

# Introducción a Robotic Operating System

Presentación del curso

Martín Llofriu y Gonzalo Tejera

Facultad de Ingeniería :: Instituto de Computación

# Contenido

- Objetivos del curso
- Temario
- Forma de trabajo
- Dedicación
- Horarios
- Aprobación del curso
- Equipamiento robótico
- Bibliografía
- Conocimiento
- Cupo

# Objetivos

Los objetivos del curso son:

- Introducir la plataforma abierta ROS
- Realizar la instalación y configuración del sistema
- Trabajar sobre los aspectos generales de la arquitectura
- Trabajar sobre los módulos más importantes sobre la plataforma robótica

<https://eva.fing.edu.uy/course/view.php?id=767>

# Temario

- Instalación de ROS en Ubuntu
- Arquitectura de ROS (paquetes, mensajes, nodos y servicios)
- Práctico: Creando un paquete ROS
- Herramientas útiles de ROS
- Práctico: Comunicando dos nodos
- Práctico: Usando ROS con Butiá
- Revisión de algunos paquetes importantes de ROS (visión, marcas, navegación, entre otros)
- Filtros de partículas - Fusión de sensores

# Forma de Trabajo

- El curso está orientado principalmente al trabajo en el laboratorio.
- A este se le suman clases teóricas.
- El trabajo de laboratorio y las clases teóricas se intercalan durante todo el curso.
- Se trabajará sobre el robot butia.

# Créditos y Dedicación

- 7 créditos.
- Horas
  - 10 teórico.
  - 60 práctico y laboratorio.
  - 20 estudio.
  - 15 proyecto final y monografía.

# Horarios

## **Teórico**

- Lunes 8 de Diciembre al 12 de Diciembre del de 9 a 12 hs.
- En el Laboratorio de Robótica y Sistemas Embebidos.

## **Laboratorio**

Cuando no hay clases teóricas,

- Lunes 8 de Diciembre al 17 de Diciembre del de 9 a 12 hs.
- Horario amplio de febrero a mediados de marzo.
- En el Laboratorio de Robótica y Sistemas Embebidos.

# Aprobación

- La asistencia al teórico y al laboratorio es obligatoria.
- Realización de Laboratorios
  - Implementación
  - Reporte técnico y documento multimedia.
  - Presentación y prueba del producto en el entorno definido (mediados de marzo)

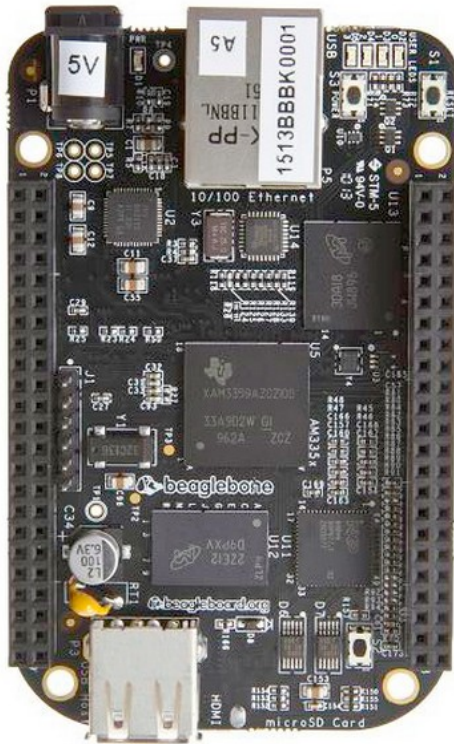


# Plataforma robótica

- Cómputo
- Sensores
- Actuadores
- Chasis

# Single Board Computer

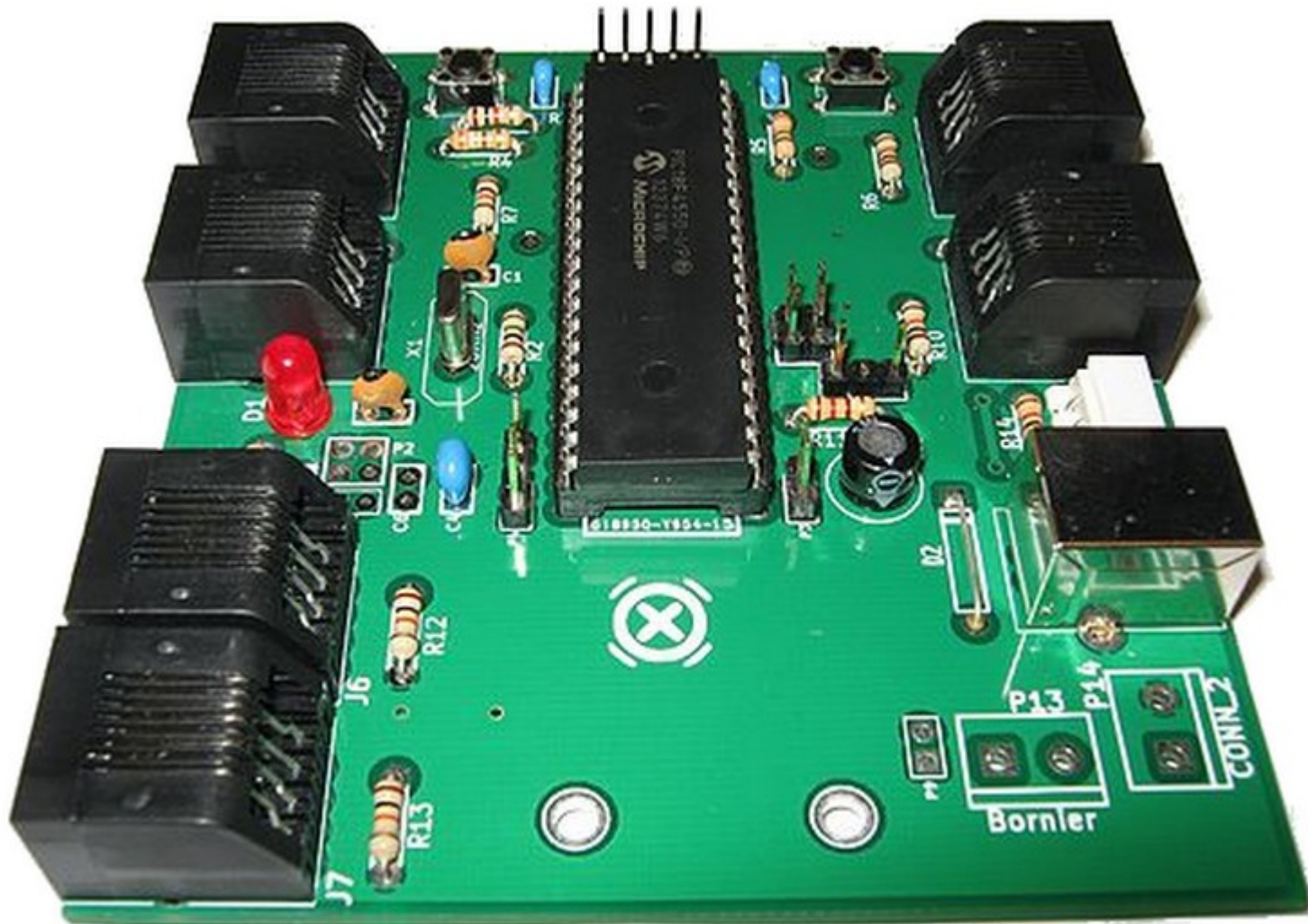
## Beagle Bone Black



Processor: AM335x 1GHz ARM® Cortex-A8

- 512MB DDR3 RAM
- 2GB 8-bit eMMC on-board flash storage
- 3D graphics accelerator
- NEON floating-point accelerator
- 2x PRU 32-bit microcontrollers
- E/S:
  - USB client for power & communications
  - USB host
  - Ethernet
  - HDMI
  - 2x 46 pin headers

# Placa E/S USB4Butia

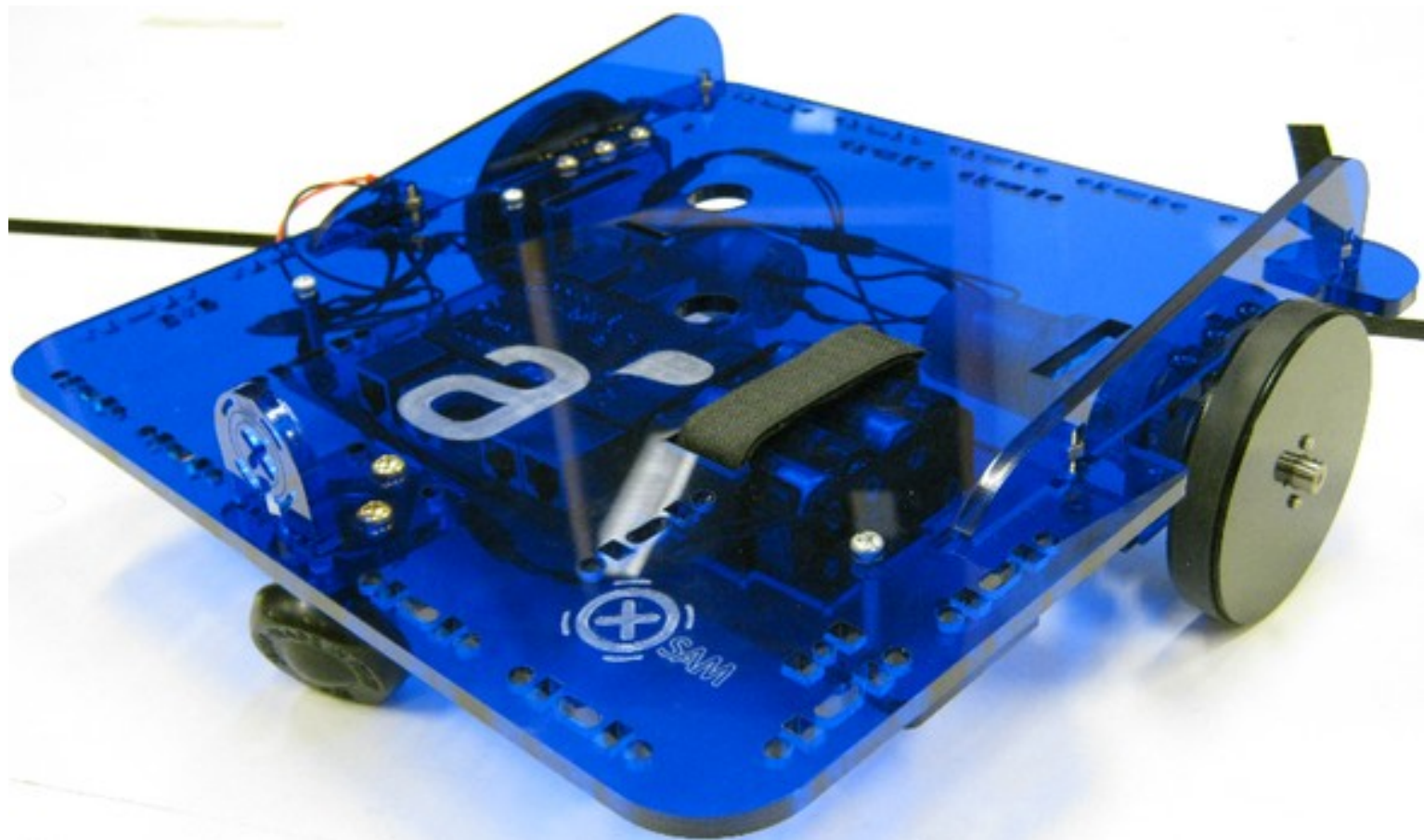


# Sensores

Contacto  
Distancia  
Sonido  
Cámaras

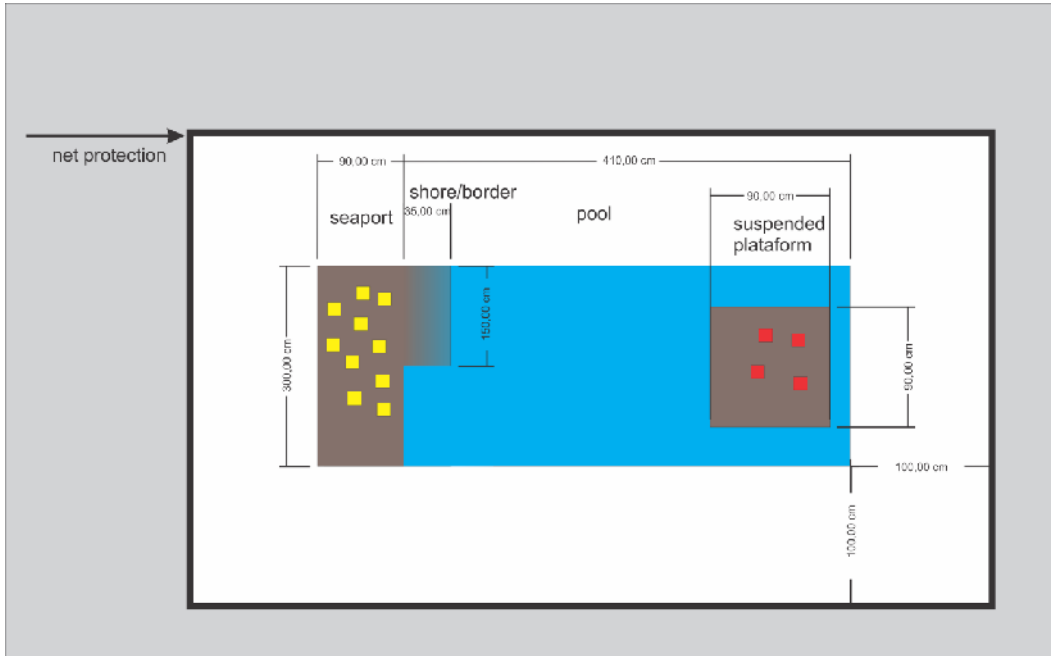
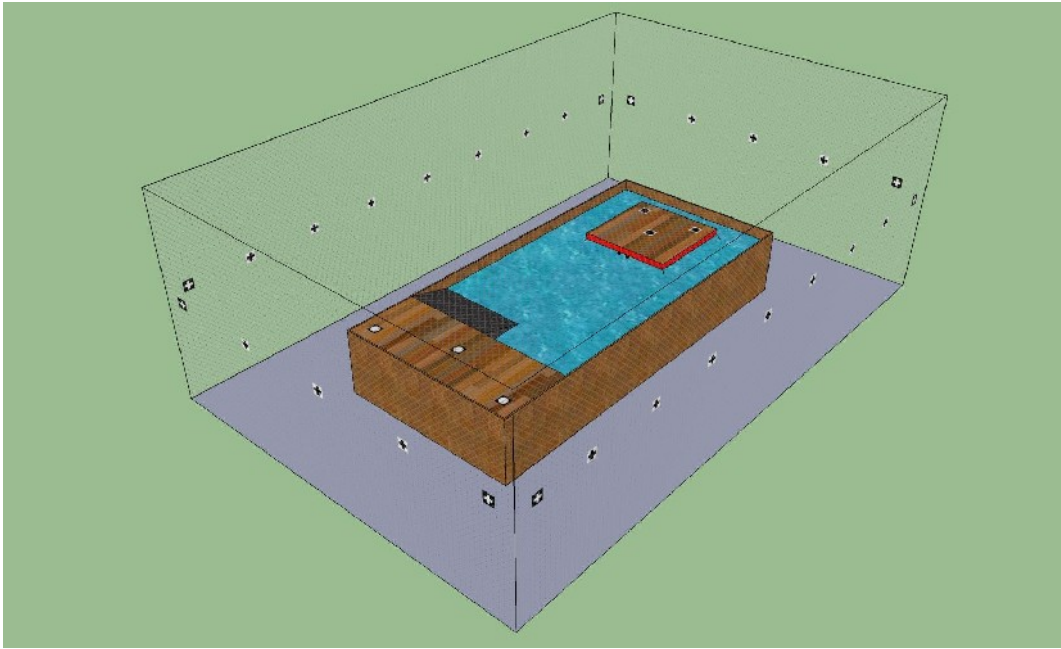


# Chasis Butiá

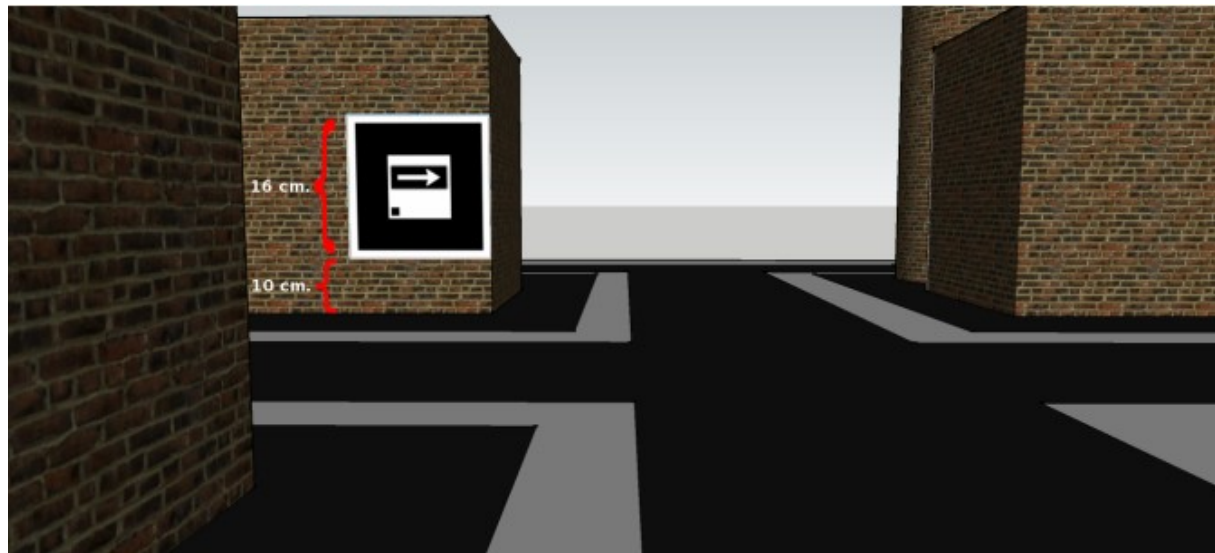
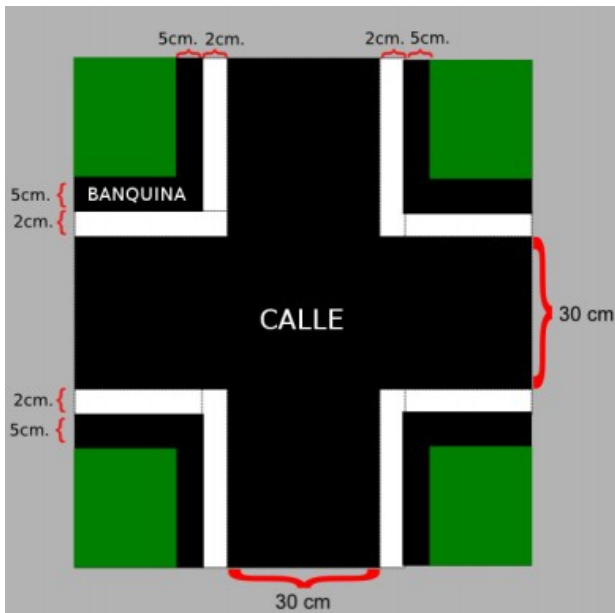




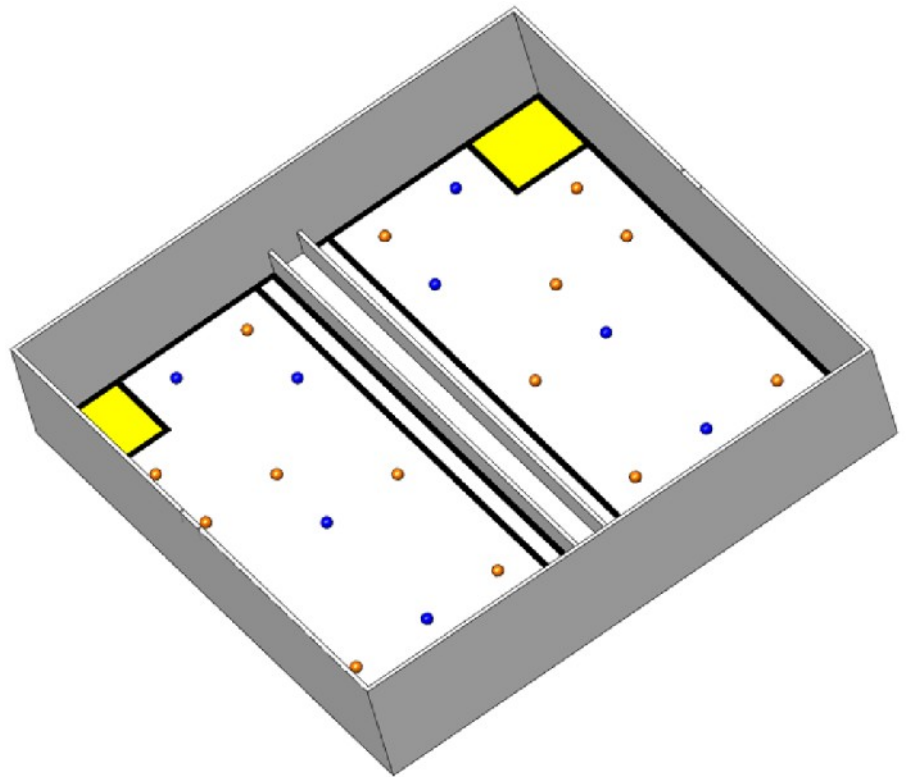
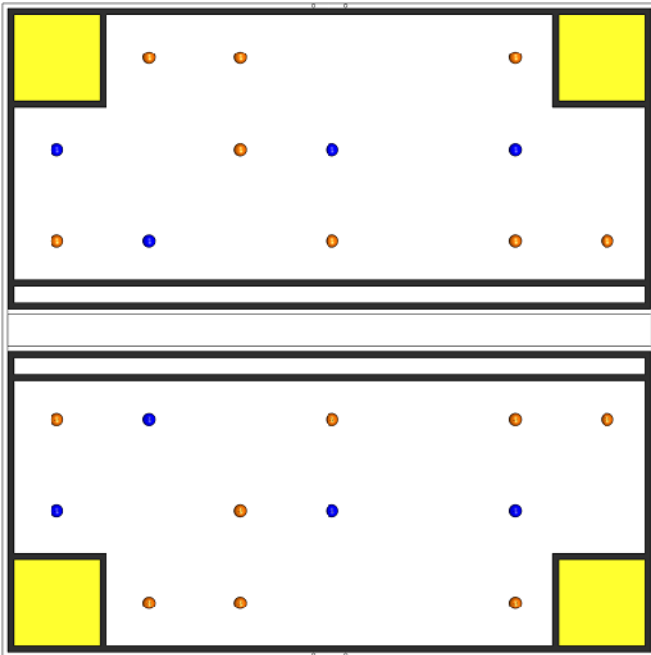
# Escenarios (1/3)



# Escenarios (2/3)



# Escenarios (3/3)





# Bibliografía

- web:
  - [wiki.ros.org/ROS/Tutorials](http://wiki.ros.org/ROS/Tutorials)
  - [www.fing.edu.uy/inco/proyectos/butia/mediawiki/index.php/Página\\_principal](http://www.fing.edu.uy/inco/proyectos/butia/mediawiki/index.php/Página_principal)
  - [wiki.ros.org/vision\\_opencv](http://wiki.ros.org/vision_opencv)
  - [wiki.ros.org/navigation](http://wiki.ros.org/navigation)
  - [wiki.ros.org/tf](http://wiki.ros.org/tf)
  - [wiki.ros.org/rviz](http://wiki.ros.org/rviz)

# Conocimientos

- Conocimientos previos exigidos:
  - Conocimientos en programación
- Conocimientos previos recomendados:
  - Conocimientos en robótica
    - Introducción
    - Agente entorno
    - Paradigmas
  - Conocimientos básicos de Linux
    - uso de terminal y comandos de base
    - gestor de paquetes
    - gcc y makefile

# Cupo

- 18 estudiantes.
- Para ser considerados en el cupo deben asistir a la primera clase introductoria.
- Tendrán prioridad los estudiantes inscriptos en algún programa de posgrado.
- Selección en base a sorteo.

# Tareas domiciliarias

- Instalar ROS Indigo o descargar máquina virtual
- Realizar los tutoriales básicos  
(Beginner Level en <http://wiki.ros.org/ROS/Tutorials>)
  - Crear un workspace (catkin)
  - Crear y compilar un paquete
  - Publicar/subscribir un mensaje
  - Publicar/subscribir un servicio (opcional)
- Language a utilizar:
  - C++ es preferible por permitir un uso más eficiente de los recursos. Indispensable si se pretende continuar con el uso de ROS en plataformas embebidas.
  - Python permite un rápido prototipado y es un lenguaje más fácil de aprender.

¡Bienvenidos!

# Preguntas

¿?