# Laboratorio 2: Control y visión

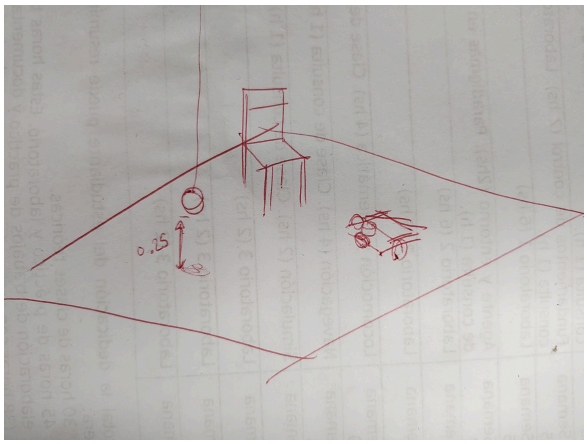
## 1 Objetivos

Inspirados en la categoría Open de la competencia de robótica LARC de la IEEE [1], que desafía a construir un robot recolector de café, en este laboratorio proponemos dar los primeros pasos hacia el desarrollo de una solución.

El objetivo es entonces aplicar conceptos de control y sensado para implementar algunas de las funcionalidad básicas que puedan servir a futuro para resolver el desafío Open.

## 2 Escenario

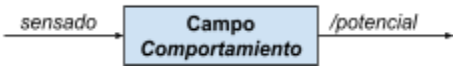
Se asume que se dispone del robot Francesca[2] equipado con un LIDAR y una cámara. El escenario consiste de un ambiente cerrado donde hay obstáculos que se pueden detectar con el LIDAR, tales como sillas, mesas, personas, etc.
Asimismo hay un objetivo en la forma de una pelota de tenis, ubicada a una altura de aproximadamente 25cm del suelo, de forma de ser visible para la cámara pero no ser considerada un obstáculo por el LIDAR.




## 3 Problema a resolver

Se implementarán dos campos de potencial que expresen dos comportamientos primitivos. Ambos comportamientos se integrarán para resolver un desafío. Para esto deberán implementarse varios nodos ROS 2.

**Comportamientos.**
Los comportamientos generarán una salida en forma de un vector de potencial. Cada comportamiento podrá ser ejecutado independientemente, o ambos podrán combinarse sumando sus vectores. La entrada de un comportamiento sera uno o varios tópicos de sensores, y la salida será un vector[3] que representa la dirección y velocidad en la que debería moverse el robot.



**Controlador**


El o los vectores de potencial deberán ser convertidos en comandos de velocidad /cmd_vel para el robot. Note que /cmd_vel contiene tanto una velocidad lineal como una angular, y es posible reusar parte del código del *Taller 5 - PID*. Asimismo, tenga en cuenta que pueden llegar múltiples tópicos de potencial originados en distintos comportamientos. El controlador deberá realizar la suma vectorial de todos ellos. Se acepta una implementación donde el controlador conoce a los comportamientos, y espera a tener por lo menos un vector de cada uno de ellos antes de hacer la suma vectorial y computar cmd_vel.

Se admite experimentar con la implementación del controlador. Por ejemplo, el robot podría siempre girar hasta alinearse con la dirección para luego avanzar, o podría seguir una curva suave girando continuamente mientras avanza.

# 3.A) Comportamiento "*acercarse a un objetivo*"

Se espera que bajo los efectos de este comportamiento el robot se mueva en dirección a un objetivo identificado con la cámara, reduciendo su velocidad a medida que se acerca hasta detenerse completamente en su proximidad.

Para realizar la detección del objetivo (una pelota de tenis) se usará el paquete `tennis_detector_pkg`. Este paquete está disponible el el branch `tennis_detector` de Francesca. Este paquete contiene dos nodos:

## Nodo detector

- Se ejecuta como `ros2 run tennis_detector tennis_ball_detector`
- Este nodo consume tópicos `/image_raw` de la cámara.
    - Levantar la cámara con Francesca o manualmente:
      `ros2 run v4l2_camera v4l2_camera_node`
- Emite tópicos `/ball_box` del tipo `vision_msgs.msg.BoundingBox2D`.
- Si en el código se declara la variable `DEBUG_DRAW=True` abrirá una ventana mostrando los frames con el bounding box dibujado.

## Nodo debug

- Se ejecuta como: `ros2 run tennis_detector tennis_ball_debug`
- Este nodo consume tanto el video como los bounding boxes para dibujarlos superpuestos.
- Sirve como ejemplo de código que consume bounding boxes.

La salida del nodo detector (el tópico /ball_box) será consumida por el nodo del comportamiento y, la usará para construir un vector de un campo potencial atractor. El vector

deberá apuntar hacia el objetivo, y la magnitud deberá variar de manera apropiada según la distancia estimada al objetivo.


## 3.B) Comportamiento "*esquivar obstáculos*"

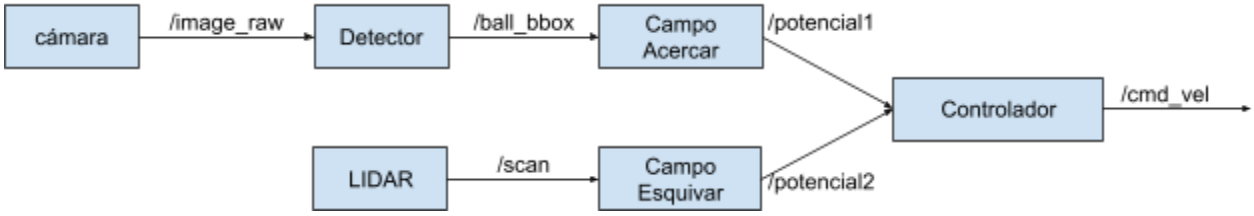
Se espera que bajo los efectos de este comportamiento el robot se aleje de los obstáculos detectados con el LIDAR que estén demasiado cerca, y que evite que el robot se acerque a los obstáculos bajo los efectos de otro campo de potencial.


El nodo del LIDAR se levantará con Francesca o manualmente:

```
ros2 run hls_lfcd_lds_driver hlds_laser_publisher
```

## 3.C) Integración de comportamientos.

Se espera poder ejecutar los dos comportamientos en simultáneo. En este caso el robot deberá acercarse al objetivo evitando los obstáculos. Puede asumir que el objetivo es visible desde la posición inicial.
Se valorará proponer o implementar soluciones para el caso donde el robot no ve el objetivo desde la posición inicial, por ejemplo porque el objetivo está fuera de su campo visual.




# 4 Entrega

La fecha límite de entrega es el día 3/6/2025. La entrega consistirá en una Wiki de Gitlab en su proyecto privado. Esta Wiki deberá extender la del Laboratorio 1 incluyendo, de la forma más pedagógica posible, asuntos tales como:

- Modificaciones a la construcción del robot.
- Diseño conceptual de los campos de potencial definidos.
- Documentación de los nodos creados.
- Descripción y resultados de las pruebas realizadas
- Tutoriales que describan como implementar y levantar las distintas funcionalidades y experimentos que realizaron.
- Se acepta y promueve el uso de fotos, esquemas y gráficas para ilustrar.

- La estructura de la Wiki es libre, pero intente ser ordenado y legible. La Wiki seguirá expandiéndose en los siguientes Laboratorios.

Para entregar deberán *pushear* una versión con un *tag* con la etiqueta “lab2entrega”, antes de la fecha de cierre del Laboratorio. Esta será la versión evaluada.

**Nota**: Si tiene mejoras al software de base de Francesca, se apreciará que envíen *pull-request* upstream para incorporarlas al proyecto común. El proponer mejoras se considerará parte del trabajo de Laboratorio. Las mejoras de interés pueden ser funcionalidades, estructura, bug-fixes, documentación, etc.

# 4 Referencias

[1] https://femexrobotica.org/tmr2025/larc-open-challenge/
[2] https://gitlab.fing.edu.uy/jvisca/francesca_ws
[3] https://docs.ros.org/en/jazzy/p/geometry_msgs/msg/Vector3.html