Universidad Tecnológica Nacional
SEPTIEMBRE - DICIEMBRE 2026 / Año 24 - Nº 57

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

Un laboratorio flexible y configurable para la formación de técnicos e ingenieros en sistemas embebidos

A flexible and configurable laboratory for training technicians and engineers in embedded systems

Presentado: 08/04/2026

Aprobado: 03/09/2026

Publicado: 23/09/2026

Maximiliano E. Véliz
https://orcid.org/0000-0002-4874-0855
Laboratorio de Investigación Aplicada a la Producción y el Trabajo, Universidad Nacional de Hurlingham, Argentina.
Instituto de Industria, Universidad Nacional de General Sarmiento, Argentina.
mveliz@campus.ungs.edu.ar

Gustavo E. Real
https://orcid.org/0000-0003-1146-8074
Laboratorio de Investigación Aplicada a la Producción y el Trabajo, Universidad Nacional de Hurlingham, Argentina.
Instituto de Industria, Universidad Nacional de General Sarmiento, Argentina.
greal@campus.ungs.edu.ar

Gonzalo P. Ribera
https://orcid.org/0009-0003-7831-0041
Instituto de Industria, Universidad Nacional de General Sarmiento, Argentina.
gribera@campus.ungs.edu.ar

Resumen

En este artículo se presenta el diseño e implementación de una propuesta educativa, centrada en la fabricación de equipamiento didáctico para la formación por competencias en ingeniería en el ámbito de los sistemas embebidos. La propuesta integra el desarrollo de un sistema embebido personalizable con actividades prácticas directamente vinculadas a la integración del hardware con el diseño del firmware con el objetivo de explorar estrategias formativas que preparan a futuros técnicos e ingenieros para los desafíos de su ejercicio profesional. Este enfoque busca alinearse con las demandas de la transformación digital en la industria y la creciente integración de la inteligencia artificial en el ámbito industrial y académico.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

El sistema embebido propuesto combina un hardware de adquisición de datos y control, el cual integra una placa Grand Central M4 Express y una placa de medición de energía eléctrica con el chip de Atmel M90E36A. Este sistema, en su conjunto, funciona como un laboratorio experimental flexible que permite adaptar los bloques de hardware y el diseño de software y de firmware de acuerdo a la aplicación. Fue concebido para enseñar conceptos claves como la abstracción funcional y el modelado de objetos en ingeniería, utilizando CircuitPython como lenguaje interpretado.

En el artículo, se detallan las características del hardware, se presenta un firmware base y otro de aplicación, ambos orientados a prácticas docentes. El firmware base permite acceder a las funcionalidades esenciales del sistema, mientras que el firmware de aplicación facilita la creación de un intérprete en CircuitPython para un lenguaje estructurado de Controladores Lógicos Programables (del inglés, PLC). Para ello, se integra la plataforma de libre acceso OpenPLC con la placa de adquisición y control, aprovechando las capacidades del chip M90E36A.

Palabras clave: Didáctica, Sistema Embebido, CircuitPython.

Abstract

This article presents the design and implementation of an educational proposal focused on the fabrication of didactic equipment for competency-based engineering training in the field of embedded systems. The proposal integrates the development of a customizable embedded system with practical activities directly linked to hardware integration and firmware design. The aim is to explore training strategies that prepare future technicians and engineers for the challenges of their professional practice. This approach seeks to align with the demands of digital transformation in industry and the increasing integration of artificial intelligence in both industrial and academic settings.

The proposed embedded system combines data acquisition and control hardware, integrating a Grand Central M4 Express board and an electrical energy measurement board with the Atmel M90E36A chip. This assembly functions as a flexible experimental laboratory, allowing for the adaptation of hardware blocks and software and firmware design according to the application. It was conceived to teach key concepts such as functional abstraction and object modeling in engineering, using CircuitPython as the interpreted language.

This article details the hardware specifications and presents both a base firmware and an application firmware, both designed for educational purposes. The base firmware provides access to the system's essential functionalities, while the application firmware facilitates the creation of a CircuitPython interpreter for a structured language used in Programmable Logic Controllers (PLCs). This is achieved by integrating the open-source OpenPLC platform with the data acquisition and control board, leveraging the capabilities of the M90E36A chip.

Keywords: Didactics, Embedded System, CircuitPython

Introducción

Según Fisher et al. (2021), el uso de software de código abierto como CircuitPython junto a microcontroladores modernos simplifica enormemente el desarrollo de instrumentos científicos personalizados. Por su parte, la investigación de Denny et al. (2024) sobre la

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

integración de asistentes digitales en la enseñanza de la programación revela que, si bien los estudiantes valoran la inmediatez y disponibilidad constante de estas herramientas, existe una demanda crítica por entornos que no se limiten a proporcionar soluciones directas. Los resultados destacan la preferencia por asistentes que actúen como tutores, priorizando explicaciones conceptuales, retroalimentación sobre la lógica del código y sugerencias incrementales que fomenten el pensamiento crítico. El estudio subraya que un diseño efectivo de estos entornos educativos debe equilibrar el apoyo técnico con la preservación de la agencia del estudiante, evitando la dependencia excesiva y asegurando un aprendizaje profundo en lugar de una simple resolución de tareas.

Por otro lado, la investigación de Wang et al. (2025) propone una reestructuración profunda de la enseñanza de microcontroladores mediante un enfoque que desplaza el foco desde el aprendizaje tradicional basado en hardware hacia un modelo de cooperación tripartita entre la guía del docente, la práctica del estudiante y el empoderamiento mediante recursos tecnológicos avanzados.

En este marco, reflexionamos sobre cómo diseñar las nuevas propuestas de laboratorios didácticos que vayan más allá de lo que los estudiantes pueden resolver con herramientas de IA de libre acceso. Sostenemos que el desafío es crear propuestas experimentales que, sin rechazar el uso de la IA o la programación asistida, exijan a los estudiantes de tecnicaturas e ingenierías un esfuerzo intelectual mayor.

El laboratorio propuesto en este trabajo aborda una arista específica de los sistemas embebidos, en donde a través de un hardware prediseñado se introduce al alumno al diseño del firmware mediante el lenguaje interpretado (CitcuitPython), orientando el aprendizaje hacia temas relacionados con la automatización y el control industrial.

En todos los casos, la programación se realizó en GitHub con asistencia de la inteligencia artificial de Copilot (Girón Jiménez et al. (2024) para los aspectos de la sintaxis del lenguaje interpretado.

Metodología

En lo que respecta a la metodología de trabajo sobre algo tan amplio como el sistema que se propone, se establecen los pasos que se ejecutarán, ya que es poco realista pretender terminar un sistema que se plantea tenga un permanente crecimiento y reformulaciones de acuerdo a las aplicaciones con las que se trabaje.

Por lo tanto, en esta primera aproximación y refiriéndonos estrictamente al alcance del presente trabajo, nos enfocaremos en enumerar los tres hitos que consideramos necesarios para cumplir con el alcance propuesto.

1. Disponibilidad de un hardware que posea los elementos necesarios para poder trabajar en un laboratorio, incorporando además distintos tipos de protecciones, varios canales de comunicación, distintos tipos de entradas y salidas, cuyas especificaciones pueden verse en el link correspondiente, y que sirven para establecer una base importante que pueda proyectar este sistema a ambientes industriales.
2. Disponer de la primera versión de un firmware elemental de adquisición y almacenamiento de datos con un tipo de acceso estándar a través de comandos, permitiendo que se puedan aprovechar sus prestaciones, tanto desde una aplicación externa, como así también a través de una terminal de texto por comunicación serial.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

3. Disponer de la primera versión de un traductor y ejecutor de esquemas estándar obtenidos de un programa libre en Ladder (OpenPLC), para el funcionamiento del conjunto como un PLC. Este paso es muy importante, ya que el alumnado involucrado cursa carreras tecnológicas vinculadas con la automatización y el control industrial.

Establecer estos hitos de trabajo no es un tema menor, porque se crean las bases de los bloques constructivos del sistema, y que sirven para que los alumnos, no solo puedan entender las distintas partes constitutivas del mismo, sino también entender que con estos bloques pueden seguir construyendo soluciones más complejas. De esta forma se pretende que, al disponerse de distintos elementos constructivos de hardware y software, se facilite el abordaje y la incorporación de conocimientos acerca de estos sistemas, evitando el hecho de tener que comenzar todo desde cero. Una vez incorporados los conceptos necesarios de forma paulatina, ellos servirán de soporte para poder encarar otros desafíos con bases más sólidas.

De las competencias hacia la construcción de las consignas en base al laboratorio flexible

El Consejo de Decanos de Ingeniería (CONFEDI) ha establecido un Enfoque Basado en Competencias (EBC) para la formación de ingenieros, formalizado en su Libro Rojo (2018). Como señala Tobón (2013), desde la perspectiva del Pensamiento Complejo, la competencia se concibe como un proceso complejo donde el individuo moviliza e integra el saber ser (actitudes), el saber conocer (conceptos) y el saber hacer (procedimientos) para actuar de forma creativa y con autonomía intelectual ante problemas de la vida cotidiana.

A partir de esta perspectiva, específicamente para que se articulen e integren las actitudes, conceptos y los procedimientos, en las nuevas propuestas didácticas con integración de IA, proponemos las siguientes dimensiones para construir la propuesta técnica - pedagógica: hardware, firmware y aplicativo.

- **Dimensión de hardware**

El alcance de este aspecto en términos de competencia adquirida va más allá de los componentes individuales y su conectividad, sino entender profundamente que cualquier diseño que involucre trabajar bajo condiciones exigentes de operación, indefectiblemente debe ser protegido adecuadamente para no comprometer la integridad de la solución. Esto significa que la solución no se limita a unir placas de desarrollo de open hardware, sino a una solución completa e integrada, en donde cada señal que ingresa o sale del sistema debe ser conformada y limitada para ser procesada.

- **Dimensión del firmware**

La construcción del firmware consiste en transformar el código fuente escrito por un programador en un archivo binario que pueda ser entendido por un microcontrolador. Lo que se propone en este sentido es evitar en un principio el proceso de compilación y generación de un código ejecutable, por la utilización de un intérprete como es el CircuitPython. Esto no pretende afirmar que con la técnica tradicional (compilar-enlazar-grabar) sea imposible realizar una evaluación empírica. La distinción crucial radica en la inmediatez y el carácter directo del ciclo de retroalimentación. Al eliminar la necesidad de recompilar y cargar en la flash un nuevo binario completo, la generación de código se vuelve interactiva, instantánea y de menor dificultad.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

- **Dimensión del aplicativo**

El aporte en esta dimensión tiene dos aristas importantes. Por un lado, generar una aplicación en donde el código interpretado (CircuitPython) ya se encuentra completo y que ante distintos accesos y excitaciones realiza acciones de control. En esta línea, cuando el alumno interactúa con una aplicación donde el código ya está completo y responde a estímulos, desarrolla competencias de carácter analítico y operacional.

Por otro lado, la posibilidad de generar un código que, a partir de un archivo de texto, pueda transformar el mismo en una gestión operativa, lo que implica que el CircuitPython interpreta un código que a su vez interpreta sobre la marcha (del inglés, "on the fly") unas especificaciones en texto que terminan en una ejecución concreta en el PLC. Al operar en un entorno de abstracción multicapa, el estudiante se distancia de la manipulación directa de registros y del hardware específico (bajos niveles de abstracción procedimental). Según el marco de formación por competencias de Tobón (2013) y las adaptaciones de CONFEDI (2018), resolver un problema en una capa lógica superior obliga al alumno a movilizar saberes complejos de diseño arquitectónico, modelado conceptual y lógica formal (niveles cognitivos de Analizar, Evaluar y Crear). El estudiante ya no piensa en "cómo configurar un temporizador en un registro de 16 bits", sino en "cómo representar el comportamiento de un temporizador en un sistema abstracto que procesa texto.

En la dimensión del aplicativo el enfoque propuesto permite desdoblar las competencias del alumno. Por un lado, las competencias de carácter analítico y operacional (bajo Nivel) vinculadas a la lectura de señales, acondicionamiento, visualización de datos y diagnóstico de fallas del entorno físico. Por otro lado, las competencias de abstracción de alto nivel centradas en el diseño algorítmico puro, mediante la interpretación de las especificaciones lógicas (como los archivos .st del PLC), apoyado en el marco del Enfoque Basado en Competencias (EBC) del CONFEDI (2018) y los postulados de Tobón (2013).

Este enfoque ayuda al estudiante a interpretar si las señales provenientes del entorno real (tensiones, corrientes provenientes del chip Atmel M90E36A, entradas analógicas optoacopladas) guardan coherencia con las leyes físicas del problema y con el estado del hardware. De esta forma puede discernir, por ejemplo, si una anomalía en una medición se debe a un ruido electromagnético, a una saturación del sensor, a una falla de aislamiento o a un error en el algoritmo. Esta afirmación se alinea con la literatura de entornos formativos embebidos e integrados.

En el artículo como Stefanov y Terziyski (2024) y Wang et al. (2025) enfatizan la transición de laboratorios abstractos hacia sistemas embebidos completos orientados a la práctica empírica con variables de campo. Por otro lado, los resultados empíricos preliminares obtenidos se basan en observaciones de campo y análisis cualitativos de las prácticas realizadas por estudiantes de la cátedra Desarrollo Avanzado de Microcontroladores de la Universidad Nacional de General Sarmiento.

Diseño y fabricación del hardware

El sistema didáctico está integrado por una placa de Adquisición de Datos y Control (SADyC) que tiene la finalidad de proteger y expandir las características de la placa de desarrollo Grand Central M4 Express para su utilización en entornos industriales o en prácticas de laboratorio exigentes y, además, integra una placa de medición de corrientes y tensiones trifásicas que integra el chip M90E36A para las funciones de desarrollos de sistemas de medición de calidad de energía.

En el detalle del diseño, en el SADyC se han implementado múltiples etapas de protección

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

y robustez industrial dividida en subsistemas. En lo que respecta a la entrada de alimentación, se utilizan sendos diodos para una rectificación de onda completa, colocando un diodo supresor de tensión transitoria (TVS, por sus siglas en inglés) de tipo bidireccional diseñado para proteger componentes electrónicos sensibles contra picos de alta tensión y descargas electrostáticas. Además, se incluye un fusible polimérico reiniciable (PPTC, por sus siglas en inglés) que actúa como una protección contra sobre corrientes y sobre temperaturas y se utiliza un filtro pasivo con núcleo de ferrita para suprimir el ruido electromagnético de alta frecuencia.

En lo que respecta a las protecciones de entradas y salidas (I/O), para aislar el microcontrolador del entorno industrial, se añade en el diseño: aislamiento galvánico en entradas digitales a través de opto acoplador, de forma que no haya conexión eléctrica directa entre la señal externa y el pin de la Grand Central. Respecto de las salidas digitales están limitadas a 1A y las salidas relé soportan hasta 2A. Por su parte, en las salidas analógicas se colocaron fusibles poliméricos reiniciables para limitar corrientes peligrosas en circuitos de muy bajo consumo, ofreciendo protección automática contra cortocircuitos y sobre corrientes sin necesidad de reemplazo físico.

Finalmente, en lo que respecta a las comunicaciones, en la comunicación RS485 se utilizan diodos supresores de tensión transitoria y fusibles poliméricos reseteables cumplimentando funciones a las ya descritas. El Bus CAN incluye un protector específico para este bus que consiste en un arreglo de diodos dobles de protección contra descargas electrostáticas y transitorios de tensión, no interfiriendo la señal de comunicación en cuestión.

Finalmente, en lo que respecta a la medición de señales analógicas, se incorporó una referencia de tensión de precisión que está diseñada específicamente para este tipo de mediciones. Asimismo, en todas las etapas de potencia se colocan capacitores de filtrado para mantener el voltaje estable.

La premisa de diseño del sistema en su conjunto, teniendo en cuenta su aplicación en la construcción de nuevos enfoques didácticos, estuvo centrada en el diseño de sistemas embebidos con lenguaje interpretado, con lo cual fue necesaria la selección de una placa de desarrollo que contenga el intérprete. Se optó por la Adafruit Grand Central M4 Express dado que integra un núcleo ARM Cortex-M4F (con FPU- unidad de punto flotante) para tareas de procesamiento de datos, su velocidad de reloj es de 120 MHz, su memoria Flash de 1MB y su SRAM de 256 KB. La Grand Central M4 Express (GCM4), además, contiene un sistema de almacenamiento de archivos (CIRCUITPY) con acceso por USB como disco.

Por otro lado, la razón del diseño de un hardware que nuclea la GCM4 fue pensar en características de diseño que amplíen las propiedades de la placa. En este sentido, además de las entradas analógicas, entradas y salidas digitales, la solución provee de salidas analógicas que se utilizan para los lazos de control, ya que permite la modificación en forma continua, dentro del rango de las especificaciones, de dos actuadores que pueden intervenir en sendos lazos. También se diseñó con buses RS485, Ethernet y comunicación WiFi y para ampliar los alcances del uso del sistema, se incluyó la comunicación CAN posibilitando, de esta forma, la vinculación con controles en el sector automotriz.

A continuación, en la Tabla 1, se detallan los componentes utilizados en el hardware del SADyC y sus respectivas especificaciones técnicas:

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

Características	Descripción / Especificaciones
PCB (Placa de Circuito Impreso)	**Montaje superficial, Cuatro capas, Material: FR4**
Salidas	**• 4 Salidas a relé (24V @ 1A) • 4 Salidas Open Drain (24V @ 0.5A) • 4 Salidas PWM (0 a 3,3V @ 10mA) • 2 Salidas analógicas (DAC) (0 a 10V)**
Entradas	**• 8 Entradas digitales opto acopladas (0 a 24V) • 8 Entradas analógicas (0 a 10V) • 4 Entradas analógicas de instrumentación (0 a 3.3V @ 0.5mA, ganancia programable)**
Buses de Datos	**I2C, SPI, GPIO genérica (1-wire, interrupciones y triggers)**
Comunicación	**• RS485 / CAN / Ethernet • Serial USB (vía placa Adafruit) • Conector para WiFi (basado en ESP8266)**
Almacenamiento	**• Conector para memoria SD (vía placa Adafruit) • Memoria EEPROM (I2C)**
Reloj de Tiempo Real (RTC)	**Por I2C con respaldo de pila de litio**
Alimentación	**• Entrada AC: Transformador 9Vac + 9Vac (40VA) • Entrada DC: 15Vdc (40VA)**

Tabla 1: Componentes del sistema.

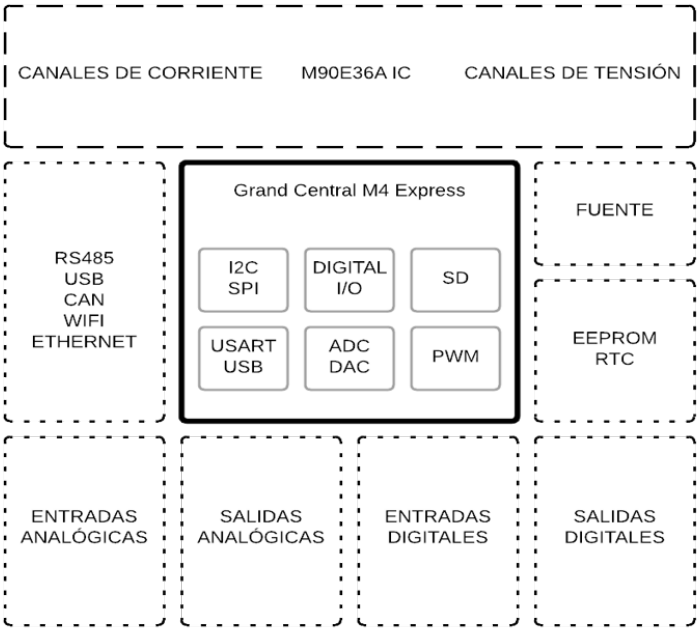

Figura 1: Esquema del diseño del laboratorio flexible. Se identifican los grupos de hardware con un tipo de borde diferente (Grand Central M4 Express - Adquisición de Datos y Control - Medición de la Energía Eléctrica + M90E36A). Fuente Elaboración propia.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

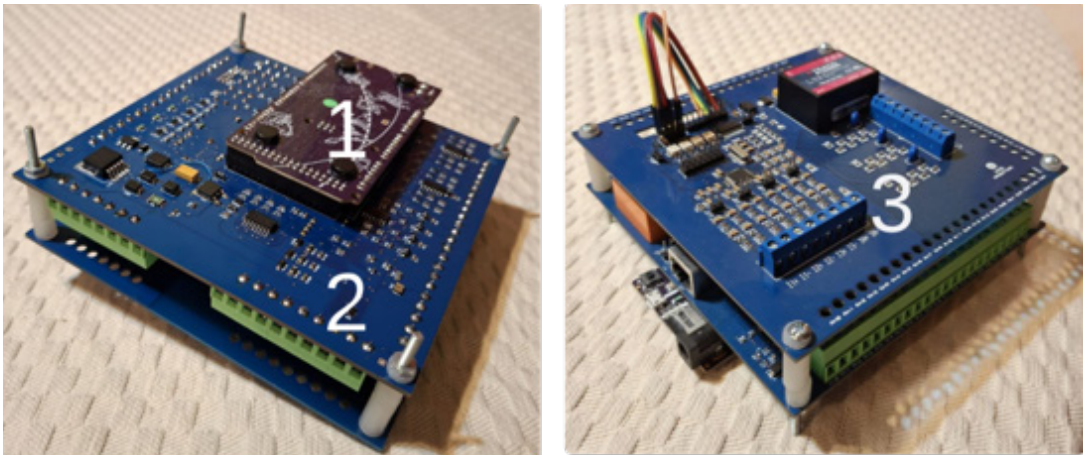

Figura 2: Fotos en reverso y anverso de la solución completa montada.
1.- Grand Central M4 Express; 2.- Placa de adquisición de datos y control;
3.- Placa de medición de energía eléctrica con el integrado M90E36A.

En https://github.com/gribera/acq-iii se pueden encontrar los planos, PCB y todos los componentes de firmware de acuerdo con la siguiente referencia de ítems:

- Documentación
  - Lista de materiales SADyC32_Adafruit_M4_GC_W5500 (ACQIII).
  - Lista de materiales ACQ-2_M90E36A3P4W.
  - SADyC esquemáticos y 3D.
  - M90E36A esquemático y 3D.
- Firmware SADyC.
- Firmware PLC.
- Firmware Test.

Diseño y desarrollo del firmware

1. Programa de Test

La puesta en marcha de cualquier tipo de hardware es relativamente compleja ya que hace al trabajo conjunto de una serie de componentes, como así también a un adecuado proceso de montaje y soldado de los mismos.

Probar por separado cada una de las funcionalidades de un sistema sin la ayuda de algún elemento que lo ordene y asegure su funcionalidad específica, no es un tema menor. Por lo tanto, lo que se propuso es disponer de un proceso específico que de acuerdo con una secuencia ordenada de pasos, nos permite probar una a una, y en su conjunto, todas las funcionalidades previstas para el sistema.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

Así, se diseñó y programó un firmware que solo sirve específicamente para que, tanto el proveedor del hardware como los alumnos e investigadores puedan corroborar que cada módulo del sistema completo funciona como estaba previsto por el diseño.

En este mismo sentido, este firmware otorga una ventaja adicional que va más allá de la prueba inicial de disponibilidad, que es mejorar el seguimiento de problemas. Cuando se está desarrollando un nuevo firmware y se producen fallas de funcionamiento, es muy deseable poder separar si las mismas son de software o de hardware; así, disponer de un firmware de prueba totalmente operacional y probado, ayuda a zanjar esa duda y focalizar más rápido la solución al problema. Ante una falla de un subsistema, se puede montar el programa de testing y verificar su funcionamiento; si funciona, el problema será del nuevo firmware, sino, será de hardware. Este concepto es aplicable a cada subsistema como así también a la operación del sistema en su conjunto.

Los módulos que integran el programa de test se muestran en la Figura 3, incluyen: pruebas analógicas, pruebas digitales, comunicaciones, RTC, EEPROM y la comunicación con la placa analizadora de energía.

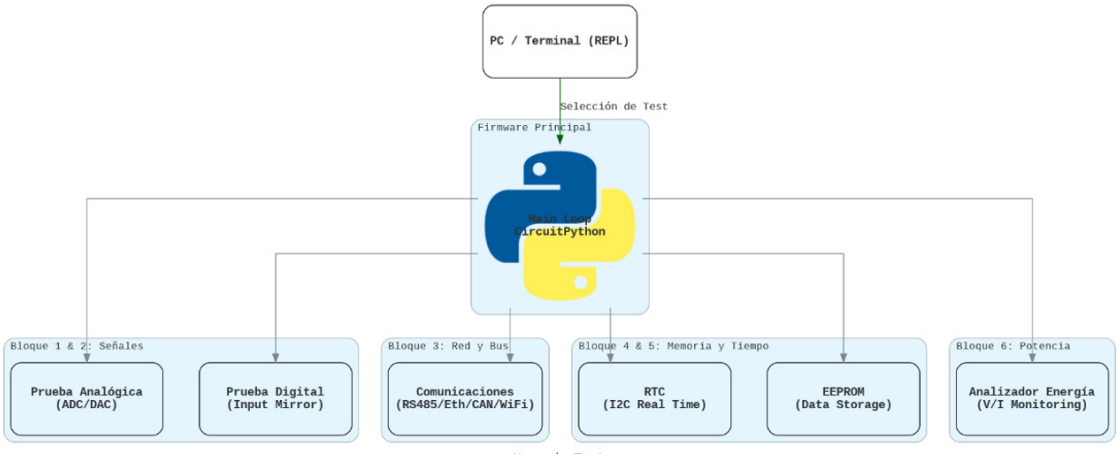

Figura 3: Módulos del programa de Test.
Fuente Elaboración propia.

De esta manera, conectando todos los elementos de hardware externos de acuerdo a lo indicado en el correspondiente procedimiento escrito, se corre el programa de testing en secuencia y el mismo informa al usuario a través de la terminal conectada al port monitor de la Grand Central M4 Express, los valores de las lecturas y se pueden visualizar las anomalías encontradas o bien un correcto funcionamiento de cada subsistema.

2. Firmware para adquisición de datos y control

El diseño del firmware para esta modalidad de trabajo del sistema se basa en una arquitectura tipo ejecución de comandos, los que se implementan formalmente a través de un protocolo de comunicación basado en paquetes con marcas de inicio y fin, a través

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

de los cuales se puede realizar todo tipo de acciones sobre el sistema, como configurar, consultar, leer entradas, cambiar estado de las salidas, etc.

Cada comando responde a una comunicación específica, sin embargo, todos ellos se estructuran de la misma forma, logrando un cierto grado de estandarización, facilitando el agregado o eliminación de estos.

El motor se implementa mediante una Máquina de Estados Finitos (FSM, del inglés) en CircuitPython. Esta FSM evalúa los bytes entrantes por los puertos de comunicación serie para identificar secuencialmente: 1) el byte de inicio, 2) el identificador del comando, 3) la información a transmitir, y 4) el byte de fin. Esta FSM, es no bloqueante y está estructurada usando la biblioteca Asyncio bajo el modelo de multitarea cooperativa en un solo hilo de ejecución, permitiendo en todo momento que los procesos cedan el control y no queden tomados los recursos del microcontrolador, sobre todo cuando procesos lentos conviven con otros más veloces.

De acuerdo con estas especificaciones, se definió como formato genérico de los comandos lo que se detalla a continuación en la Tabla 2.

Inicio de comando	Código	Subcomando	Datos	Fin de comando
ESC (1Bh)	**Tipo de comando**	**Variante del comando (opcional)**	**Parámetros (opcionales)**	**CR (0Dh)**

Tabla 2: Comandos genéricos

El campo subcomando está presente únicamente en aquellos comandos que agrupan más de una operación bajo el mismo código principal (por ejemplo, W para WiFi, L para transmisión, R para registro). En estos casos, el subcomando es obligatorio e indica la variante específica a ejecutar; en el caso de que los parámetros sean más de uno, se enviarán separados por coma. El campo Datos aparece cuando el comando requiere parámetros; si son varios, se separan por comas. Cuando ninguno de los dos campos opcionales está presente, el comando se resuelve directamente al recibir el CR.

De esta forma, no solo se logra disponer de un método standard para establecer una comunicación con otros dispositivos, sino que cualquier servicio adicional que se quiera incorporar a la estructura del sistema, y que implique una comunicación del tipo descripto, se podrá realizar bajo este mismo método agregando un comando más a los ya existentes.

En la Tabla 3 se especifican alguno de los comandos existentes al día de hoy, en donde se puede observar claramente el formato explicado anteriormente.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

Descripción	Comando	Subcomando	Parámetros	Ámbito
Muestra la ayuda	?	-	-	General
Datos y versión del equipo	e	-	-	
Consulta servicios de arranque	s	-	-	Startup
Habilita/deshabilita servicio de arranque	S	-	[w]	
Lee fecha y hora del RTC	h	-	6 valores numéricos: yyyy,mm,dd,hh,mm,ss	RTC
Configura fecha y hora del RTC	H	-	-	
Información sobre conexión WiFi	w	-	-	WiFi
Configura SSID	W	s	cadena de texto	
Configura contraseña de Wifi	W	p	cadena de texto	
Conecta a red WiFi	W	c	-	
Desconecta de red WiFi	W	d	-	
Configura el modo de trabajo	E	-	modo: 1 a 4	ACQ
Configura la cantidad de canales analógicos (automatización)	A	-	n: 1 a 8	
Configura la cantidad de canales analógicos (instrumentación)	B	-	n: 1 a 4	
Inicia transmisión temporizada de canales analógicos	L	s	tiempo: 0 (única) o tmin a 10000000 µs	
Detiene transmisión temporizada	L	p	-	
Transmite estado de canales digitales	L	u	-	
Inicia registro periódico en EEPROM	R	s	tiempo: 1 a 3600 seg.	
Detiene registro en EEPROM	R	p	-	
Descarga datos de EEPROM	R	d	-	

Tabla 3: Comandos específicos.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

En cuanto al funcionamiento, el sistema trabaja por modos. Esto está directamente asociado con cada una de las prestaciones que dispone, de acuerdo con la medición y/o control que se quiera realizar. Dichos modos son mutuamente excluyentes.

En este sentido, se definió que el sistema puede trabajar en los siguientes cuatro modos, lo que no es una limitante, ya que en el futuro se podría incorporar más modos, siempre respetando la exclusividad de los mismos y la comunicación por comandos:

Adquisición de datos “on line” (ADQ)

Adquisición y registro de datos (ARQ)

A continuación, en la Tabla 4, se especifica el detalle de los modos de trabajo.

Modo	Nombre	Canales Analógicos	Canales digitales	Descripción
1	ACQ1 “On Line”	Hasta 8 canales 12 bits de resolución Automatización	Hasta 8	Consulta en tiempo real vía canal de comunicación. Datos: 2 bytes por canal (byte alto primero). Intervalo mínimo: t_min = 20000000 * N / 115200 µs. Tiempo = 0 -> Transmisión única
2	ACQ1 “Registro”	Hasta 8 canales 12 bits de resolución Automatización	Hasta 8	Registro periódico en EEPROM. Encabezado: bytes por sesión (time stamp BCD + config). Muestra: 1 byte digital + N*2 bytes analógicos (LSB primero).
3	ACQ2 “On Line”	Hasta 4 canales 12 bits de resolución instrumentación (InAmp)	Hasta 8	Igual que Modo 1 pero usando los canales InAmp
4	ACQ2 “Registro”	Hasta 4 canales 12 bits de resolución instrumentación (InAmp)	Hasta 8	Igual que Modo 2 pero los datos analógicos corresponden a los canales InAmp

Tabla 4: Síntesis de Resultados y Validación del Sistema.

3. Firmware de aplicación didáctica PLC

Al igual que el entorno interpretado de CircuitPython facilita la ejecución directa de código fuente sin compilación previa, se desarrolló un motor de firmware capaz de interpretar un archivo de Texto Estructurado (.st) basado en el estándar IEC 61131-3. Este motor procesa el archivo lógicamente bajo un ciclo de scan determinístico homólogo al de un PLC industrial, lo que permite actualizar la lógica de control del sistema simplemente modificando el archivo de texto almacenado en la memoria Flash (actuando como unidad de almacenamiento masivo USB).

Para lograrlo, se implementó una capa de intérprete basada en CircuitPython, diseñada específicamente para traducir el lenguaje de texto estructurado utilizado en PLC. Como punto de partida, se empleó OpenPLC, el primer software de código abierto para la programación de PLC, según Rodrigues Alves, (2014). A partir del manual de usuario de OpenPLC y de los fundamentos de programación de CircuitPython, se construyó un firmware que actúa como intérprete que analiza la lógica de texto estructurado generada al compilar un proyecto en OpenPLC. El firmware de aplicación didáctica, alojado en la memoria Flash de la placa Adafruit Grand Central M4 Express, opera como un motor de ejecución dinámico. Al igual que el entorno interpretado de CircuitPython facilita la ejecución directa de código fuente sin requerir una compilación previa,

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

este firmware actúa como una capa de abstracción intermedia capaz de procesar un archivo de texto plano con extensión .st basado en el estándar normativo internacional IEC 61131-3 (Texto Estructurado). Cada vez que el estudiante guarda o modifica este archivo a través de la interfaz de almacenamiento masivo USB (unidad virtual CIRCUITPY), el motor detecta el cambio, y ejecuta las instrucciones en memoria RAM de forma cíclica y determinística. Esta rutina emula el ciclo de scan tradicional de un PLC industrial, permitiendo alterar por completo la estrategia de automatización en tiempo real sin someter al dispositivo al ciclo tradicional de compilación, enlazado y carga de binarios en la memoria flash, tal como se modela analíticamente en la Figura 4.

Este enfoque persigue dos objetivos principales: por un lado, al basar el firmware del PLC en CircuitPython, cada bloque de Ladder puede representarse como una función o clase en este lenguaje, lo que facilita la comprensión y el análisis del lenguaje industrial de automatización según el estándar IEC 61131. Por otro lado, se logra una mayor flexibilidad y transparencia en el desarrollo y depuración de aplicaciones de control.

La construcción del intérprete se originó en tres etapas: la primera consistió en realizar diversas simulaciones en OpenPLC para analizar el lenguaje de texto estructurado y así construir un parser comprensible en CircuitPython. Seguidamente se construyeron pseudo-códigos para representar los elementos del estándar IEC 61131 originando así las clases de CircuitPython para representar cada uno de los elementos. En todos los casos, la programación se realizó en GitHub con asistencia de Copilot (Girón Jiménez et al., 2024).

Las pruebas se realizaron con programas que contenían: contactos normalmente abiertos, contactos normalmente cerrados, bobinas, bobinas de set y reset, temporizadores, contadores y funciones de transferencia. En el repositorio GitHub se encuentra el firmware PLC y 8 (ocho) ejemplos de aplicación.

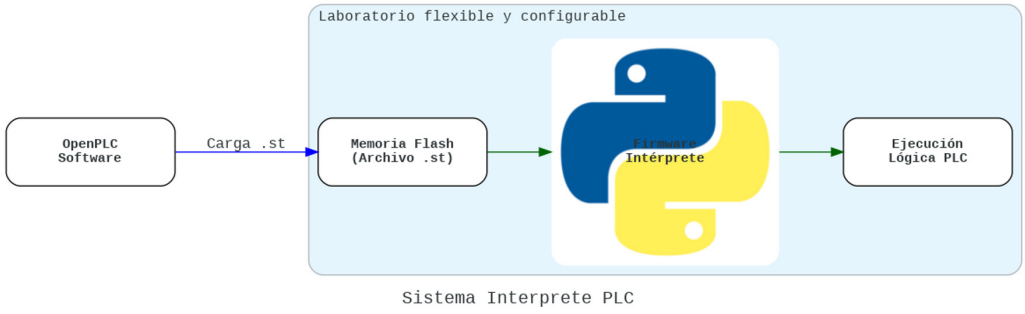

Figura 4: Intérprete PLC.
Fuente: elaboración propia.

4. Secuencia didáctica

La secuencia didáctica se estructura en dos etapas progresivas orientadas a la integración técnico-cognitiva del alumno. En la primera etapa, se establecen las bases mediante la presentación de los objetivos y el reconocimiento del hardware, profundizando en la modalidad de trabajo sobre el firmware base provisto. Los estudiantes manipulan este código preexistente modificando sus funciones de adquisición y control para comprender la arquitectura. En esta fase, se analizan los módulos de software en su rol de bloques funcionales lógicos de un PLC (bajo norma IEC 61131-3), desafiando a los estudiantes a diseñar e implementar lazos cerrados de control proporcional (P) utilizando variables físicas del entorno y ejecutando dichas lógicas de control mediante scripts interpretados de OpenPLC.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

La segunda etapa busca potenciar la autonomía intelectual y la creatividad técnica; en ella, los alumnos deben extender la gestión del hardware mediante la creación de nuevos comandos del SADyC y el desarrollo de módulos inéditos en OpenPLC. Esta etapa culmina con la interpretación "on-line" de dichas funciones, consolidando la capacidad del estudiante para intervenir sistemas complejos y adaptar la lógica de control en tiempo real sobre la plataforma embebida de forma recursiva.

Resultados

Los resultados se abordaron bajo la premisa metodológica de la recursividad, buscando determinar si la arquitectura desarrollada poseía la flexibilidad necesaria para integrarse como un elemento constitutivo en procesos de creación continua. La validación del impacto de la secuencia didáctica se realizó mediante un estudio de caso de enfoque mixto sobre la cohorte de la comisión de Desarrollo Avanzado de Microcontroladores de la UNGS. La "autonomía intelectual" se evidenció observando la capacidad de resolución de fallas sin intervención docente a través de una secuencia de actividades. Los datos consolidados de esta experiencia, registrados sistemáticamente por el equipo docente según los marcos de competencias de CONFEDI (2018), sirvieron de base para la construcción de los indicadores cualitativos mostrados en la Tabla 5.

Dimensión de Análisis	Hallazgos de Éxito (Lo construido)	Áreas de Mejora (El crecimiento)
Arquitectura de Firmware	**Se corroboró que la arquitectura propuesta trabajó de manera transparente, tanto como adquisidor de datos y control como en el modo PLC.**	**Necesidad de ampliar el conjunto de instrucciones compatibles para funciones de control más avanzadas tanto en el modo de adquisición y control como en el modo PLC.**
Modo SADyC	**Los estudiantes integraron con éxito sensores (ej: de corriente ACS712) y actuadores (ej: Servomotor SG90) mediante los comandos nativos de la capa de adquisición.**	**Optimizar la velocidad de respuesta (latencia) en el modo de interpretación "on-line" bajo cargas de trabajo pesadas, en cuanto al incremento de los puntos de monitoreo y el aumento de la frecuencia de muestreo.**
Interacción con OpenPLC	**La traducción de lógica industrial a hardware de bajo costo fue validada por estudiantes de diversas especialidades. Cabe destacar que el intérprete logró procesar archivos.st de forma transparente, permitiendo actualizaciones lógicas sin re-compilación.**	**Mejorar la retroalimentación de errores (debugging) en la consola REPL cuando el archivo.st presenta fallos de sintaxis. También se deberá incorporar nuevos elementos que forman parte del estándar utilizado en la herramienta Ladder, para lograr una mayor variedad en el control aplicado, como operaciones aritméticas y lógicas.**
Autonomía Intelectual	**Los alumnos propusieron ideas sobre nuevos módulos, demostrando que la herramienta es lo suficientemente amigable, permitiendo la creación recursiva.**	**Proveer plantillas de código más documentadas para reducir la curva de aprendizaje inicial en la extensión del firmware.**

Tabla 5: Síntesis de Resultados y Validación del Sistema

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

Conclusión

La experiencia directa con los estudiantes demostró que, al verse liberados de la complejidad sintáctica y de configuración de bajo nivel (compiladores y registros), alumnos que habitualmente quedaban rezagados por el problema de la operativa inicial, pudieron asimilar y aplicar conceptos avanzados de lógica industrial en la primera sesión. La democratización radica en que estudiantes con diversos niveles de programación lograron alcanzar los objetivos de diseño tecnológico en tiempos similares.

La implementación de un intérprete de lógica industrial sobre CircuitPython demuestra que la simplificación de los procesos tecnológicos no implica una reducción de su complejidad, sino una democratización del acceso al conocimiento técnico. Al mitigar la dificultad con la sintáctica inicial, este entorno formativo facilita que los estudiantes de ingenierías y tecnicaturas transiten de ser usuarios pasivos a desarrolladores activos de estrategias y lógicas de automatización. Esta transición se evidencia de forma concreta en las actividades de la primera etapa de la secuencia didáctica, donde los alumnos logran diseñar e implementar de forma autónoma lazos cerrados de control proporcional y lógicas de interbloqueo sobre variables físicas reales utilizando Texto Estructurado.

Esta experiencia muestra que el "saber hacer" no se limita a la abstracción del PLC; requiere que el alumno gestione restricciones propias de los sistemas embebidos, tales como el direccionamiento de pines físicos del microcontrolador, la interacción con la memoria Flash embebida y el diagnóstico a través del puerto serie (REPL). El valor de la propuesta radica justamente en que el estudiante utiliza la flexibilidad de un sistema embebido abierto para estudiar y experimentar cómo se comportaría en un entorno industrial (PLC).

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.57.1-16.2026 - ISSN 1666-6933

Referencias

Consejo Federal de Decanos de Ingeniería. (2018). Propuesta de estándares de segunda generación para la acreditación de carreras de ingeniería en la República Argentina: "Libro Rojo de CONFEDI" (R. Giordano Lerena & S. Cirimelo, eds.; 1ª ed.). Universidad FASTA Ediciones.

Denny, P., MacNeil, S., Savelka, J., Porter, L., & Luxton-Reilly, A. (2024). Desirable characteristics for AI teaching assistants in programming education. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (pp. 408-414). https://doi.org/10.1145/3649217.3653574

Fisher, D. K., Fletcher, R. S., & Anapalli, S. S. (2021). Python software integrates with microcontrollers and electronic hardware to ease development for open-source research and scientific applications. *Advances in Internet of Things, 11*(1), 42-58. https://doi.org/10.4236/ait.2021.111004

Girón Jiménez, A., Valero Redondo, M., Martín García, A., & Panizo Lledot, Á. (2024). El impacto de herramientas de programación inteligente en ingeniería: Evaluando el uso de Github Copilot. En *Revolucionando la docencia universitaria: Innovación educativa en la era de la IA y la Gamificación*. Madrid, Dykinson, 2024, pp. 201-218. http://hdl.handle.net/10396/32397

Rodrigues Alves, T., Buratto, M.G., de Souza, F.M., & Rodrigues, T.V. (2014). OpenPLC: An open source alternative to automation. *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, 585-589. https://doi.org/10.1109/GHTC.2014.6970342

Tobón, S. (2013). *Formación integral y competencias: Pensamiento complejo, currículo, didáctica y evaluación.* (4ta ed.). Bogotá: ECOE Ediciones.

Wang, D., Yuan, C., & Guo, R. (2025). Exploration of AI-Enhanced Teaching Models in Microcontroller Courses. *Journal of Education and Educational Research*, *14*(3), 39-42. https://doi.org/10.54097/336me943

Contribución de los Autores

	Colaboración Académica													
Nombres y Apellidos del autor	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Maximiliano Véliz	x	x	x	x	x	x	x	x		x	x	x	x	x
Gustavo Real	x	x		x	x	x	x	x	x	x	x	x	x	x
Gonzalo Ribera			x		x	x	x				x		x	x

1-Administración del proyecto, 2-Adquisición de fondos, 3-Análisis formal, 4-Conceptualización, 5-Curaduría de datos, 6-Escritura - revisión y edición, 7-Investigación, 8-Metodología, 9-Recursos, 10-Redacción - borrador original, 11-Software, 12-Supervisión, 13-Validación, 14-Visualización.