

UNIVERSIDAD COMPLUTENSE DE MADRID



## PROYECTO DE INNOVACIÓN

Convocatoria 2022/2023

*Número 313*

### UCM-Robotarium

Desarrollo y construcción de una plataforma de experimentación robótica multiagente para la realización de prácticas presenciales y remotas.

Coordinador: Jesús Chacón Sombría

Facultad de Ciencias Físicas

Departamento de Arquitectura de Computadores y Automática

# Índice

<b>1. Objetivos propuestos en la presentación del proyecto</b>	<b>2</b>
<b>2. Objetivos alcanzados</b>	<b>2</b>
<b>3. Metodología</b>	<b>3</b>
<b>4. Recursos Humanos</b>	<b>3</b>
<b>5. Desarrollo de las actividades</b>	<b>4</b>
5.1. Robotarium-UCM: los robots . . . . .	4
5.1.1. Hardware . . . . .	4
5.2. Software . . . . .	5
5.3. Robotarium-UCM como Laboratorio Remoto . . . . .	6

## 1. Objetivos propuestos en la presentación del proyecto

El objetivo de este proyecto es el montaje de un laboratorio remoto multirobot que nos permita mejorar las prácticas de laboratorio de las asignaturas del área de Ciencias e Ingeniería. Este objetivo lo hemos dividido en dos objetivos generales: 1) construir un Robotarium-UCM, con una flota de 10 robots de bajo coste, un sistema de localización visual basado en marcas y un Robot Arena, y 2) dotar al Robotarium-UCM de capacidades de programación remota.

**Objetivo 1** Construir un Robotarium-UCM, con una flota de 10 robots de bajo coste, un sistema de localización visual basado en marcas y un entorno (mesa) de trabajo. Para realizarlo deberemos alcanzar los siguientes subobjetivos:

- O1.1** Tener 10 robots funcionales en el robotarium.
- O1.2** Montar y poner en marcha la cámara cenital de localización.
- O1.3** Poner en marcha el sistema de localización de los robots.
- O1.4** Montar la pista de trabajo de los robots: Robot Arena.
- O1.5** Desarrollar la API de programación que usarán los estudiantes.

**Objetivo 2** Dotar al Robotarium-UCM de capacidades de programación remota. Para realizarlo deberemos alcanzar los siguientes subobjetivos:

- O2.1** Identificar una práctica piloto para realizar con el Robotarium-UCM.
- O2.2** Habilitar la programación y visualización remota de la práctica seleccionada.

## 2. Objetivos alcanzados

Los objetivos propuestos al inicio del proyecto han sido alcanzados satisfactoriamente, como se refiere a continuación. El objetivo **O1**, construir un Robotarium-UCM, con una flota de 10 robots de bajo coste, un sistema de localización visual basado en marcas y un entorno (mesa) de trabajo, se ha alcanzado a través de la consecución de todos sus objetivos específicos. El objetivo **O1.1** se ha cumplido satisfactoriamente, ya que actualmente disponemos de 10 robots funcionales en el robotarium, 5 de tipo A y 5 de tipo B. Así mismo, hemos alcanzado los objetivos **O1.2** y **O1.3**, la cámara cenital está actualmente en funcionamiento y junto con el software desplegado en el Robotarium-UCM proporciona la localización de los robots mediante marcadores ArUCO. La pista de trabajo de los robots: *Robot Arena* se ha situado en el laboratorio 02.237 de la facultad de Ciencias Físicas, perteneciente al departamento de Arquitectura de Computadores y Automática. Finalmente, el desarrollo de la API de programación que utilizarán los estudiantes (**O1.5**), aunque por su naturaleza es un trabajo en continuo desarrollo, ha alcanzado un nivel de madurez razonable para dar por alcanzado el objetivo.

Respecto al segundo objetivo general (**O2**), dotar al Robotarium-UCM de capacidades de programación remota, también se ha alcanzado. Se selecciono como práctica piloto y se implemento (**O2.1**) un experimente basado en el problema de *rendezvous*, también conocido como problema de consenso, en el que  $n$  robot móviles deben desplazarse a una localización común utilizando control distribuido. Este problema es un punto de entrada al problema más complejo del control distribuido en Robótica.

Además, la plataforma Robotarium-UCM permite actualmente funciones básicas de programación y visualización remota (**O2.2**) de los robots. No obstante, queda trabajo por delante para mejorar las capacidades del Robotarium-UC, y se prevé alcanzar un grado de madurez y robustez mucho mayor a lo largo del siguiente curso académico.

### 3. Metodología

El trabajo se estructuró en las siguientes tareas:

**Tarea 1** Revisión bibliográfica sobre bancos de pruebas multirobots accesibles remotamente para analizar el hardware empleado (O1.2, O1.3, O1.4) y el software desarrollado (O1.5).

**Tarea 2** Montar los robots que se incorporan al robotarium (O1.1).

**Tarea 3** Seleccionar una cámara y un sistema de marcas para la localización de los robots (O1.3).

**Tarea 4** Programar el ordenador central para que realice las tareas de comunicación y localización (O1.3).

**Tarea 5** Desarrollar la API para que los estudiantes puedan programar los robots para las prácticas (O1.5).

**Tarea 6** Revisar las prácticas de las diferentes asignaturas y seleccionar aquellas que podrían ampliarse/mejorarse con el uso de robots en grupo. Seleccionar una de ellas que será nuestra práctica piloto (O2.1)

**Tarea 7** Desarrollar, realizar y evaluar la práctica piloto seleccionada (O2.1)

**Tarea 8** Diseñar y desarrollar la programación remota de los robots a través de la herramientas ReNoLabs y EJS (O2.2)

**Tarea 9** Publicitar y publicar el Robotarium-UCM, tanto en nuestro ámbito más próximo (la Facultad de Ciencias Físicas) como en el mundo académico educativo (mediante publicaciones en conferencias o revistas especializadas en herramientas educativas).

### 4. Recursos Humanos

Las personas que han participado en la realización de este proyecto se listan a continuación:

**Responsable** Prof. Jesús Chacón Sombría.

**PDI** UCM.

- Prof. Eva Besada Portas
- Prof. Lía Garcia Perez
- Prof. Juan Francisco Jiménez Castellanos
- Prof. Jose Antonio López Orozco
- Prof. Segundo Esteban San Román
- Prof. Raúl Fernández Fernández

**PAS** UCM.

- Félix Marquez Vidal
- David Sánchez Foces
- Leticia Docampo Granados

## Alumnos UCM.

- Alejandro Gutierrez Fontan
- Daniela Alejandra Goncalvez López
- Leonor Real López
- Gonzalo Carazo Barbero

Las responsabilidades y participación de los miembros del proyecto en cada tarea fueron distribuidas atendiendo a sus experiencias previas:

- Las tareas 1 y 6 se realizaron de forma coordinada entre todos los profesores participantes, por medio de reuniones preliminares en las que se aportó toda la información disponible sobre las prácticas y se acordó una práctica piloto.
- Los profesores Juan Francisco Jiménez y Lía García fueron los responsables de coordinar las tareas 2, 3, 4.
- El profesor Segundo Esteban fue responsable de coordinar la tarea 5.
- Los profesores Jesús Chacón, José Antonio López Orozco y Eva Besada coordinaron las tareas 7 y 8, debido a la experiencia de los tres en el desarrollo y aplicación de diferentes metodologías para la creación de laboratorios remotos y virtuales, la participación en el desarrollo de la herramienta existente, y la experiencia de los dos últimos en metodologías docentes y del primero en el diseño y desarrollo de aplicaciones web.
- El personal técnico compuesto por Alejandro Gutierrez Fontán (cuyo trabajo fin de grado es la semilla de este proyecto) David Sánchez y Félix Márquez proporcionaron el apoyo necesario para realizar diferentes labores: instalación de las herramientas necesarias, montaje del hardware de las nuevas prácticas, conexión de los equipos a la red, y otras dificultades técnicas que surgieron en el transcurso del proyecto.
- Los alumnos que participaron en el proyecto, Leticia Docampo, Leonor Real, Daniela Goncalves y Gonzalo Carazo colaboraron en la puesta en marcha de los robots y del sistema de localización, bajo la supervisión de los profesores Juan Francisco Jiménez y Lía García.
- Finalmente, la Tarea 9 fue realizada por todos los profesores del proyecto, y la documentación final del proyecto fue responsabilidad del coordinador del mismo. Para dar visibilidad al trabajo, se presentó en la reunión anual del *Grupo de Educación en Automática* del *Comité Español de Automática* (CEA) [2], y en el congreso internacional *Robotics in Education* (RiE) [3].

## 5. Desarrollo de las actividades

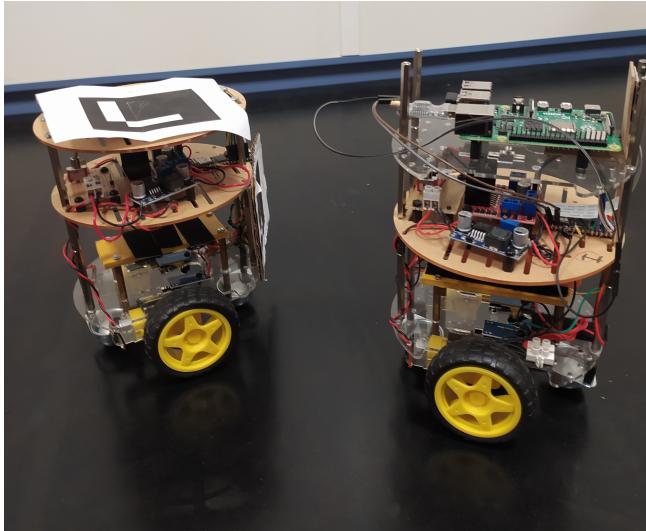
### 5.1. Robotarium-UCM: los robots

#### 5.1.1. Hardware

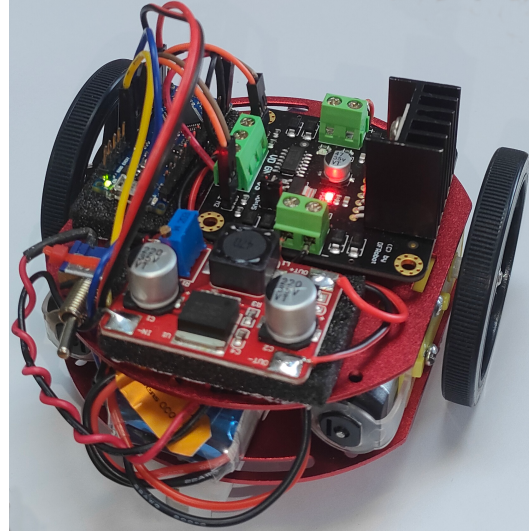
Los robots del Robotarium-UCM son holonómicos, con dos ruedas motrices y una de libre movimiento. Actualmente tenemos disponibles dos modelos de robots, según tamaño y materiales del chasis. El modelo inicial, tipo A, es más grande y está compuesto por un chasis de madera y metacrilato (Fig 1a). El otro modelo, tipo B, tiene un chasis metálico y es ligeramente más pequeño (Figura 1b). Los tamaños de ambos modelos se detallan en la Tabla 1.

Cuadro 1: Tamaño de los dos modelos de robot

Modelo	Diámetro (cm)	Altura (cm)
<b>Robot A</b>	15	20
<b>Robot B</b>	10	10



(a) Robot (tipo A).



(b) Robot (tipo B).

Figura 1: Robots del Robotarium-UCM.

Los robots tienen dos motores DC controlados por Modulación de Ancho de Pulso (*Pulse-Width Modulation*, PWM) como únicos actuadores, como es habitual en robots educativos. Por el momento, los únicos sensores incorporados a los robots son los codificadores (*encoders*) unidos al eje de cada rueda, que permiten estimar la velocidad de cada rueda con una resolución de 20 pulsos por vuelta.

El control de los motores, mediante el cálculo de la señal apropiada de PWM, y la lectura de los *encoders* se realiza desde la placa de control *Arduino Nano 33 IoT*. En los modelos más potentes, este Arduino comunica vía puerto serie con una Raspberry Pi incorporada al robot. En ese caso, la Raspberry Pi es la encargada de comunicarse con el servidor del Robotarium-UCM para transmitir y recibir datos y órdenes. Cada *agente* robot tiene asignado un identificador único, un código ArUCO [4], (Fig. 1a) que permite al servidor identificar y localizar al agente, utilizando una cámara cenital. El coste total de las partes ensambladas (chasis, placas de circuito, motores, *encoders* y baterías) se encuentra en torno a 150 € para ambos modelos.

## 5.2. Software

El software que da soporte al funcionamiento del Robotarium-UCM se compone de tres pilares: *Firmware*, *Agent* y *Hub*. La división conceptual en estos tres componentes resulta conveniente y mantenerlos desacoplados facilita tanto en el diseño como en el posterior mantenimiento. No obstante, el despliegue de cada subsistema en el hardware puede realizarse de diversas maneras, y en última instancia dependerá de la configuración elegida para la implementación de cada robot. Por ejemplo, para robots basados únicamente en Arduino Nano podría tener sentido ejecutar en el

mismo dispositivo físico el firmware del robot y el agente, mientras que para aquellos dotados de Raspberry Pi puede ser más conveniente desplegar el agente sobre la Raspberry.

**Firmware.** El firmware es el código de más bajo nivel, que en nuestra configuración se ejecuta habitualmente en Arduino Nano, y es responsable de controlar las funciones básicas del robot. Se le proporcionan una serie de parámetros necesarios para configurar el robot (identificador, IP, controlador, etc...). Debe realizar al menos las siguientes tareas:

- Lectura y comunicación de las medidas de los sensores.
- Control de los actuadores (p. ej. cálculo de los valores PWM).
- Comunicación con el agente para realizar la transmisión de datos y órdenes.

**Agent.** El *agente* es responsable de controlar las comunicaciones entre el *concentrador* y el robot. Por un lado, el agente recibe los mensajes que se originan en el firmware y los reenvía hacia el concentrador. Estos mensajes habitualmente contienen información sobre el estado del robot, como puede ser la velocidad de las ruedas medida por los *encoders*. Por otro lado, toma los mensajes provenientes del *Hub* y los hace llegar al *firmware*. Estos mensajes pueden haber sido generados en el propio Hub, otros agentes, o bien entidades externas que han hecho uso de la API del Hub. Normalmente contendrán órdenes dirigidas al agente o información que este pueda aprovechar de alguna manera, por ejemplo su propia posición enviada por el sistema de posicionamiento.

La motivación de separar el firmware del agente es que las comunicaciones podrían no ser gestionadas desde el sistema de control empotrado ((p. ej. Arduino Nano) sino desde un computador de abordo tal como Raspberry PI o NVIDIA Jetson Nano. De esta manera, se abre la posibilidad no solo de usar diferentes arquitecturas, sino de otras capacidades que pueden ser interesantes como puede ser la reprogramación remota (*Over The Air*, OTA) del controlador (Arduino Nano).

**Hub.** El Hub es una aplicación que (1) realiza un seguimiento de los diferentes agentes presentes en el Robotarium-UCM, (2) recopila datos de los agentes y los retransmite, y (3) proporciona una API para que los clientes externos puedan interactuar con el Robotarium-UCM. Puede verse como un *middleware* que cohesiona los agentes (robots, cámaras y otros elementos del sistema) y proporciona interoperabilidad con aplicaciones y elementos externos. Un ejemplo de estos es VRISA-ReNoLabs, el software que proporciona una plataforma para experimentación remota y que existe como un proyecto independiente.

El Hub mantiene una lista de agentes que se conectan dinámicamente y proporciona ciertas capacidades básicas de autodescubrimiento de dispositivos y configuración automática de las comunicaciones. Cada nuevo agente debe solicitar registro en el sistema, y de esta manera el Hub puede establecer las comunicaciones y acciones necesarias para integrar al agente en la red de comunicación del Robotarium-UCM, de manera que este sea capaz de enviar a, y recibir información de, los restantes agentes.

Además, la existencia del Hub es conveniente ya que proporciona flexibilidad al permitir definir topologías virtuales de comunicaciones, un registro de datos y eventos centralizados y otras funciones añadidas.

Este software se encuentra en desarrollo activo, y es de código abierto y distribución gratuito. Está disponible para descargar en el repositorio <https://github.com/UCM-237/robotarium-hub>.

### 5.3. Robotarium-UCM como Laboratorio Remoto

Una contribución importante del Robotarium-UCM es que ofrece a los estudiantes la posibilidad de llevar a cabo experimentos con los robots móviles de forma remota, con el único requisito de

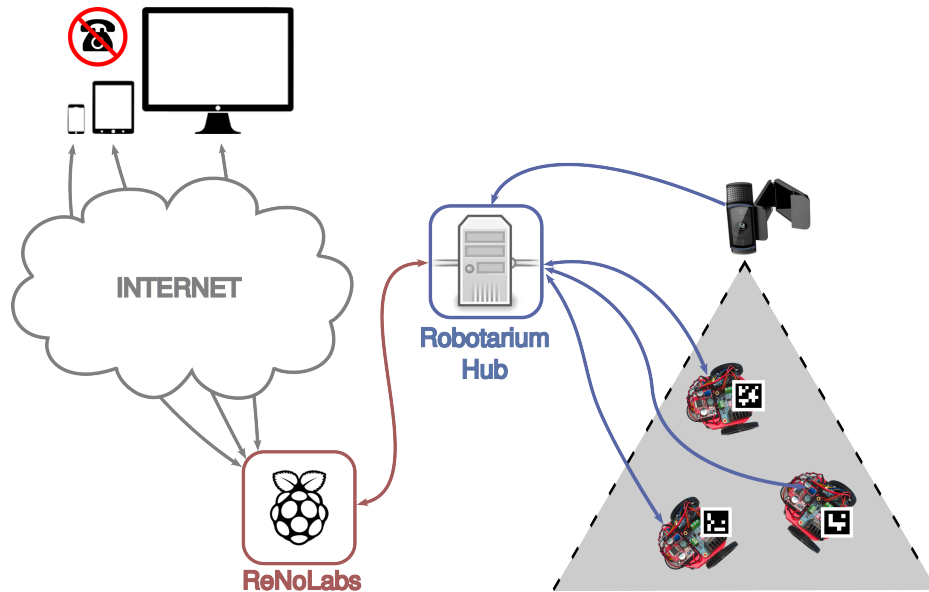


Figura 2: Una visión global de la arquitectura del Robotarium-UCM.

disponer de una conexión estable a Internet y cualquier dispositivo personal capaz de ejecutar un navegador web moderno.

La figura 2 representa la arquitectura software/hardware que habilita el acceso remoto al laboratorio y soporta la experimentación con los robots móviles y cualquier dispositivo que este disponible en el Robotarium-UCM. El funcionamiento se basa en *ReNoLabs* ([1]), un *Remote Laboratory Framework* (RLF). Como se discute y analiza en el trabajo citado, la creación y puesta en producción de un laboratorio remoto habitualmente no es un problema trivial, ya que involucra muchos aspectos técnicos que deben ser resueltos. El RLF simplifica y racionaliza el proceso.

Centrándonos en nuestro caso de uso, la experimentación con los robots móviles requiere una plataforma software que debe ser capaz de proporcionar diferentes servicios, entre los que se encuentran:

- Autenticación y Autorización.
- Planificación del acceso a los recursos del laboratorio.
- Interfaz de Usuario Web (WebUI).
  - Monitorizar el estado del laboratorio.
  - Edición de código y programación de los robots.

Algunos de estos servicios son ya proporcionados por ReNoLabs, como la autenticación y autorización, alojamiento web, y un entorno de desarrollo que permite construir y desplegar los laboratorios remotos.

ReNoLabs proporciona una interfaz web (WebUI) a la que los estudiantes pueden acceder con sus credenciales de usuario. Una característica importante del sistema es que, gracias a esta interfaz web, puede accederse al sistema desde prácticamente cualquier dispositivo, siempre que este pueda ejecutar un navegador web, por ejemplo cualquier dispositivo personal como un teléfono móvil, tableta u ordenador portátil. Dentro de la aplicación web, los alumnos visualizan el Robotarium-UCM a través de las cámaras situadas en el laboratorio (que poseen una doble función, ya que como se menciono antes permiten localizar a los robots mediante el uso de marcadores ArUCO).

Los estudiantes pueden seleccionar para el control uno de tres niveles diferentes de granularidad, en función del tipo de actividad que se desee llevar a cabo.

**Control incorporado** El robot se controla mediante el código incorporado, con algoritmos pre-implimentados que aseguran un comportamiento robusto y seguro del robot.

**Control asistido** El estudiante proporciona órdenes de alto nivel al robot, pero éste aún es responsable de garantizar las funciones básicas tales como evitación de obstáculos u operación dentro de los límites de la plataforma.

**Control completo** Este es el modo de operación que ofrece más libertad, ya que las funciones esenciales pueden ser expuestas directamente a los estudiantes. No obstante, debería ser utilizado únicamente con alumnos avanzados y en un entorno controlado, para evitar causar daño al Robotarium-UCM.

## Referencias

- [1] Chacon, J., E. Besada, L. Garcia, J.A. Lopez-Orozco. Efficient deployment of remote laboratories with TwinCAT-PLCs and EjsS Plugins IFAC papersOnline, 55 (17). pp. 326-331. ISSN 2405-8963, 2022.
- [2] Chacón, J., and L. García. IX Seminario de Educación en Automática, Comité Español de Automática, CEA. Badajoz, España, 2023.
- [3] García, L., Chacón, J., Gutierrez-Fontán, A., and J. F. Jiménez. Collaborative construction of a multi-robot remote laboratory: Description and experience. Robotics in Education (RiE 2023), Cyprus, 2023.
- [4] Garrido-Jurado, S., R. Muñoz-Salinas, F.J. Madrid-Cuevas and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition 47(6), pp. 2280-2292, 2014.