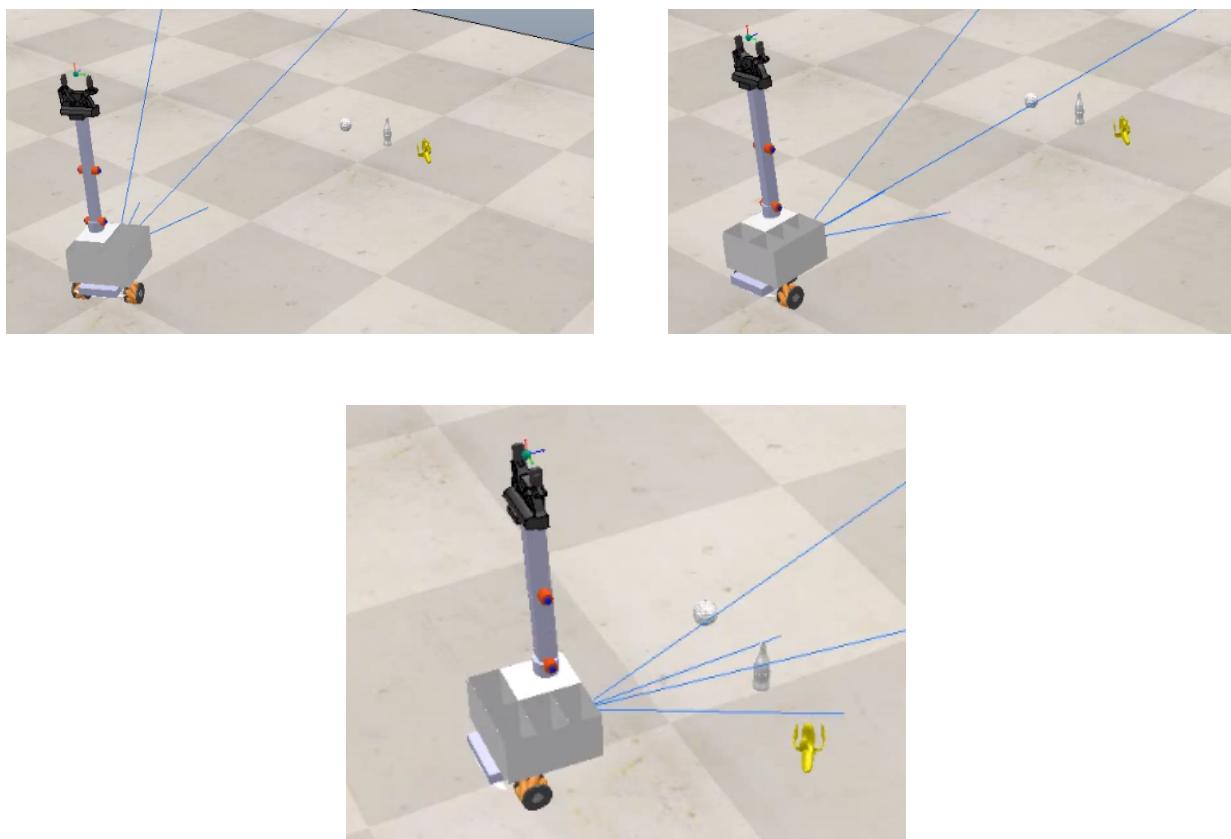


Se han definido las propiedades físicas necesarias para imitar lo más fielmente posible un comportamiento real.



Segunda simulación

Para probar que el robot es capaz de llegar hasta el objeto a recoger se utiliza visión por computador. A través de esta el robot gira sobre sí mismo hasta encontrar el objeto a recoger. Una vez lo localiza, se mueve hacia él, calculando continuamente la distancia restante hasta su llegada. Cuando la distancia es lo suficientemente pequeña como para que el brazo robótico pueda trabajar con comodidad el robot se detiene.

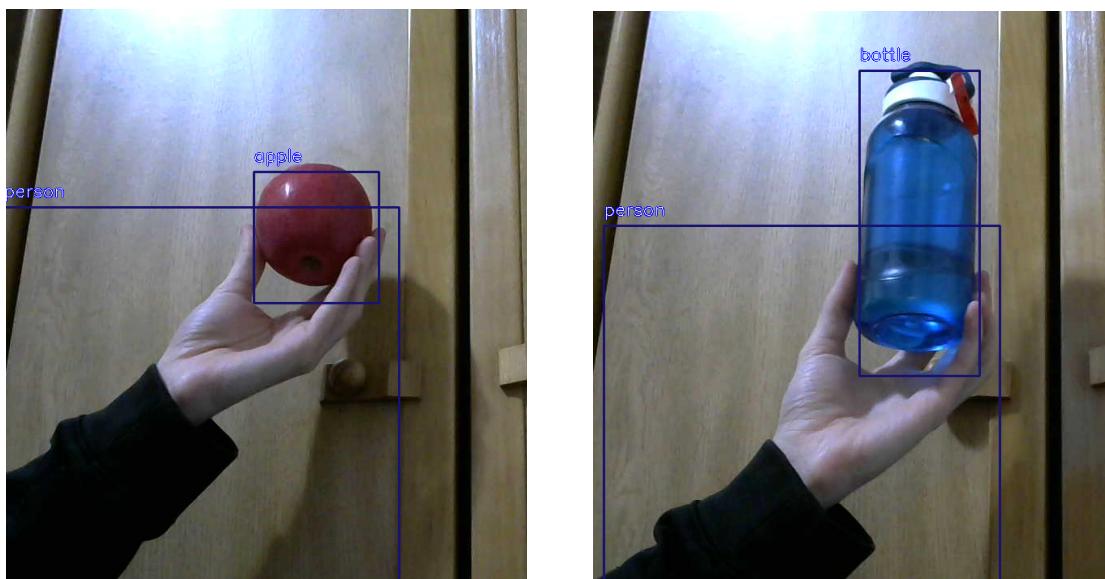


Como se puede observar en las imágenes el brazo robótico mantiene una posición de reposo cuando el vehículo se mueve para evitar la colisión con objetos del entorno.

Reconocimiento de objetos

Primera simulación

La primera simulación de reconocimiento de objetos ha sido realizada utilizando una webcam de ordenador. Se han recogido frames de la cámara en los que se sostienen diferentes objetos que deberían poder ser identificados por el robot para poder realizar su clasificación.

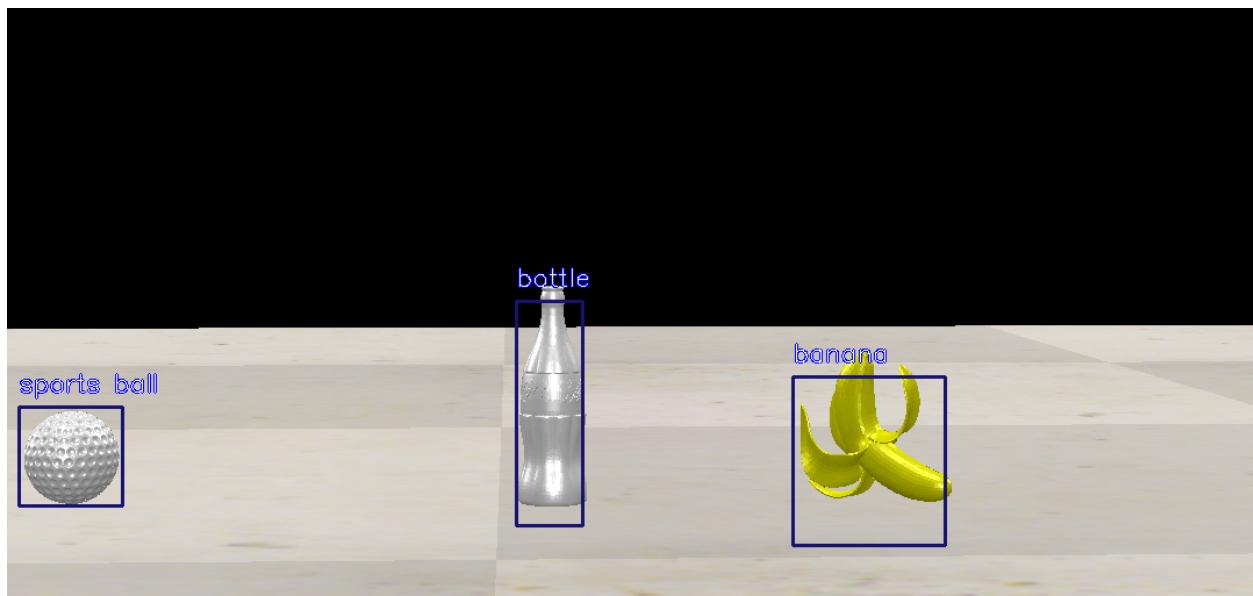


Como se puede observar en las imágenes el modelo de reconocimiento obtiene de manera bastante exacta el tipo de objeto que es y su posición actual.

Simulación en CoppeliaSim

Se ha utilizado el mismo modelo de aprendizaje para realizar el reconocimiento de objetos en el simulador CoppeliaSim. Han sido necesarios varios ajustes para conseguir que el modelo consiga entender qué objetos tiene delante. Al ser una simulación los objetos son menos detallados de lo que son en las imágenes de fotos.

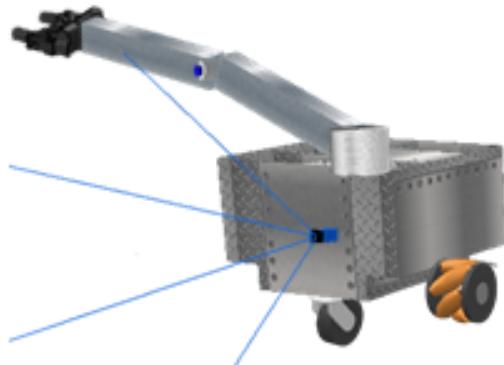
Para que el modelo reconozca los objetos se ha tenido que bajar el límite de threshold del modelo para que aunque la probabilidad de que sea ese objeto sea más baja, se muestren los datos de este y su posición. También se utiliza una resolución de 1080x720 en la cámara. Además, se utilizan objetos fácilmente reconocibles para que el modelo pueda identificarlos correctamente.



El robot cuenta con 4 compartimentos. En la simulación presentada, el robot recoge la botella, que al considerarse envase lo almacena en el compartimento correspondiente.

Texturas

Para ofrecer una simulación lo más realista posible se han añadido texturas y elementos de terreno a la simulación. Entre estos destacan la textura del propio vehículo.



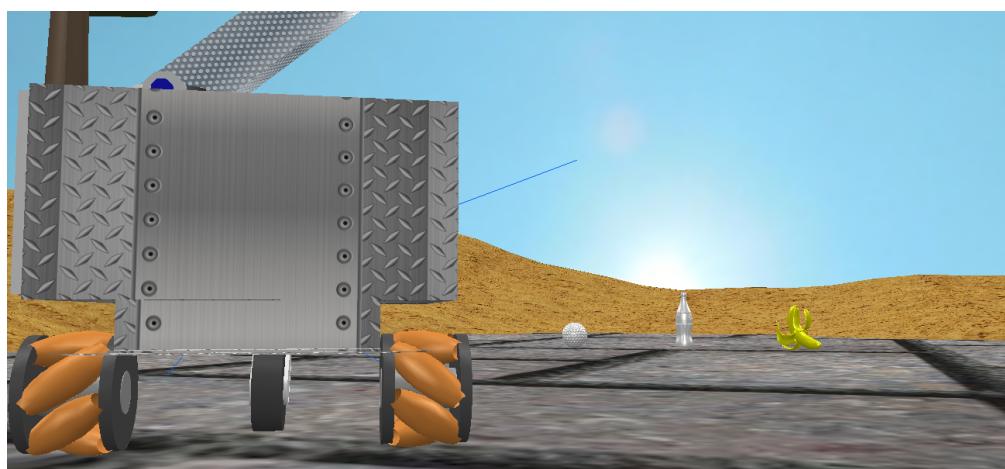
Otro de los elementos que se ha decidido añadir es una textura al suelo de la simulación, para que parezca que el robot se está probando en un entorno real y no en uno controlado.



También se ha decidido añadir terreno vertical para probar la detección de objetos en escenarios no estrictamente horizontales. El escenario recreado pretende ser una zona costera, en el que las dunas de la playa obstruyen parcialmente la vista al horizonte.



Finalmente se han añadido objetos decorativos y una cúpula de cielo con iluminación a modo de hacer más atractiva la simulación en el video promocional del proyecto.



Consideraciones

Para guardar los objetos en los compartimentos se utilizan planos con propiedades de colisión, estos permiten contener el cilindro de la botella en el interior del vehículo.

Riesgos previstos y plan de contingencia

Riesgo #	Descripción	Probabilidad (Alta/Media/Baja)	Impacto (Alto/Medio/Bajo)	Plan de contingencia
1	Fallo en identificación de objeto	Media	Medio	Depósito del objeto en un compartimento “default”
2	Caída de objetos agarrados por la pinza	Baja	Alto	Asociación padre-hijo entre pinza y objeto
3	Colisión del brazo mecánico con posibles objetos y paredes cuando el vehículo está en movimiento	Media	Alto	Definir posición “default” cuando se mueva el robot
4	Brazo pasa por la posición en la que se sitúan los componentes electrónicos	Baja	Alto	Modificar límites en el área de trabajo del brazo mecánico
5	Pérdida de estabilidad por el peso del brazo o de un objeto	Baja	Medio	Colocar contrapesos en la parte trasera del robot
6	Funcionamiento erróneo del robot debido a la velocidad de un movimiento	Baja	Medio	Añadir limitador de velocidad a los controladores de cada motor

Referencias

Este proyecto ha sido inspirado por los siguientes proyectos de Internet:

http://www.iearobotics.com/wiki/index.php?title=Freecad:_Pinza_mecanica

<https://github.com/chillibasket/walle-replica>

Github del proyecto:

<https://github.com/JordiCampoy/SweeperBot>