
Software libre en robótica: ROS y JdeRobot

José María Cañas

josemaria.plaza@urjc.es



@RoboticsLabURJC, @JdeRobot

Contenidos

- La robótica es útil (y mola!)
- Software en robótica
- Un caso de (mucho) éxito: ROS
- JdeRobot

La robótica es útil (y mola!)

- Robótica ficción vs Robótica real
- *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



¿Qué es un robot? Componentes



Sistema informático con:

- Sensores
- Actuadores
- Computador

Hay que **programarlo** para que consiga sus objetivos y sea sensible a la situación.

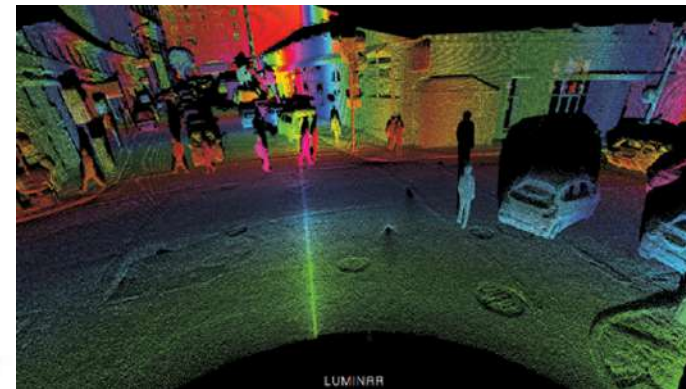
La inteligencia reside en su software

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++, python...
- Los sistemas robóticos son sistemas complejos (el tamaño importa)

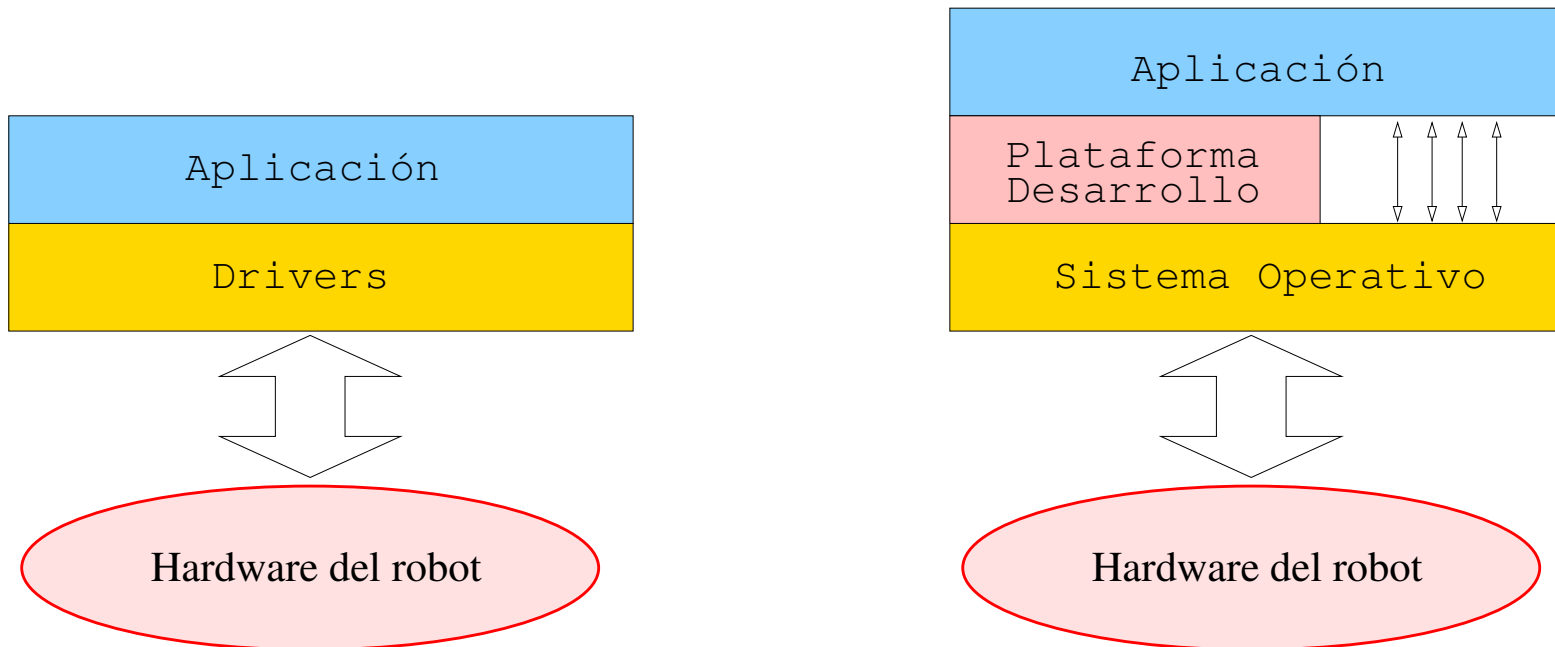
Requisitos específicos

- Vivacidad, agilidad (tiempo real)
- Multitarea (conurrencia, múltiples fuentes de actividad)
- Distribuido, comunicaciones
- Interfaz gráfica, depuración
- Expandible
- Conectado a la realidad física
- Heterogeneidad dispositivos hardware
- Encapsular funcionalidad o comportamientos es difícil

Tendencias

- Antes: cada robot su entorno de programación
- Ingeniería software: orientación a objetos, distribución
- Software orientado a componentes
- Interfaces explícitos
- Reutilizar software es difícil, pero muy ventajoso
- Plataformas software robótico

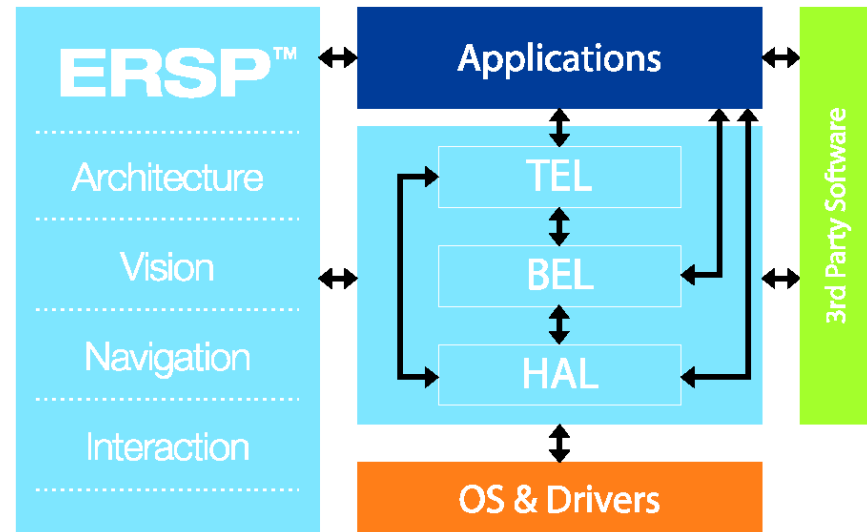
Plataformas de software robótico



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

¿Qué proporciona una plataforma?

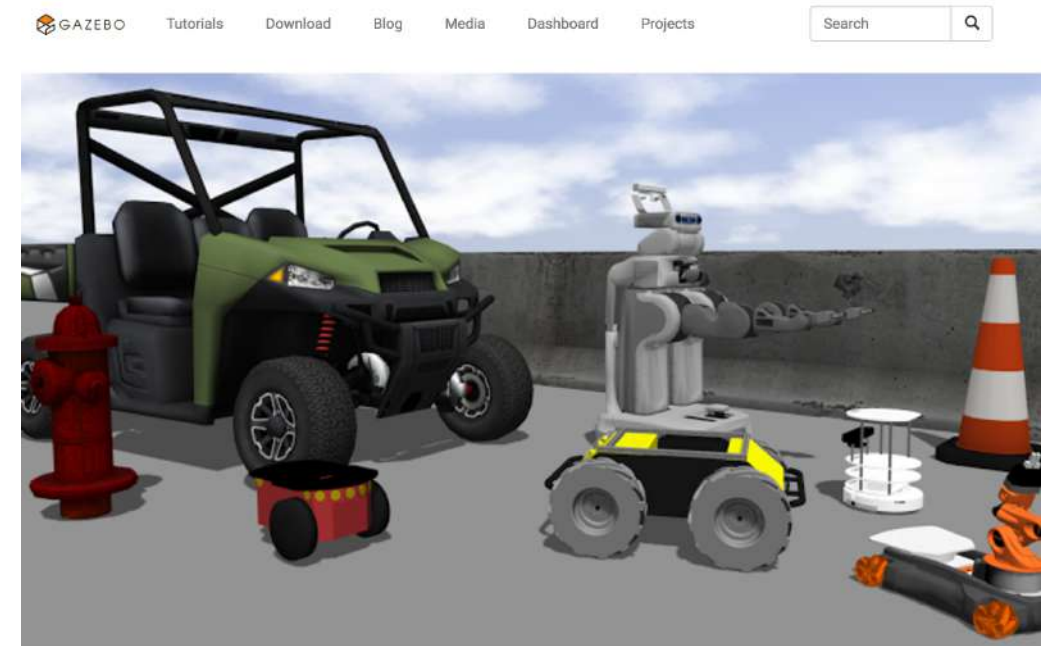
- Abstracción hardware (HAL)
- Arquitectura software
- Funcionalidades de uso común
- Herramientas



- Comerciales, investigación, software libre
- ROS, Urbi, YARP, Orca, OROCOS, Player/Stage, Clarity, MSRS

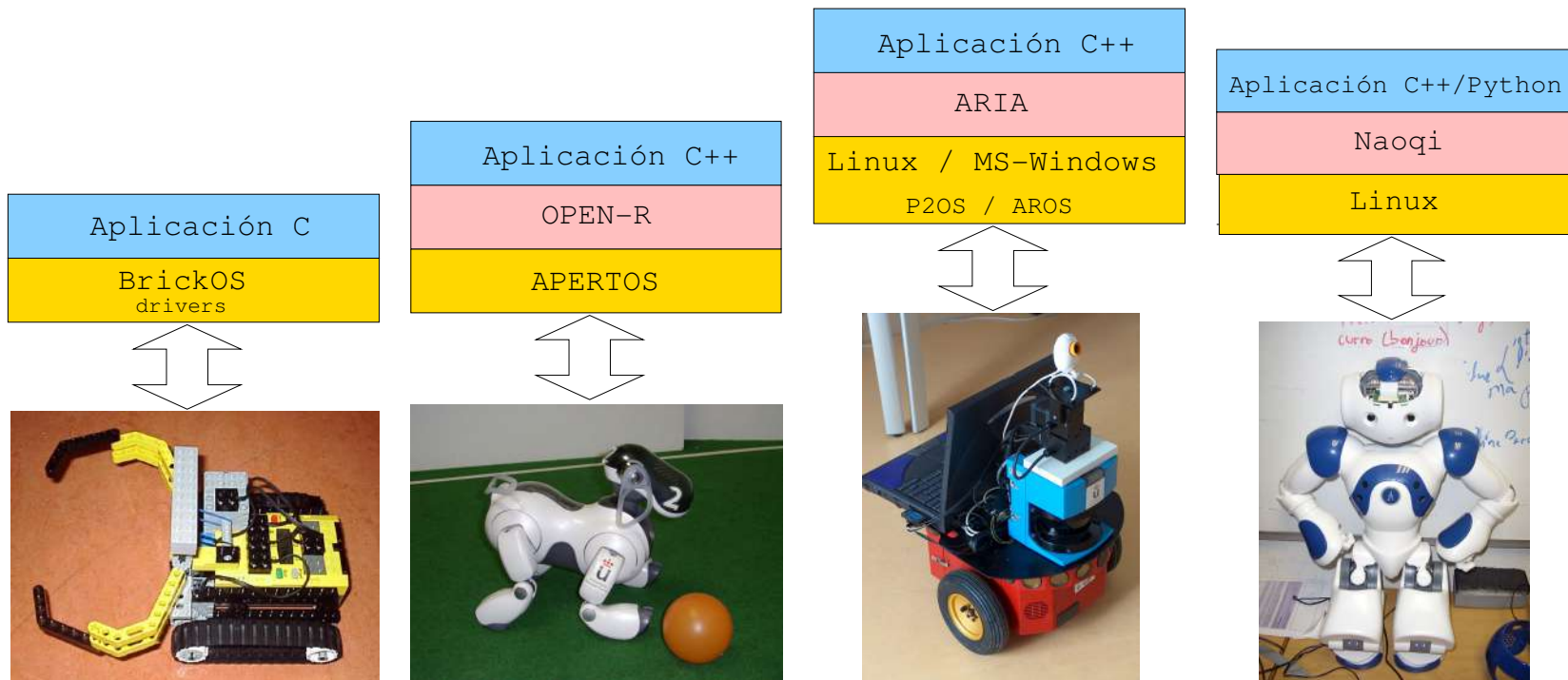
Simuladores

- Madurar algoritmos
- Comodidad trabajar sin robot
- Las caídas no duelen
- Mundo, sensores y actuadores
- Motor físico: ODE



- Gazebo, V-REP, Stage, Webots, MORSE

Ejemplos



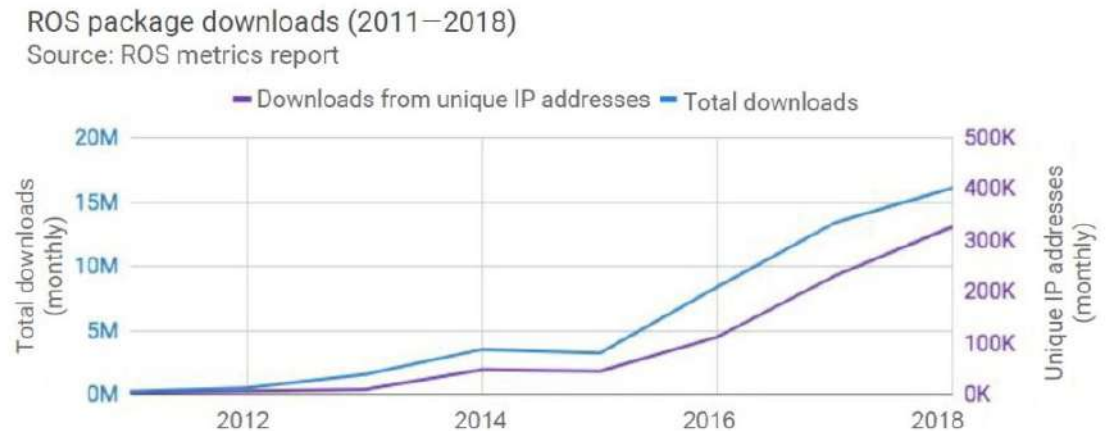
Robot Operating System (ROS)

ROS.org

- *Middleware* para software robótico
- Evitar la reinención 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)

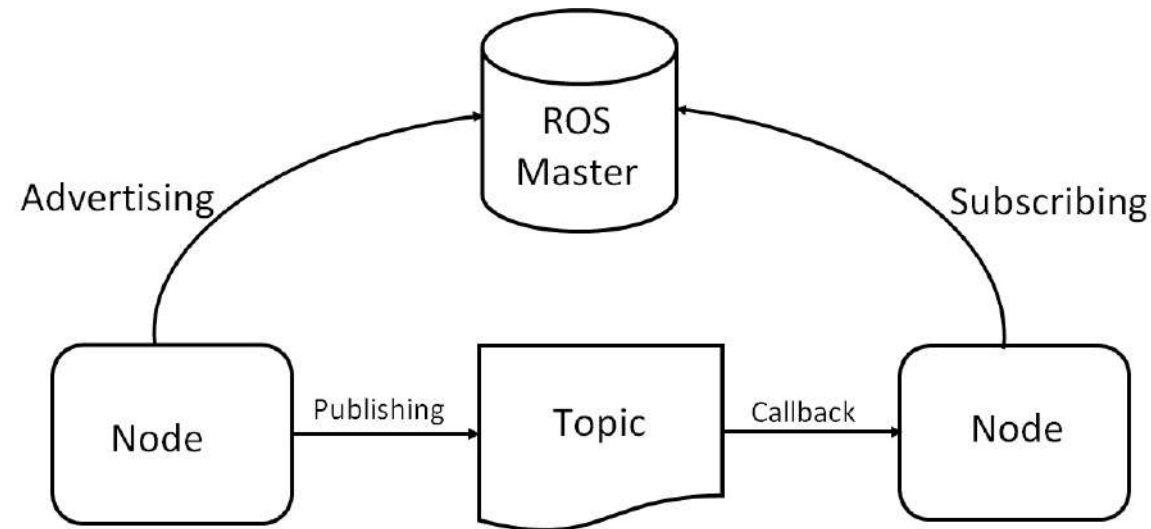


Características

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

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

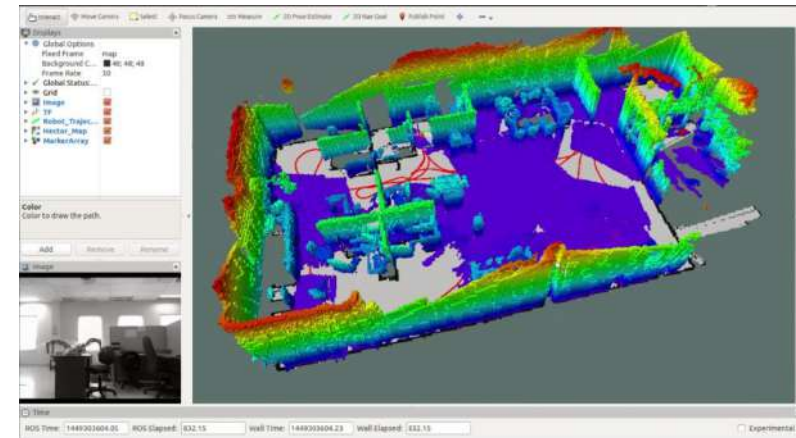
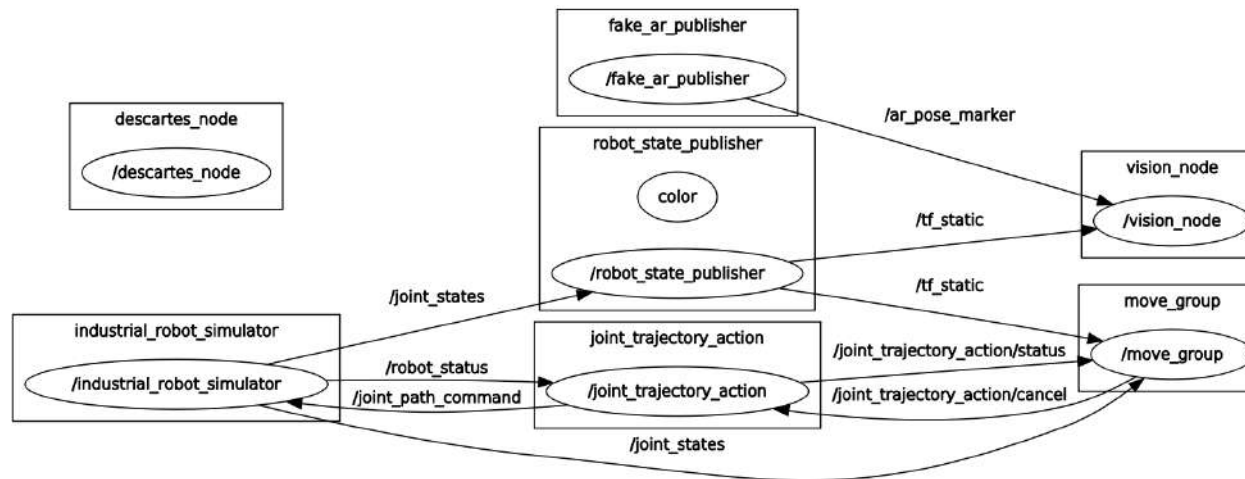
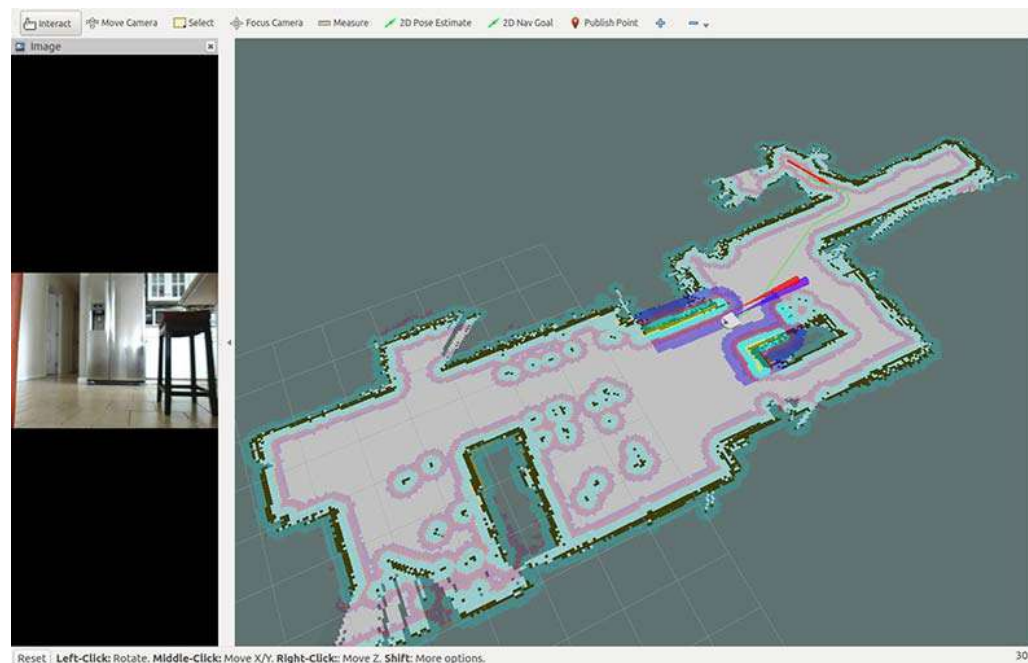


Figure 7. Visualiza 3D & 2D simulada con RVIZ



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)
- Ignition simulator
 - cloud



JdeRobot



- Gratis y software libre: <https://jderobot.org>
- Nació en la URJC
- C/C++, Python, JavaScript
- Linux
- De plataforma a toolkit
- Adaptación a ROS, de drivers ICE a drivers ROS

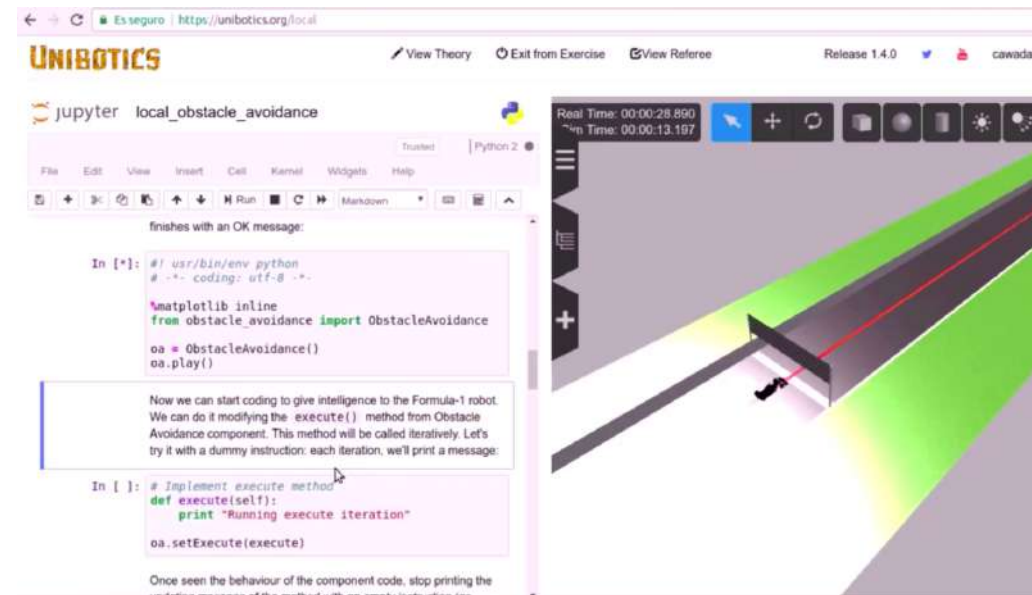
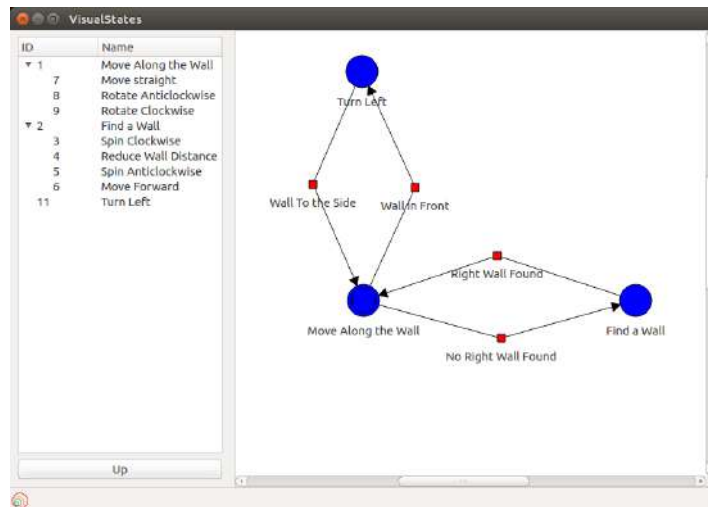
Proyectos

- Herramientas de programación de robots
- *DeepLearning*, redes neuronales
percepción y comportamientos
- FPGAs en robótica
- VisualSLAM
- **Educación en robótica**
RoboticsAcademy, Unibotics
- Drones



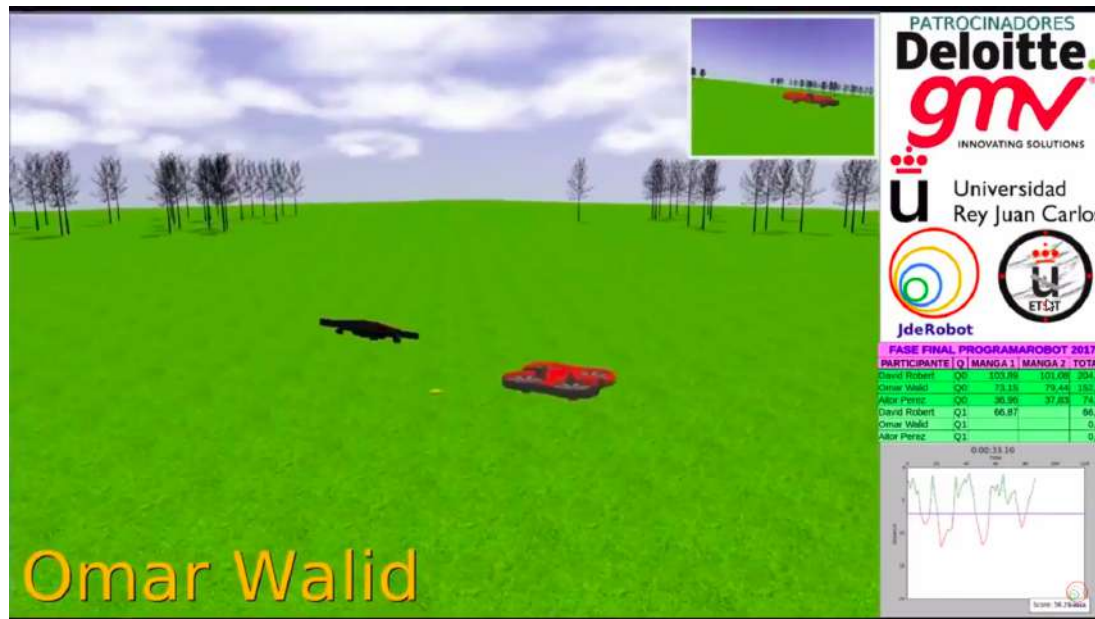
Algunos productos

- VisualStates
- Unibotics
- Conducción autónoma



Actividades

- RoboticsClub
- Competiciones: IROS 2018
- Google Summer of Code 2015,2017,2018,2019



The screenshot displays a 3D simulation of a competition arena with a green field and a line of trees. Two robots are visible: a black one and a red one. The interface includes logos for sponsors: Deloitte, gmV, Universidad Rey Juan Carlos, and JdeRobot. A results table for the 'FASE FINAL PROGRAMAROBOT 2017' is shown, along with a line graph of a robot's path.

FASE FINAL PROGRAMAROBOT 2017				
PARTICIPANTE	Q	MANGA 1	MANGA 2	TOTAL
David Robert	Q0	100,00	100,00	200,00
Omar Walid	Q0	73,33	73,33	146,66
Alfonso Perez	Q0	90,90	90,90	181,80
David Robert	Q1	66,67		66,67
Omar Walid	Q1			0,00
Alfonso Perez	Q1			0,00

Omar Walid

Infraestructura de desarrollo

- Muuuuchas líneas de código (<https://github.com/JdeRobot>)
- De svn a gitlab y github: incidencias y parches
- Documentación: de mediawiki a github pages
- CI-CD, jenkins...
- Slack, Hangouts
- YouTube channel
- Twitter @JdeRobot