

# Entorno de Contenedores de Sistemas Embebidos con Conexión a Dispositivos Externos

*Waldo Valiente, Esteban Carnuccio, Mariano Volker, Matías Adagio, Micaela Antelo*  
*Departamento de Ingeniería e Investigaciones Tecnológicas*  
*Universidad Nacional de La Matanza*  
*Dirección: Florencio Varela 1703 – CP 1754 –*  
*{wvaliente, ecarnuccio, mvolker, matadagio, mantelo}@unlam.edu.ar*

## RESUMEN

Los Sistemas Ciber Físicos toman relevancia, con el surgimiento de tecnologías populares como Internet de las cosas, Industria 4.0, Internet industrial, máquina a máquina, Internet de todo y la capa Fog. Su contexto general de trabajo se basa en las interacciones dadas entre las funciones de control y los mecanismos de comunicación entre los componentes que forman el sistema. Esta interacción afecta al ambiente externo, que luego el sistema se adapta a ese cambio que él mismo produce. En la actualidad hay diferentes iniciativas y proyectos educativos que buscan enseñar estas tecnologías. Uno de los métodos de enseñanza es a través de prácticas con el sistema físico. Esto tiene problemas inherentes a la compra de los componentes y/o por la inexperiencia en la utilización de su electrónica. Como alternativa se busca que el uso de emuladores de sistemas embebidos permita enfocar directamente sobre el aprendizaje. Así facilita la tarea de abordar diferentes escenarios sobre los mecanismos de comunicación utilizando los protocolos Bluetooth y WiFi. Para sostener la infraestructura se utilizarán contenedores Docker, que ayudan en la construcción, gestión y pruebas de complejas topologías de sistemas.

**Palabras clave:** Sistemas Embebidos, Emulación, Comunicación.

## CONTEXTO

Nuestra Línea de Investigación es parte del proyecto “*Entorno de Contenedores para sistemas embebidos con conexión a dispositivos externos*”, dependiente de la Unidad Académica del *Departamento de Ingeniería e Investigaciones Tecnológicas*, perteneciente al programa de Investigaciones CyTMA2 de la Universidad Nacional de La Matanza, el cual es formado por docentes e investigadores de la carrera de ingeniería en informática. Este proyecto es continuación de los trabajos que viene realizando el grupo de investigación en sistemas operativos y en el área de Internet de las cosas.

## 1. INTRODUCCIÓN

La noción de los Sistemas Ciber Físicos, denominación que proviene del inglés Cyber-Physical Systems (CPS). Surgió en el año 2006 por Helen Gill en su discurso en la Fundación Nacional de Ciencias de EE. UU. Aunque en [1] interpreta que esta definición posee mayor antigüedad. Ya que, establece que es un término que deriva de la definición de “cibernética”.

Cuyo origen proviene desde mediados del siglo XX, nacido en el estudio de sistemas complejos de control de retroalimentación. El contexto de trabajo de los CPS se centra en hallar las interacciones dadas entre las funciones de control y los mecanismos de comunicación entre los componentes del sistema. Esta interacción afecta al ambiente externo, luego el sistema se debe adaptar al cambio que él mismo produce. En la actualidad el término CPS toma nueva relevancia, con el surgimiento de tecnologías populares como Internet de las cosas (IoT), Industria 4.0, Internet industrial, máquina a máquina (M2M), Internet de todo (IT) y la capa Fog (una capa intermedia de procesamiento en la topología de computación en la nube). Estos reflejan una visión de una tecnología que conecta profundamente el mundo físico con la información necesaria para interactuar en el día a día. El término CPS es una herramienta fundamental y duradera, ya que permite unir a la ingeniería tradicional del mundo cibernético y físico con las personas sin importar la distancia.

En los últimos años creció de forma exponencial la tecnología de Internet de las Cosas. El término IoT toma relevancia cuando se superó la cantidad de dispositivos conectados a internet, que el número de personas que existían en el mundo en ese momento. Según las proyecciones recompiladas por [2], se estima que actualmente hay 75.000 millones de dispositivos conectados y en el año 2025 habrá aproximadamente 100 mil millones. Además, que el instituto McKinsey Global sugiere que el impacto financiero de IoT en la economía global puede ser de entre 3,9 a 11,1 billones de dólares para 2025. El potencial anticipado de IoT se puede realizar cuando las “cosas” de Internet comienzan a interactuar de forma automática e inteligente con otras “cosas” de Internet.

Algunas de esas “cosas” como las computadoras y los teléfonos inteligentes pueden conectarse directamente a Internet a través de Ethernet o protocolos de comunicaciones móviles estándar, mientras que otros dispositivos se conectarán a Internet a través de una compuerta de enlace o enrutador local. De hecho, la compuerta incluso podría ser un teléfono inteligente que se comunica con otros sensores y dispositivos en proximidad. Por lo que el SE es claramente una plataforma flexible, capaz de desempeñarse como “Cosa” y/o la compuerta de enlace, en la medida en que sea capaz de conectarse directamente a Internet a través de Ethernet o, para dispositivos portátiles y más pequeños, capaz de conectarse con la compuerta a través de Bluetooth o comunicación inalámbrica como WiFi [3].

Por otro lado, el objetivo primario de la virtualización es recrear un componente físico o hardware, mediante un programa (software), emulando así todas las funcionalidades de este. Se pueden virtualizar: dispositivos, redes, componentes hardware, sistemas operativos, servidores, entre otros. Las tecnologías de virtualización como QEMU permiten a los desarrolladores emular un dispositivo físico, así se puede comenzar el desarrollo antes de fabricar el hardware [4]. El emulador y visualizador de la máquina QEMU permite a los desarrolladores probar el dispositivo de forma segura, incluso permite corregir defectos que pueden bloquear todo el sistema. El desarrollo y depuración de programas o controladores dentro de un emulador, hace que sea una tarea similar al desarrollo de aplicaciones comunes. Ya que, en el peor de los casos, los errores pueden provocar que el emulador se bloquee, se corrige el error y se vuelve a iniciar la simulación, sin mayores contratiempos [5].

Hay varias iniciativas y proyectos educativos que buscan enseñar las tecnologías mencionadas, orientándolas a estudiantes universitarios y de pregrado. En [6] se recapitulan las diferentes formas de enseñanza de estos temas, diferencia a los enfoques principales para los tópicos de gestión de proyecto, el diseño del sistema o las técnicas de redes. Enfatiza que el uso de una plataforma flexible en donde se pueda ejecutar los Sistemas Embebidos (SE) con conectividad de red. La conexión es una característica vital, ya que los SE utilizan distintos esquemas que les permiten comunicarse con otros dispositivos externos. Para esto se utilizan mecanismos tales como WiFi, Bluetooth, Ethernet, entre otros. De esta manera pueden intercambiar datos con servidores u otros dispositivos. Para así formar una topología de computación en la nube [7] y [8]. Para gestar esta idea, los contenedores Docker son una de las herramientas que ayudan en la construcción, gestión y pruebas de complejas topologías de sistemas. Ellos brindan un ambiente aislado que permiten trabajar con paquetes de software sin utilizar virtualización del hardware. La principal diferencia del uso de contenedores con una máquina virtual. Es que está realiza una abstracción completa del dispositivo físico y necesita ejecutarse sobre un Sistema Operativo (SO) que es redundante. Mientras que los contenedores se ejecutan sobre el motor de contenedores, que a su vez ejecuta integrado al SO anfitrión, que lo emplea como base para poder funcionar y acceder a los recursos del equipo [9]. Gracias a sus virtudes tiene un impacto que va más allá de su tradicional origen, dedicado al desarrollo o pruebas de programas. Los contenedores de Docker se están extendiendo a ámbitos como la investigación científica o la comunidad docente [10] y [11].

## 2. LINEAS DE INVESTIGACIÓN Y DESARROLLO

La enseñanza de los sistemas embebidos tiene como dificultad que no existen herramientas de simulaciones con características completas de licenciamiento abierto. Esta dificultad surge porque las herramientas de acceso libre tienen limitados sus recursos. Las herramientas profesionales, se enfocan a un mercado donde prevalece el licenciamiento del tipo comercial. Sin embargo, ofrecen versiones destinadas a alumnos, pero muy limitadas. Estas limitaciones, tanto en las herramientas de acceso libre como las comerciales, entorpecen la tarea de la enseñanza plena. Por este motivo, en esta investigación, se plantea la construcción de un entorno de emulación, que permita la conexión del SE con sistemas externos. Formando un ecosistema integro mucho más amplio. Así se presentan pruebas sobre escenarios complejos de funcionamiento. De esta forma, se dispone de entornos listos para enseñar en forma completa sobre el amplio espectro del universo CPS. Esto se sustenta, en el hecho de que existen diferentes métodos de enseñanza. El más utilizado es el tradicional. El cuál consiste en realizar las prácticas con el sistema físico. Esto tiene problemas inherentes a la compra de los componentes y/o por la inexperiencia en la utilización de su electrónica. Como alternativa surge el uso de SE emulados. Esto permite enfocar las prácticas directamente sobre el aprendizaje, dejando de lado las problemáticas antes mencionadas. En este sentido existen diferentes simuladores, que presentan determinadas características y limitaciones, que dificultan el aprendizaje completo, ya que no están preparados para ello.

El simulador Thinkercad, es utilizado en la educación a distancia de sistemas embebidos

[12]. Esta es una plataforma web que permite simular el SE Arduino Uno. De tal manera que ofrece la posibilidad de conectarla con distintos sensores y actuadores virtuales [13]. Pero su falencia es que no presenta conectividad con el exterior. Ya que no permite emular un módulo de Bluetooth ni de WiFi. Por otro lado, se encuentra el simulador Proteus, que permite simular los componentes de Bluetooth y WiFi. Pero, no posee conexión con el exterior. Además, presenta una licencia del tipo comercial. Aunque brinda una versión para estudiantes, resulta bastante restringido su uso, por ejemplo, no permite guardar lo realizado, entre otras limitaciones [14]. Por otra parte, está la página Wokwi que permite simular algunas placas de desarrollo básicas [15]. Pero, la emulación de la conectividad del WiFi y Bluetooth no se encuentra implementada. Siguiendo la misma línea, existe un simulador específico para el SE Raspberry Pi, que ofrece Microsoft, desde su plataforma Azure [16]. Sin embargo, este proyecto se encuentra en una versión preliminar, ya que posee un único ejemplo con funcionalidades muy restringidas. Esto se debe a que no permite al usuario reemplazar sus componentes de hardware. Otro de los emuladores que existen es el Android Emulator, que está integrado en la herramienta Android Studio, que permite emular un dispositivo móvil incluyendo al SO Android [17]. La herramienta facilita las pruebas, dentro de un simulador, de la aplicación Android mientras se está desarrollando.

Por ende, se pretende desarrollar una herramienta de enseñanza y trabajo, que permita a la comunidad llevar a cabo sus proyectos de IoT en forma simulada. De forma tal, que permita construir e implementar sistemas simulados comunicándolos con dispositivos externos reales. Para ello se proyecta que se

podrá llevar a cabo, a través de algunas de las formas de comunicación antes mencionadas. Esto se realizará empleando emuladores, tales como Qemu, configurado dentro de contenedores. A fin de que pueda ser de rápido uso e instalación.

### **3. RESULTADOS OBTENIDOS/ESPERADOS**

Se espera mejorar el proceso de aprendizaje y de pruebas sobre los CPS. A través de un único entorno modular. Que este permitirá la integración de distintos dispositivos formando topologías complejas de múltiples dispositivos. Permitiendo así emular el comportamiento de la plataforma completa. Todo el proyecto se espera que sea disponible y documentado sobre repositorios públicos, bajo la metodología de acceso abierto, permitiendo el uso a toda la comunidad científica y de alumnos.

### **4. FORMACIÓN DE RECURSOS HUMANOS**

La presente línea de investigación, dentro del departamento de Ingeniería e Investigaciones Tecnológicas, forma parte del trabajo que uno de los investigadores se encuentra realizando para su maestría. Completan el grupo de investigación dos de docentes de categoría V y un ingeniero en formación de investigador.

### **5. BIBLIOGRAFÍA**

- [1] E. A. Lee y S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, Cambridge, Massachusetts, Estados Unidos: The MIT Press, 2017.
- [2] T. Al-Rousan, «The Future of the Internet of Things,» de *Journal of Computing*,

- Communications & Instrumentation Engg*, 2017.
- [3] R. Toulson y T. Wilmshurst, *Fast and Effective Embedded Systems Design: Applying the ARM Mbed*, 2 ed., Newton, MA: Newnes, 2012.
- [4] qemu, 09 2019. [En línea]. Available: <https://www.qemu.org/>.
- [5] A. Kotovsky, *How to Develop Embedded Software Using the QEMU Machine Emulator*, Wilmington, Estados Unidos: Apriorit Inc., 2019.
- [6] S. Martin, *Teaching and Learning Advances on Sensors for IoT*, Basel, Suiza: MDPI, 2021.
- [7] P. Waher, *Learning Internet of Things*, Packt Publishing, 2015.
- [8] Sachan, *Internet de las cosas (IoT) y sus aplicaciones*, 2020.
- [9] J. M. Ortega, *DOCKER. Seguridad y monitorización en contenedores e imágenes*, RCLibros, 2019.
- [10] C. Boettiger, «An introduction to Docker for reproducible research, with examples from the R environment,» *ACM SIGOPS Operating Systems Review, Special Issue on Repeatability and Sharing of Experimental Artifacts*, pp. 1-24, 2015.
- [11] D. N. Nüst, V. Sochat, B. Marwick, S. J. Eglén, T. Head, T. Hirst y B. Evans, «Ten Simple Rules for Writing Dockerfiles for Reproducible Data Science,» *PLOS Computational Biology*, pp. 1-24, 2020.
- [12] J. Lopez Cisneros y G. Torales, «Experiencia de transición de una clase presencial a virtual en tiempos de Covid-19: una visión desde el docente,» 2020.
- [13] Autodesk, «<https://www.tinkercad.com/>,» 2022. [En línea].
- [14] Labcenter, 2022. [En línea]. Available: <https://www.labcenter.com/downloads/>.
- [15] CodeMagic, 2019. [En línea]. Available: <https://wokwi.com/>.
- [16] Microsoft, 2022. [En línea]. Available: <https://docs.microsoft.com/es-es/azure/iot-hub/iot-hub-raspberry-pi-web-simulator-get-started>.
- [17] G. Developers, 2022. [En línea]. Available: <https://developer.android.com/studio/run/emulator?hl=es-419>.