



**Metodología CDIO para el Desarrollo de
Productos Electrónicos del Internet de las
Cosas**

Johan Julián Molina Mosquera

Docente Universidad Surcolombiana

Doctorando Universidad Americana de Europa,

UNADE

Rodrigo Cadena Martínez

Profesor-Investigador, Universidad Americana de
Europa, UNADE

Eduardo Eloy Loza Pacheco

Doctorando Universidad Americana de Europa, UNADE

RESUMEN

En esta tesis se presenta la utilización de la
metodología CDIO (Concebir, Diseñar, Implementar

y Operar) para desarrollar productos electrónicos del Internet de las Cosas. Debido a la desactualización del plan de estudios del programa de Ingeniería Electrónica de la Universidad Surcolombiana, y al desconocimiento en temas de diseño electrónico con prototipado de circuitos aplicados a las soluciones de IoT, es necesario aplicar las etapas de la iniciativa CDIO. Esto permitirá que los estudiantes de sexto semestre, que cursen “Aplicaciones en la nube con sistemas embebidos”, adquieran las habilidades y competencias técnicas y blandas necesarias para desarrollar productos de IoT que sean robustos, escalables, seguros y personalizados. El diseño y uso de una herramienta tecnológica intuitiva en la etapa de concepción permitirá diagnosticar el contexto de la aplicación, seleccionar dispositivos y plataformas IoT, y analizar los costos según el dimensionamiento de la solución IoT.

Palabras claves: CDIO, IoT, Producto Electrónico.

ABSTRACT

This study titled "Knowledge Transfer and Innovation in the Spanish ICT Sector: An Analysis of Internal and External Sources (2008-2016)" by Dr. Carlos Eduardo Herrera Avendaño and Jesús María Martín Reina explores how internal and external sources of information contribute to innovation in the Information and Communication Technologies (ICT) sector in Spain. Using panel data from the Spanish National Institute of Statistics (INE), the

research seeks to understand the impact of these knowledge sources on process and product innovation during the 2008-2016 period. The study has three specific objectives: analyzing the factors influencing knowledge transfer in Spanish ICT companies, examining the complementarity between internal and external information sources in collaboration processes, and identifying the motivations for internal and external collaboration within the sector. Based on the literature on Knowledge-Based Economy (KBE) and Global Innovation Networks (GIN), the research posits that both internal and external knowledge sources significantly influence innovation. The methodology employs a deductive approach with multivariate causal hypotheses, using bivariate and multivariate probit models. Data were collected through the Technological Innovation Panel (PITEC) and the Community Innovation Survey (CIS) adapted for Spain. The study highlights the importance of understanding how ICT companies can enhance their innovative capacity through effective knowledge transfer and collaboration.

Keywords: ICT Industry, Innovation, Knowledge Sources

INTRODUCCIÓN

El programa de Ingeniería Electrónica de la Universidad Surcolombiana de Neiva (Huila) esta en proceso de actualizar su plan de estudios y mejorar las metodologías, didácticas, estrategias de

enseñanza aprendizaje y evaluación formativa. Esto se basa en un diagnóstico aplicado a 327 estudiantes matriculados, de los cuales respondieron 104 (31,8%), con un nivel de confianza del 95% y un margen de error del 8%. La encuesta se enfocó en el “Desarrollo de Proyectos del Internet de las Cosas (IoT)”. Los resultados fueron los siguientes:

- El 65,4 % de los estudiantes ha conocido el tema IoT en alguno de los cursos del programa académico.
- El 69,2% de los estudiantes no ha desarrollado ningún proyecto de curso sobre IoT.
- El 83,2% de los estudiantes no ha utilizado ninguna metodología como referencia para desarrollar su proyecto de IoT.
- El 19,6% de los estudiantes que realizaron un proyecto de curso han utilizado sitios web como referencia.
- El 95,3% de los estudiantes no ha logrado que su proyecto de IoT se convierta en producto electrónico robusto, escalable, seguro y a la medida.

Esto genera la pregunta de investigación:

¿De qué manera los estudiantes del programa de Ingeniería Electrónica de la Universidad Surcolombiana adquieren las competencias para desarrollar productos electrónicos aplicados al Internet de las Cosas?

METODOLOGÍA

Metodología CDIO: La metodología CDIO (Conceive-Design-Implement-Operate) (Crowley et al., 2014; Liu et al., 2017) es una iniciativa educativa para la enseñanza de la ingeniería que se enfoca en el desarrollo de habilidades y competencias prácticas a través del desarrollo de proyectos. Este enfoque se ha utilizado con éxito en una variedad de campos, incluyendo la ingeniería mecánica, eléctrica y de software. En el contexto del Internet de las cosas (IoT), la metodología CDIO puede ser utilizada para desarrollar soluciones integradas de hardware y software que se ajusten a las necesidades de los usuarios. **Concebir:** La fase de concepción de un proyecto de IoT implica la identificación de las necesidades y los objetivos de los usuarios, así como la definición de los requisitos técnicos y de negocio. Se pueden utilizar técnicas como el análisis de casos de uso y la investigación de mercado para determinar las características y funcionalidades que debe tener un producto electrónico de IoT.

Diseñar: En la fase de diseño, se desarrollan modelos y prototipos que permiten visualizar la solución de IoT. En esta fase, se utilizan herramientas de diseño y simulación para validar los conceptos y probar diferentes opciones de diseño. Se deben considerar aspectos como la arquitectura del sistema, la selección de componentes, la seguridad y la privacidad de los datos.

Implementar: La fase de implementación se enfoca en la construcción y pruebas de la solución de IoT. Se utilizan herramientas de desarrollo de software y hardware para implementar la solución, y se realizan pruebas para verificar el correcto funcionamiento del sistema. Se deben considerar aspectos como la interoperabilidad entre los diferentes componentes y dispositivos, la capacidad de escalar la solución y el consumo de energía.

Operar: En la fase de operación, se realiza el despliegue y la gestión de la solución de IoT. Se deben considerar aspectos como la configuración, el mantenimiento y la actualización del sistema, así como la monitorización y el análisis de los datos generados por la solución.

En la actualización del syllabus CDIO V2.0 en la que se declara la actualización de los objetivos para la educación de ingeniería (Gustavsson & Stolterman, 2015) se menciona que Concepción, Diseño, Implementación y operación de sistemas en el contexto empresarial, social y ambiental presenta una visión de como el desarrollo de productos o sistemas se mueve a través de esas cuatro metafases. Los términos elegidos son descriptivos de las industrias de hardware, software y procesos. De acuerdo con los objetivos específicos propuestos en el proyecto de tesis, se construye la tabla con las respectivas actividades, que servirán como fases de referencia y ruta metodológica para el desarrollo de un producto electrónico de IoT:

Tabla 1

Objetivos y actividades para el desarrollo metodológico CDIO

Objetivo General:

Diseñar la metodología CDIO (Concebir, Diseñar, Implementar y Operar) para desarrollar productos electrónicos de IoT. (Internet de las Cosas) robustos, escalables, seguros y personalizados.

ACTIVIDADES EN ETAPAS DE METODOLOGÍA CDIO					
Objetivos Específicos	Etapa de Concebir	Etapa de Diseño	Etapa de Implementación	Etapa de Operación	
Diseñar la herramienta tecnológica informática que permita diagnosticar el contexto de la aplicación, seleccionar el uso de dispositivos electrónicos y plataforma IoT de código abierto, definir las etapas de desarrollo según metodología CDIO y estimar los costos del producto electrónico de IoT.	Consultar el estado del arte de las plataformas del internet de las Cosas de código abierto. Seleccionar sistemas embebidos robustos que permitan la conexión por protocolo wifi. Seleccionar sensores y actuadores que se puedan adaptar a los sistemas embebidos.	Diseñar herramienta informática de acuerdo con la consulta de plataforma y selección de sistemas embebidos, sensores y actuadores, realizada en la etapa de concepción.	Implementar la herramienta de software como aplicativo web de consulta en línea para que muestre las recomendaciones de la solución IoT y estimación de costos del producto terminado, incluyendo el costo del uso de plataforma IoT.	Verificar conectividad al aplicativo de software y realizar pruebas de funcionalidad y posibles mejoras en herramienta tecnológica.	
Desarrollar un producto electrónico de IoT que cumpla con las etapas de la metodología CDIO determinada por la herramienta informática.	Identificar el problema que requiera la solución de IoT. Agregar el Marco teórico de la plataforma de IoT y los dispositivos de hardware como sistema embebido, sensor y actuador, utilizados para la solución de IoT	Diseñar el circuito esquemático que se conectara con plataforma IoT. Diseñar el algoritmo como pseudocódigo del sistema embebido que se ajusta a la plataforma IoT. Diseñar la placa de circuito impreso PCB del producto electrónico de IoT	Implementar en placa de pruebas el circuito que se conectara con plataforma IoT. Implementar la codificación del algoritmo en el lenguaje que requiera el sistema embebido. Ensamblar componentes en circuito impreso PCB para obtener un producto robusto y escalable.	Redactar manual de operación del producto electrónico	
Validar el uso del producto electrónico de IoT.		Diseñar la interfaz de usuario Gráfica para interacción máquina a máquina M2M entre aplicativo móvil y producto electrónico	Implementar la interfaz de usuario gráfica dependiendo de las funcionalidades que tenga la plataforma de IoT seleccionada.	Verificar conectividad del producto electrónico con plataforma IoT. Evaluar el uso del producto y determinar posibles mejoras de diseño.	

RESULTADOS

Resultados de producto electrónico

En la etapa de concepción se diseñó un aplicativo web tipo responsive con lenguajes de software como HTML, JavaScript y CSS, permitiendo la interacción con el usuario para la consulta a base de datos en formato JSON (JavaScript Object

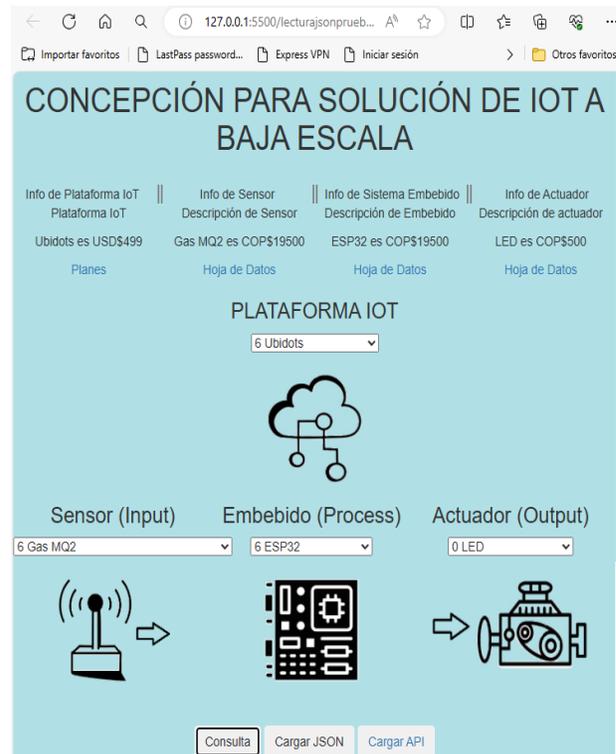


Notation) y una API REST. Este formato es ligero para el intercambio de datos y su estructura basada en pares clave-valor es flexible para manejarse en bases de datos como NoSQL y MongoDB.

La interfaz de usuario gráfica denominada “Concepción para solución de IoT a baja escala”(Figura 1), muestra cómo se estructura el diseño general de un producto electrónico de IoT en la que se consideran las cosas como sensores y/o actuadores y los dispositivos de borde como sistema embebido, medio de conexión, protocolos de comunicación y plataforma en la nube. Permitiendo que desde el botón “Consulta” se gestione la consulta de datos tipo JSON de forma local y servicio API REST a las hojas de datos de dispositivos, planes de plataformas de IoT y la respectiva estimación de costos.

Figura 1

Aplicativo Web para etapa de concepción



El consumo completo de datos en formato JSON puede obtenerse desde el botón “Cargar API”, ofrecido desde GitHub para el servicio API Rest, y accediendo a la dirección url “https://my-json-server.typicode.com/julmolina/IoT/db”

El producto de IoT desarrollado con la metodología CDIO es el “Medidor de gas propano”, y se describe cada etapa metodológica de la siguiente manera:

Etapas de Concepción

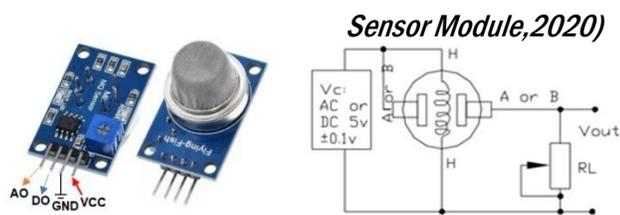
La cosa utilizada como dispositivo principal para la medición de gas butano es el sensor de gas MQ2, el cual se puede usar para detectar la presencia de gas licuado de petróleo “LPG”, propano e hidrógeno, también se puede usar para detectar metano, gas inflamable y humo (MQ2 Gas Sensor Module,2020).



El sensor MQ2 posee una capa sensible constituida por un óxido metálico, en este caso dióxido de estaño (SnO_2). En presencia de un gas, el óxido metálico provoca la disociación del gas en iones ocasionando transferencia de electrones y variación en la conductividad de la capa. Como este proceso depende de la temperatura, el sensor contiene un calefactor para mantener la temperatura de trabajo. Comercialmente, el sensor MQ2 generalmente se encuentra en módulo, lo que simplifica la parte de las conexiones y facilitan su uso. Su salida es analógica a través de un divisor de voltaje, como se muestra en la Figura 2. Este se polariza a 5 voltios DC, La resistencia sensible R_s del sensor (medida entre los terminales A y B) y R_L forman un divisor de voltaje (Nagy et al., 2020). Los terminales H se utilizan para proveer la corriente de calentamiento del sensor. Este módulo también tiene una salida digital, la cual permite su uso como detector (ON-OFF) al variar la sensibilidad con el potenciómetro.

Figura 2

Módulo de sensor Circuito de medición (MQ2 Gas



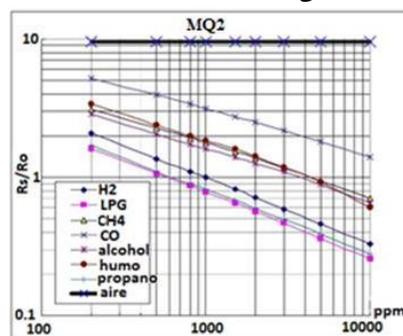
Las características de sensibilidad del sensor se muestran en la Figura 3, donde se observa en escalas logarítmicas la variación de la sensibilidad

R_s/R_o en función de la concentración de los gases que se miden. Con el MQ-2 se puede medir la concentración de la mayoría de los gases, pero su selectividad es menor, ya que el mismo valor de la relación R_s/R_o puede corresponder a diferentes gases, sin poder determinar a cuál de ellos se debe el resultado.

Las curvas de sensibilidad presentan el valor de R_s/R_o medido para el aire limpio, lo que puede ser un indicador importante en diversas aplicaciones.

Figura 3

Características de sensibilidad del MQ2 a diferentes concentraciones de gases



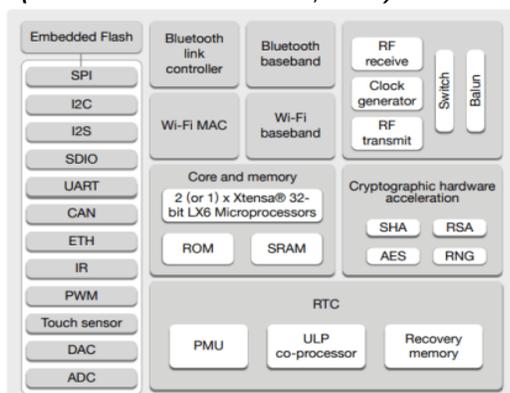
Las curvas características de sensibilidad a diferentes concentraciones de gases, se presentan en las condiciones ambientales como: temperatura a 20°C , Humedad relativa al 65% y Oxígeno O_2 al 21%. Al querer analizar el comportamiento R_s/R_o del aire. Primero se ubica el tipo de gas, en este caso "aire", luego se sitúa en el eje de la abscisa en el valor de 1000 ppm (partículas por millón), y por último se observa que en el eje de la ordenada se obtiene un $R_s/R_o=1$, en donde R_o es la resistencia del sensor para este gas en el aire limpio y R_s es la

resistencia del sensor a varias concentraciones de gases.

Como dispositivo de borde, que realiza el procesamiento de los datos de entrada del sensor de gas y transmite esos datos a la plataforma IoT, se encuentra la tarjeta ESP32. Este es un sistema en Chip SoC (System on Chip) que ofrece conectividad por señales WiFi y Bluetooth, con frecuencia de operación a 2,4 GHz, con prestaciones computacionales por disponer de CPU, memorias, Periféricos de entrada y salida de datos, reloj en tiempo real RTC, soporte de interfaces de comunicaciones como SPI, I2C, I2S; operando a baja potencia y con bloques de hardware utilizados para la seguridad en un solo chip (Bertoleti, 2019). En la Figura 4 se muestran los bloques funcionales del sistema embebido ESP32.

Figura 4

Bloques funcionales del Sistema embebido ESP32 (ESP32 Series Datasheet, 2023)



1. RIMI © 2023 by Eliz

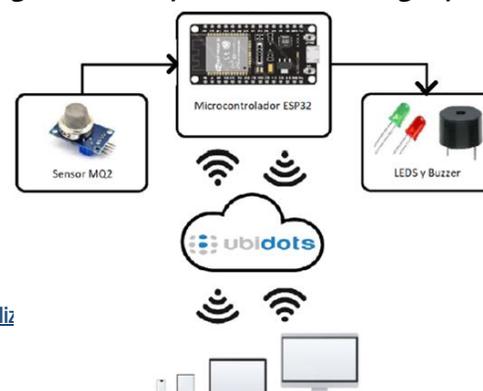
Como plataforma en la nube se utiliza “Ubidots” (es.ubidots.com/plans , 6 de junio de 2024), que permite la recopilación, el almacenamiento, la visualización y el análisis de datos en tiempo real. Diseñado para crear aplicaciones y soluciones de Internet de las cosas (IoT), incluye herramientas y servicios que facilitan la gestión de datos y la creación de paneles interactivos y personalizados. La plataforma Ubidots ofrece una amplia gama de características y funciones que permiten a los usuarios recopilar datos de diversas fuentes, como sensores, dispositivos y sistemas existentes, y verlos de forma clara y significativa. Proporciona una interfaz de usuario intuitiva para configurar y administrar el dispositivo, así como para crear reglas y notificaciones basadas en eventos.

Etapa de Diseño

En esta etapa se diseña el diagrama en bloques del producto electrónico “medidor de gas propano”, el circuito esquemático de las conexiones entre periféricos de entrada - salida de datos y el módulo ESP32, con el respectivo pseudocódigo.

Figura 5

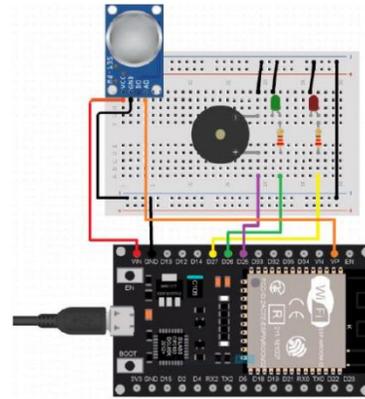
Diagrama en bloques de medidor de gas propano



En la figura 5 se muestra la conexión del sensor MQ2 como periférico de entrada al módulo ESP32, y la conexión del sistema embebido a los periféricos de salida, que serían los actuadores “Leds y Buzzer”, los cuales indicaran las alertas según la detección del gas propano.

El chip wifi con el que dispone el módulo, se conectara a la plataforma de IoT en la nube conocida como “ubidots”, para que desde la interfaz de usuario gráfica construida con sus componentes, se muestre la medición del gas y se generen las notificaciones al correo electrónico.

El diagrama circuital de conexión entre el sistema embebido ESP32 y el periférico de entrada (sensor MQ2) y periférico de salida (leds y buzzer), se muestra en la figura 6.



El algoritmo del sistema embebido ESP32 con el módulo sensor MQ-2, se presenta como pseudocódigo en la tabla 2 (Molina et al., 2023).

Algoritmo ESP32_Sensor_Gas

Declaración de variables Cloud, de tipo int para “condición”, de tipo bool

Para “presencia” y de tipo float para “valores_gas”

Definición de funciones para:

1. Configuración de velocidad por monitor serial
2. Inicio de propiedades y conexión a plataforma IoT de Ubidots
3. Ejecución cíclica del algoritmo
4. Ejecución de funciones creadas para los cambios de estado de las variables “condicion”, “presencia” y “valores_gas”.

En función de configuración se declara el pin GPIO36, para la entrada del

dato analógico entregado por el módulo sensor MQ-2.

También se declaran los pines GPIO25 y GPIO26 como salidas de datos digitales.

En función de ejecución cíclica se actualiza la conexión a plataforma

IoT y se realiza el proceso de lectura analógica en pin GPIO36.

Se convierte el dato de lectura analógica en valor de voltaje.

//Se declara las estructuras de decisión para las condiciones 0 y 1 del

//widget Value Dropdown y las estructuras de decisión anidadas.

Si (condicion==0) Entonces

Si (sensor_dato_ADC > 1900)

//Detección de gas propano, presencia=true;

//Muestra en dashboard el dato ADC y retardo de 1 seg;

FinSi

Si (sensor_dato_ADC <= 1900)

//No detección de gas propano, presencia=false;

//Muestra en dashboard el dato ADC y retardo de 1 seg;

Figura 6

Diagrama circuital del medidor de gas propano

```

FinSi
FinSi
Si (condicion==1) Entonces
  Si (sensor_dato_ADC > 1900)
    //Detección de gas propano, presencia=true;
    //Muestra en dashboard el dato de voltaje y retardo de 1
seg;
  FinSi
  Si (sensor_dato_ADC <= 1900)
    //No detección de gas propano, presencia=false;
    //Muestra en dashboard el dato de voltaje y retardo de 1
seg;
  FinSi
FinSi

```

Etapa de Implementación

Teniendo como base el diagrama circuital de conexión entre el sistema embebido ESP32 y periféricos de entrada y salida de datos, se realiza el diseño de la placa de circuito impreso PCB con el uso del software KiCad (Figura 7) , programación de tarjeta con lenguaje Arduino, y proceso de impresión de pistas en placa de cobre y ensamble de componentes en tarjeta de cobre (Figura 8).

Figura 7

Vistas frontal y trasera del diseño de circuito impreso PCB

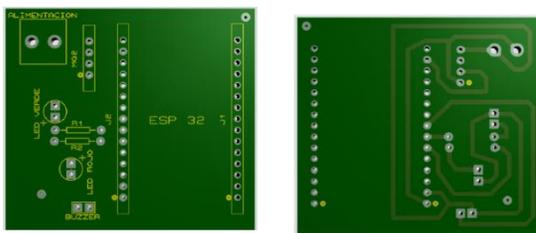


Figura 8

Circuito impreso en placa de cobre y ensamble de componentes

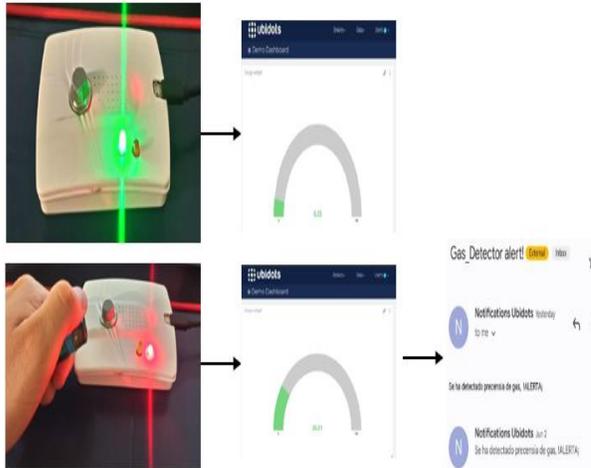


Etapa de Operación

Para la validación de la etapa operativa del diseño de la interfaz de usuario gráfica GUI generada desde la plataforma de Ubidots, se procede a la interacción máquina a máquina M2M como se muestra en la Figura 9 (No Detección y detección de gas con notificación) entre la aplicación web y el circuito del montaje de la Figura 8. En esta interacción y los diferentes componentes, se muestran las opciones de detección y valores de Partículas por millón PPM, nivel ADC o de voltaje, y el indicador de estado ON o de estado OFF de los actuadores que depende de la condición para la cantidad de gas propano detectado, siendo este gas el aplicado en la capa de oxido metálico del módulo sensor MQ-2.

Figura 9

No Detección y detección de gas con notificación a correo electrónico



El sistema embebido ESP32 permite la integración con otros sensores para ser escalable y mejorar la capacidad del sistema detector de gas. Por ejemplo, se pueden agregar sensores que miden variables como temperatura, humedad u otros parámetros ambientales para obtener una visión más completa del entorno y detectar posibles condiciones adversas.

Los estudiantes desde el sexto semestre del programa de Ingeniería Electrónica de la Universidad Surcolombiana adquieren las habilidades necesarias para desarrollar soluciones de IoT siguiendo la metodología CDIO. Esto incluye dominar diversas fases de desarrollo, desde el uso de herramientas de software y hardware hasta el diseño de algoritmos de programación, circuitos impresos PCB e interfaces de usuario gráficas GUI. Este enfoque les capacitará para crear productos de IoT robustos, adaptables y personalizados.



CONCLUSIONES

El producto electrónico cumplió con las expectativas de funcionamiento, resultando ser un dispositivo que da una solución eficaz y versátil para monitorear la presencia de gases en diversos ambientes.

REFERENCIAS

Crawley, E. F., Malmqvist, J., Östlund, S., & Brodeur, D. R. (2014). Rethinking engineering education: The CDIO approach. Springer.

Liu, X., Li, G., & Wan, J. (2017). IoT-based smart rehabilitation system: A framework for ubiquitous exercise monitoring and real-



time feedback. *IEEE Internet of Things Journal*, 4(1), 183-194.

Gustavsson, I., & Stolterman, E. (2015). The CDIO syllabus: A statement of goals for undergraduate engineering education. *International Journal of Engineering Education*, 31(4), 1074-1086.

MQ2 Gas Sensor Module. (2020). MQ2 gas sensor module. Retrieved from <https://5.imimg.com/data5/BG/QX/MY-1833510/mq2-gas-sensor-module.pdf>

Nagy, A. S., Polanco Risquet, A., Martínez de la Cotera, O. L., & Carralero Ibargollen, O. (2020). Medición simultánea de gases con sensores MQ. *Ingeniería Electrónica, Automática y Comunicaciones*, 41(1), 34-43. Epub 09 de marzo de 2020. Retrieved from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59282020000100034&lng=es&tlng=es

Bertoletti, P. (2019). *Proyectos com ESP32 y LoRa*. Brasil: Editora NCB.

ESP32 Series Datasheet. (2023). ESP32 datasheet. Retrieved from https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Ubidots. (n.d.). Retrieved from <https://es.ubidots.com/plans>

Molina, J, Salgado, J., & Bravo, M. (2023). *Circuitos Electrónicos con ESP32 y Plataforma IoT de Arduino*. editorial académica española.