

VALIDACIÓN DE UN MODELO PEDAGÓGICO PARA LA ENSEÑANZA DE IOT EN EDUCACIÓN MEDIA: SCRUM, PROGRAMACIÓN VISUAL Y HARDWARE DE BAJO COSTO

VALIDATION OF A PEDAGOGICAL MODEL FOR TEACHING IOT IN SECONDARY EDUCATION: SCRUM, VISUAL PROGRAMMING AND LOW-COST HARDWARE

José Hernán Ramírez Ramírez.

Profesor invitado de Robótica Aplicada. Colegio Sagrada Familia
Mérida- Venezuela. 5101
hernan.ramirez@gmail.com

Recibido: 11-08-2025

Aceptado: 12-10-2025

RESUMEN

Este artículo describe la implementación y validación pedagógica de una arquitectura distribuida robusta basada en MQTT y ESP32, diseñada específicamente para proyectos de investigación de Internet de las Cosas (IoT) en la Educación Media General. Para la gestión del proyecto, se adaptó la metodología ágil Scrum, la cual resultó fundamental para guiar a los estudiantes y fomentar la colaboración efectiva. El trabajo culminó con el desarrollo exitoso de tres proyectos principales y un proyecto piloto, empleando programación visual (Kodular) y un broker.

La validación se realizó mediante la triangulación de datos (cualitativos y cuantitativos) a partir de: una encuesta Likert de 5 puntos, métricas técnicas y observaciones del profesor. Los resultados confirman tanto la viabilidad técnica de la arquitectura como su impacto pedagógico. A través de encuestas y retroalimentación, se validó que la adopción de Scrum fue crucial para incrementar la motivación de los estudiantes (92%), mejorar la comunicación (83%) y desarrollar habilidades de resolución de problemas (83%).

Estos hallazgos demuestran que este modelo no solo facilita la asimilación de conceptos técnicos complejos, sino que también equipa a los estudiantes con habilidades blandas esenciales para el trabajo en equipo, transformando el aula en un entorno de aprendizaje dinámico y efectivo.

Palabras clave: Domótica, ESP32, IoT, MQTT, Programación Visual, Metodología Ágil, Scrum.

José Hernán Ramírez Ramírez: Doctor en Ecología del Desarrollo Humano, Sub área: Ingeniería de Software. Universidad Politécnica Territorial de Mérida "Kleber Ramírez" (UPTMKR). Maestría en Educación mención Informática y Diseño Instruccional Universidad de los Andes (ULA). Licenciado en Computación Tecana American University (TAU), USA. Técnico Superior en Informática IUFROnt Merida-Venezuela

VALIDACIÓN DE UN MODELO PEDAGÓGICO PARA LA ENSEÑANZA DE IOT EN EDUCACIÓN MEDIA: SCRUM, PROGRAMACIÓN VISUAL Y HARDWARE DE BAJO COSTO

VALIDATION OF A PEDAGOGICAL MODEL FOR TEACHING IOT IN SECONDARY EDUCATION: SCRUM, VISUAL PROGRAMMING AND LOW-COST HARDWARE

José Hernán Ramírez Ramírez.

Profesor invitado de Robótica Aplicada. Colegio Sagrada Familia
Mérida- Venezuela. 5101
hernan.ramirez@gmail.com

Recibido: 11-08-2025

Aceptado: 12-10-2025

ABSTRACT

This article describes the implementation and pedagogical validation of a robust distributed architecture based on MQTT and ESP32, specifically designed for Internet of Things (IoT) research projects in General Secondary Education. For project management, the Scrum agile methodology was adapted, which proved essential for guiding students and fostering effective collaboration. The work culminated in the successful development of three main projects and one pilot project, using visual programming (Kodular) and a broker.

The validation was carried out through data triangulation (qualitative and quantitative) based on: a 5-point Likert survey, technical metrics, and teacher observations. The results confirm both the technical viability of the architecture and its pedagogical impact. Through surveys and qualitative feedback, it was validated that the adoption of Scrum was crucial for: increasing student motivation (92%), improving communication (83%), and developing problem-solving skills (83%).

These findings demonstrate that this model not only facilitates the assimilation of complex technical concepts but also equips students with essential soft skills for teamwork, transforming the classroom into a dynamic and effective learning environment.

Key words: Domotics, ESP32, IoT, MQTT, Visual Programming, Agile Methodology, Scrum.

José Hernán Ramírez Ramírez: Doctor en Ecología del Desarrollo Humano, Sub área: Ingeniería de Software. Universidad Politécnica Territorial de Mérida "Kleber Ramírez" (UPTMKR). Maestría en Educación mención Informática y Diseño Instruccional Universidad de los Andes (ULA). Licenciado en Computación Tecana American University (TAU), USA. Técnico Superior en Informática IUFRONT Merida-Venezuela

Introducción

El Internet de las Cosas (IoT) ha emergido como un paradigma transformador en la sociedad contemporánea, fundamentado en la interconexión de objetos y la gestión eficiente de la información.¹ Este avance permite a los dispositivos cotidianos enviar y recibir datos para automatizar tareas y optimizar procesos, siendo la domótica (automatización del hogar) un campo clave que mejora el confort, la seguridad y la eficiencia energética en los espacios habitacionales.²

Este artículo aborda la necesidad de introducir estos conceptos avanzados en la Educación Media General mediante el desarrollo y la validación de una arquitectura distribuida. Esta arquitectura aprovecha el protocolo MQTT (Message Queue Telemetry Transport), el cual, a diferencia de las arquitecturas cliente-servidor tradicionales que pueden generar cuellos de botella,³ ofrece una solución escalable, flexible y basada en eventos.⁴ Para garantizar la accesibilidad técnica y económica del modelo, la implementación se basa en microcontroladores de bajo costo como el ESP32⁵ y plataformas de programación visual como Kodular.^{6,7}

La iniciativa del Colegio Sagrada Familia persigue un doble objetivo pedagógico: primero, sumergir a los estudiantes en la práctica de la ingeniería de sistemas distribuidos; y segundo, cimentar las bases para su desarrollo académico y la selección de futuras carreras universitarias. La selección de MQTT, ESP32 y programación visual no es arbitraria; estas herramientas, por su naturaleza ligera, económica e intuitiva, reducen significativamente la barrera de entrada, permitiendo a los estudiantes de este nivel abordar sistemas IoT complejos de manera efectiva.

Una contribución fundamental de este trabajo es la introducción y validación de la metodología ágil Scrum para la gestión de estos proyectos educativos. Scrum proporciona un marco estructurado que facilita la gestión de tareas de aprendizaje a largo plazo y promueve la colaboración

efectiva en el aula.⁸ La combinación de una arquitectura técnica accesible con un marco de gestión riguroso como Scrum establece un modelo replicable para otras instituciones que busquen integrar conceptos tecnológicos avanzados en la educación secundaria, fomentando la innovación y habilidades blandas esenciales.

Este estudio valida el modelo mediante la triangulación de datos cualitativos (observaciones del profesor) y cuantitativos (encuestas y métricas técnicas), asegurando un análisis robusto.

En el contexto venezolano, este modelo busca democratizar el acceso a tecnologías avanzadas en la Educación Media General, fomentando habilidades técnicas y blandas

Esta sección proporciona el fundamento teórico de los componentes técnicos y los conceptos subyacentes a la arquitectura propuesta y la metodología de desarrollo.

Internet de las Cosas (IoT) y Domótica

El Internet de las Cosas (IoT) se define como una red de dispositivos físicos, vehículos, electrodomésticos y otros elementos equipados con sensores, software y conectividad, que les permite intercambiar datos a través de internet.¹ Su núcleo es la comunicación Máquina-a-Máquina (M2M, por sus siglas en inglés: Machine to Machine), que posibilita la colaboración entre dispositivos sin intervención humana directa. La domótica, por su parte, es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda.² Al integrar IoT, la domótica se potencia, permitiendo la gestión remota y programación autónoma de sistemas de iluminación, climatización, seguridad y electrodomésticos, adaptándose a las necesidades del usuario.

La arquitectura de un sistema IoT se describe comúnmente mediante un modelo de tres capas: el Dominio de Sensores (o capa de percepción), encargado de la recolección de datos; el Dominio de Red (o capa de red), responsable de la transmisión de información; y el Dominio de Aplicación (o

capa de aplicación), que procesa los datos y proporciona servicios al usuario final.^{1,2}

Protocolo MQTT y Principios de Sistemas Distribuidos

MQTT (Message Queue Telemetry Transport) es un protocolo de mensajería asíncrona diseñado para ser extremadamente ligero y flexible, idóneo para dispositivos con recursos computacionales limitados y redes con ancho de banda restringido, características comunes en sistemas IoT.⁴

Opera sobre TCP/IP y sigue un modelo de publicación/suscripción. En este modelo, la comunicación no es directa entre los dispositivos (clientes), sino mediada por un componente central llamado broker. Los clientes que envían información (publicadores) la difunden en un canal específico, denominado tópico. Los clientes interesados en recibir esa información (suscriptores) se suscriben a dicho tópico. El broker se encarga de recibir los mensajes de los publicadores y reenviarlos a todos los suscriptores correspondientes.³

Esta estructura de comunicación tipo estrella, con el broker como punto central, permite que los sistemas sean no bloqueantes y desacoplados. El desacoplamiento implica que los publicadores no necesitan conocer la identidad ni el estado de los suscriptores, y viceversa. Además, el uso de topics jerárquicos (ej. hogar/sala/temperatura) facilita la organización lógica de la comunicación y la escalabilidad eficiente del sistema. Al ser un estándar de código abierto, reduce significativamente los costos de implementación.⁴

Un sistema distribuido es una colección de computadoras autónomas conectadas a través de una red que colaboran para realizar una tarea, presentándose ante el usuario como un único sistema coherente.⁹

Los principales objetivos de un sistema distribuido son la transparencia (ocultar la complejidad de la distribución al usuario) y la escalabilidad (capacidad de crecimiento sin pérdida de rendimiento).¹⁰ El modelo de publicación-suscripción de MQTT aborda

directamente varios desafíos de los sistemas distribuidos, al lograr un desacoplamiento referencial (los componentes no se conocen entre sí) y temporal (no necesitan estar activos al mismo tiempo), lo cual es fundamental para construir sistemas distribuidos resilientes y escalables.^{3,10}

Programación Visual (VPL) en Entornos Educativos

La Programación por Lenguajes Visuales (VPL, por sus siglas en inglés) es un paradigma donde el desarrollo de software se realiza mediante la manipulación de elementos gráficos (como bloques o iconos) en lugar de escribir código textual.⁶

Estas herramientas están diseñadas para ser intuitivas y fáciles de comprender, reduciendo la curva de aprendizaje para programadores novatos. Plataformas como Scratch, MakeCode, ArduinoBlocks y App Inventor (o sus derivados como Kodular) son ejemplos destacados. Sus ventajas en el ámbito educativo son notables: facilitan la implementación de lógica compleja, ayudan a desarrollar el pensamiento algorítmico y aumentan el interés y la motivación de los estudiantes por la informática y la ingeniería.⁶ En este proyecto, se utilizaron VPL como Kodular para que los estudiantes diseñaran y programaran aplicaciones móviles de control de forma rápida y visual, enfocándose en la lógica del sistema en lugar de la sintaxis de un lenguaje de programación tradicional.

Componentes Hardware (ESP32) y Aplicaciones Móviles para IoT

Para la implementación física de los proyectos, se seleccionó el microcontrolador ESP32, un sistema en un chip (SoC) de bajo costo que integra un potente microprocesador de doble núcleo con conectividad WiFi y Bluetooth, ideal para el desarrollo de sistemas IoT.⁵ A estas placas se conectan diversos sensores para capturar datos del entorno, como sensores de temperatura y humedad (DHT22), de pH y turbidez para líquidos, de humedad del suelo, ultrasónicos (HC-SR04) y cámaras para registro fotográfico. Para interactuar con el

entorno, se utilizan actuadores como relés (para controlar dispositivos de corriente alterna), servomotores (para controlar aletas o compuertas), LEDs y pantallas LCD.

Los teléfonos inteligentes se han consolidado como la interfaz de usuario por excelencia para los sistemas domóticos. A través de aplicaciones móviles, el usuario puede monitorear el estado de su hogar y enviar comandos a los dispositivos.⁷

Estas aplicaciones pueden desarrollarse con entornos nativos o, de manera más accesible para principiantes, mediante herramientas VPL como App Inventor o Kodular. La comunicación entre la aplicación móvil y los dispositivos IoT se realiza a través del broker MQTT. La aplicación se suscribe a los topics relevantes para recibir datos en tiempo real de los sensores y publica mensajes en otros topics para enviar comandos, como activar un dispensador o encender una bomba de riego.⁷

La combinación del protocolo MQTT, el microcontrolador ESP32 y la simplicidad de la programación visual (Kodular) genera un efecto sinérgico que reduce significativamente la barrera técnica para que los estudiantes de Educación Media General implementen sistemas IoT distribuidos complejos.

Cada componente es accesible, pero su integración permite a los estudiantes concentrarse en la lógica y aplicación del IoT, en lugar de complejidades de bajo nivel de red, programación embebida o desarrollo de interfaces de usuario tradicionales. Este conjunto tecnológico integrado (ESP32-MQTT-Kodular) representa una herramienta pedagógica eficaz para la educación práctica en IoT, con potencial de escalabilidad a otros niveles educativos o programas de formación profesional.

Métodos y Materiales

La implementación de los proyectos se llevó a cabo siguiendo una metodología rigurosa que integró un diseño de arquitectura distribuida con un enfoque ágil de gestión de proyectos, específicamente Scrum.

Diseño de la Arquitectura Distribuida

La arquitectura distribuida se fundamentó en el uso del microcontrolador ESP32 como cliente MQTT, un broker central MQTT (CristalML) y aplicaciones móviles desarrolladas con Kodular para la interfaz de usuario y el control. El flujo de datos se estableció así: los datos recolectados por los sensores se envían desde el ESP32 (publicador MQTT) al broker CristalML; desde allí, el broker reenvía los mensajes a la aplicación Kodular (suscriptor MQTT), donde los estudiantes diseñaron interfaces para visualizar los datos en tiempo real. Inversamente, los comandos del usuario desde la aplicación Kodular se publican en topics específicos en el broker, y el ESP32 (suscriptor) recibe estos comandos para activar los actuadores correspondientes.

Esta naturaleza desacoplada de la comunicación, mediada por el broker, permitió un desarrollo modular y paralelo de los componentes de hardware y software.

Proceso de Desarrollo de Proyectos con Enfoque Ágil (Scrum)

La metodología Scrum, originada en el desarrollo de software,¹¹ fue adaptada para la gestión de estos proyectos educativos de IoT, que involucran tanto hardware como software. Este enfoque se seleccionó por su idoneidad para el prototipado rápido y el desarrollo iterativo,¹² así como por su capacidad para gestionar tareas de aprendizaje a largo plazo y fomentar la colaboración en un entorno de aula.⁸

Adaptación del Marco Scrum para Proyectos Educativos de Educación Media General

Aunque se adhirió a los principios fundamentales de la Guía Scrum oficial,¹¹ el marco se ajustó al contexto de Educación Media General:

Roles:

- Product Owner: El profesor/mentor asumió este rol, definiendo la visión general del proyecto y priorizando las funcionalidades.

- Developers:** Los estudiantes formaron equipos autoorganizados, responsables de todas las tareas de desarrollo.

- Scrum Master:** El profesor/mentor también actuó como Scrum Master, facilitando el proceso, eliminando impedimentos y asegurando la adhesión a los principios de Scrum.

⊗**Eventos:**

- Sprint Planning:** Al inicio de cada sprint (períodos de trabajo fijos, por ejemplo, de 1 a 2 semanas), los estudiantes y el profesor definían los "objetivos del sprint", que eran tareas más pequeñas derivadas del objetivo general del proyecto.⁸

- Daily Scrum:** Se realizaban reuniones diarias cortas (5-8 minutos) donde cada equipo de estudiantes revisaba el progreso ("Hecho", "En Progreso", "Por Hacer"), identificaba impedimentos y planificaba las tareas del día.⁸ Estas reuniones fomentaron la comunicación y la transparencia, permitiendo la detección temprana de problemas y la asignación equitativa de la carga de trabajo. La naturaleza estructurada pero flexible de Scrum, con su énfasis en las reuniones diarias y las revisiones de sprint, abordó directamente desafíos comunes en proyectos grupales estudiantiles, como la distribución desigual de la carga de trabajo, la falta de rendición de cuentas y la identificación tardía de obstáculos. Este ciclo de retroalimentación constante y de corta duración, inherente a Scrum, impulsó la resolución inmediata de problemas y promovió la responsabilidad compartida.

- Sprint Review:** Al final de cada sprint, los equipos demostraban su "producto funcional" (un incremento de hardware/software operativo) a sus compañeros y al profesor.¹³

- Sprint Retrospective:** Los equipos reflexionaban sobre lo que funcionó bien, lo que podría mejorarse y cómo optimizar la colaboración.¹²

⊗**Artefactos:**

- Product Backlog:** Una lista priorizada de funcionalidades para el proyecto completo, mantenida por el Product Owner (profesor/mentor).

- Sprint Backlog:** Tareas seleccionadas del Product Backlog para el sprint actual, desglosadas en elementos más pequeños y manejables por los estudiantes.⁸ Se utilizó un tablero físico y/o digital para visualizar el progreso.

- Incremento:** El "producto funcional" entregado al final de cada sprint.¹³ Para los proyectos IoT, esto significaba un componente funcional de hardware y software, como una lectura de sensor o un control de actuador.

Integración de Hardware y Software en el Ciclo de Desarrollo Ágil

La naturaleza iterativa de Scrum demostró ser particularmente adecuada para fusionar el desarrollo de hardware y software en el contexto de IoT.¹² Se enfatizó el "Desarrollo Guiado por Pruebas y Datos" y la importancia de los prototipos para validar suposiciones en etapas tempranas.¹³ La modularidad en el diseño del hardware (por ejemplo, módulos separados para sensores y actuadores) facilitó el desarrollo paralelo y la integración continua dentro de los sprints.¹³

La implementación de Scrum en este contexto educativo no solo facilitó la culminación de los proyectos, sino que también cultivó habilidades blandas esenciales, como la comunicación, la colaboración, la resolución de problemas y la adaptabilidad.

Estas habilidades son altamente valoradas tanto en el ámbito académico como en las carreras STEM profesionales, transformando la metodología en una experiencia de aprendizaje holística.

Recolección y Análisis de Datos

El proceso de recolección de datos se diseñó para obtener una validación robusta a través de la triangulación metodológica, combinando enfoques cualitativos y cuantitativos.

Métodos Cualitativos

Se empleó la observación participante no estructurada por parte del profesor/Scrum Master durante los eventos fundamentales de Scrum: las reuniones diarias (Daily Scrum), la Revisión de Sprints y las Retrospectivas. El objetivo fue registrar la dinámica de equipo, la evolución en la articulación y resolución de problemas técnicos, la colaboración, y la gestión autónoma del proyecto.

El registro de la observación se realizó a través de un cuaderno de notas o bitácora de observación, capturando narrativas descriptivas de las interacciones grupales, los diálogos sobre decisiones técnicas y las respuestas emocionales ante los desafíos.

La atención se centró en los siguientes artefactos y elementos de Scrum:

⊗ Manejo del Product Backlog y Sprint Backlog: Observación de cómo los estudiantes priorizaban tareas, descomponían funcionalidades complejas (por ejemplo, definir la arquitectura distribuida vs. programar un sensor) y estimaban el esfuerzo.

⊗ Corrección de Errores y Troubleshooting: Seguimiento de los procesos internos de los equipos para diagnosticar fallas, evaluar la causa raíz (si era un problema de hardware, firmware o la red MQTT), y la evolución hacia soluciones independientes.

⊗ Roles y Responsabilidades: Registro de cómo el profesor/Scrum Master facilitaba, removía impedimentos y guiaba la autogestión, en lugar de proporcionar soluciones directas. Por ejemplo, en lugar de corregir un fallo de conexión MQTT, el profesor preguntaba: "¿Qué parte del sistema (ESP32, red, o broker) está fallando y cómo lo verificarían?", forzando a los equipos a aplicar el conocimiento de arquitectura distribuida para la solución. También se registraron ejemplos de mediación cuando surgían conflictos por la distribución desigual de la carga de trabajo, promoviendo la responsabilidad compartida.

Las observaciones fueron codificadas

manualmente por el profesor mediante un enfoque temático. El análisis identificó patrones clave como una mayor colaboración en equipo, aumento de la autonomía y resolución efectiva de problemas técnicos, proporcionando patrones recurrentes de aprendizaje, motivación y desarrollo de habilidades blandas, esenciales para el éxito del modelo pedagógico.

Instrumento de Percepción Estudiantil (Evaluación Cuantitativa)

El principal instrumento cuantitativo utilizado fue la Encuesta de Percepción de la Metodología, administrada a los 12 estudiantes que participaron en los proyectos piloto y demostrativos, después de completar al menos un ciclo completo de Sprint. El instrumento busca cuantificar la percepción del estudiante respecto al impacto de la metodología ágil y el enfoque práctico en su aprendizaje y desarrollo de habilidades blandas.

Título del Instrumento: Encuesta de Percepción de la Metodología Ágil (Scrum) en Proyectos de IoT. Instrucciones: Por favor, marque con una 'X' la opción que mejor refleje su nivel de acuerdo con cada una de las siguientes afirmaciones, considerando su experiencia durante el desarrollo de los proyectos de IoT.

Tabla 1: Escala de Respuesta (Escala Likert de 5 Puntos):

Opción	Descripción
1	Muy en Desacuerdo (MED)
2	En Desacuerdo (ED)
3	Neutral (N)
4	De Acuerdo (DA)
5	Muy de Acuerdo (MDA)

Tabla 2: Cuestionario:

N°	Pregunta	MED (1)	ED (2)	N (3)	DA (4)	MDA (5)
1	La metodología Scrum facilitó la organización y el avance de las tareas en equipo.					
2	Las reuniones diarias (Daily Scrum) ayudaron a mejorar la comunicación y a resolver problemas rápidamente.					
3	La entrega de un "producto funcional" al final de cada sprint fue motivadora.					
4	La metodología Scrum fomentó el desarrollo de habilidades de resolución de problemas.					
5	Me sentí más motivado con este enfoque práctico que con los métodos de enseñanza tradicionales.					

La encuesta Likert de 5 puntos incluyó preguntas como 'La metodología Scrum aumentó mi motivación para trabajar en el proyecto y Las reuniones diarias mejoraron mi capacidad de comunicación con el equipo (ver la Tabla 2 del instrumento completo).

Nota: Los resultados de esta encuesta, consolidando las categorías De Acuerdo y Muy de Acuerdo, se presentan en la Sección 4 (Resultados) y son los que alimentan la Tabla de Encuestas.

Triangulación de Datos

Para ir más allá de los porcentajes y construir un argumento sólido, se empleó la triangulación metodológica. Esta estrategia fue crucial para asegurar la validez interna del estudio, especialmente dada la naturaleza de muestra pequeña ($N=12$), permitiendo que la evidencia se respalde mutuamente. El proceso consistió en el cruce sistemático de tres tipos de lentes de observación:

1.El Lente de la Percepción (Datos Cuantitativos): La voz de los estudiantes a través de la encuesta Likert (3.3.2). Esta fuente estableció qué tan motivados se sintieron o si percibieron una mejor comunicación (satisfacción subjetiva).

2.El Lente del Proceso (Datos Cualitativos):

La bitácora del profesor/Scrum Master (3.3.1), que sirvió como un registro de la realidad en el aula. Esta fuente respondió al por qué y al cómo de los cambios: ¿Se observó realmente el aumento de la colaboración o solo se reportó?

3.El Lente Objetivo (Métricas de Rendimiento): Las tasas de finalización de sprints y la latencia MQTT, que proporcionaron hechos medibles sobre la eficiencia técnica y de gestión.

La convergencia de estos tres lentes permitió validar cada hallazgo. Por ejemplo, si un equipo reportó una alta motivación (Fuente 1), verificamos que este sentimiento se tradujera en una mayor autonomía en la resolución de problemas (Fuente 2) y, finalmente, en una alta tasa de sprints completados (Fuente 3).

De esta manera, cada resultado se soporta en una prueba triple: lo que el estudiante sintió, lo que el profesor observó y lo que el sistema técnico confirmó, otorgando una validación robusta y multifacética a la propuesta IoT-Scrum.

Resultados

La validación de la arquitectura distribuida

Tabla 3: Resultados de la Encuesta de Percepción de Estudiantes sobre la Metodología Scrum (Porcentaje de Acuerdo)

Pregunta (Escala Likert: Acuerdo Total o Parcial)	Porcentaje de Acuerdo
La metodología Scrum facilitó la organización y el avance de las tareas en equipo.	92%
Las reuniones diarias (Daily Scrum) ayudaron a mejorar la comunicación y a resolver problemas rápidamente.	83%
La entrega de un "producto funcional" al final de cada sprint fue motivadora.	92%
La metodología Scrum fomentó el desarrollo de habilidades de resolución de problemas.	83%
Me sentí más motivado con este enfoque práctico que con los métodos de enseñanza tradicionales.	92%
<i>Leyenda: Muestra de 12 estudiantes. Los porcentajes combinan respuestas "Muy de Acuerdo" y "De Acuerdo".</i>	

y la metodología Scrum se demostró a través del desarrollo exitoso de un proyecto piloto y tres proyectos de investigación más complejos, todos completados por estudiantes de Educación Media General.

Evaluación del Aprendizaje y la Metodología

Los resultados de la evaluación pedagógica y técnica se presentan a continuación, divididos en hallazgos cuantitativos y su validación cualitativa, demostrando el impacto del enfoque IoT-Scrum.

Resultados Cuantitativos

Los resultados de la encuesta de percepción aplicada a los 12 estudiantes y las métricas de desempeño técnico y de gestión, presentados en la Tabla 3 y la Tabla 4, demuestran una alta percepción positiva y una alta eficiencia de gestión de proyectos.

La Tasa de Finalización de Sprints del 88%

demuestra la efectividad de la planificación y el flujo de trabajo ágil. El bajo Número Promedio de Iteraciones (1.6) necesarias para resolver un problema crítico indica una curva de aprendizaje rápida en el diagnóstico y la corrección de errores de la arquitectura distribuida. La latencia promedio del sistema distribuido se mantuvo en 195 ms, asegurando una experiencia de usuario fluida, lo cual valida la programación visual como una herramienta eficiente para el troubleshooting.



Figura 1: Distribución Porcentual de Percepciones de los Estudiantes.

Tabla 4: Métricas de Desempeño Cuantitativo de Proyectos y Viabilidad Técnica

Métrica	Proyecto 1: Domótica Básica	Proyecto 2: Gestión de Riego	Proyecto 3: Monitoreo Ambiental	Promedio General
Tasa de Finalización de Sprints (% de tareas completadas a tiempo)	88%	85%	91%	88%
Tiempo de Respuesta (Latencia) Promedio MQTT (ms)		210 ms	190 ms	195 ms
Número Promedio de Iteraciones para Resolver Fallo Crítico (por Sprint)	1.5	1.8	1.4	1.6
Leyenda: Las métricas técnicas validan la estabilidad de la arquitectura distribuida (MQTT/ESP32) y la eficiencia de la gestión ágil.				

La Figura 1, basada en la Tabla 1, muestra la distribución porcentual de los estudiantes que estuvieron "Muy de Acuerdo" o "De Acuerdo" con el impacto de la metodología. Se observa que el 92% de los estudiantes reportó mayor motivación y facilidad en la organización de tareas, destacando la aceptación del enfoque ágil. El 83% de acuerdo en el desarrollo de habilidades de comunicación y resolución de problemas es un hallazgo clave, sustentando la efectividad pedagógica del modelo.

Resultados Cualitativos

Los datos cualitativos se complementan y se validan mediante el análisis temático de la bitácora de observación del profesor/Scrum Master (3.3.1), proporcionando un análisis de las dinámicas en el aula. Los hallazgos se estructuraron en torno a los patrones recurrentes de aprendizaje, gestión del Backlog y resolución de errores que se identificaron durante los sprints.

⊕Aumento de la Autonomía y Metacognición: Inicialmente, los equipos mostraron una gran dependencia del profesor, especialmente en la definición del Product

Backlog y el diagnóstico de errores. Sin embargo, gracias al rol de facilitador del profesor y a la técnica de guía basada en preguntas (3.3.1), se observó que, para el tercer sprint, el 75% de los equipos lograban resolver problemas técnicos de manera independiente (por ejemplo, recalibrar un sensor DHT22 o identificar la causa de un fallo de conexión MQTT). La observación participante registró este cambio en la metacognición. Un estudiante reflejó esta evolución: "Las reuniones diarias me ayudaron a organizarme mejor, a entender que esto no es como una tarea que se entrega al mes, sino un proceso de mejora continua." (Comentario de estudiante).

⊕Mejora Significativa en la Colaboración y Comunicación: La implementación obligatoria de los Daily Scrums forzó la comunicación diaria. El profesor registró en la bitácora un cambio progresivo de discusiones superficiales a debates técnicos profundos sobre la arquitectura distribuida y el manejo del Sprint Backlog. Esta evidencia conductual corrobora el 83% de los estudiantes que reportaron una mejor comunicación en la encuesta (Tabla 1). Por ejemplo, se observó a un equipo debatiendo

si la latencia del sensor se debía a un problema de código (firmware en el ESP32) o a un cuello de botella en el broker MQTT, demostrando una comprensión holística del sistema distribuido.

⊙ **Resolución Efectiva de Problemas Técnicos (Troubleshooting):** El patrón temático más fuerte se centró en la capacidad de troubleshooting. La bitácora indica que, al ser guiados, los estudiantes aplicaban el conocimiento de la arquitectura distribuida (capas de red, publicación/suscripción MQTT) para diagnosticar fallas. La observación participante mostró cómo los equipos pasaron de la frustración a la aplicación de pasos lógicos (ej. "Primero verificamos si el ESP32 está publicando algo en el topic 'sensor/temperatura' antes de revisar la aplicación Kodular"), lo que llevó a una reducción en la necesidad de intervención directa del docente. Estos hallazgos cualitativos complementan la métrica objetiva de 1.6 iteraciones promedio necesarias para resolver impedimentos críticos (Tabla 2), demostrando que la eficiencia de gestión fue correlacionada con la mejora de las habilidades blandas.

Discusión

Los resultados obtenidos validan la propuesta de integrar una arquitectura distribuida (MQTT/ESP32) con una metodología ágil (Scrum) en el contexto de la enseñanza de IoT a nivel de Educación Media General.

La viabilidad técnica de la arquitectura queda confirmada por las métricas de rendimiento. El tiempo de respuesta promedio de MQTT de (Tabla 4) demuestra que la latencia del sistema es adecuada para la mayoría de las aplicaciones de IoT en entornos educativos, superando los requerimientos de interacción en tiempo real típicos de proyectos pedagógicos. Esta estabilidad subraya que el broker CristalML y los microcontroladores ESP32 son herramientas robustas y accesibles para replicar sistemas distribuidos a bajo costo.

Desde la perspectiva pedagógica, los hallazgos cuantitativos de la encuesta (Tabla

3) son altamente positivos, con un 92% de motivación y un 83% en comunicación y resolución de problemas. Esta alta percepción se correlaciona fuertemente con la eficiencia de gestión, evidenciada por la Tasa de Finalización de Sprints del 88% (Tabla 4). Este alineamiento entre la percepción positiva y el alto rendimiento objetivo sugiere que Scrum no solo mejora la experiencia, sino que es un impulsor efectivo de la productividad en proyectos tecnológicos.

La triangulación de datos es crucial para interpretar estos resultados. La observación del profesor (4.1.2) sobre el aumento de la autonomía estudiantil, donde el 75% de los equipos resolvieron problemas de forma independiente, se alinea con el 83% de los estudiantes que reportaron habilidades mejoradas de resolución de problemas (Tabla 3).

Este fenómeno apoya directamente estudios previos sobre el rol de Scrum en fomentar el aprendizaje autodirigido y la responsabilidad compartida en entornos de ingeniería.⁸ La práctica de la gestión del Backlog y la corrección de errores observada en la bitácora demuestra un desarrollo de competencias que va más allá del simple cumplimiento técnico, sino que se enfoca en el desarrollo de la metacognición.¹²

Al comparar estos hallazgos con la literatura, la combinación IoT-Scrum logra sus objetivos. Este estudio es pionero al integrar MQTT, ESP32 y Scrum en Educación Media General, a diferencia de entornos universitarios o profesionales,¹² ofreciendo un modelo accesible para estudiantes con experiencia técnica limitada. Mientras que otros estudios se centran solo en la viabilidad técnica del hardware de bajo costo o la aplicación de metodologías ágiles en general, esta investigación demuestra que la estructura de Scrum es particularmente efectiva para gestionar la complejidad inherente de los sistemas distribuidos.

Los Daily Scrums y las Retrospectivas actuaron como mecanismos de mitigación de riesgos de integración que son comunes en los proyectos de IoT, permitiendo a los

estudiantes identificar la causa raíz de las fallas entre las capas del sistema (código, red o broker).

Limitaciones Metodológicas: La principal limitación del estudio es el tamaño muestral reducido (N=12) y su naturaleza de estudio de caso único en un contexto específico (Educación Media General). Si bien la

triangulación de datos minimiza el impacto de esta limitación, se debe reconocer la posible subjetividad en la observación participante del profesor. Investigaciones futuras deberían incluir grupos de control y evaluaciones ciegamente codificadas para validar los hallazgos en muestras más amplias y diversas.

Conclusiones

El objetivo de validar la arquitectura distribuida (MQTT/ESP32) junto a la metodología Scrum para la enseñanza de IoT fue alcanzado satisfactoriamente.

Viabilidad Técnica

La arquitectura distribuida basada en ESP32, Programación Visual (Kodular) y el broker MQTT CristalML demostró ser una solución técnica robusta, escalable y viable a bajo costo para proyectos de Educación Media General. La latencia promedio de 195 ms garantiza la estabilidad de la comunicación y valida que los estudiantes pueden implementar sistemas distribuidos funcionales, cumpliendo con los objetivos técnicos del curso.

Impacto Pedagógico

El impacto pedagógico es significativo. La implementación de Scrum fomentó un entorno de aprendizaje activo donde los estudiantes experimentaron una mejora palpable en habilidades clave:

⊕ **Colaboración y Comunicación:** El 83% de los estudiantes percibieron una mejor comunicación (Tabla 1), lo cual fue corroborado cualitativamente por la evidencia conductual de debates técnicos profundos sobre la arquitectura en los Daily Scrums (4.1.2).

⊕ **Autonomía y Metacognición:** El rol del Scrum Master, centrado en guiar en lugar de corregir, promovió la independencia. La observación del profesor de que los estudiantes comenzaron a proponer soluciones de troubleshooting de manera independiente durante las retrospectivas ilustra el desarrollo de habilidades metacognitivas necesarias para abordar la complejidad de IoT. La alta tasa de motivación del 92% (Tabla 3) se tradujo en una gestión eficiente de los proyectos.

⊕ **Habilidades Técnicas Integradas:** El enfoque en la gestión del Backlog y la resolución de errores (troubleshooting) guió la aplicación del conocimiento de arquitectura distribuida, resultando en una métrica objetiva de eficiencia de 1.6 iteraciones promedio para resolver impedimentos críticos (Tabla 4).

Replicabilidad y Trabajo Futuro

Los resultados sugieren que este modelo es altamente replicable en instituciones de educación media y técnica. Como trabajo futuro, se recomienda desarrollar un modelo de evaluación cuantitativa del desempeño técnico individual, con el fin de correlacionar con mayor precisión la percepción de las habilidades blandas (Tabla 3) con la contribución de código y la resolución de fallos en el sistema IoT.

También se podría explorar la integración de una plataforma Agile digital más robusta para

optimizar la gestión del Product Backlog. Se recomienda validar el modelo en muestras más grandes y contextos diversos para confirmar su generalización.

Agradecimientos

El más sincero agradecimiento a la Unidad Educativa Sagrada Familia por facilitar el espacio y los recursos necesarios para la implementación de este proyecto piloto. Un reconocimiento especial se extiende a los estudiantes de 5to año de Educación Media General, promoción XII (período 2024-2025), cuya dedicación y entusiasta participación fueron fundamentales para la recolección de datos y la validación del modelo pedagógico propuesto.

Referencias

- 1.- Patel KK, Patel SM. Internet of Things-IOT: Definition, characteristics, architecture, enabling technologies, application & future challenges. *Int J Eng Sci Comput.* 2016;6(5):6122-31.
- 2.- Al-Ali AR, Al-Rousan M. Java-based home automation system. *IEEE Trans Consum Electron.* 2004;50(2):498-504. doi:10.1109/TCE.2004.1309449
- 3.- Thangavel D, Ma X, Valera A, Tan HX, Tan CKY. Performance evaluation of MQTT and CoAP via a common middleware. En: 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP); 2014 Apr 21-24; Singapore. Piscataway, NJ, USA: IEEE; 2014. p. 1-6. doi:10.1109/ISSNIP.2014.6827678
- 4.- Hunkeler U, Truong HL, Stanford-Clark A. MQTT-S: A publish/subscribe protocol for Wireless Sensor Networks. En: 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing; 2008 Jun 11-13; Taichung, Taiwan. Los Alamitos, CA, USA: IEEE Computer Society; 2008. p. 23-30. doi:10.1109/SUTC.2008.10
- 5.- Schwartz M. Internet of Things with ESP32: Build amazing IoT projects using the ESP32 chip. Birmingham, UK: Packt Publishing Ltd; 2018.
- 6.- Fraser N. Ten things we've learned from Blockly. En: 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond); 2015 Oct 24; Atlanta, GA, USA. Atlanta, GA, USA: IEEE Computer Society; 2015. p. 49-50. doi:10.1109/BLOCKS.2015.7394334
- 7.- Wolber D, Abelson H, Spertus E, Looney L. App Inventor: Create your own Android apps. Beijing: O'Reilly Media, Inc.; 2011.
- 8.- Three Teachers Talk. Using Scrum in the Classroom [Internet]. 2019 Jan 25 [cited 2025 Aug 10]. Available from: <https://threeteacherstalk.com/2019/01/25/using-scrum-in-the-classroom/>
- 9.- Tanenbaum AS, Van Steen M. Distributed systems: Principles and paradigms. 3rd ed. Upper Saddle River, NJ: Prentice Hall; 2017.
- 10.- Coulouris G, Dollimore J, Kindberg T, Blair G. Distributed systems: Concepts and design. 5th ed. Boston: Addison-Wesley; 2011.
- 11.- Schwaber K, Sutherland J. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game [Internet]. Scrum.org; 2020 [cited 2025 Aug 10]. Available from: <https://scrumguides.org/>

- 12.- Weiss T. Scrum in device development: new trend or old hat? [Internet]. doubleSlash Blog; [cited 2025 Aug 10]. Available from: <https://blog.doubleslash.de/en/iot-and-connected-things/scrum-in-device-development-new-trend-or-old-hat/>
- 13.- Justice J. Scrum in Hardware Guide [Internet]. Scrum Inc.; [cited 2025 Aug 10]. Available from: <https://www.scruminc.com/scrum-in-hardware-guide/>
- 14.- Mitchell TM. Machine learning. New York: McGraw-Hill; 1997.