

# Tablero didáctico para prácticas académicas del protocolo CAN BUS

1<sup>st</sup> Daniel Márquez

Dpto. de Sistemas e Informática (DSI.) (FCEIA.)

Rosario, Argentina

dmarquez@fceia.unr.edu.ar

2<sup>nd</sup> Walter Wagner

Dpto. de Sistemas e Informática (DSI.) (FCEIA.)

Rosario, Argentina

wagner@fceia.unr.edu.ar

3<sup>rd</sup> Agustín Zuliani

(FCEIA.)

Rosario, Argentina

amzuliani02@gmail.com

**Abstract**—Esta publicación presenta el diseño e implementación de un tablero didáctico para simulación de un sistema en campo de comunicación CAN BUS para estudiantes de diferentes niveles; desde escuelas secundarias hasta Ingeniería Electrónica, pasando por cursos de capacitación. Esta herramienta permite realizar actividades prácticas específicas para la comprensión del protocolo y complementar los conocimientos incorporados desde los aspectos teóricos. También se pretenden mostrar las estrategias metodológicas en el proceso enseñanza/aprendizaje para lograr un aprendizaje significativo.

**Palabras clave**—CAN BUS, didáctica, protocolo de comunicación, sistemas embebidos.

## I. INTRODUCCIÓN

El protocolo CAN BUS (Controller Area Network) es un estándar de comunicación desarrollado para redes de control distribuidas. Originalmente creado por la empresa Bosch en 1983, el mismo surgió como respuesta a la creciente complejidad de los sistemas electrónicos automotrices, donde múltiples dispositivos y sensores necesitaban intercambiar información de manera rápida, fiable y robusta (ver [1] y [2]).

Desde su introducción en 1986, CAN BUS se ha convertido en un pilar fundamental para la comunicación en redes multiplexadas en vehículos, extendiéndose posteriormente a aplicaciones industriales, médicas y sistemas embebidos en general.

Una gran ventaja fue la de reemplazar los sistemas de cableado punto a punto, que aumentaban el peso, el costo y la complejidad de la comunicación dentro de los sistemas de transporte terrestre. La empresa Bosch propuso una arquitectura de red serie multiplexada que permitiera la comunicación entre unidades de control electrónicas (ECUs) utilizando un único par de cables (Bus de comunicación), reduciendo significativamente la cantidad de conexiones físicas.

Se emplea de manera predominante en el sector automotriz y de maquinaria agrícola, pero se ha comenzado a utilizar en aplicaciones para sistemas médicos y entornos de automatización industrial, consolidándose como una tecnología versátil y confiable para la comunicación entre dispositivos electrónicos.

Ha comenzado a reemplazar el uso de UART con el estándar RS-485 en los nuevos diseños y desarrollos de sistemas embebidos de aplicaciones generales, por sus grandes ventajas frente a este, y dado que no existen diferencias de costos en su uso. Un aspecto importante a tener en cuenta, es que la comunicación CAN BUS es orientada a mensajes con el paradigma productor/consumidor, a diferencia del típico esquema maestro/esclavo de las conexiones multipunto orientada a comunicación basada en direcciones de nodos, como la comunicación RS-485.

Con todo lo ya mencionado, CAN BUS es un protocolo que debe ser enseñado en los ambientes académicos, debido a su importancia en las diferentes industrias donde se encuentra presente. Por tal motivo es importante la elaboración de estrategias metodológicas durante el proceso de enseñanza para lograr un aprendizaje significativo. Dentro de estas estrategias metodológicas para abordar su estudio, es posible utilizar el aprendizaje basado en proyectos (ABP).

En el contexto académico, se debe trabajar en un correcto abordaje de la temática, donde el estudiante pueda incorporar conocimientos teóricos, conceptuales y prácticos sobre el protocolo. Principalmente generando un ambiente real de operación, en el cual en base a experimentaciones y observaciones, se pueda lograr el aprendizaje significativo.

El objetivo de la presente publicación es mostrar el recorrido realizado para el desarrollo de recursos didácticos para la enseñanza del protocolo CAN BUS. El enfoque es a través de una simulación en un ambiente real de operación y la elaboración de las estrategias metodológicas. De esta forma permite al estudiante, adquirir una base sólida sobre dicho protocolo, desde una mirada práctica y experimental.

## II. PROTOCOLO CAN BUS

### A. Introducción a CAN BUS

El protocolo de comunicación CAN BUS, por sus siglas en inglés Controller Area Network (CAN), es un protocolo de comunicación serie. Está orientado a aplicaciones en tiempo real y se caracteriza por su robustez, bajo costo y tolerancia

a fallas. Es ampliamente utilizado en sectores como el sector vehicular [3], aeronáutico, maquinaria agrícola, industria, medicina, entre otros.

Su diseño permite que distintos dispositivos conectados al bus (*Nodos*) se comuniquen entre sí mediante un medio compartido sin la necesidad de un dispositivo central (*Maestro*), logrando un sistema distribuido confiable y eficiente. Esto lo convierte en una solución ideal para entornos donde la comunicación debe mantenerse funcional incluso en presencia de errores o interferencias.

### B. Ventajas y similitudes de CAN respecto a RS-485

La comunicación CAN presenta varias ventajas frente a RS-485, entre las que se destacan:

- Arquitectura multi-maestro, que permite a cualquier nodo iniciar una transmisión.
- Mecanismos avanzados de detección y manejo de errores (CRC, monitoreo de bit, ACK, confinamiento de fallas).
- Escalabilidad: facilidad para añadir o retirar nodos sin reconfigurar el bus.
- Priorización de mensajes mediante identificadores, garantizando tiempos de respuesta predecibles.
- Mayores velocidades efectivas de comunicación, especialmente con CAN FD.

Como similitud relevante, ambos emplean transmisión diferencial, lo que incrementa la inmunidad frente a interferencias electromagnéticas (EMI). Esto resulta clave en entornos ruidosos como vehículos y plantas industriales, donde componentes como motores eléctricos, actuadores, interruptores o sistemas de audio generan ruido que puede acoplarse a las líneas de datos.

En la Figura 1, a continuación, se muestra un ejemplo de señal afectada por ruido eléctrico. Este tipo de perturbaciones puede introducir bits erróneos en la trama, provocando interpretaciones incorrectas del mensaje. Gracias a la transmisión diferencial y a los mecanismos de verificación de CAN, es posible detectar y gestionar estos errores para mantener la integridad de la comunicación.

### C. Fundamentos del protocolo CAN BUS

Recordando el modelo OSI (Open System Interconnection), el protocolo CAN se define en la *capa de enlace de datos* y parte de la *capa física* [4], como puede observarse en la Figura 2.

La lógica de control de enlaces (LLC) administra el control de sobrecarga y notificaciones, así como el filtrado de mensajes. El control de acceso al medio (MAC) se encarga de encapsular / desencapsular, del control y detección de errores, de los bits de stuff y la serialización y deserialización.

Respecto a la capa física, las especificaciones del protocolo CAN, solamente abarca la Señal Física (PS). El resto mostrados no se definen mediante CAN.

### D. Capa física

1) *Elementos físicos*: Un módulo CAN se compone de dos elementos básicos:

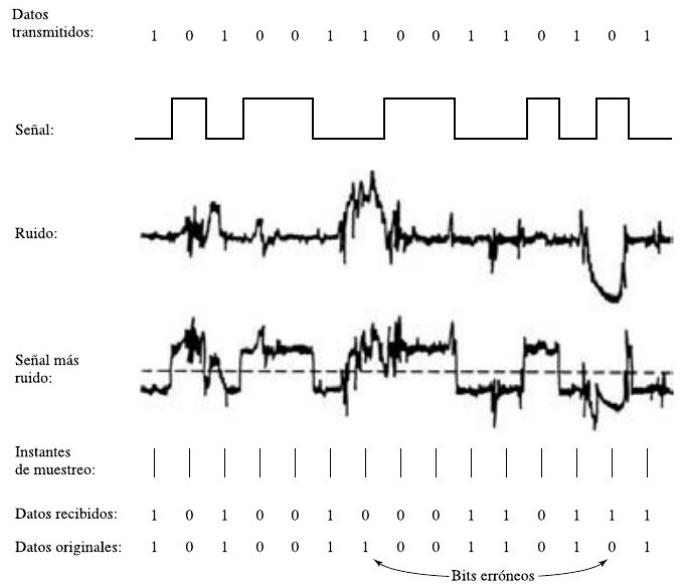


Figura 1. Representación de señal a la que se le adiciona ruido eléctrico. Observar los bits erróneos producidos.

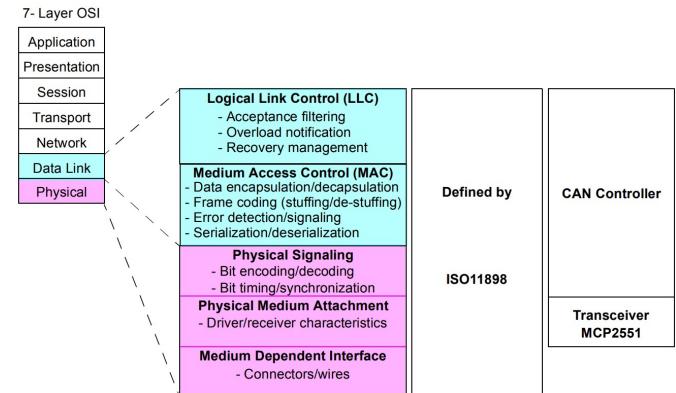


Figura 2. Capas del modelo OSI en las que trabaja CAN BUS.

- **Controlador**: Gestiona las tramas CAN, realiza la comprobación de errores en transmisión y detecta colisiones de mensajes. Un ejemplo es el MCP2515 [5], que se utilizará en las siguientes secciones para mostrar el tablero didáctico CAN.
- **Transmisor/Receptor**: También llamado transceptor. Se encarga de la codificación y decodificación de los mensajes en el bus. Un ejemplo de estos transceptores es el TJA1050 [6].

Tanto el controlador CAN como el transceptor son independientes del dispositivo conectado al bus (nodo), que como veremos más adelante, representa un microcontrolador específico. Esto tiene una ventaja significativa, ya que libera de carga adicional a la CPU del nodo conectado. Gracias a esto nos permite filtrar mensajes con identificadores específicos y evitar problemas de sincronización y colisión de mensajes.

Como mencionamos en la introducción no existe un maestro o esclavo, debido a esto cualquier nodo conectado al bus puede

enviar mensajes. Esta arquitectura responde a una filosofía de productor/consumidor. Donde el productor genera el mensaje CAN y los consumidores escuchan los mensajes transmitidos en el bus y en base a un identificador único (ID) del mensaje, deciden si deben procesarlo o descartarlo.

2) *Estándares fijados por la industria:* El protocolo CAN está diseñado bajo el estándar ISO 11898, el cual presenta diversas normas que abarcan distintos aspectos del funcionamiento. En particular, la ISO 11898-2 estandariza el protocolo CAN de alta velocidad, utilizado comúnmente en automóviles, permitiendo alcanzar velocidades de hasta 1 Mbps. Por otro lado, la ISO 11898-3 está diseñada para bajas velocidades y es tolerante a fallos, siendo adecuada para aplicaciones en maquinaria pesada o entornos con mucho ruido eléctrico.

A continuación, en la Tabla I: se presenta una comparación entre longitud de cable y velocidad de transmisión:

TABLA I  
RELACIÓN ENTRE LONGITUD DEL BUS Y TASA DE TRANSFERENCIA EN CAN

Longitud del bus (m)	Tasa de transferencia (kbit/s)
40	1000
100	500
200	250
500	100
1000	50

3) *Características eléctricas del bus:* La transmisión de señales en un bus CAN se lleva a cabo a través de dos cables trenzados (esto se realiza para disminuir el acoplamiento magnético entre ambos). Las señales se denominan CAN\_H (CAN high) y CAN\_L (CAN low), en los cuales en sus extremos se conectan una resistencia terminadora, normalmente de  $120\ \Omega$  (para evitar la reflexión de señales). El bus tiene dos estados definidos: estado dominante y estado recesivo. En la Figura 3 se puede ver una interpretación de esto. Observemos que cuando ambos cables del bus se encuentran al mismo nivel de tensión, se denomina estado recesivo y cuando existe una diferencia de tensión (de al menos 1.5 V), se denomina estado dominante.

#### E. Capa de enlace de datos

1) *Control de acceso al medio:* Una característica distintiva de CAN, es el denominado arbitraje, el cual se basa en el principio de CSMA/CA con AMP (Carrier Sense Multiple Access with Collision Avoidance and Arbitration on Message Priority), lo que significa que los nodos escuchan el estado del bus antes de transmitir y resuelven conflictos según la prioridad del mensaje. Este proceso es no destructivo, lo que garantiza que el mensaje de mayor prioridad se transmita sin interrupciones ni retrasos significativos. El arbitraje en CAN ocurre durante el campo de arbitraje de la trama de datos, que incluye el identificador del mensaje (ID). En CAN, la prioridad de un mensaje está determinada por su identificador:

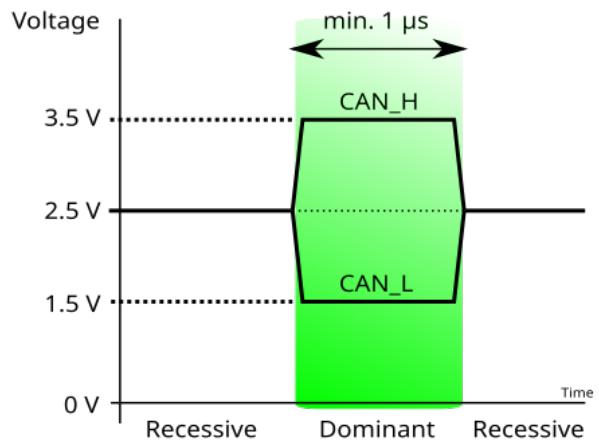


Figura 3. Estado del BUS CAN.

un identificador con un valor numérico más bajo tiene mayor prioridad. Ej: supongamos que tenemos dos mensajes uno con Id = 0x01 y otro con Id = 0x1A, cuando intentamos transmitir ambos mensajes, el primero en transmitirse será el de Id = 0x01.

#### F. Trama del protocolo CAN BUS

Una trama de datos en CAN se compone de los siguientes campos:

- **SOF (Start of Frame):** 1 bit dominante que marca el inicio.
- **Arbitration Field:**
  - Identificador (11 bits estándar o 29 bits extendido).
  - Bit RTR (Remote Transmission Request).
- **Control Field:**
  - IDE (indica estándar o extendido).
  - DLC (Data Length Code, 4 bits).
- **Data Field:** hasta 8 bytes (CAN clásico) o 64 bytes (CAN FD).
- **CRC Field:** 15 bits de verificación + delimitador.
- **ACK Field:** bit de confirmación por los receptores + delimitador.
- **EOF (End of Frame):** 7 bits recesivos que indican el fin.

#### G. Usos e Importancia en la Industria

El protocolo CAN BUS al ser tan confiable y robusto, se ha transformado en un valioso recurso para la Industria al momento de tener que establecer comunicaciones en sistemas críticos operando en entornos industriales exigentes. Entre los usos destacados podemos mencionar:

##### 1) Industria Automotriz:

- a) *Control de sistemas automotrices:* Comunicación entre unidades de control electrónico (ECUs) que gestionan el motor, la transmisión, los frenos, los airbags, el sistema de climatización y las luces, entre otros.

- b) *Sistemas avanzados de asistencia al conductor (ADAS)*: Coordina sensores de radar, cámaras y sistemas de frenado automático o control de crucero adaptativo.
- c) *Diagnósticos a bordo (OBD)*: Facilita el monitoreo de fallos y el diagnóstico de vehículos mediante la comunicación estandarizada entre componentes. Muy utilizado con scanners automotrices.

## 2) Automatización Industrial:

- a) *Control de maquinaria*: Permite la comunicación entre PLCs (Controladores Lógicos Programables), sensores y actuadores en líneas de producción automatizadas.
- b) *Robótica*: Coordina el movimiento de robots industriales, como en líneas de ensamblaje, mediante la transmisión en tiempo real de datos de sensores y comandos.
- c) *Sistemas de monitoreo*: Integra sensores de temperatura, presión y flujo en entornos industriales para monitoreo y control en tiempo real.

## 3) Industria Aeroespacial:

- a) *Sistemas de aviación*: CAN se usa en aeronaves para conectar sistemas de navegación, control de vuelo y monitoreo de motores, debido a su robustez y tolerancia a interferencias.
- b) *Satélites y vehículos espaciales*: Gestiona la comunicación entre subsistemas en entornos extremos, aprovechando su bajo consumo de energía y fiabilidad.

## 4) Equipos Médicos:

- a) *Dispositivos de monitoreo*: Conecta sensores en equipos como monitores de signos vitales, ventiladores mecánicos y máquinas de resonancia magnética.
- b) *Sistemas quirúrgicos robóticos*: Facilita la comunicación entre componentes para movimientos precisos y coordinados durante procedimientos quirúrgicos.

## 5) Transporte y Maquinaria Pesada:

- a) *Vehículos pesados*: En camiones, autobuses y maquinaria agrícola, CAN gestiona sistemas como el control de tracción, suspensión y monitoreo de combustible.
- b) *Ferrocarriles*: Coordina sistemas de control de trenes, como frenos, puertas y sistemas de señalización.

## 6) Energía y Redes Inteligentes:

- a) *Gestión de energía*: En sistemas de energía renovable, como turbinas eólicas o paneles solares, CAN conecta inversores, sensores y controladores.
- b) *Redes inteligentes*: Permite la comunicación en sistemas de distribución eléctrica para monitoreo y control distribuido.

## 7) Marina y Náutica:

- a) *Sistemas de navegación*: Integra sensores y sistemas de control en embarcaciones para monitorear motores, GPS y sistemas de propulsión.
- b) *Sistemas de monitoreo remoto*: Gestiona datos de sensores en barcos para garantizar la seguridad y eficiencia

operativa.

- 8) **Internet de las Cosas Industrial (IoT)**: CAN se utiliza en aplicaciones de IoT industrial para conectar dispositivos en redes de sensores y actuadores, especialmente en entornos donde se requiere robustez y baja latencia.

## III. ESTRATEGIAS METODOLÓGICAS DE ENSEÑANZA CAN BUS

La formación académica en la carrera Ingeniería Electrónica, el proceso enseñanza/aprendizaje involucra conceptos abstractos que desafían la vinculación de aspectos teóricos con los prácticos. Por tal motivo es necesario establecer estrategias que permitan al estudiante consolidar los conocimientos y apropiarse de la experiencia de campo que aplicará en un futuro en el ámbito profesional.

Es importante que el estudiantado alcance un aprendizaje significativo en cada temática abordada, el cual consiste de un proceso en el que se relaciona nueva información con conocimientos previos, construyendo activamente un entendimiento profundo y personal que facilita la retención y aplicación de los saberes incorporados. Por este motivo, y dado que el tema “Protocolo CAN BUS” curricularmente prosigue a otros protocolos y estándares de comunicación como UART, SPI, I2C, entre otros, es que se plantea un abordaje de enseñanza donde se relacionan conceptos anteriormente estudiados.

Las estrategias de enseñanza, se elaboran con los objetivos formativos del estudiante. Por tal motivo, planteamos: ¿Cuáles serían los objetivos formativos de un estudiante de Ingeniería Electrónica en cuanto al protocolo de comunicación CAN BUS?

- a) Vincular los conceptos teóricos, normas y estándares para la comprensión del funcionamiento de la comunicación estudiada.
- b) Comprender los ámbitos de utilización según las ventajas frente a otros protocolos.
- c) La correcta implementación de la comunicación en diferentes plataformas de sistemas embebidos.
- d) Incorporar el recurso de comunicaciones basadas en identificadores de mensajes y no en direcciones de nodos.
- e) Comprender el paradigma de comunicación productor/consumidor.
- f) Abordar prácticas para consolidar el funcionamiento del protocolo.
- g) Obtener experiencia de campo con una simulación de un ámbito real de operación.

Con los objetivos planteados, ahora nos posicionamos en las dificultades y desafíos para la confección de estrategias de enseñanza:

- a) **Abstracción conceptual**: CAN BUS opera a nivel lógico y de señalización, lo que dificulta su visualización y comprensión directa por parte de los estudiantes.
- b) **Distancia entre teoría y práctica**: Para los estudiantes puede ser un recorrido complejo de transitar para conectar los conceptos teóricos aprendidos con su implementación

y funcionamiento en sistemas reales, lo que puede ser un obstáculo para alcanzar un aprendizaje significativo.

- c) **Limitaciones de equipamiento:** Los laboratorios pueden no disponer de recursos para que todos los estudiantes puedan realizar las prácticas que los nutran de experiencia de campo para consolidar los conocimientos.

Teniendo en claro los objetivos formativos y teniendo en cuenta las dificultades y desafíos en el proceso de enseñanza aprendizaje, preguntamos ¿Qué recurso tecnológico basado en las TIC podría sernos de utilidad?

Poder contar con un recurso tecnológico de simulación de un ambiente real de operación y funcionamiento del protocolo CAN BUS permite trabajar en múltiples estrategias en el proceso enseñanza/aprendizaje. Cada estrategia dependerá de los alcances y objetivos de la unidad curricular donde se trata la temática.

Un tablero didáctico para la simulación de un ambiente real de operación del Protocolo CAN BUS es un recurso que puede estar conformado por múltiples plataformas de diferentes arquitecturas con distintos sensores y actuadores. Se pretende que todos los dispositivos de la red puedan interactuar entre sí de múltiples maneras en función de los diferentes firmwares utilizados.

Esto permite:

- a) Visualización de un sistema real operando para resolver requerimientos concretos.
- b) Comprender la interacción entre dispositivos de la red.
- c) Visualizar los mensajes de la implementación para una mejor comprensión del funcionamiento.
- d) Analizar la capa física observando con osciloscopio señales, niveles de tensión, transmisión de datos bit a bit, etc.
- e) Que los estudiantes interactúen con el tablero, realizando implementaciones de firmwares sobre dispositivos existentes o agregando otros nodos al bus CAN.
- f) Comprender que el protocolo es multiplataforma, permitiendo conectar dispositivos de diferentes arquitecturas, siempre y cuando se respete el estándar de comunicación CAN BUS.

Además de todo lo ya expuesto en esta sección, dentro de las estrategias en el proceso enseñanza/aprendizaje, la utilización del tablero permite abordar metodologías que forman al estudiante para situaciones que puede experimentar en su recorrido profesional:

- a) **Aprendizaje Basado en Proyectos (ABP):** Con esta metodología, se pretende que el estudiante incorpore los conocimientos del campo teórico y de experiencia práctica desarrollando un proyecto significativo. El tablero didáctico permite tratarlo como una simulación de un ambiente real de operación, y desarrollar dispositivos CAN que se anexen al mismo con requerimientos determinados. El proyecto puede llevarse adelante desde un estudiante o grupos de trabajo. Trabajando en grupos, se extiende la experiencia de los saberes propios de la

temática al desarrollo de habilidades en las relaciones interpersonales.

- b) **Aprendizaje Colaborativo:** Trabajando en grupos, en muchas oportunidades se da un aprendizaje entre pares. La utilización del tablero para diferentes consignas didácticas, puede ayudar a que los estudiantes puedan responder inquietudes de entre ellos, motivando las relaciones entre ellos y la obtención de respuestas específicas de la temática. En ambientes profesionales, el aprendizaje entre pares es algo muy habitual, por lo que este abordaje intenta entender esta dinámica de grupos.

- c) **Aprendizaje Invertido (aula invertida):** En esta estrategia, se pretende el cambio de roles en el aula, dando más espacio al trabajo activo por parte de los estudiantes. Promueve la autonomía y el aprendizaje activo. La utilización del tablero podría aplicarse para el estudio en detalle del protocolo CAN desde sus aspectos conceptuales y que los estudiantes ofrezcan una presentación con demostración práctica sobre el tablero para lograr dar una explicación del tema. Se pretende que el estudiante incorpore conocimientos de la temática mientras estudia y experimenta con el objetivo de poder explicarlo luego.

- d) **Aprendizaje Basado en Competencias:** Se centra en el desarrollo de habilidades y competencias específicas, evaluando el dominio de estas en lugar de conocimientos aislados. Se pretenden obtener resultados prácticos. Los estudiantes avanzan al demostrar competencia. El tablero didáctico puede utilizarse con una simulación concreta de un ambiente real de funcionamiento (es decir con todos sus dispositivos corriendo una aplicación específica) y el abordaje puede orientarse a la detección de fallas, mediciones estratégicas, análisis de mensajes y señales, diagnósticos de operación, mediciones de voltajes y desarrollo de protocolos para determinar estado de funcionamiento. Todo esto, con el objetivo de obtener competencias de campo en el protocolo CAN BUS.

- e) **Aprendizaje Experiencial o basado en experiencias:** El aprendizaje se basa en experiencias directas, como prácticas, simulaciones o actividades en contextos reales. La utilización del tablero didáctico es un recurso fundamental para el desarrollo de esta estrategia de enseñanza. Conecta la teoría con la práctica, promoviendo la reflexión sobre la experiencia viva. Se pueden plantear prácticas en la simulación de un ambiente real de operación con el tablero didáctico, y que dichas prácticas persigan como objetivo la vinculación de la comprensión del protocolo en sus aspectos teóricos y conceptuales.

- f) **Aprendizaje por Descubrimiento:** Los estudiantes exploran y descubren conocimientos por sí mismos, guiados por el profesor mediante actividades prácticas, en lugar de recibir información de forma directa como en una clase magistral. Estimula la curiosidad, la creatividad y el pensamiento crítico. Se puede utilizar el tablero didáctico en una aplicación específica para este abordaje en cada dispositivo, y que los estudiantes experimenten su funcionamiento, mediante observación y análisis. Luego, con

lo experimentado y con la comparación de conocimientos previos, como por ejemplo comunicación RS-485, pueden descubrir las características de la comunicación CAN y así mismo lograr un aprendizaje significativo.

#### IV. ESPECIFICACIONES TÉCNICAS DEL TABLERO

El tablero está construido por una base sólida de madera, que asegura su durabilidad y facilita la posibilidad de trasladarlo para el uso en distintos espacios educativos, y además, por un soporte que permite que el tablero quede a una inclinación acorde para permitir que los estudiantes estando sentados o parados puedan ver las explicaciones y luego trabajar en él; todo esto puede visualizarse a continuación, en la Figura 4. Además cuenta con un BUS CAN tendido y distintos nodos interconectados en él; en la Tabla II, a continuación, se resume el funcionamiento de cada nodo y luego se detalla cada uno de ellos:

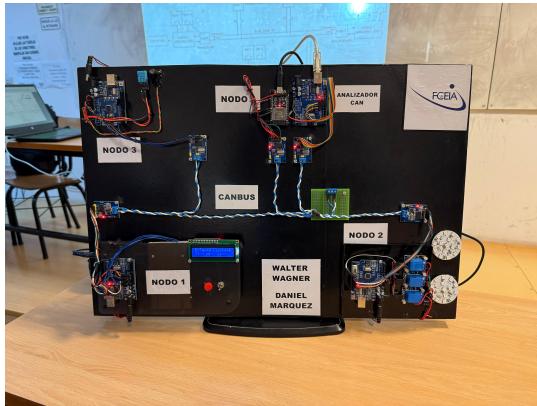


Figura 4. Tablero CAN en el cursado de Sistemas Digitales II año 2025.



Figura 5. Tablero CAN en curso de capacitación a la empresa Agronorte 2025.

TABLA II  
TIPOS DE NODOS EN LA RED CAN

Tipos de Nodos						
Nodo	ID Recibido	ID Enviado	Consumidor	Productor	Dispositivo	
1	0x120 0x130	0x110	X	X	Arduino Uno	
2	0x110 0x130	0x120	X	X	Arduino Uno	
3	-	0x130	-	X	Arduino Uno	
4	0x130	-	X	-	ESP32	
5	-	0x150	-	X	KL46Z	

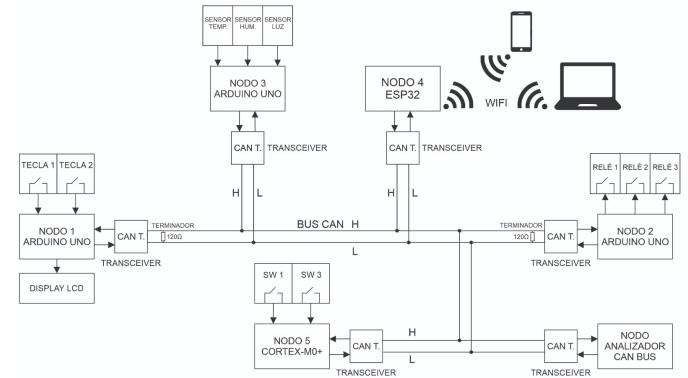


Figura 6. Diagrama de bloques del tablero.

#### A. Nodo 3 – Nodo sensor ambiental:

El Nodo 3 constituye el dispositivo de menor complejidad funcional dentro de la topología de red del tablero didáctico, lo cual lo convierte en un excelente punto de partida para la comprensión progresiva del sistema CAN implementado. Su función principal es actuar como nodo emisor periódico de datos ambientales, contribuyendo con información sensórica al tráfico general del bus.

Este nodo, que puede verse a continuación en la Figura 7, está basado en un microcontrolador **ATmega328P**, alojado en una placa **Arduino Uno**, que se comunica mediante el protocolo **SPI** con un controlador **MCP2515**, el cual, a su vez, se acopla a la red mediante un transceptor CAN. Este controlador actúa como interfaz entre el microcontrolador y el bus CAN físico, permitiendo la encapsulación y el despacho de tramas.

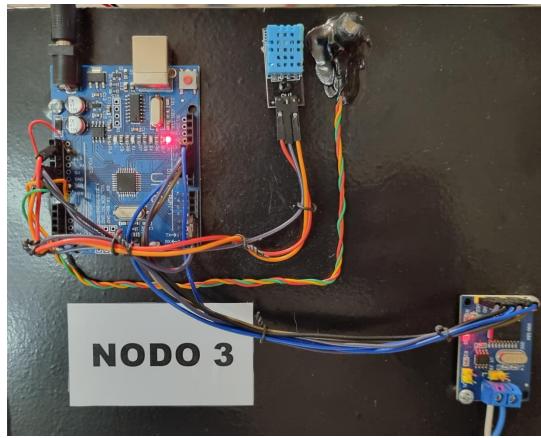


Figura 7. Representación física del Nodo 3.

La identificación del nodo en la red está dada por el identificador 0x130, el cual se utiliza para etiquetar todas las tramas originadas por este dispositivo. Cada mensaje transmitido posee una longitud fija de 8 bytes (campo DLC = 8), y es emitido con una frecuencia aproximada de 1 Hz, lo que garantiza una actualización periódica y estable de los valores sensados.

En cuanto a sensórica, el nodo está dotado de dos sensores:

- Un sensor DHT11, encargado de medir temperatura ( $^{\circ}\text{C}$ ) y humedad relativa (% HR) del ambiente. Este dispositivo entrega sus datos en formato digital a través de una única línea de datos, y sus lecturas son convertidas a valores enteros sin decimales para optimizar el empaquetado en la trama CAN.
- Una fotoresistencia LDR conectada a una entrada analógica del Arduino, que permite estimar el nivel de luminosidad ambiental. El valor leído se escala proporcionalmente al rango 0–100 % para representar el porcentaje relativo de luz, el cual se discretiza como un número entero sin signo de 8 bits.

El empaquetado de los datos se organiza de la siguiente forma en la trama CAN:

- **Byte 0:** Temperatura ambiente (entero con signo de 8 bits, en grados Celsius).
- **Byte 1:** Humedad relativa (entero con signo de 8 bits, en porcentaje).
- **Byte 2:** Nivel de luminosidad (entero sin signo de 8 bits, en porcentaje).
- **Bytes 3 a 7:** No utilizados, asignados a cero.

El nodo incluye lógica básica de verificación de lecturas inválidas (como NaN en el caso del sensor DHT11), y en tales casos, omite la transmisión para evitar introducir datos erróneos en el bus. Esta decisión de diseño enfatiza la robustez del sistema frente a fallas de lectura intermitentes, habituales en sensores de bajo costo.

Finalmente, la configuración del controlador MCP2515 incluye el establecimiento de una tasa de transferencia de **125 Kbps**.

## B. Nodo 2 – Actuadores digitales y control de la red

El Nodo 2, que puede verse en la Figura 8, representa una unidad funcional intermedia dentro del tablero didáctico basado en red CAN BUS, combinando capacidades de adquisición, procesamiento, actuación y retransmisión. Su diseño lo posiciona como un nodo mixto o productor-consumidor, ya que no solo publica datos en el bus, sino que además reacciona ante la información proveniente de otros nodos de la red.

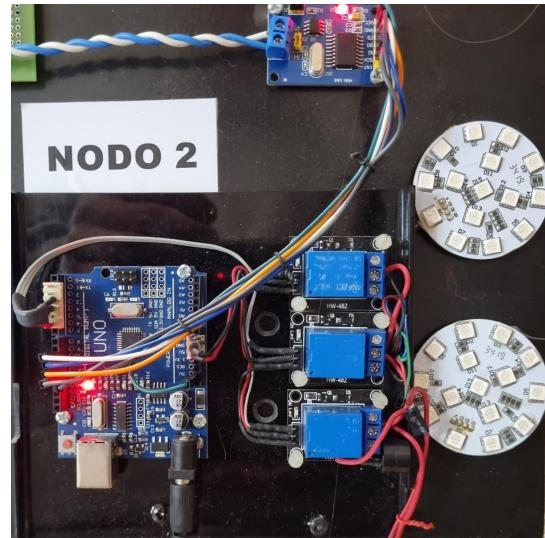


Figura 8. Representación física del Nodo 2.

El sistema se estructura en torno a una placa **Arduino Uno** con microcontrolador **ATmega328P**, equipada con un módulo controlador **MCP2515** más transceptor CAN, conectado al microcontrolador a través de una interfaz **SPI**. Al igual que el Nodo 3, el nodo opera a una velocidad de transmisión de **125 Kbps**, y se encuentra configurado en **modo normal**, habilitado tanto para emisión como recepción activa de mensajes.

Este nodo se encarga de recibir y procesar datos provenientes de los siguientes nodos:

- **Nodo 1:** Mediante el identificador **0x0110**, este nodo transmite el estado de dos pulsadores. El Nodo 2 interpreta el estado lógico y, según corresponda, acciona o desactiva dos **salidas digitales** conectadas a relés electromecánicos, que a su vez comandan alarmas lumínicas externas conectadas a 12V.
- **Nodo 3:** A través del identificador **0x0130**, el Nodo 3 envía datos ambientales. El Nodo 2 extrae de la trama los tres primeros bytes, correspondientes a temperatura, humedad y nivel de luz ambiental. Aunque recibe todos los datos, su lógica interna se concentra en la variable de luminosidad. Si el valor supera un umbral predeterminado (50 %), se activa un tercer relé asociado a una alarma sonora (zumbador). En caso contrario, esta salida se desactiva. Este comportamiento define una lógica umbral de activación clásica en sistemas de control embebidos.

Adicionalmente, el nodo incorpora un mecanismo de **deteción de fallo de comunicación** con el Nodo 3. Si durante 10 ciclos consecutivos no se recibe el mensaje correspondiente, se asume pérdida de señal y se desactiva el relé asociado a la alarma lumínica. Esto agrega un nivel básico de tolerancia a fallos.

De forma periódica, el Nodo 2 transmite una trama CAN con el identificador **0x120**. Dicha trama consta de 2 bytes que comunican el estado interno del nodo respecto a:

- **Byte 0:** Indica si se ha recibido correctamente información de temperatura en el último intervalo.
- **Byte 1:** Refleja si la alarma de nivel de luz ha sido activada (umbral excedido).

Esta funcionalidad permite que el nodo actúe como **fuentede diagnóstico o telemetría básica**, facilitando a otros nodos (o una interfaz de monitoreo) verificar el estado operativo del sistema en tiempo real.

#### C. Nodo 1 – Nodo supervisor con interfaz de usuario tipo HMI

El Nodo 1 constituye la unidad de mayor complejidad funcional del sistema didáctico basado en red CAN BUS, integrando capacidades de producción y consumo de datos, monitoreo visual local mediante una interfaz de visualización, y generación de comandos. Este nodo ilustra con claridad el paradigma de control distribuido con interacción hombre-máquina y puede verse a continuación en la Figura 9:

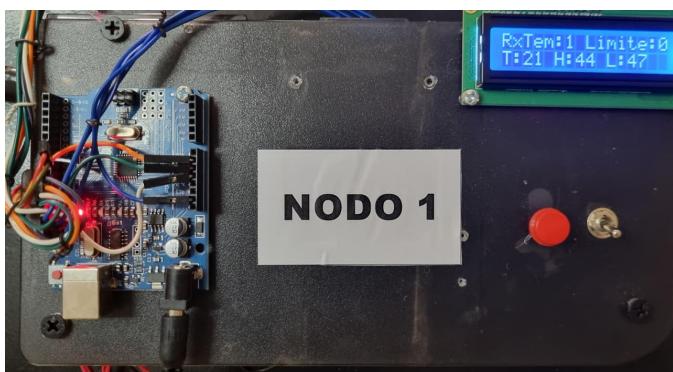


Figura 9. Representación física del Nodo 1.

El núcleo de procesamiento está conformado por una placa **Arduino Uno**, basada en el microcontrolador **ATmega328P**, que se comunica con un controlador **MCP2515** y su correspondiente transceptor CAN, conectados por una interfaz **SPI**. La tasa de transferencia configurada es de **125 Kbps**, conforme al estándar **CAN 2.0B** y operando en **modo normal**.

El nodo incorpora dos dispositivos de entrada de usuario:

- 1) **Switch 1** (pulsador): Un pulsador configurado con resistencia interna de pull-up, cuyo accionamiento produce un cambio de estado lógico.
- 2) **Switch 2** (interruptor con enclavamiento): Permite mantener su estado alto o bajo de manera persistente hasta ser accionado nuevamente.

Estos dos dispositivos representan señales digitales binarias que el nodo convierte en mensajes CAN periódicos con identificador **0x110**. Cada transmisión contiene un Data Length Code (DLC) de **2 bytes**, distribuidos de la siguiente forma:

- **Byte 0:** Estado lógico del pulsador momentáneo (0: no presionado, 1: presionado).
- **Byte 1:** Estado lógico del interruptor enclavado (0: posición de reposo, 1: activado).

La lectura del estado de las teclas se efectúa cada **200 ms**, garantizando la actualización frecuente del estado de los comandos generados por el operador.

El Nodo 1 se encuentra diseñado para recibir y procesar información proveniente de los nodos 2 y 3, con el propósito de informar al usuario sobre el estado general del sistema y las condiciones ambientales:

- 1) Mensajes provenientes del Nodo 2 (ID 0x120):
  - a) **Byte 0:** Bandera de estado de recepción de temperatura. Un valor de 1 indica que el Nodo 2 recibe periódicamente datos de temperatura/humedad; un valor de 0 implica pérdida de comunicación o falla en la adquisición.
  - b) **Byte 1:** Estado de la alarma de nivel de luz, indicando si el Nodo 2 ha detectado que el umbral luminoso fue excedido.

Estos datos se muestran en la primera línea del display **LCD alfanumérico 16x2**, accesible mediante una interfaz **I2C** configurada en la dirección **0x27**. El display refleja el mensaje de estado en tiempo real, permitiendo la supervisión de las alarmas y del correcto funcionamiento de los nodos distribuidores de datos ambientales.

- 2) Mensajes provenientes del Nodo 3 (ID 0x130):

- a) **Byte 0:** Temperatura medida.
- b) **Byte 1:** Humedad relativa.
- c) **Byte 2:** Nivel de luz.

El Nodo 1 actualiza estos valores en la segunda línea del **LCD**, mostrando los parámetros ambientales actuales. En caso de que el Nodo 2 notifique la ausencia de datos de temperatura (bandera de recepción en cero), el Nodo 1 sustituye los valores por indicadores de falta de información ("T:- H:- L:-").

#### D. Nodo 4 – Visualización Web Asíncrona de Datos

El Nodo 4, que puede verse a continuación en la Figura 10, está constituido por un microcontrolador **ESP32**, el cual se comunica con el bus CAN mediante un transceptor **SPI** basado en el circuito **MCP2515**. Su función principal es la de **nodo consumidor**, especializado en la **visualización remota de datos** ambientales (temperatura y humedad) recolectados y transmitidos por el Nodo 3.

Este nodo implementa un servidor web asíncrono utilizando la biblioteca **ESPAsyncWebServer**, que permite la entrega de una interfaz gráfica en tiempo real sin bloquear el bucle principal del sistema. A través de esta página, accesible desde cualquier navegador conectado a la misma red local, se exponen los valores de temperatura y humedad, con una **frecuencia de actualización de 1 Hz** mediante llamadas

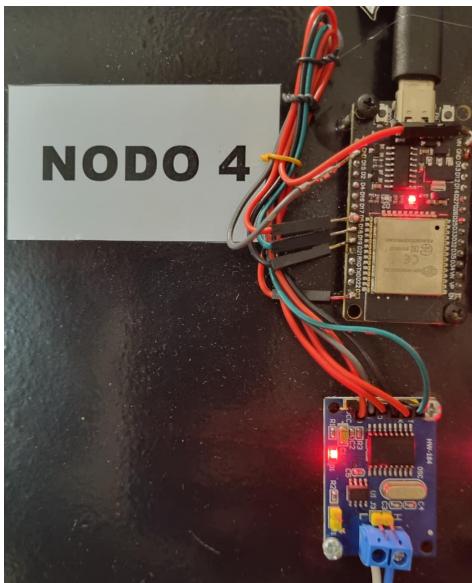


Figura 10. Representación física del Nodo 4.

fetch() a un endpoint REST que sirve los datos en formato JSON.

- **ESP32:** Actúa como unidad de procesamiento principal. Posee conectividad Wi-Fi integrada, lo que lo habilita para servir contenido web sin requerir hardware adicional.
- **MCP2515 + transceptor CAN:** Recibe los mensajes del bus CAN en modo SPI. La tasa de transmisión configurada es de 125 Kbps, en coherencia con los otros nodos.
- **Bus CAN:** El nodo filtra e interpreta únicamente los mensajes con ID 0x130, emitidos por el Nodo 3, los cuales contienen tres valores en los primeros tres bytes del campo de datos: temperatura (byte 0), humedad (byte 1), y nivel de luz (byte 2). Este nodo descarta el tercer byte (nivel de luz), dado que no lo utiliza.

Los datos recibidos son almacenados en variables globales (temperatura y humedad) y luego renderizados dinámicamente en una interfaz HTML contenida en memoria flash (PROG-MEM). Esta interfaz incluye:

- Encabezados institucionales, proporcionando el marco académico del proyecto.
- Dos tarjetas con íconos, unidades y formato gráfico profesional para la visualización de temperatura y humedad.
- Un pie de página que identifica a los autores del desarrollo.

El uso de técnicas modernas de desarrollo web, como la combinación de **HTML**, **CSS** y **JavaScript** con programación asíncrona, convierte a este nodo en una herramienta didáctica de alta accesibilidad y funcionalidad, particularmente útil para laboratorios de instrumentación, redes industriales y cursos de sistemas embebidos.

Aunque no genera mensajes propios ni actúa sobre la red, su función como **cliente pasivo de monitoreo web** es fundamental para ilustrar una de las aplicaciones más relevantes de CAN en contextos actuales: la visualización remota de variables medidas por sensores distribuidos. Representa, por tanto, un punto de integración entre el mundo físico de sensores embebidos y la capa de presentación orientada al usuario.

vantes de CAN en contextos actuales: la visualización remota de variables medidas por sensores distribuidos. Representa, por tanto, un punto de integración entre el mundo físico de sensores embebidos y la capa de presentación orientada al usuario.

#### E. Nodo 5 – Emisor digital de eventos basado en KL46Z

El Nodo 5, mostrado luego en la Figura 11, está implementado sobre una placa de desarrollo **Freescale/NXP FRDM-KL46Z**, equipada con el microcontrolador **MKL46Z256VLL4** de arquitectura **ARM Cortex-M0+**. Este nodo se incorpora a la red CAN como unidad exclusivamente transmisora, especializada en la detección de eventos digitales producidos por la interacción del usuario con pulsadores físicos.

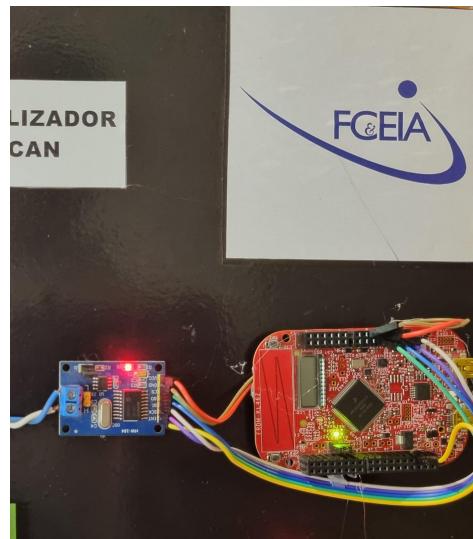


Figura 11. Representación del Nodo 5.

Su rol dentro del sistema es transmitir periódicamente, cada 500 ms, un mensaje con el estado lógico de los pulsadores **SW1** y **SW3**, permitiendo así que otros nodos (como los de monitoreo o visualización) puedan reaccionar ante acciones humanas distribuidas.

- **Microcontrolador MKL46Z:** Encargado de la lógica de adquisición digital, control del tiempo (vía SysTick) y la generación del mensaje CAN.
- **MCP2515 + transceptor CAN:** Controlador externo de red conectado mediante SPI al KL46Z, responsable de gestionar el protocolo CAN y la transmisión física de los mensajes.
- **Entradas digitales (SW1 y SW3):** Pulsadores conectados al puerto GPIOC, pines 3 y 12 respectivamente, configurados como entradas con pull-up internas activas, lo que permite una lógica negativa (presionado = 1 lógico).

El nodo transmite en forma periódica un mensaje con las siguientes características:

- a) **Identificador (ID):** 0x150 correspondiente al nodo transmisor.

b) **Data Length Code (DLC)**: 2 bytes.

c) **Campo de datos**:

- **Byte 0**: Estado lógico invertido del pulsador SW1 (1: presionado, 0: liberado).
- **Byte 1**: Estado lógico invertido del pulsador SW3 (1: presionado, 0: liberado).

Este diseño permite que los demás nodos interpreten fácilmente las acciones del usuario sin necesidad de conocer el hardware subyacente.

El envío periódico de los mensajes se implementa mediante la utilización del **SysTick Timer**, que genera interrupciones cada 1 ms. A partir de este mecanismo, se decrementa una variable de retardo (Delay) que define la frecuencia de envío (500 ms). Esta técnica evita el uso de retardos bloqueantes, manteniendo el diseño eficiente y no intrusivo.

El nodo incluye salida por consola serie para fines de diagnóstico, usando la biblioteca **fsl\_debug\_console**. Cada vez que se transmite un mensaje CAN, se imprime en el puerto serie la siguiente información:

- Confirmación del envío.
- Identificador del mensaje.
- DLC.
- Valores de los bytes transmitidos.

El Nodo 5 representa un ejemplo didáctico del modelo **productor-pasivo**, en el cual un nodo emite datos sin requerir retroalimentación ni ejecutar funciones de recepción. Su simplicidad lo hace idóneo para ilustrar principios fundamentales del modelo CAN.

#### F. Nodo analizador/sniffer CAN

El nodo analizador, también denominado **sniffer o monitor pasivo**, tiene como objetivo fundamental la observación del tráfico del bus CAN sin intervenir activamente en la comunicación ni modificarla. Este nodo fue implementado mediante un microcontrolador **Arduino Uno**, el cual se encuentra conectado a un transceptor **MCP2515**, un controlador CAN externo que se comunica con el microcontrolador a través del protocolo **SPI**.

El código embebido en el microcontrolador configura al MCP2515 en **modo normal** de operación, estableciendo una tasa de baudios de **125 Kbps**, acorde al resto de la red de prueba. Una vez inicializado el módulo, el nodo se dedica a consultar en forma continua si existe un nuevo mensaje disponible en el bus mediante la función `readMessage`. En caso de recepción exitosa (`ERROR_OK`), se accede a la trama CAN y se imprimen sus campos esenciales a través del puerto serie: el identificador del mensaje (`can_id`), el campo de longitud de datos (`can_dlc`) y el contenido del campo de datos (`data[]`), todo en formato hexadecimal.

Esta implementación permite obtener en tiempo real una traza del tráfico que circula por el bus, facilitando tareas de diagnóstico, verificación del protocolo implementado, análisis de latencias y detección de errores. Otro beneficio de este nodo es la obtención de una herramienta de diagnóstico de muy bajo costo, si lo comparamos con analizadores CAN comerciales.



Figura 12. Representación física del analizador CAN.

Dado que el sniffer no transmite mensajes, no requiere ID propio ni configuración de filtros, lo que lo convierte en una herramienta no intrusiva. No obstante, dado que el MCP2515 puede actuar como nodo receptor activo, es esencial garantizar que no se envíen ACKs ni interrupciones no deseadas, condición que se satisface mediante una adecuada configuración del entorno.

#### V. USO DE REPOSITORIOS COMO RECURSO DIDÁCTICO

A medida que se fue desarrollando el tablero con la mirada puesta en la enseñanza, se hizo evidente que el software es un recurso que hay que administrar y gestionar de forma prolífica con el objetivo de formar a los estudiantes. El tablero puede tener diferentes funcionalidades de acuerdo a los firmware de los distintos nodos que lo componen. Además, con el paso del tiempo se desarrollaron y podrían desarrollarse más mejoras o modificaciones; incluso los estudiantes, dependiendo la consigna pueden revertir el manejo de la comunicación entre los nodos. Con esto presente, se llegó a la conclusión que era necesario una herramienta que permita acceso a estas diferentes funcionalidades, junto con la documentación de cada caso.

Como recurso complementario al hardware del tablero, se plantea el uso de un repositorio de GitHub [7] para trabajar de forma colaborativa las diferentes versiones de cada funcionalidad del tablero, y para tener acceso centralizado a la documentación.

Desde el aspecto formativo, la utilización de un repositorio de GitHub prepara a los estudiantes para el manejo de versiones de código en un ambiente profesional, lo cual es muy demandado en el ámbito laboral.

Desde la mirada de los proyectos que pueden llevar adelante los estudiantes sobre el tablero, la utilización de repositorios permite mantener un historial de sus avances, realizar trabajos colaborativos y compartir de una forma sencilla con sus pares y docentes el código realizado para su análisis y estudio.

## VI. USOS DEL TABLERO

La utilización del tablero puede tener diferentes objetivos académicos. La herramienta dispone de mucha versatilidad, y aquí se mencionan los usos en los que se ha aplicado:

- **Cursos de capacitación de empresas:** Se utiliza con el objetivo de capacitar los aspectos conceptuales de operación del protocolo CAN BUS en relación con maquinaria agrícola. Esto es útil para que los técnicos puedan aprender metodologías sistemáticas para el diagnóstico. Persiguiendo ese objetivo, en una etapa formativa de estas características, el tablero se presenta con los Firmware cargados y, una vez comprendido el protocolo, se simulan fallas de funcionamiento.
- **Facultad:** Se utiliza en la actualidad en la asignatura Sistemas Digitales II de la carrera de Ingeniería Electrónica de la FCEIA (UNR) para la enseñanza del protocolo CAN BUS en profundidad. En este caso la herramienta permite vincular los aspectos teóricos con los prácticos, relacionando conocimientos previos de los estudiantes para alcanzar un aprendizaje significativo. En este contexto, desde la teoría se utiliza demostrativamente con todos los nodos ya programados, pero puede utilizarse en clases de práctica para que los estudiantes realicen modificaciones o mejoras sobre la comunicación ya existente. Adicionalmente, es posible utilizarlo en otros ámbitos de educación formal.
- **Colegios técnicos secundarios:** El tablero es posible utilizarlo en instituciones técnicas secundarias, y según el objetivo formativo, plantear las estrategias metodológicas acordes. El colegio secundario técnico es un lugar donde se prepara a los estudiantes para cuestiones prácticas, donde se pueda alcanzar la comprensión de lo conceptual y al mismo tiempo obtener herramientas de diagnóstico. Disponer del tablero permite un ahorro de tiempo en el armado de circuitos para realizar ensayos.
- **Colegios técnicos superiores terciarios:** El tablero es posible utilizarlo en instituciones técnicas terciarias, donde según la especialidad del título es posible diagramar diferentes estrategias de aprendizaje. En una institución terciaria donde la tecnicatura se orienta al control y automatización, es posible trabajar en los aspectos prácticos de CAN BUS para comprender protocolos industriales basados en él, como CANopen y DeviceNet. Si la institución tiene orientación en técnicas digitales, es posible abordar el tablero para comprender el funcionamiento de CAN BUS en cuanto a la estructura lógica de operación, firmware del microcontrolador, controladores CAN stand alone y otras cuestiones de los sistemas digitales.

La ventaja de ser una herramienta portátil, es que permite transportarla a diferentes lugares sin inconvenientes, por ejemplo para cursos de capacitación que se dictan en diferentes localidades.

## VII. PRÓXIMOS PASOS Y MEJORAS

Si bien el tablero se ha diseñado para cubrir la mayor cantidad de demandas en cuanto a la enseñanza académica en las áreas donde se utiliza, siempre es posible agregar más nodos con otras especificaciones al BUS de comunicación CAN para obtener variantes. Podría pensarse como módulos de expansión.

Por otro lado, una mejora a esta propuesta del tablero, es el diseño y construcción de otro tablero con el mismo concepto, pero con nodos y funcionalidades de hardware diferentes. Todo esto con el objetivo de simular ambientes de operación con otras características. Esto permitiría que cada grupo de estudiantes lleve adelante experiencias variadas en la forma de trabajo del protocolo. Lo mencionado, también permitiría la interconexión de tableros mediante el BUS de comunicación CAN, logrando escalar la propuesta didáctica, y volverla más o menos compleja en función de los objetivos de aprendizaje requeridos.

## VIII. CONCLUSIONES

La presente publicación describe una herramienta didáctica tecnológica, que fue desarrollada e implementada como una necesidad de mejorar la experiencia del proceso enseñanza/aprendizaje.

Se expone la importancia de la simulación de ambientes reales de operación del protocolo CAN BUS para lograr un aprendizaje significativo, basándose en diferentes estrategias metodológicas.

El desarrollo del tablero estuvo nutrido en base a las experiencias académicas y profesionales de cada uno de los autores, donde la heterogeneidad de las edades y recorridos personales permitieron diseñar la herramienta con una mirada amplia, posicionándonos en las perspectivas de estudiante, profesor y profesional de campo específico.

## IX. AGRADECIMIENTOS

- Ing. José Ignacio Sosa: Ingeniero eléctrico con orientación en electrónica de la Universidad Nacional de Rosario. Ex Director del Departamento de Sistemas e Informática de la Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Fue Director académico del curso de capacitación dictado en la empresa Agronorte desde la Escuela de Posgrado y Educación Continua, donde el tablero didáctico CAN BUS fue concebido para alcanzar los objetivos didácticos de la capacitación.
- Ing. José Luís Simón: Ingeniero electrónico de la Universidad Nacional de Rosario. Actual Director del Departamento de Sistemas e Informática de la Facultad de Ciencias Exactas, Ingeniería y Agrimensura.

## REFERENCES

- [1] D. Bortolami. (2020) Can-bus: Introduction and history. Altium Learning Hub. Created: August 28, 2020; Updated: September 28, 2020. [Online]. Available: <https://resources.altium.com/p/Controller-Area-Network-Bus-Introduction-and-History>
- [2] Wikipedia contributors, "Bus can," Wikipedia, la encyclopédie libre, 2025. [Online]. Available: [https://es.wikipedia.org/wiki/Bus\\_CAN](https://es.wikipedia.org/wiki/Bus_CAN)

- [3] YPF Ruta, “¿qué es y cómo funciona el can bus en nuestra flota?” *Blog YPF Ruta*, Jul. 2022. [Online]. Available: <https://ruta.ypf.com/que-es-y-como-funciona-el-can-bus-en-nuestra-flota.html>
- [4] P. Richards, “A can physical layer discussion,” Microchip Technology Inc., Application Note AN228A, 2002. [Online]. Available: <https://ww1.microchip.com/downloads/en/appnotes/00228a.pdf>
- [5] M. T. Inc., “Mcp2515 stand-alone can controller with spi interface data sheet,” Microchip Technology Inc., Data Sheet DS20001801J, 2019. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>
- [6] N. Semiconductors, “Tja1050 high-speed can transceiver data sheet,” NXP Semiconductors, Data Sheet, 2003, supersedes May 16, 2002 edition; Released: October 22, 2003. [Online]. Available: <https://www.nxp.com/docs/en/data-sheet/TJA1050.pdf>
- [7] SD2-DSI-FCEIA, “Repositorio sd2\_tablero\_can: Red can con múltiples nodos (frdm-kl46z, arduino, esp32),” [https://github.com/sd2-dsi-fceia/SD2\\_TABLERO\\_CAN](https://github.com/sd2-dsi-fceia/SD2_TABLERO_CAN), 2025, [En línea].