Robótica del siglo XXI y educación

José María Cañas Plaza

josemaria.plaza@gmail.com

PROSEGUR, 22 de octubre de 2020

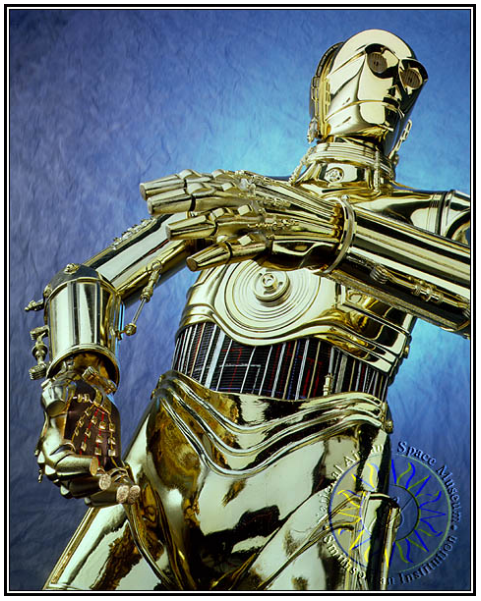
Contenidos

- Introducción
  - Robótica, una tecnología en auge
  - ¿Qué es un robot?
  - Software es importante en robótica
- Educación robótica para niños e iniciación
- Educación robótica superior, profesionales

Introducción

Robótica, una tecnología en auge

Robótica ficción vs Robótica real

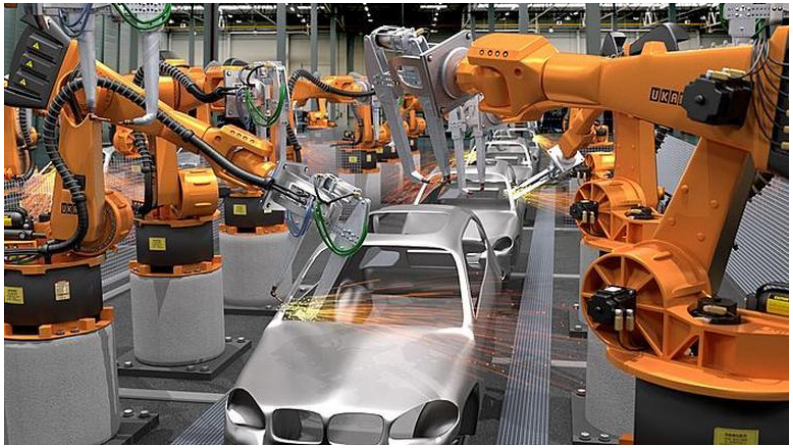


- Cada vez más aplicaciones y productos robóticos útiles en la sociedad

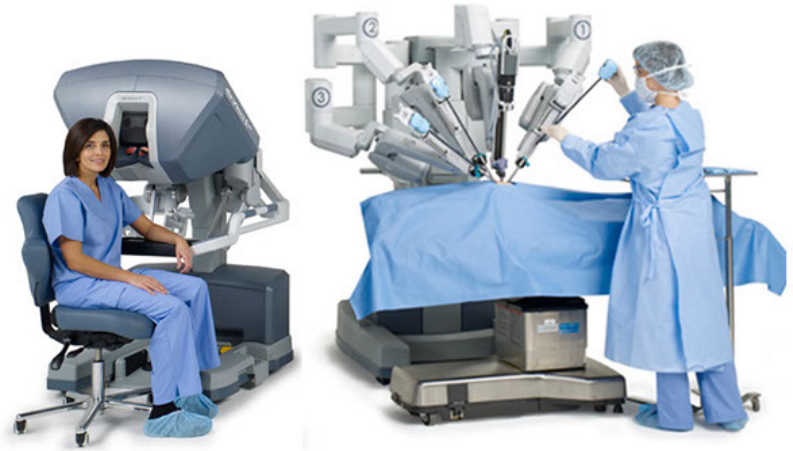
Aplicaciones reales

- *Dull, Dirty, Dangerous*
- ¡La robótica ha salido de los laboratorios!
- Aplicaciones reales, masivas

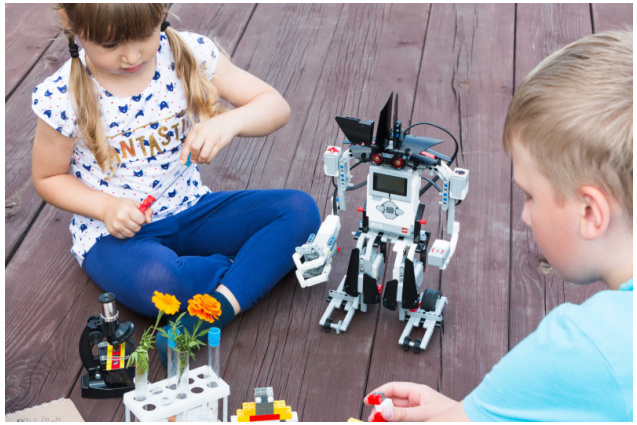
- Industria automovilística
- Coches autónomos
- Gestión de almacenes

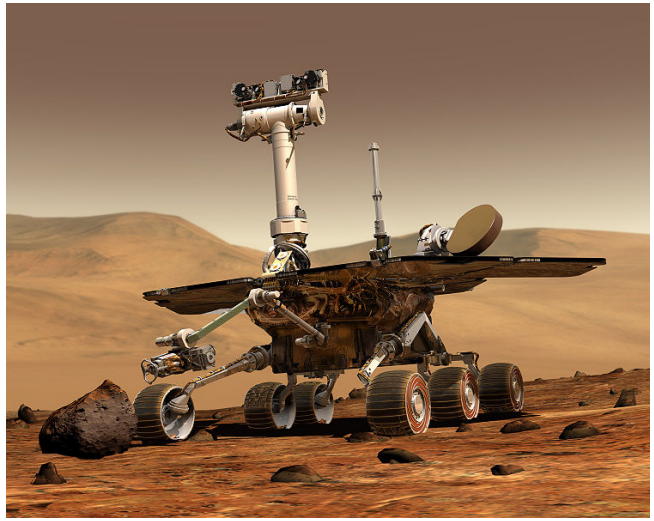


- Hogar: aspiradoras
- Medicina
- Envasado de alimentos



- Entretenimiento: LEGO...
- Usos militares: PackBot...
- Espacio: Opportunity...
- Prestige, limpieza centrales nucleares





¿Qué es un robot?

- $robot = hardware + software$
- (Hardware) Sistema informático con:
  - Sensores
  - Actuadores
  - Computadoras
- (Software) Hay que programarlo para que consiga sus objetivos y sea sensible a la situación.

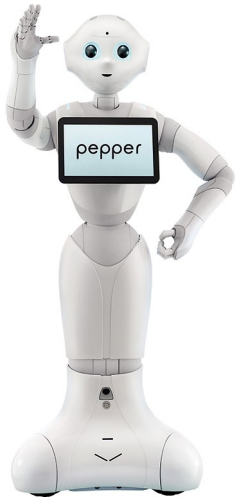
- La inteligencia reside en su software

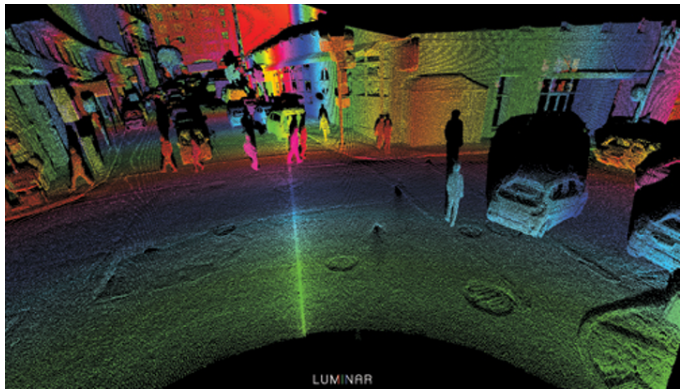
Sensores

- Cámaras, RGBD
- US, Láser, LIDAR
- Encoders

Actuadores

- Motores eléctricos
- Locomoción
- Manipulación





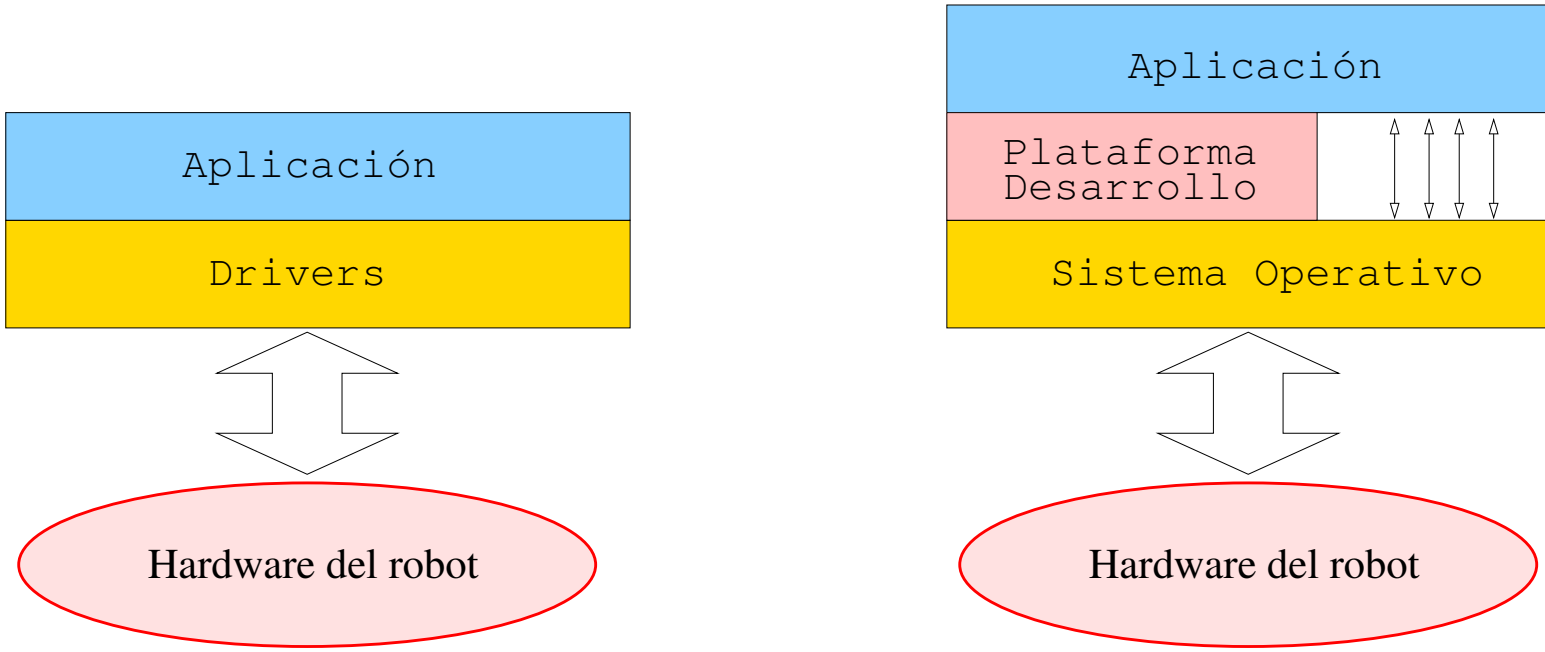
Software para robots

- Determina el comportamiento del robot
- Establece cómo se coordinan la percepción y la actuación
- No hay una manera universalmente aceptada de programarlos
- Lenguajes: ensamblador, C, C++ (de bajo y alto nivel)
- Heterogeneidad
  - Dispositivos hardware
  - Encapsular funcionalidad
- Requisitos específicos
- Sistemas operativos y plataformas
- Simuladores

Requisitos específicos

- Vivacidad, agilidad (tiempo real)
- Multitarea (concurrencia, múltiples fuentes de actividad)
- Distribuido, comunicaciones
- Interfaz gráfica, depuración
- Expandible
- Conectado a la realidad física
- Hardware heterogéneo
- Reutilizar software es difícil

Sistemas operativos y plataformas

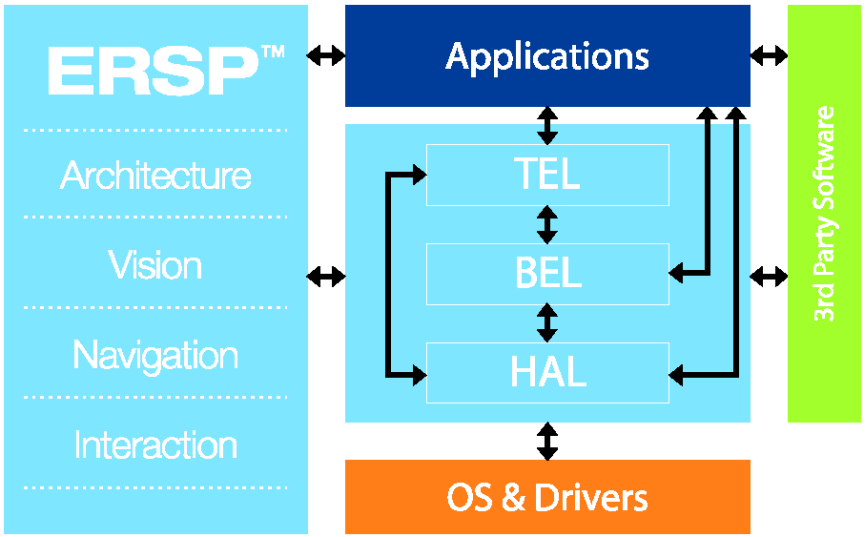


- Procesadores empotrados (robots pequeños) o PC (medianos-grandes).
- Sistemas operativos: dedicados o generalistas
- *Middleware* para simplificar la creación de aplicaciones robóticas

- Empezar de cero cada vez, cada robot su entorno de programación
- Reutilizar software es difícil
- Encapsular funcionalidad o comportamientos es difícil
- Tendencia a software orientado a componentes
- Tendencia a interfaces explícitos

¿Qué proporciona una plataforma sw para robots?

- Abstracción del hardware (HAL)
- Arquitectura software
- Funcionalidades de uso común
- Arquitectura cognitiva

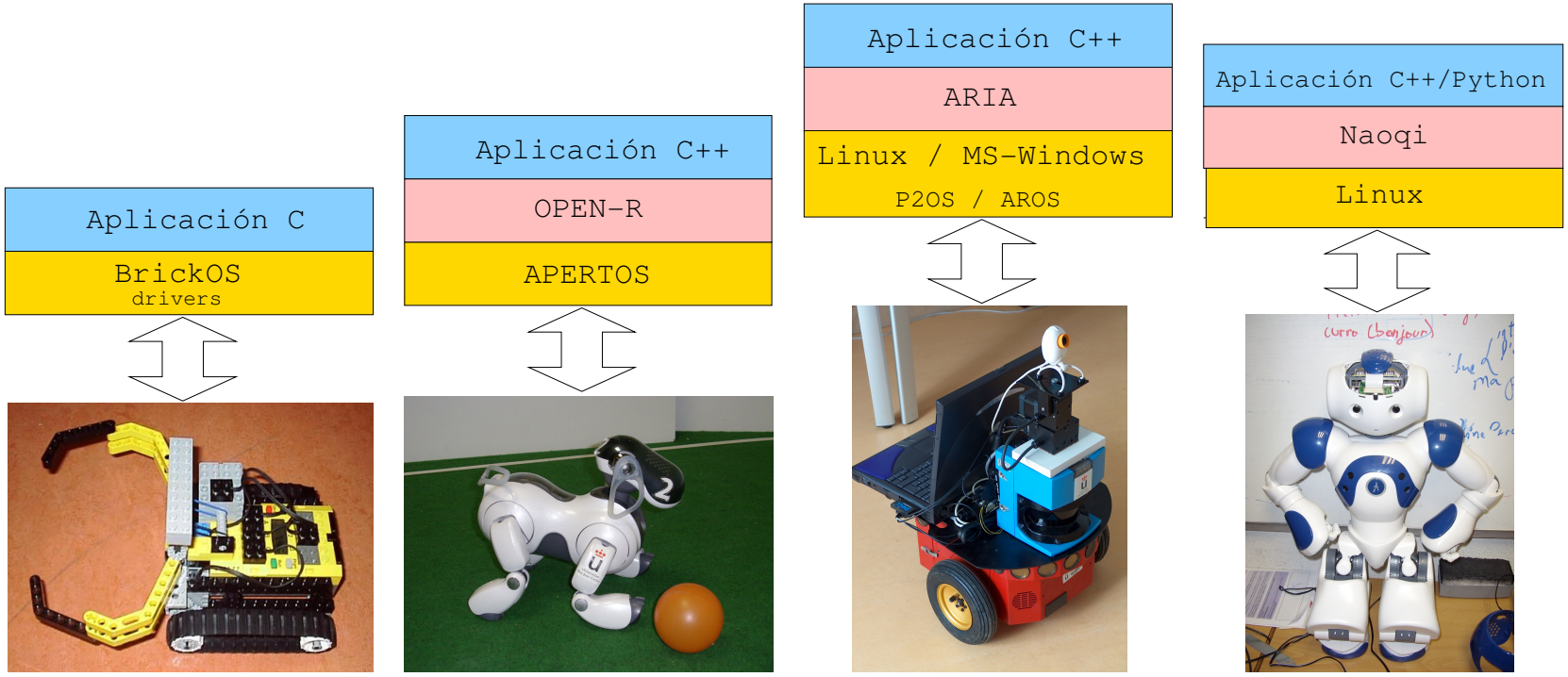


- Comerciales, investigación, software libre
- Ingeniería software: orientación a objetos, distribución
- ROS, Orca, Carmen, OROCOS, ERSP, Player/Stage, Claraty, etc.

Robot Operating System (ROS)

- *Middleware* para software robótico
- Evitar la reinvención de la rueda: comunicaciones, drivers...
- Gratis y software libre: `https://ros.org`
- Colección de paquetes
- C/C++, Python
- Comunidad enorme (usuarios, desarrolladores, soporte...)
- Standard de facto en robótica de servicios
- Sobre Linux principalmente

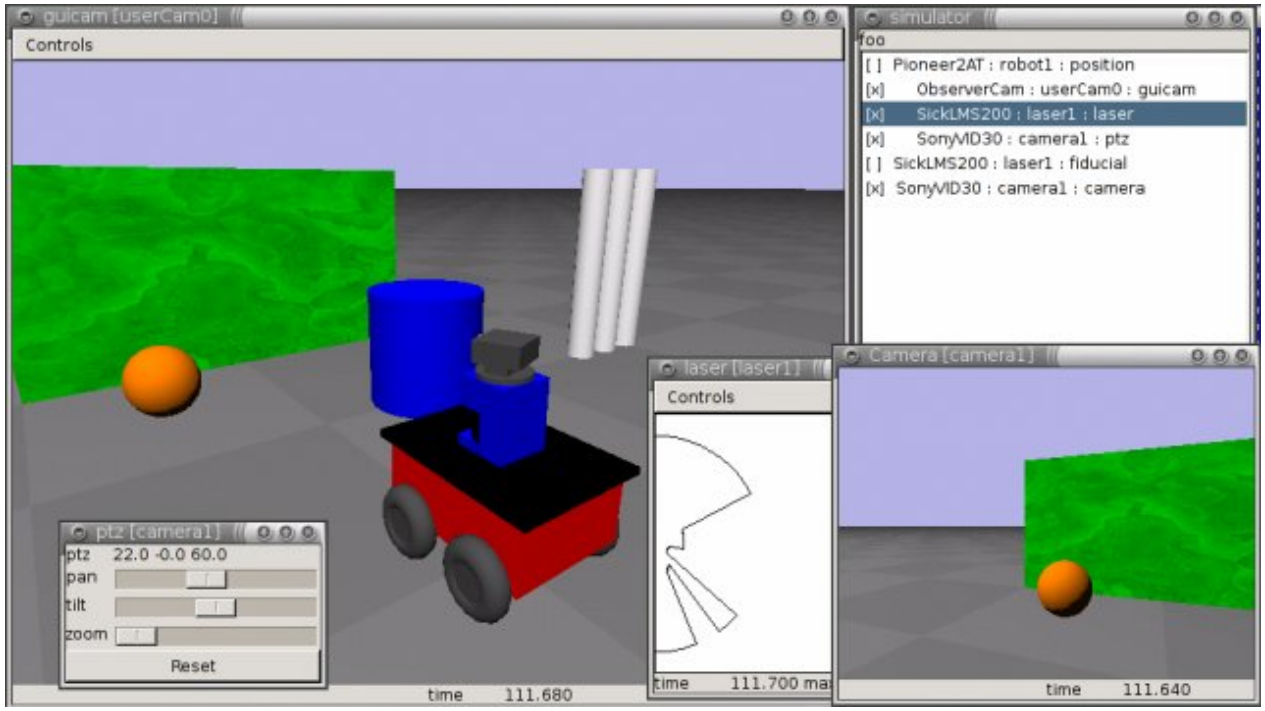
Ejemplos de plataformas sw



Simuladores

- Madurar algoritmos
- Comodidad trabajar sin robot
- Las caídas no duelen
- Mundo, sensores y actuadores
- OpenGl (OGRE) para imágenes
- Motor físico: ODE (*Open Dynamics Engine*), Bullet...

- Gazebo, CoppeliaSim, Webots, Stage, MORSE

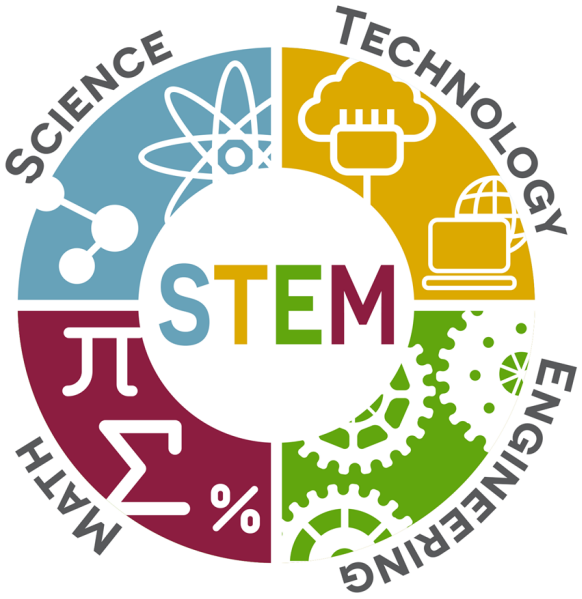

guicam [userCam0]
Controls
simulator
foo
[ ] Pioneer2AT : robot1 : position
[x] ObserverCam : userCam0 : guicam
[x] SickLMS200 : laser1 : laser
[x] SonyVID30 : camera1 : ptz
[ ] SickLMS200 : laser1 : fiducial
[x] SonyVID30 : camera1 : camera
ptz [camera1]
ptz 22.0 -0.0 60.0
pan
tilt
zoom
Reset
time 111.680
laser [laser1]
Controls
time 111.700
Camera [camera1]
time 111.640

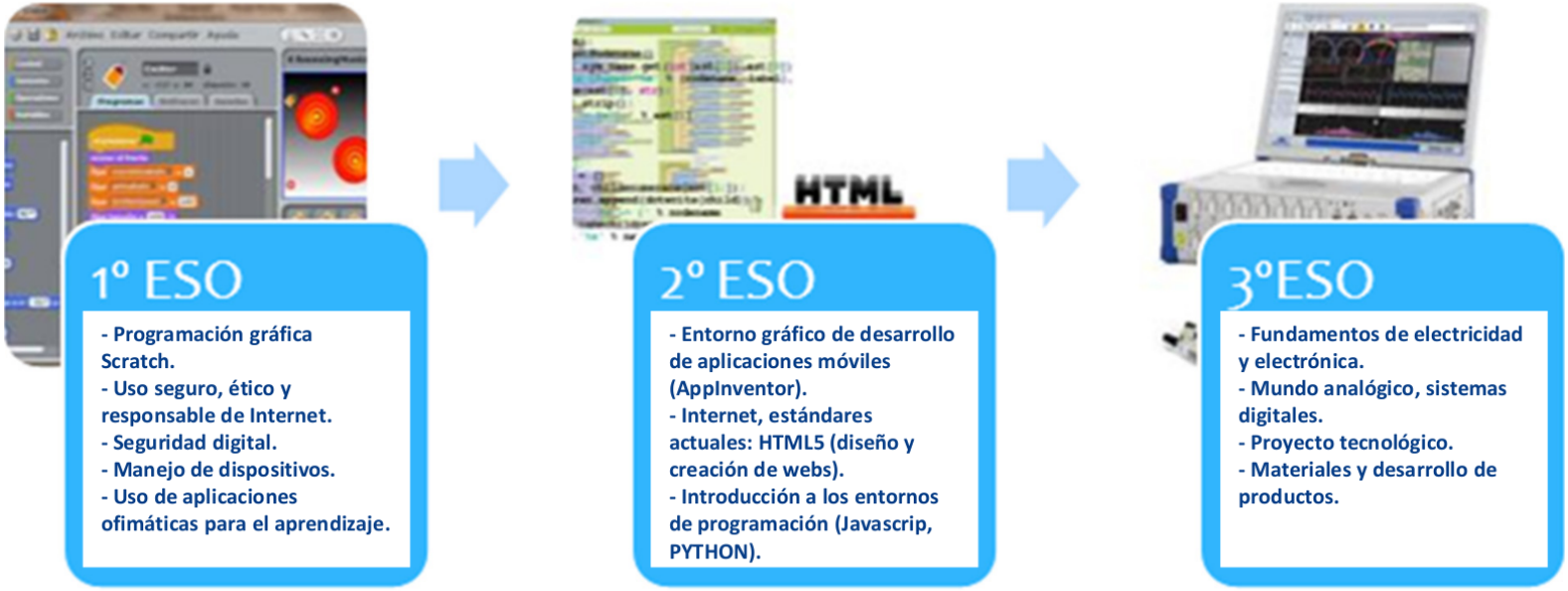

Stage: worlds/fasr.world
File View Run Help

Educación robótica para niños e iniciación

Demanda creciente de formación en robótica

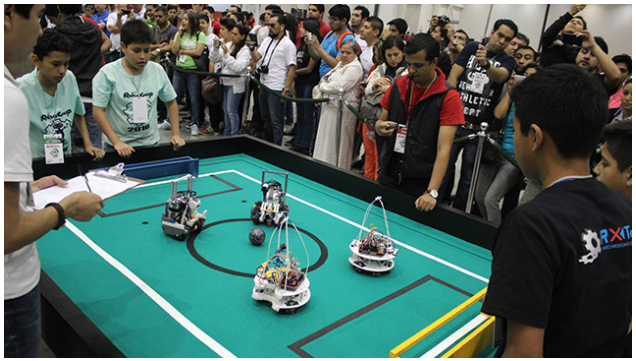
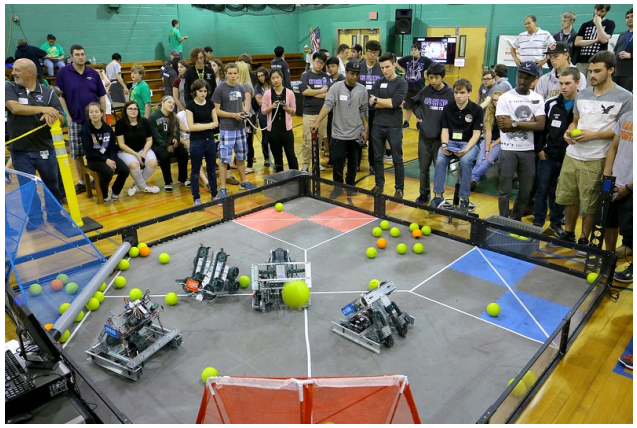
- Nociones básicas útiles para todos en XXI
- Introducción divertida a la tecnología
- Disciplinas STEM
  Science, Technology, Engineering, Math
- + Permite docencia por proyectos
- + Permite *gamificación*
- + Facilita trabajo en equipo
- Pensamiento Computacional





- Curriculo oficial Enseñanza Secundaria (Com.Madrid, Galicia...)
  - 1º, 2º, 3º ESO "Tecnología, Programación y robótica" (2h/sem)
  - 4º ESO "Tecnología" (3h/sem)
- Actividades extraescolares

Competiciones robóticas infantiles


- First LEGO League
- Robocup Junior
- VEX
- RoboCampeones

Robots educativos

LEGO MindStorms EV3

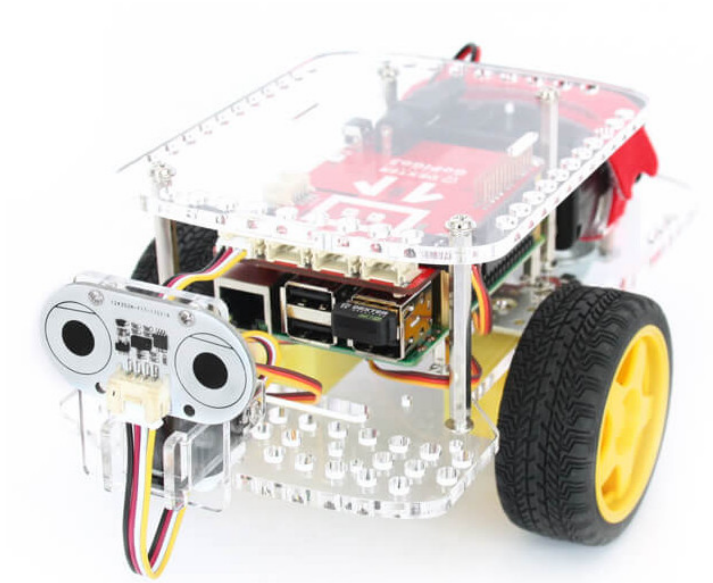
- Comercial LEGO
- Fácilmente adquirible
- Robusto y estable
- Control en posición
- Control en velocidad

Robot mBot

- Comercial Makeblock
- Barato, fácilmente adquirible
- Robusto
- Arduino
- Motores
- Ultrasonidos, Infrarrojos
- mBlock, lenguaje Arduino

Robot GoPiGo3

- Comercial, Dexter industries
- Abierto, construible
- RaspberryPI
- Motores con encoders
- Ultrasonidos, Infrarrojos
- (PiCam cámara)

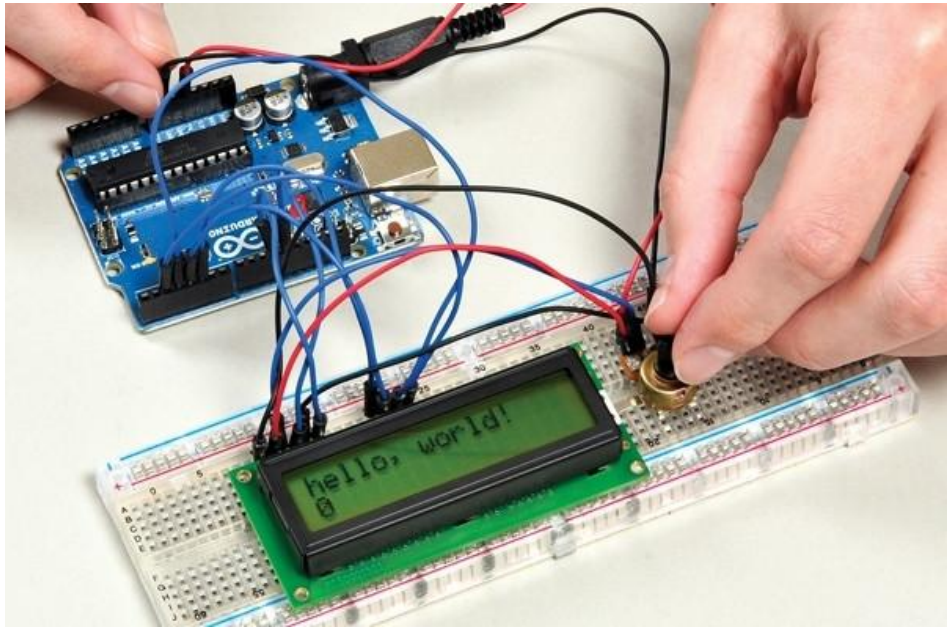


Drone Tello

- Comercial DJI+Intel
- Barato, fácilmente adquirible
- Robusto y estable
- Cámara
- Control en posición
- Control en velocidad

Microcontrolador Arduino

- Sensores, Motores
- Pulsadores
- Pantalla
- Electrónica
- Programación
- Lenguaje Arduino



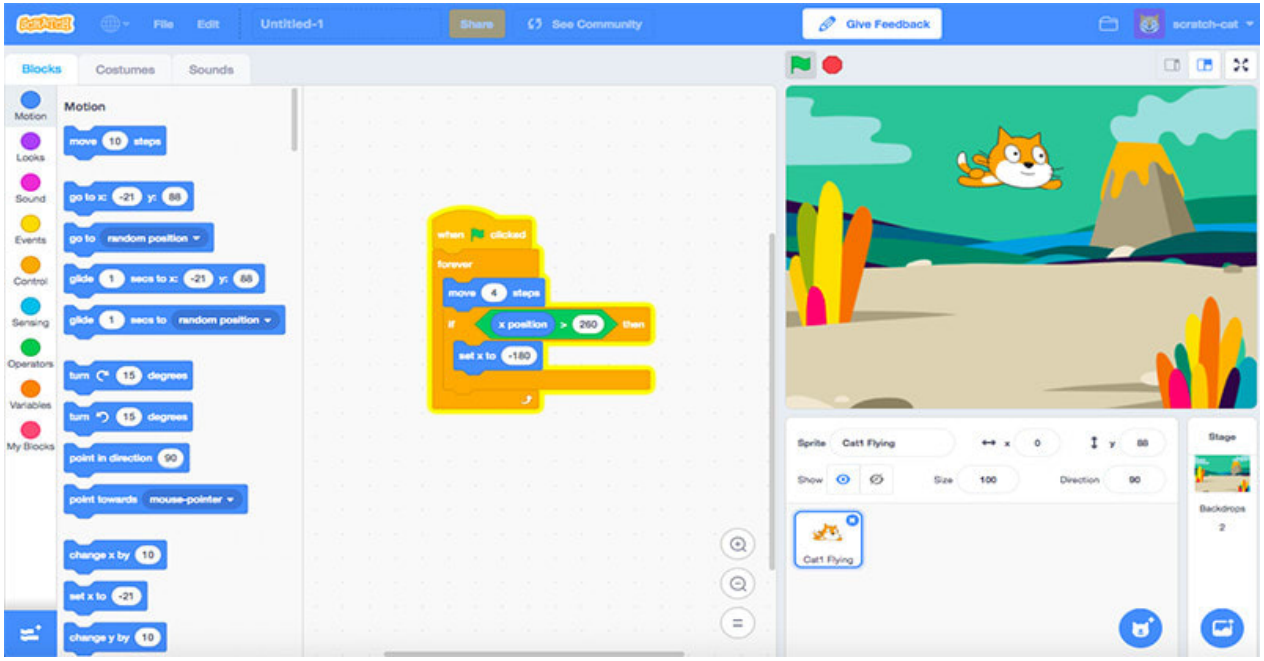
Plataformas educativas infantiles

- Permiten programar los robots
- Cada fabricante ofrece el suyo
- Tiene que ser fácil, para niños
- Tiene que ser estimulante
- Lenguajes gráficos o de texto

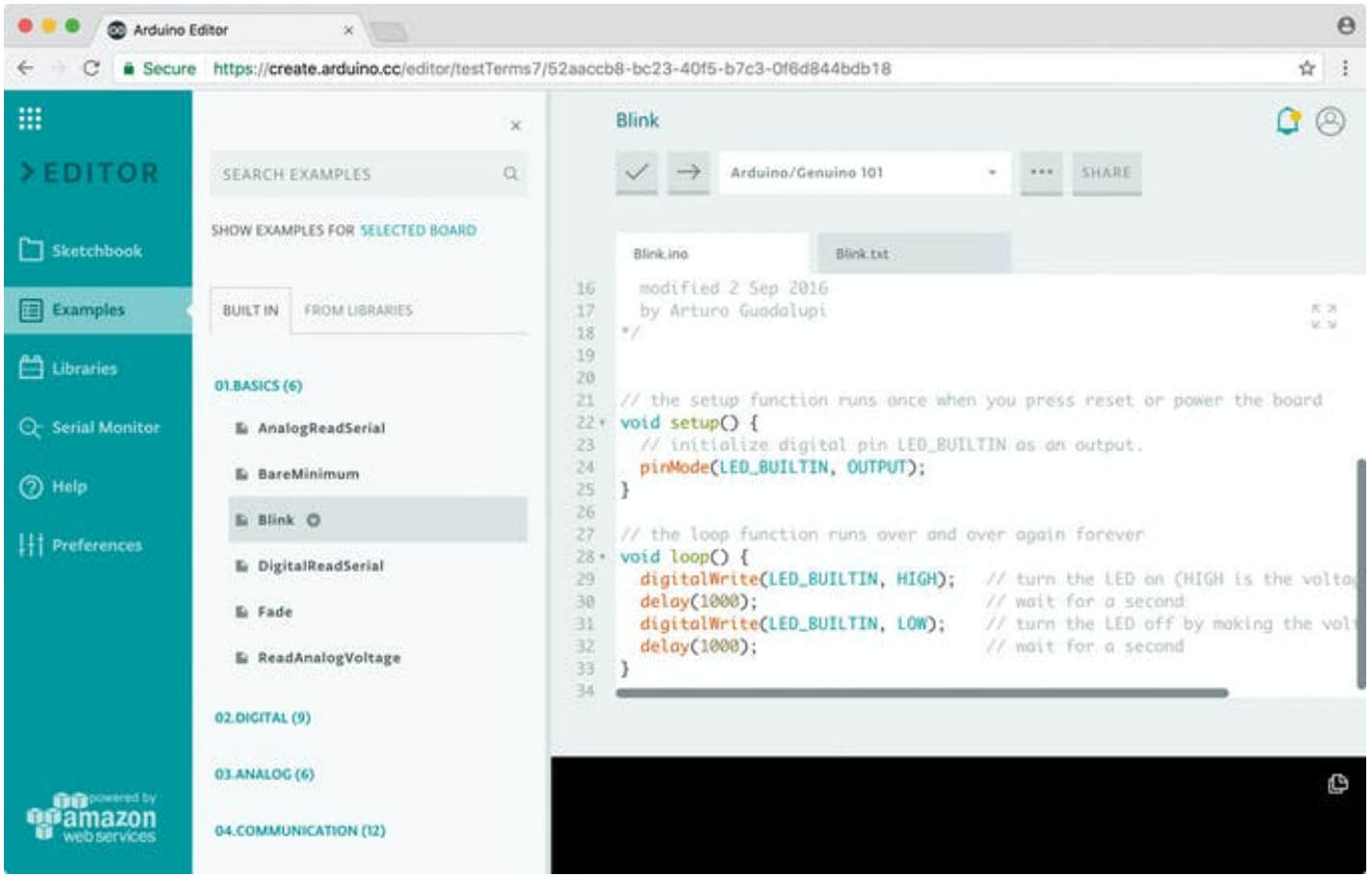
- *LEGO*
- *VEX*
- *Arduino web-IDE*
- *mBlock IDE*
- *Scratch*
- *OpenRoberta Lab*

Scratch

- MIT, https://scratch.mit.edu
- Abierto, gratuito
- Bloques visuales
- Muy intuitivo
- Conceptos básicos de pensamiento computacional
- Gato en una pantalla 2D
- Bloques de control de flujo

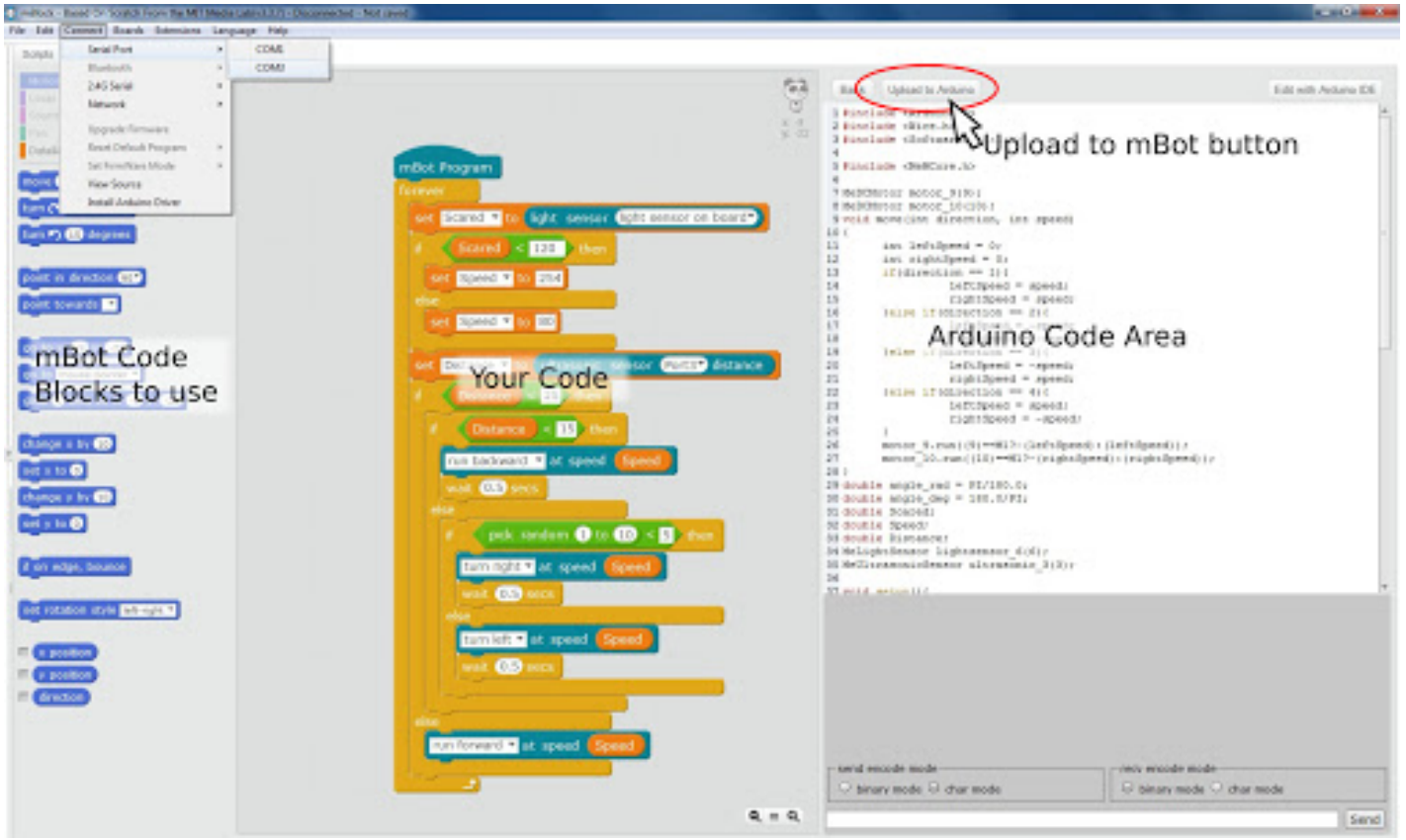


Arduino web-IDE



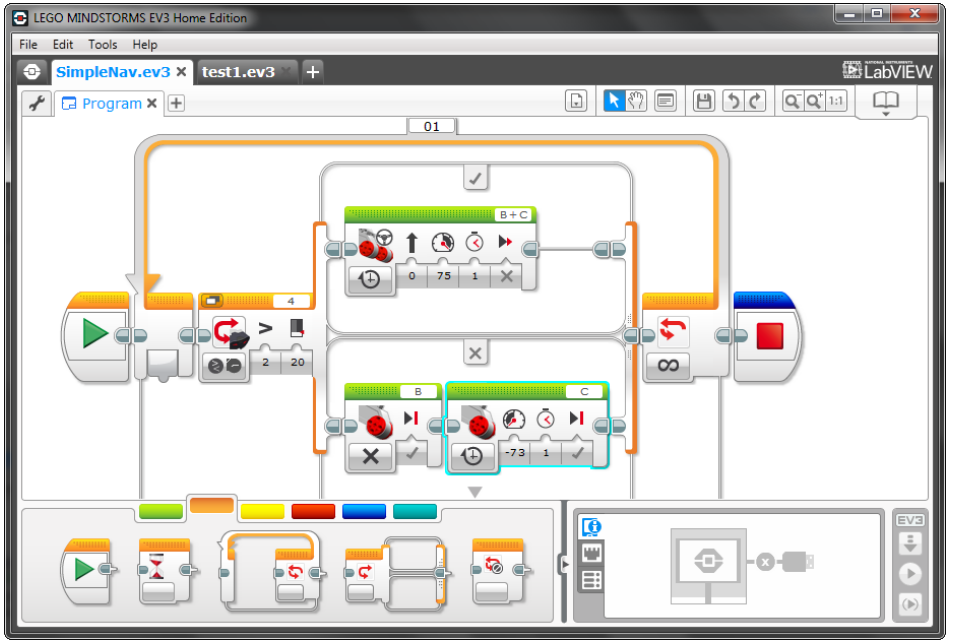
- Versión local y versión web `https://create.arduino.cc/`

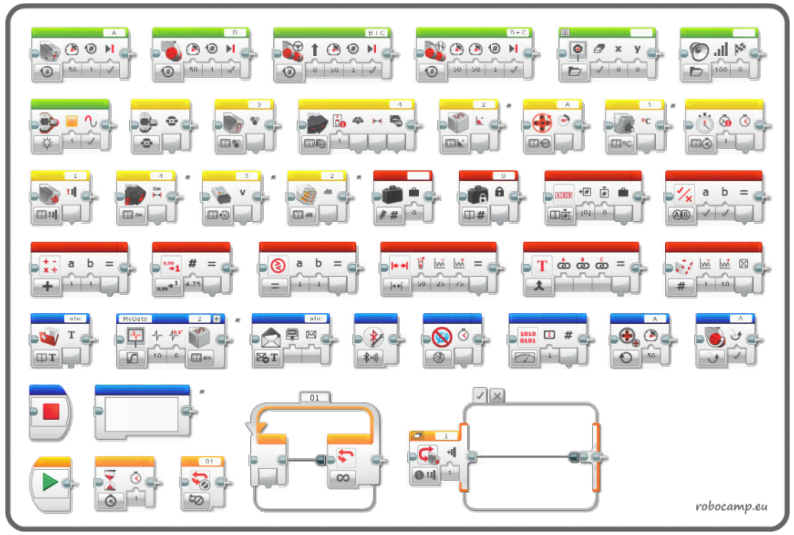
mBlock IDE


mBot Code
Blocks to use
Your Code
Upload to mBot button
Arduino Code Area

LEGO EV3 Home Edition

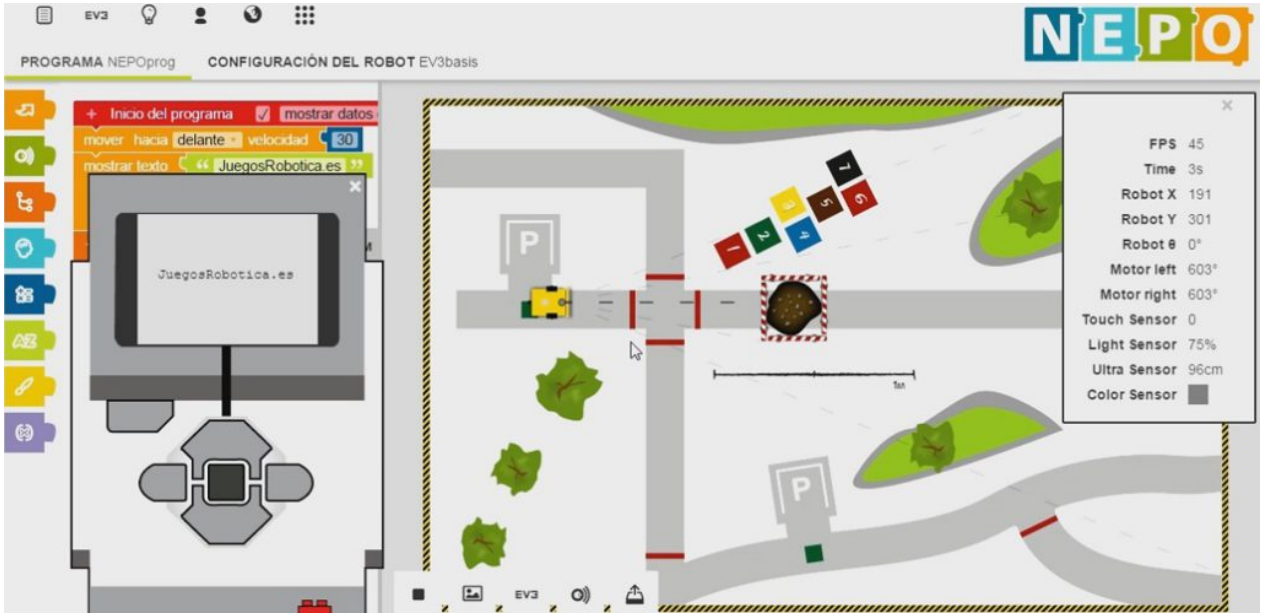
- Almacén bloques visuales
- Intuitivo
- Se descarga en robot LEGO EV3
- Bloques de sensores, de actuadores





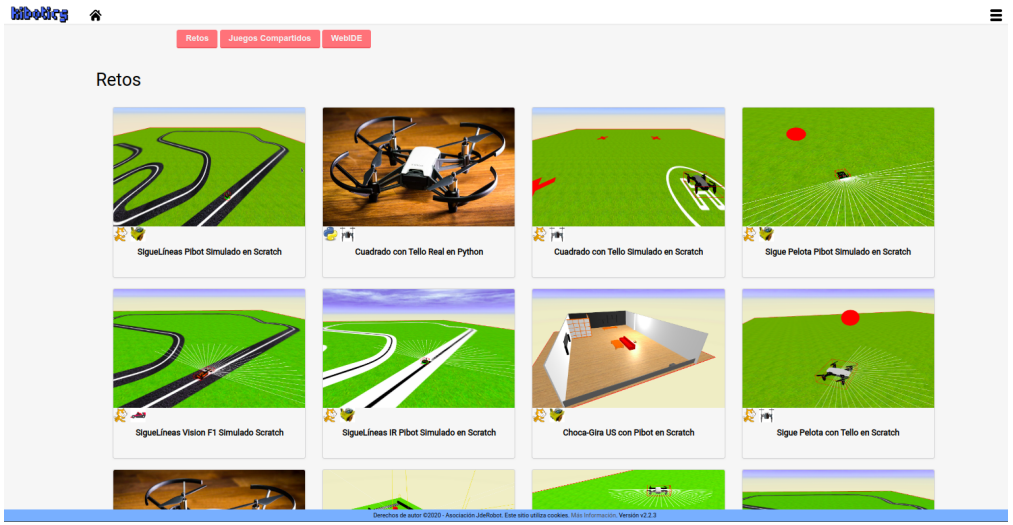
OpenRoberta Lab

- https://lab.open-roberta.org/
- Iniciativa educativa alemana, Fraunhofer IAIS
- Abierto, gratuito
- Soporta varios robots
- Lenguaje visual NEPO
- Simulador 2D



Plataforma Kibotics

- En línea, navegador web
- Primaria, secundaria y bachillerato
- Énfasis en programación
  Lenguajes Scratch y Python
- Robots físicos y simulados
  mBot, drone, PiBot, visión...
- Contenidos educativos
- Interacción social, foro



Plataforma web: http://kibotics.org

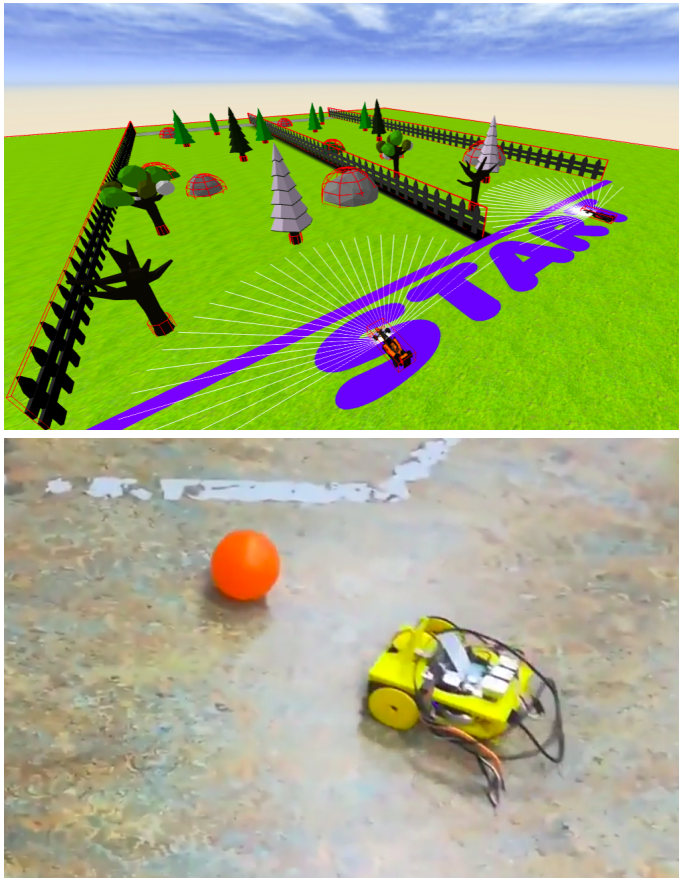
- No hay que instalar nada
- Desde cualquier sitio, a cualquier hora :-)
- Multiplataforma: Linux, Windows, Tablets, MacOS, Android

Ventajas: cuándo y dónde quieras

- Incluye evaluación automática
- Completamente en línea
- Los alumnos pueden acceder también fuera de las horas de clase
- Pueden acceder desde clase y también desde su casa
- Se puede enseñar robótica sin (o con menos) robots físicos
- Ahorro de costes

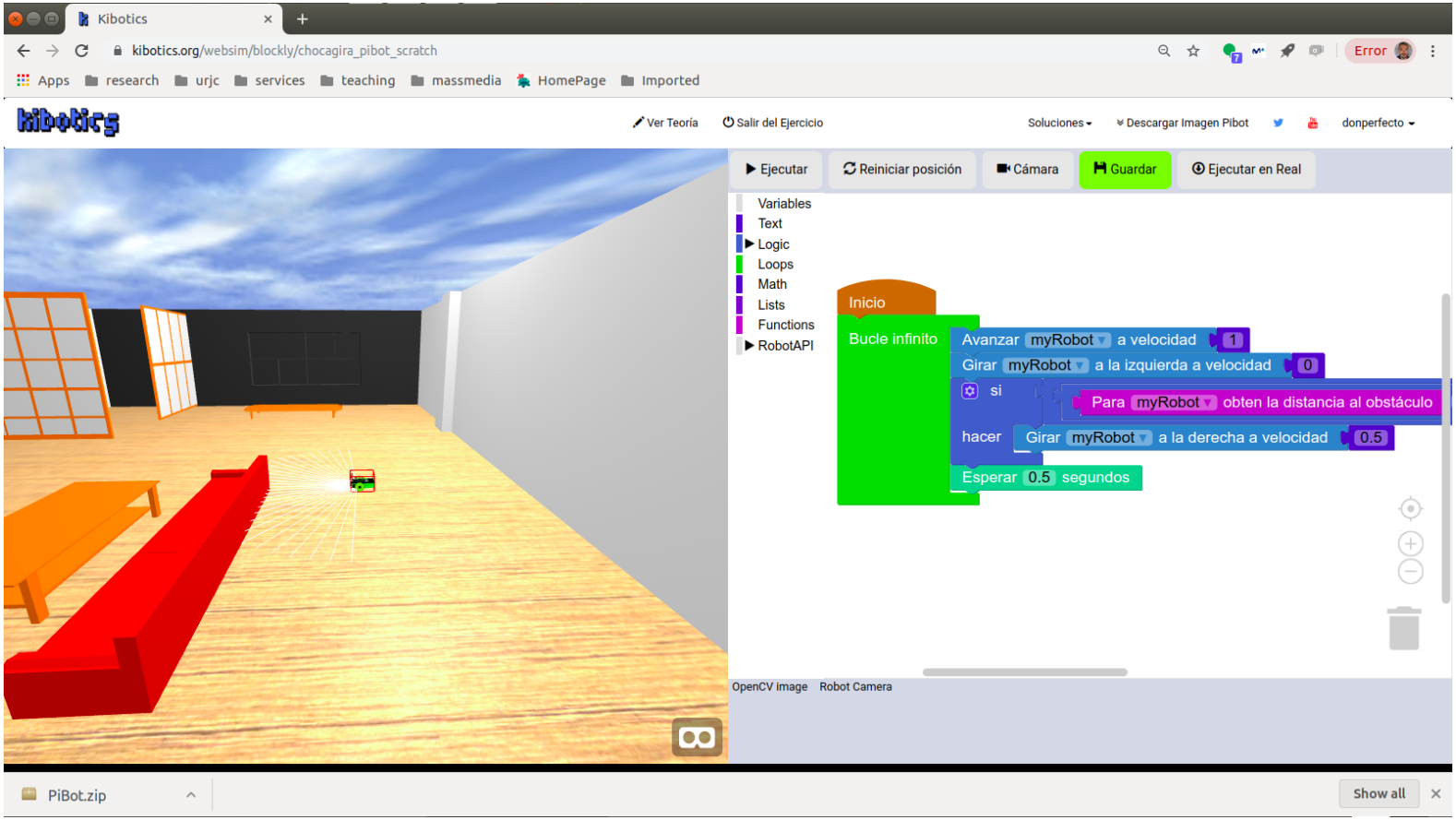
Contenidos educativos

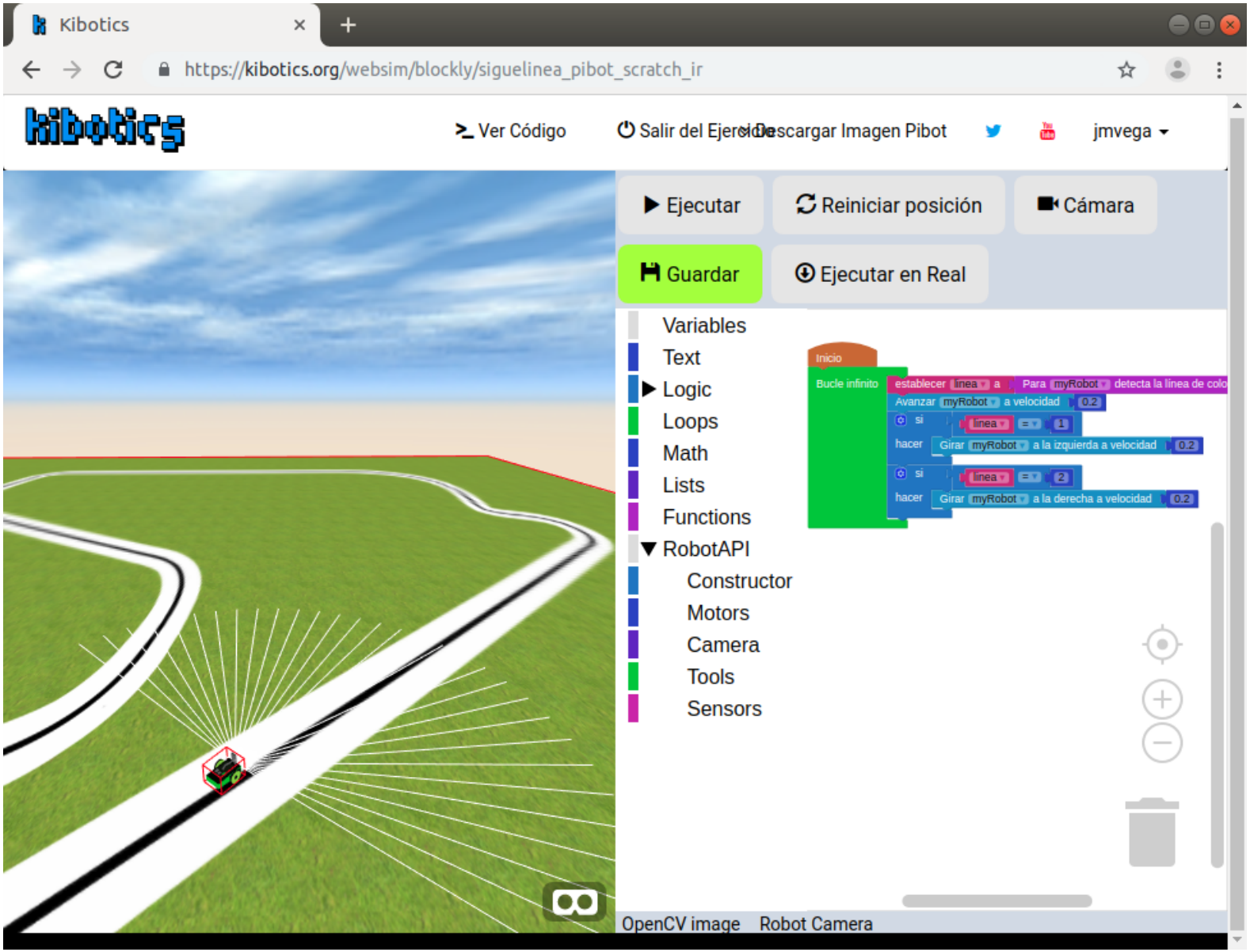
- Adaptados a curricula oficiales
- *Learn by doing*
- ¡Divertidos! *Gamificación*
- Complejidad dosificada
- Ejercicios prácticos
  - Choca-gira con US
  - Sigue-líneas con IR
  - Sigue objeto con visión ...
- Lecciones de teoría



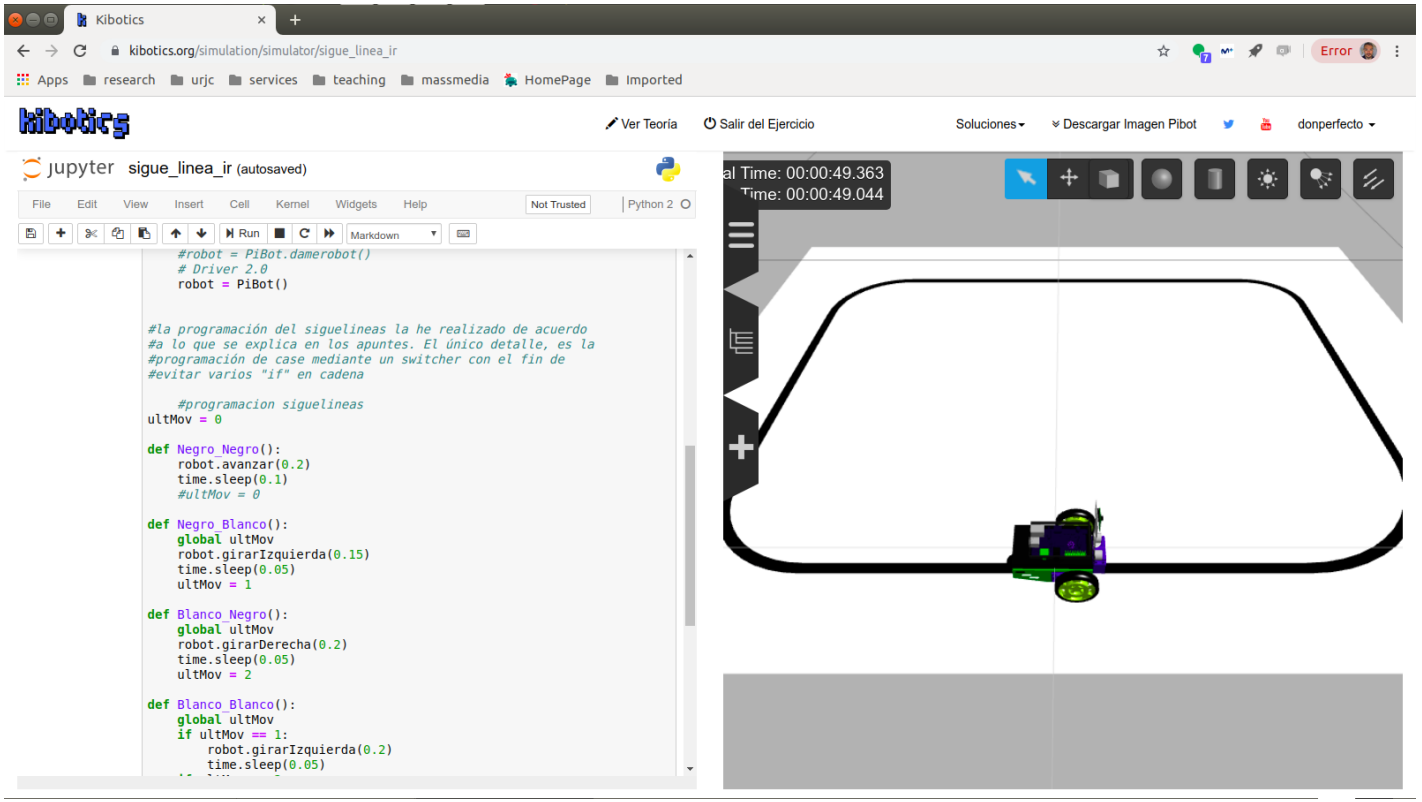
Lenguajes de programación: Scratch y Python

- Scratch: lenguaje visual, fácil de aprender

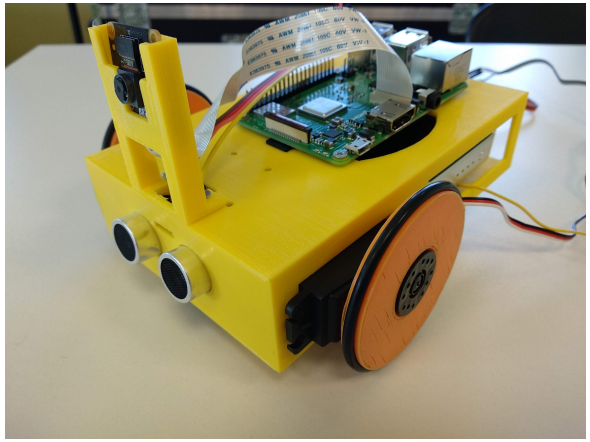


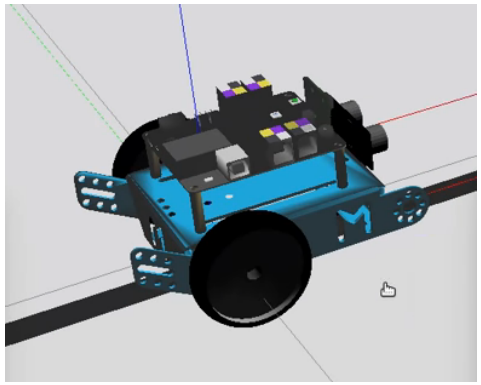

Kibotics
https://kibotics.org/websim/blockly/siguelinea_pibot_scratch_ir
kibotics
Ver Código
Salir del Ejercicio
Descargar Imagen Pibot
jmvega
Ejecutar
Reiniciar posición
Cámara
Guardar
Ejecutar en Real
Variables
Text
Logic
Loops
Math
Lists
Functions
RobotAPI
Constructor
Motors
Camera
Tools
Sensors
Inicio
Bucle infinito
establecer linea a Para myRobot detecta la línea de colo
Avanzar myRobot a velocidad 0.2
si linea = 1
hacer Girar myRobot a la izquierda a velocidad 0.2
si linea = 2
hacer Girar myRobot a la derecha a velocidad 0.2
OpenCV image
Robot Camera

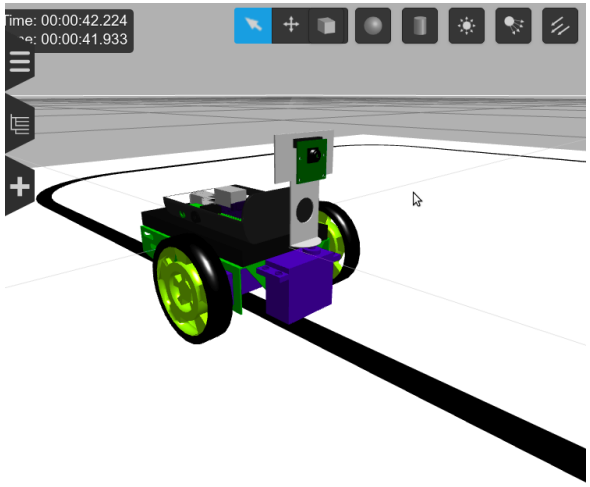
- Python: lenguaje de texto, sencillo pero potente
- Se usa también en la universidad y trabajo

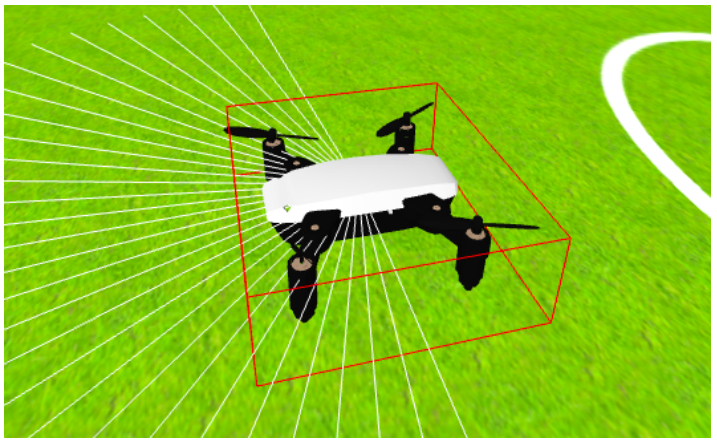


Múltiples robots soportados: mBot, drones,...









Casos de éxito

- Escuela de Pensamiento Computacional INTEF (2019)
- Ayuntamiento de Fuenlabrada (2019, 2020)
- Empresa Logix5 (2019)
- Universidad Rey Juan Carlos (2019)
- IES Martinez Uribarri (Salamanca, curso 2019-2020)
- Comunidad de Madrid (2020)

...

Educación superior en robótica, profesionales

Demanda creciente de formación en robótica

- Nuevas aplicaciones, productos y servicios robóticos
- Cada vez más ofertas laborales en este ámbito
- Ingenierías, profesionales
- Investigación
- $robot = hardware + software$
- $hardware = sensores + actuadores + computadoras$

Universidades

- Doctorado
- Másteres en las escuelas de ingeniería
  - Ing. Industrial, Ing. Informática, Ing. Telecomunicación...
  - Electrónica, Mecánica, Programación, Inteligencia Artificial
- Grados específicos
  - U.Alicante: Grado en Robótica
  - U.Rey Juan Carlos: Grado en Robótica Software

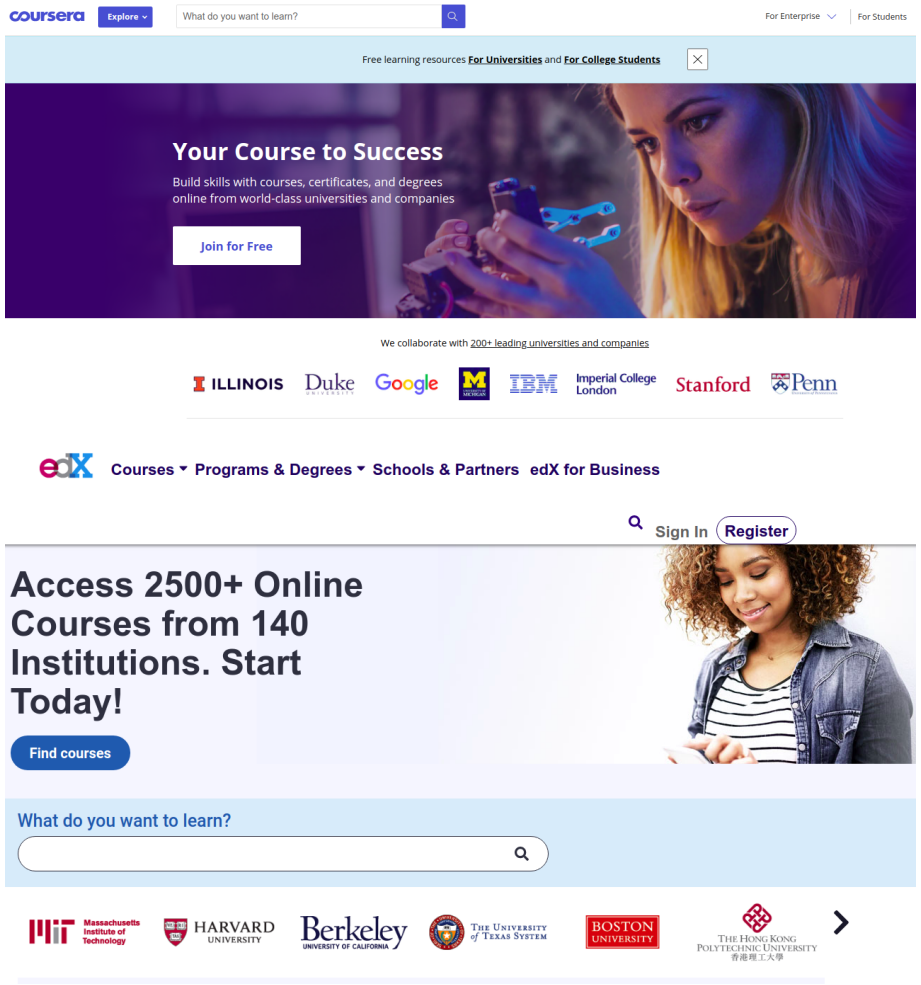
URJC: Grado Ingeniería en Robótica Software

- https://www.urjc.es/estudios/grado/3099-ingenieria-de-robotica-software
- Fundamentos genéricos ingeniería
- Fundamentos específicos de robótica
- Dominios de aplicación

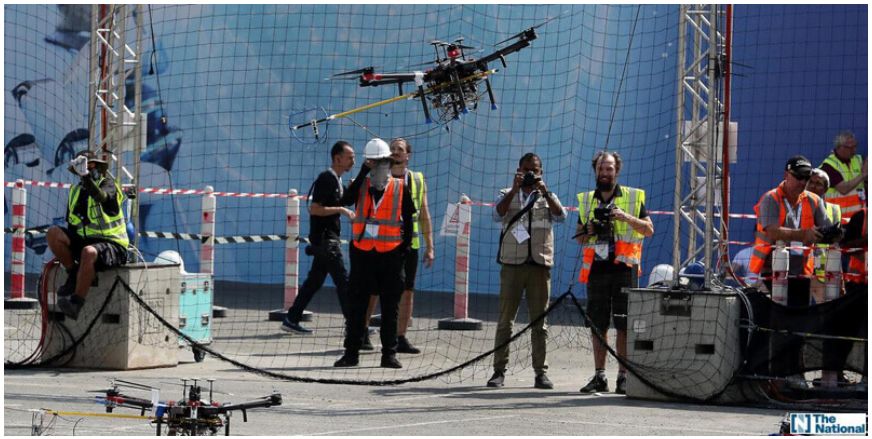
Curso 1º (60 ECTS)	**Curso 2º** (60 ECTS)	**Curso 3º** (60 ECTS)	**Curso 4º** (60 ECTS)
Emprendimiento e innovación en Robótica	Probabilidad y estadística	Robótica móvil	Aprendizaje Automático y Profundo
Álgebra	Fundamentos de Redes de Ordenadores	Sistemas distribuidos y concurrentes	Mecatrónica
Fundamentos físicos de la ingeniería	Diseño Software	Inteligencia Artificial	Robótica de Servicios
Fundamentos de la Programación	Sensores y actuadores	Redes de ordenadores para robots y máqui	Robótica aérea
Arquitectura de computadores	Idioma Moderno	Ingeniería de Control	RAC
Cálculo	Deontología y normativa en robótica	Planificación y Sistemas Cognitivos	TFG
Evolución y futuro de la Robótica	Ampliación de matemática aplicada	Sistemas empotrados y de tiempo real	Practicas
Algoritmos y estructuras de datos	Arquitecturas SW para robots	Robótica Industrial	
Electrónica digital	Fundamentos de automática	Visión artificial	
Laboratorio de Sistemas	Sistemas Operativos	Modelado y Simulación de robots	

Formación no reglada

- Internet
- Masivo, en linea
- Cursos
- MOOCs:
  - Coursera
  - EDx
- Empresas:
  - Udemy
  - Udacity
  - TheConstruct



Competiciones robóticas internacionales



- Robocup: soccer, home, rescue...
- DARPA Robotics Challenge
- MBZIRC: drones...
- Program-a-Robot

Plataforma Robot Operating System (ROS)

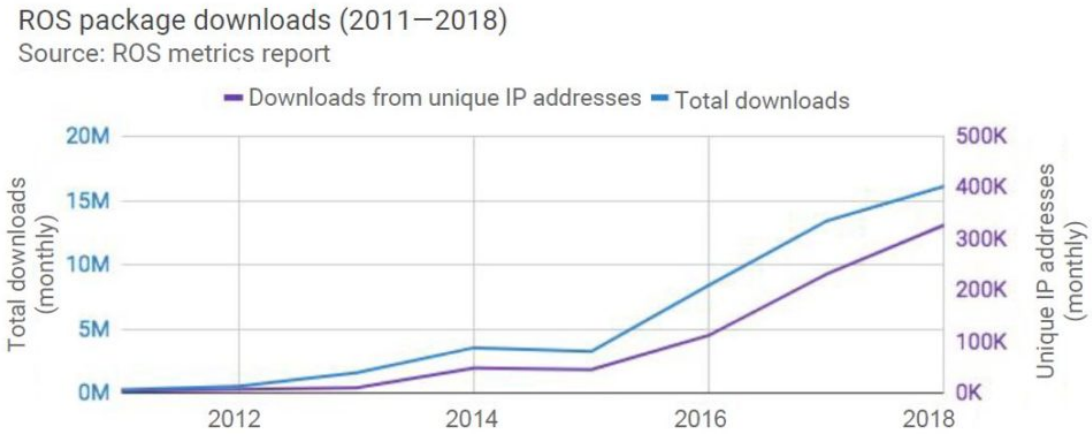
- *Middleware* para software robótico
- Evitar la reinvención de la rueda: comunicaciones, drivers...
- Gratis y software libre: `https://ros.org`
- Colección de paquetes
- C/C++, Python
- Comunidad enorme (usuarios, desarrolladores, soporte...)
- Standard de facto en robótica de servicios
- Sobre Linux principalmente

Un poco de historia (2006-2019)

- Stanford (-2008)

  Personal Robotics Program
- WillowGarage (-2014)

  Gazebo, Turtlebot
- OSRF (-2017)
- OpenRobotics (-today)

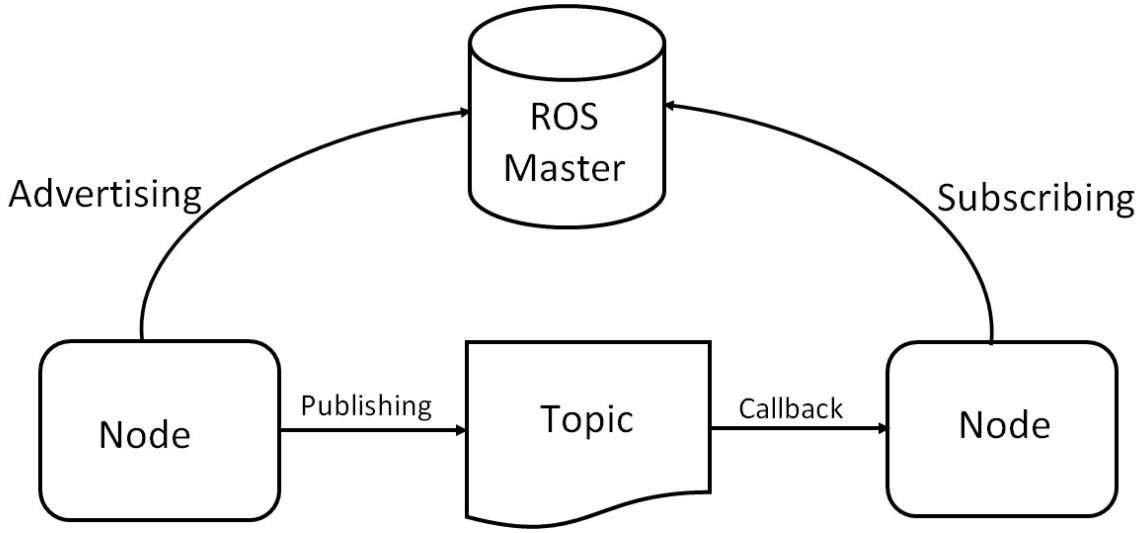


Características

- Aplicaciones robóticas distribuidas en nodos que se comunican
- Standard robot messages
- Drivers
- Herramientas
- Capacidades
- Aumenta la interoperación y reutilización de sw robótico
- Muchos robots soportados

Comunicaciones

- *Topics*: publicación-suscripción, asíncronos y anónimos
- *Services*: RPC, bloqueantes
- *Actions*: interrumpibles



Herramientas

- `ROSbags`, recording and playback
- `RViz`, visualizador 3D
- `Rqt-graph`, grafo de cómputo

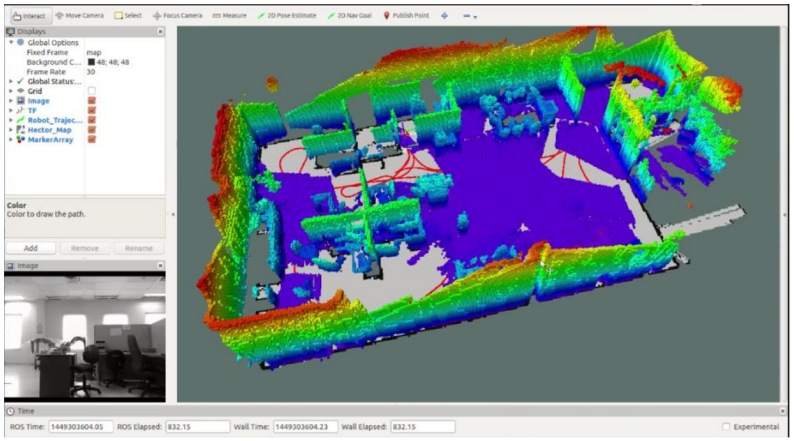
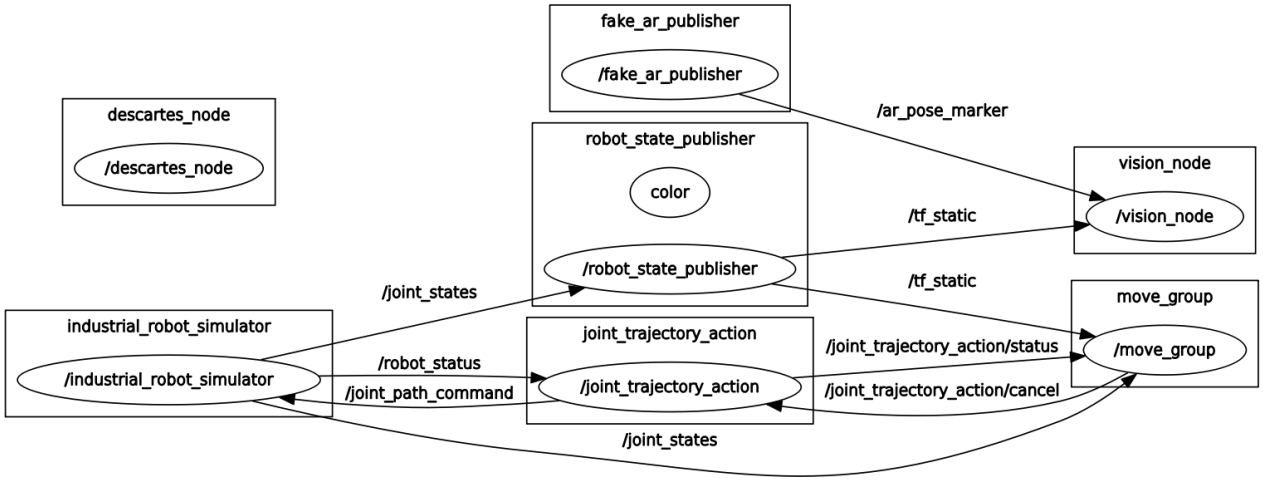
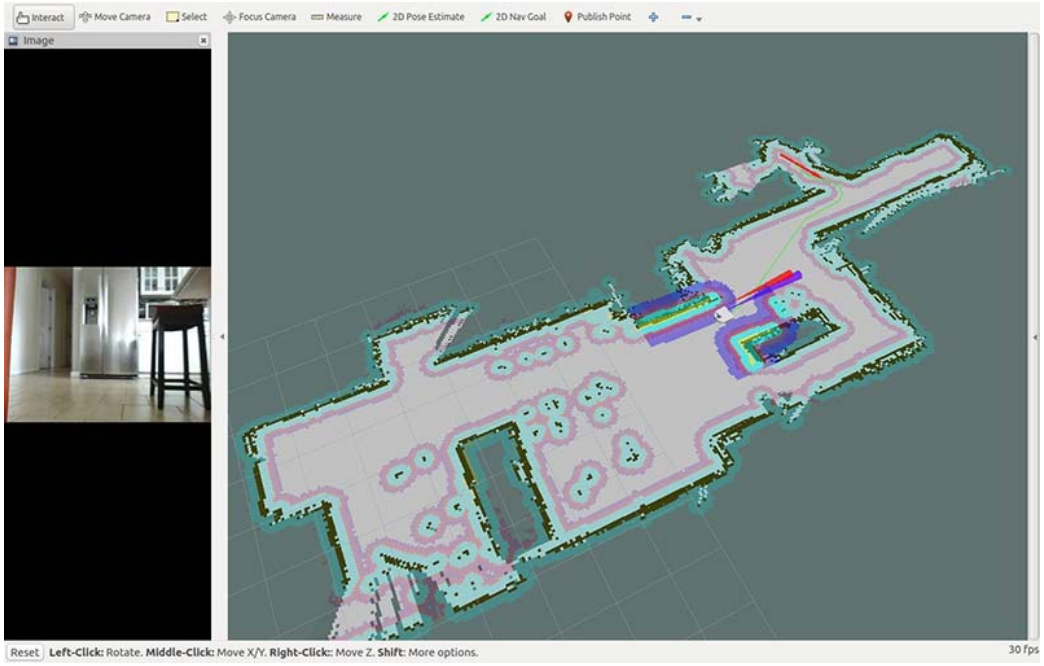


Figure 7. Visualize 2D & 3D occupied map using RVIZ



Capacidades

- Nodos (*stacks*) con implementación de algoritmos punteros
- Localización
- Construcción de mapas
- Navegación



Tendencias

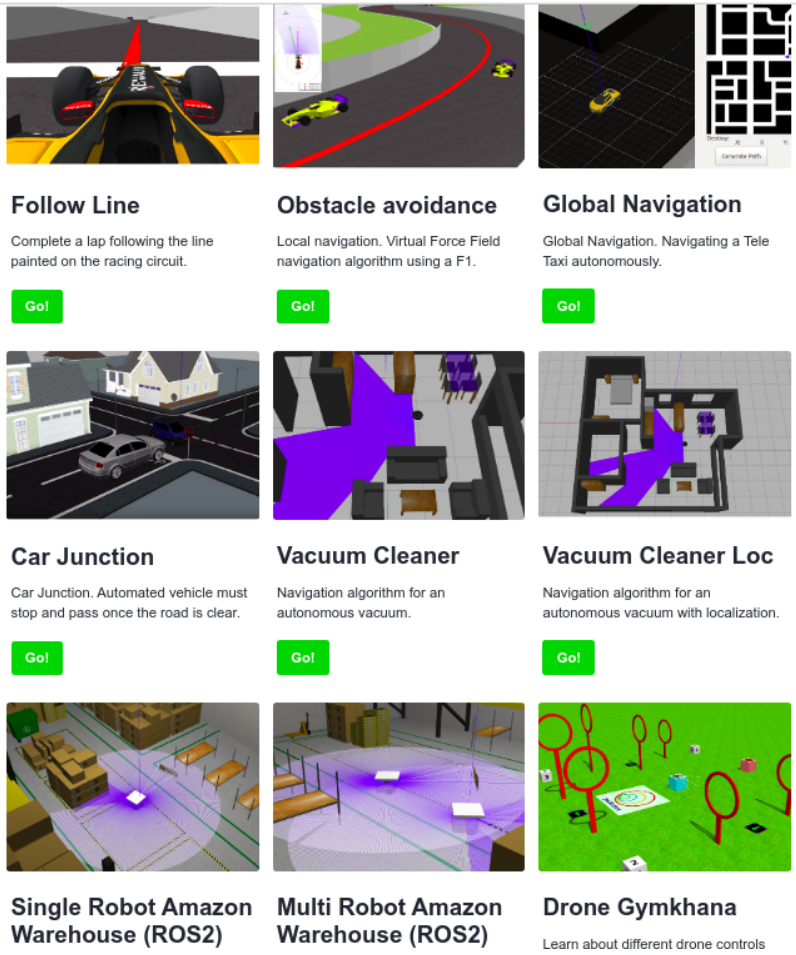
- ROS-Industrial
- ROS2: DDS
  - security
  - real-time
  - no single point of failure (roscore)
  - Multiplataforma
    Linux,Windows,MacOS
- Ignition simulator
  - cloud

Plataforma RoboticsAcademy

- https://jderobot.github.io/RoboticsAcademy/
- Contenidos educativos
- *Learn by doing*
- ¡Divertidos! *Gamificación*
- Lenguaje Python
- Simulador Gazebo
- Middleware ROS
- Foro
- Open source, internacional

Contentidos educativos: conjunto de ejercicios

- Conducción autónoma
- Robótica móvil
- Drones
- Robots industriales
- Visión Artificial



Visual Follow Line

Versions to run the exercise

Currently, there are 2 versions for running this exercise:

- ROSNode Templates
- Web Templates(Current Release)

The instructions for both of them are provided as follows.

Goal

The goal of this exercise is to perform a PID reactive control capable of following the line painted on the racing circuit.

Gallery

The students program a Formula1 car in a race circuit to follow the red line in the middle of the road.

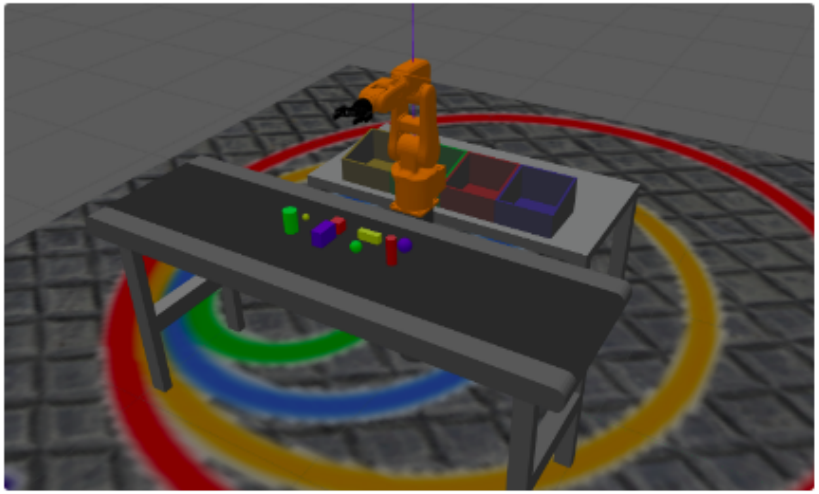
Instructions for Web Templates

This is the prefered way for running the exercise.

TOC Visual Follow Line

Pick and Place

The goal of this exercise is to learn the underlying infrastructure of Industrial Robot exercises(ROS + MoveIt + our own industrial robotics API) and get familiar with the key components needed for more complex exercises by completing the task of pick and place multiple objects and classify them by color or shape.



Gallery.

Installation

1. Install the General Infrastructure of the JdeRobot Robotics Academy.
2. Install MoveIt

```
sudo apt install ros-melodic-moveit
sudo apt-get install ros-melodic-ros-control ros-melodic-ros-cont
```