

# Gestor de tareas ágil que utiliza la mensajería distribuida, la comunicación en tiempo real y soporte en SCRUM

## Study of a Task Manager with SCRUM Support, Real-Time Communication, and Distributed Messaging for Agile Teams

Alejandro Páez Garrido<sup>1</sup>, Carolina Petro Alarcón<sup>2</sup>, Daniel Salas Álvarez <sup>3</sup>

- <sup>1</sup> **Universidad de Córdoba**, Facultad de Ingeniería, Departamento de Ingeniería de Sistemas y Telecomunicaciones, Grupo SOCRATES, Montería, Córdoba, Colombia, apaezgarrido02@correo.unicordoba.edu.co.
- <sup>2</sup> **Universidad de Córdoba**, Facultad de Ingeniería, Departamento de Ingeniería de Sistemas y Telecomunicaciones, Grupo SOCRATES, Montería, Córdoba, Colombia, cpetroalarcon47@correo.unicordoba.edu.co.
- <sup>3</sup> **Universidad de Córdoba**, Facultad de Ingeniería, Departamento de Ingeniería de Sistemas y Telecomunicaciones, Grupo SOCRATES, Montería, Córdoba, Colombia, ORCID: 0000-0002-7097-7883, danielsalas@correo.unicordoba.edu.co.

**Cómo citar:** Páez Garrido, A., Petro Alarcón, C., & Salas Álvarez, D. (2026). Gestor de tareas ágil que utiliza la mensajería distribuida, la comunicación en tiempo real y soporte en SCRUM. *Ingeniería, Sostenibilidad Y Sociedad*, 1(7), 31–42. <https://doi.org/10.24054/iss.v1i7.4478>

**Editorial:** Universidad de Pamplona.

**Recibido:** 8/diciembre/2025

**Aprobado:** 19/abril/2026

**Publicado:** 5/mayo/2026



**Resumen:** Este estudio utiliza un gestor de tareas ágil para mejorar la comunicación y la coordinación entre los miembros de un equipo de proyecto. Esto ayuda a aplicar mejor las metodologías ágiles en trabajos en equipo. Su propósito es evaluar el impacto del uso de esta herramienta en la productividad, la calidad del desarrollo y la satisfacción del equipo. Para ello, se desarrolló una investigación cuasiexperimental con grupos de estudiantes de Ingeniería de Sistemas, organizados bajo tres condiciones de trabajo: sin metodología, con SCRUM manual y con SCRUM asistido por el gestor de tareas. Durante cuatro semanas se recopilaron datos cuantitativos y cualitativos mediante registros de actividad, formularios de seguimiento y encuestas de percepción, lo que permitió comparar el desempeño y las dinámicas de colaboración entre los grupos. Los resultados muestran que los grupos que utilizaron el Task Manager lograron reducir en promedio un 25 % el tiempo de finalización, incrementar la productividad hasta en un 40 %, disminuir en más de un 60 % los errores semanales y alcanzar los mayores niveles de satisfacción. Estos hallazgos sugieren que la solución propuesta mejora de manera significativa la coordinación, la calidad y la eficiencia en proyectos de desarrollo ágil de software.

**Palabras clave:** Gestión ágil de proyectos, SCRUM, administrador de tareas, arquitectura de microservicios, comunicación en tiempo real, productividad de software.

**Abstract:** This study implements an agile task manager aimed at optimizing communication and coordination among project team members, strengthening the practical application of agile methodologies in collaborative environments. Its purpose is to evaluate the impact of using this tool on productivity, development quality, and team satisfaction. To achieve this, a quasi-experimental study was conducted with groups of systems engineering students organized under three working conditions: without methodology, with manual SCRUM, and with SCRUM supported by the task manager. Over four weeks, quantitative and qualitative data were collected through activity logs, progress reports, and perception surveys, allowing for the comparison of performance and collaboration dynamics among the groups. The results show that the groups using the Task Manager reduced completion time by an average of 25%, increased productivity by up to 40%, decreased weekly errors by more than 60%, and achieved the highest satisfaction levels. These findings suggest that the proposed solution significantly improves coordination, quality, and efficiency in agile software development projects.

**Keywords:** Agile project management, SCRUM, task manager, microservices architecture, real-time communication, software productivity.

## 1. INTRODUCCIÓN

La gestión de tareas en el desarrollo de software constituye un componente esencial para garantizar la productividad, la coordinación de equipos y la calidad de los productos generados. A medida que los proyectos se vuelven más complejos y se desarrollan en entornos distribuidos, las metodologías tradicionales de planificación han mostrado limitaciones para responder a las demandas actuales de flexibilidad y adaptación (Samadzadeh et al., 2025).

Diversos estudios han demostrado que la adopción de metodologías ágiles incrementa significativamente la productividad y la satisfacción del cliente. (Manisha et al., 2021), evidencian que el uso de Scrum reduce los defectos, mejora la calidad del producto y eleva la satisfacción del cliente, mientras que Setiana y Yunizar (2023) (Setiana & Yunizar, 2025) confirman que su implementación tiene un efecto directo y positivo sobre la productividad de los proyectos.

La gestión de tareas no debe entenderse solo desde un enfoque técnico, sino también organizacional y humano, donde la comunicación y la colaboración sean constantes. Ortiz Álvarez (2022) destaca que las herramientas basadas en Scrum fortalecen la interacción entre clientes y desarrolladores mediante flujos de trabajo estructurados y roles definidos, mejorando la eficiencia y la trazabilidad de las actividades. Y la experiencia de usuario (UX) también desempeña un papel determinante en la adopción de herramientas de gestión, ya que la facilidad de uso y la usabilidad influyen directamente en el compromiso y la productividad (Barroso Benítez et al., 2021). Sin embargo, Nikitashin et al. (2024) destacan que, a pesar de los avances, todavía hay diferencias entre lo que los usuarios esperan y lo que ofrecen las plataformas actuales. Esto muestra que se necesitan soluciones más completas y adaptables.

Por otra parte, la incorporación de arquitecturas modernas ha demostrado ser un aspecto importante para la eficiencia y escalabilidad de los

sistemas. Freire (2023) (Freire, 2025) destaca que las arquitecturas de microservicios reactivas y distribuidas permiten gestionar grandes volúmenes de trabajo mediante comunicación asíncrona y orquestación con Kubernetes, asegurando baja latencia y resistencia operativa.

Es así como el presente trabajo propone diseñar e implementar un Task Manager orientado a equipos ágiles, que integra microservicios, comunicación en tiempo real y mensajería asíncrona para optimizar la planificación y ejecución de tareas. El objetivo es ofrecer una herramienta que mejore la productividad, la trazabilidad y la colaboración en entornos distribuidos, articulando la innovación tecnológica con las necesidades organizacionales de la industria del software contemporánea.

## 2. METODOLOGÍA

La arquitectura del Task Manager se sustenta en un conjunto de tecnologías contemporáneas orientadas a garantizar la escalabilidad, la modularidad y una experiencia de usuario eficiente. En el backend se implementó ASP.NET Core, seleccionado por su robustez y capacidad para desarrollar microservicios de alto rendimiento, mientras que el frontend se construyó con Angular, framework que permite ofrecer una interfaz de usuario dinámica bajo el modelo de Aplicación de Página Única. La persistencia de datos se gestiona mediante MongoDB, base de datos NoSQL que facilita el almacenamiento flexible de tareas, subtareas y registros de actividad dentro de un entorno distribuido.

Para optimizar la comunicación y sincronización del sistema, se integraron tecnologías complementarias. SignalR habilita actualizaciones en tiempo real entre usuarios y servidores. RabbitMQ es encargado de la mensajería asíncrona y el intercambio confiable de información entre microservicios. Esta práctica es validada en arquitecturas de contenedores modernas

(Decimavilla Alarcón & Marcillo Franco, 2025). Asimismo, el uso de Docker y Docker Compose permitió estandarizar la ejecución de los servicios en entornos aislados y reproducibles (Cruz Diaz & Ramos Pacheco, 2025) mientras que Azure se adoptó como plataforma de despliegue y automatización, garantizando procesos continuos de integración y entrega. La gestión colaborativa del código se apoyó en Git y GitHub, herramientas que facilitaron la trazabilidad y coordinación del equipo de desarrollo.

La metodología de evaluación del sistema se diseñó bajo un enfoque cuasiexperimental, aplicado a grupos de estudiantes de Ingeniería de Sistemas durante la ejecución de su proyecto final de curso. Todos los equipos desarrollaron un mismo producto —una aplicación de comercio electrónico—, lo que permitió establecer condiciones homogéneas para la comparación. Se evaluaron tres escenarios: grupos sin metodología ágil, grupos que aplicaron SCRUM de forma manual y grupos que utilizaron el Task Manager con integración SCRUM. El uso de Scrum para gestionar el desarrollo de software, incluso en proyectos técnicos complejos como migraciones o implementaciones de microservicios, ha sido documentado como un enfoque viable para estructurar el trabajo (Cando Carmilema, 2025; Poveda Alzate, 2024). El objetivo del estudio fue analizar el impacto de la herramienta en variables como la productividad, la calidad del desarrollo y la satisfacción del equipo. Este diseño permitió observar diferencias significativas en el desempeño y la organización de los grupos, demostrando el potencial del Task Manager para optimizar la coordinación, reducir errores y mejorar la eficiencia en contextos académicos y de desarrollo ágil, de forma similar a como Gómez Ortega (2021) implementó Scrum y la gestión de roles para desarrollar el frontend de una aplicación empresarial de gestión de órdenes.

### 2.1. Diseño del estudio

La población estuvo conformada por estudiantes de Ingeniería de Sistemas. La muestra incluyó 15 estudiantes, organizados en cinco subgrupos de tres integrantes. Estos subgrupos se clasificaron en tres condiciones: Grupos sin uso de metodología ágil ni administrador de tareas (1A, 1B). Grupo que aplicó la metodología SCRUM de manera lógica, sin apoyo tecnológico (2A). Y grupos que emplearon el administrador de tareas integrado con SCRUM (3A, 3B). El seguimiento se realizó semanalmente durante el último corte académico, con una duración mínima de cuatro semanas.

El estudio buscó cuantificar diferencias en productividad, calidad y tiempo de entrega entre los tres tipos de grupos. Adicionalmente, se evaluó la consistencia del avance semana a semana, el nivel de comunicación y coordinación interna, y la percepción de los estudiantes sobre la experiencia de uso y la adopción de la metodología.

## 2.2. Instrumentos y recolección de datos

La recolección de información se apoyó en tres fuentes: Primero, se revisaron los registros del Task Manager y los repositorios de código de los grupos que tenían integración tecnológica. El objetivo era encontrar métricas claras sobre la productividad y el control de versiones. En segundo lugar, se recopilaron formularios semanales diligenciados por los representantes de cada equipo, donde se documentaron avances, incidencias y correcciones realizadas durante el proyecto. Y, en tercer lugar, al concluir el periodo de implementación, se aplicaron encuestas de satisfacción y entrevistas semiestructuradas orientadas a captar las percepciones de los participantes respecto a la carga de trabajo, la organización interna y la calidad de la comunicación entre los integrantes del grupo.

## 2.3. Protocolo de seguimiento

En la semana inicial se aplicó un cuestionario de línea base para registrar

experiencia previa en metodologías ágiles y estado inicial del proyecto. Durante cada semana del corte, los representantes reportaron avances y errores, y se consolidaron métricas a partir de los logs y repositorios. Finalmente, al concluir el proyecto, se aplicaron encuestas de satisfacción y entrevistas, y los datos fueron anonimizados para su análisis.

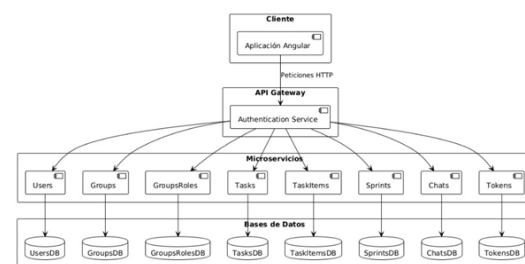
## 2.5. Consideraciones éticas

Se garantizó el anonimato de los datos de los participantes y se obtuvo su consentimiento informado antes de la aplicación de las encuestas y entrevistas. Para el estudio, se controlaron variables de confusión como la experiencia previa en metodologías ágiles y la carga académica adicional. Dado el tamaño reducido de la muestra, los resultados se interpretan de manera exploratoria, privilegiando la descripción y la triangulación de fuentes.

## 3. RESULTADOS

### 3.1. Especificaciones

El sistema se diseñó bajo un paradigma de microservicios, donde cada funcionalidad principal (gestión de usuarios, tareas, sprints, chats, etc.) opera como un servicio independiente.



**Figura 1.** Arquitectura lógica de la aplicación.

En la parte del cliente, se compone de una aplicación Angular que se comunica con el sistema mediante peticiones HTTP. Estas solicitudes son gestionadas por un API Gateway, el cual centraliza el acceso y canaliza las peticiones hacia los diferentes microservicios. Además de incluir un servicio de autenticación

(Authentication Service) que controla la validación y autorización de usuarios. Desde el API Gateway, las peticiones se distribuyen a los microservicios especializados: Users, Groups, GroupsRoles, Tasks, TaskItems, Sprints, Chats y Tokens, cada uno encargado de una parte específica de la lógica del negocio. Finalmente, cada microservicio interactúa de forma independiente con su propia base de datos —UsersDB, GroupsDB, GroupsRolesDB, TasksDB, TaskItemsDB, SprintsDB, ChatsDB y TokensDB—, garantizando la independencia y escalabilidad del sistema.

Cuando el usuario crea un grupo, se le atribuye el rol máximo en la metodología Scrum: Product Owner. Los demás usuarios se podrán unir copiando el código de grupo.

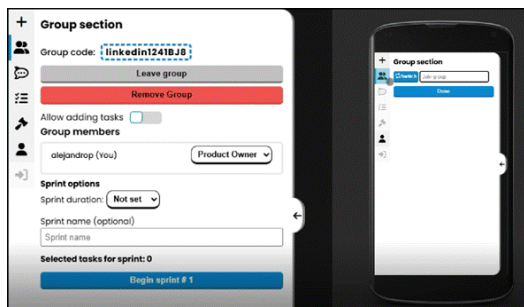


Figura 2. Interfaz de administración de sprint.

Se mostrará el usuario al cual fue asignado para esta subtarea junto con funcionalidades como eliminar subtareas y gestionar su organización de manera eficiente. Cada subtarea puede clasificarse por nivel de prioridad (baja, media o alta), al igual que la tarea principal, y se ordena automáticamente en el listado según dicha prioridad. Solo el Product Owner tiene la facultad de modificar estas prioridades. Además, el usuario puede obtener asistencia del IA Assistant, quien le ayudará a planificar y ejecutar la subtarea asignada, y también cuenta con la opción de marcarla como completada una vez haya finalizado su ejecución.



Figura 3. Interfaz de asignación de tarea.

### 3.2. Análisis comparativos de resultados

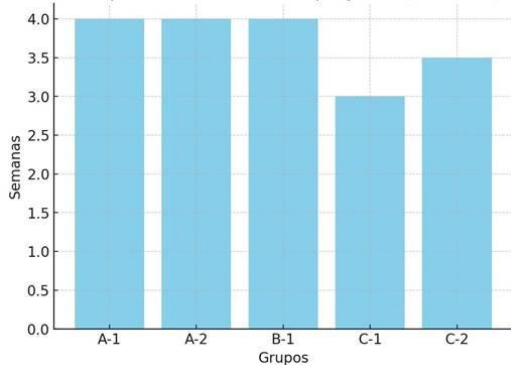
A partir del seguimiento semanal realizado a los grupos de estudiantes durante el desarrollo de sus proyectos, se identificaron diferencias sustanciales en cuanto al tiempo de finalización, la consistencia de los avances, la calidad del producto entregado y la percepción de satisfacción por parte de los integrantes.

Tabla 1. Resultados comparativos del desempeño de los grupos según metodología y uso del Task Manager.

Grupo	Tiempo de Finalización (semanas)	Consistencia (1-5)	Productividad (SP/sprint)	Errores/emana
A-1	4.0	1	95	5
A-2	4.0	2	100	4.5
B-1	4.0	3	115	3.5
C-1	3.0	5	160	1.5
C-2	3.5	5	150	2.0

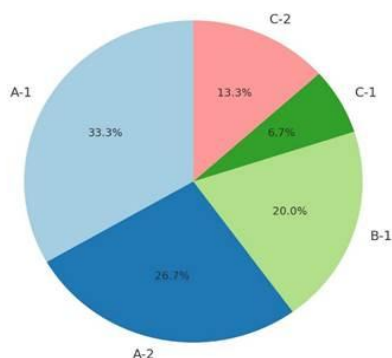
Además, se documentó que el grupo A-1 presentó avances desordenados y evidenció una falta de coordinación en su trabajo. El grupo A-2 logró cumplir con los objetivos planteados, aunque lo hizo bajo una notable presión en la etapa final. Por su parte, el grupo B-1 intentó seguir la metodología Scrum, pero terminó trabajando de una manera propia, adaptando el proceso a su conveniencia. Y en contraste con los anteriores, el grupo C-1 mostró una excelente organización y una comunicación clara entre sus integrantes. Y el grupo C-2 mantuvo un avance fluido, destacando que el uso de sprints facilitó significativamente el desarrollo de su trabajo.

Los grupos sin metodología concluyeron en promedio al límite de las 4 semanas, mientras que aquellos que aplicaron SCRUM finalizaron con una ligera anticipación. En contraste, los grupos que integraron SCRUM con el Task Manager lograron entregar sus proyectos hasta una semana y media antes.



**Figura 4.** Tiempo de finalización del proyecto de cada grupo, indicado en semanas.

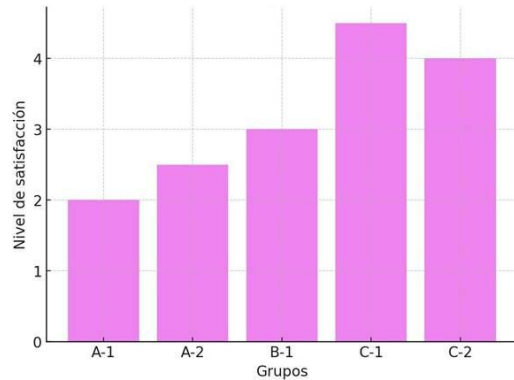
Casi la mitad de las incidencias provinieron de los equipos sin metodología, lo que refleja deficiencias en comunicación y control de calidad. Los grupos con SCRUM redujeron estos valores, pero aún reportaron fallos significativos en las fases finales. En cambio, los equipos con Task Manager presentaron el menor número de errores.



**Figura 5.** Errores promedio por grupo.

La percepción de los participantes indica que los grupos sin metodología tuvieron la menor satisfacción, reflejando estrés y desorden durante el proceso. Los equipos con SCRUM reportaron una satisfacción intermedia,

asociada al esfuerzo adicional para adaptarse a la metodología sin soporte tecnológico. En cambio, los equipos con Task Manager expresaron altos niveles de satisfacción, destacando la claridad en la asignación de roles y la transparencia en el seguimiento del proyecto.



**Figura 6.** Niveles de satisfacción por grupo en una escala de 1 a 5.

Se muestra que no tener una metodología clara (grupos A) resultó en que los tiempos de finalización estuvieran muy cerca del límite del plazo, hubo poca consistencia en los avances semanales y se cometieron más errores. Aunque estos equipos lograron completar sus entregas, el proceso estuvo caracterizado por desorden, presión acumulada y correcciones constantes hasta la etapa final.

Por su parte, el grupo que implementó SCRUM sin apoyo tecnológico (grupo B) obtuvo una ligera mejora en los tiempos de entrega y un incremento moderado en productividad. Sin embargo, se vio que el seguimiento manual de la metodología no fue suficiente para lograr un desarrollo estable, ya que los avances se concentraron en las últimas semanas y hubo problemas de comunicación y control de errores.

Los grupos que integraron SCRUM con el Task Manager (grupos C) presentaron un desempeño superior en todas las métricas evaluadas. Estos equipos no solo lograron finalizar sus proyectos hasta un 25 % antes del plazo máximo, sino que además mantuvieron un avance constante en cada sprint,

redujeron de manera notable la aparición de errores y alcanzaron niveles de productividad entre un 40 % superiores respecto a los grupos base. Adicionalmente, las observaciones cualitativas señalan que la herramienta facilitó la claridad en la asignación de roles y mejoró la comunicación interna, lo cual repercutió directamente en una mayor satisfacción de los participantes.

#### 4. DESAFÍOS Y CONSIDERACIONES FUTURAS

A pesar de la robustez de la solución, el éxito a largo plazo de un sistema distribuido como el propuesto depende de abordar proactivamente los desafíos inherentes a su complejidad. En una arquitectura de microservicios, donde una sola solicitud de usuario puede desencadenar una cascada de interacciones entre múltiples servicios, es importante la capacidad de entender el estado interno del sistema a partir de sus salidas externas. Esto va más allá del monitoreo tradicional y se sustenta en tres pilares: Logs, Métricas y Trazas (distributed tracing).

Implementar una estrategia de observabilidad efectiva requiere enfrentar el desafío de correlacionar información dispersa proveniente de distintos componentes del sistema. Para lograrlo, es fundamental la correcta instrumentación y recolección de métricas clave, conocidas como las "Cuatro Señales de Oro" propuestas por Google SRE. Estas son: latencia, que mide el tiempo de respuesta de las peticiones críticas y permite identificar degradaciones en el rendimiento. Tráfico, que monitorea la carga de los servicios y facilita la planificación de la capacidad. Errores, que calcula la tasa de fallos como el porcentaje de peticiones que dan como resultado respuestas erróneas (por ejemplo, códigos HTTP 5xx). Además, saturación, que evalúa el nivel de ocupación del sistema mediante el seguimiento del uso de CPU, memoria y profundidad de colas, ayudando a anticipar posibles puntos de sobrecarga.

Vinculado a lo anterior, la estandarización de contratos de eventos se presenta como un segundo desafío crítico. Sin un esquema de mensajería coherente y versionado, cada microservicio podría evolucionar de forma independiente hasta el punto en que la comunicación se rompa, generando errores difíciles de diagnosticar.

Es esencial tratar esto desde el principio para evitar una deuda técnica que perjudicaría las ventajas del modularidad. Este aspecto se considera un significativo obstáculo para la productividad del desarrollador (Pettersson, 2025) y su manejo adecuado se ve como una práctica excelente en Scrum (Pashchenko, 2024). Sobre la evaluación de la aplicación, aunque el estudio mostró diferencias claras entre los escenarios analizados, tiene limitaciones por el pequeño tamaño de la muestra y la falta de equilibrio en la forma en que se formaron los grupos.

La posibilidad de organizar subgrupos más homogéneos y comparables se vio limitada. Por ejemplo, en este caso no se logró contar con más de un grupo que aplicara SCRUM sin usar el Task Manager. Esta necesidad de validación empírica más robusta es una brecha reconocida en la investigación sobre prácticas ágiles (Alonso et al., 2021). En futuras investigaciones sería recomendable aplicar el estudio a una muestra mucho más amplia, extendiendo la invitación a todos los estudiantes matriculados en el curso donde se desarrolla el e-commerce como proyecto final, y no solo en la sede de Montería, sino también en las demás sedes de la universidad.

Esta convocatoria podría apoyarse en los profesores de los espacios académicos involucrados, quienes podrían facilitar la socialización y motivación de los estudiantes para participar voluntariamente en el seguimiento. Esto aseguraría que los grupos se dividan de manera más justa en cada condición experimental, lo que permitiría hacer comparaciones más

equilibradas entre los equipos que no usan metodología, los que aplican SCRUM manualmente y los que utilizan el Task Manager.

Otra línea de proyección consiste en trasladar el estudio a un contexto profesional. Resultaría de considerable valor aplicar el mismo seguimiento a equipos de desarrolladores en entornos reales, registrando su desempeño en proyectos previos donde no utilizaron la herramienta, y comparando estos resultados con otros nuevos gestionados mediante el Task Manager y bajo la metodología SCRUM.

Este tipo de evaluación longitudinal permitiría medir el impacto de la herramienta no solo en contextos académicos controlados, sino también en dinámicas de producción reales. En estas dinámicas, los desafíos de coordinación y comunicación entre múltiples equipos son aún más pronunciados (Eigner & Fahrnberger, 2025; Matcha & Kumar, 2025). Esto aportaría evidencia más sólida sobre su potencial de mejora en productividad, calidad y satisfacción de los equipos.

## 5. CONCLUSIONES

El uso del Task Manager integrado con la metodología SCRUM no solo optimiza los tiempos de desarrollo, sino que también garantiza un proceso más ordenado y colaborativo. En comparación, los equipos que trabajaron sin apoyo tecnológico o con una aplicación parcial de SCRUM enfrentaron mayores dificultades de organización y calidad. Esto es consistente con los desafíos reportados en la industria sobre las brechas de comunicación y la falta de disponibilidad de roles clave en Scrum (Faiz et al., 2025).

Este estudio ha mostrado cómo se ha diseñado y construido un gestor de tareas ágil. Esto se basa en combinar una arquitectura de microservicios, las prácticas de SCRUM y la comunicación en tiempo real. La solución muestra que es muy técnica y, mediante un análisis futuro, tiene un considerable potencial

para mejorar la forma en que trabajan los equipos de desarrollo.

La combinación de la arquitectura de microservicios con un diseño centrado en los principios de SCRUM posiciona al sistema como una base sólida y eficaz para contextos educativos y profesionales. Esto apoya la idea de que una herramienta tecnológica bien adaptada a la metodología puede ser clave para la mejora continua en el desarrollo de software. Esto potencia el impacto positivo que los roles ágiles efectivos tienen en los proyectos (Hamza, 2023).

Se evidenciaron diferencias significativas entre grupos que trabajaron sin metodología, aquellos que aplicaron SCRUM manualmente y los que integraron el Task Manager diseñado en este proyecto. La ausencia de un marco estructurado de trabajo se traduce en mayores retrasos, inconsistencias en los avances y una alta densidad de errores. Reforzando de esta manera dos premisas fundamentales, la primera, siendo la incorporación de tecnologías de soporte, tales como plataformas distribuidas de gestión y comunicación en tiempo real, constituye un catalizador para potenciar los beneficios de las metodologías ágiles; y la segunda, la experiencia de usuario y la claridad en la asignación de roles desempeñan un papel crítico para consolidar la motivación y la satisfacción de los equipos (Challa et al., 2024), donde no solo los aspectos pragmáticos, sino también los hedónicos del diseño, son imperativos para una mejor satisfacción (Ramírez García, 2025).

Si bien el estudio presenta limitaciones relacionadas con el tamaño y la composición de la muestra, los hallazgos ofrecen un punto de partida valioso para proyecciones futuras. Se reitera que es recomendable ampliar la aplicación a un número mayor de participantes y explorar su impacto en entornos profesionales. La validación longitudinal de la herramienta podría aportar evidencia aún más detallada

sobre su efectividad para transformar la dinámica de colaboración y la calidad del software producido.

## 6. REFERENCIAS

Alonso, S., Kalinowski, M., Viana, M., Ferreira, B., & Barbosa, S. D. J. (2021). A Systematic Mapping Study on the Use of Software Engineering Practices to Develop MVPs. Proceedings - 2021 47th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2021, 62–69.

<https://doi.org/10.1109/SEAA53835.2021.00017>

A. Bravo, Z. Nieto, J. Cristancho y L. Useche, «Perspectivas de investigación en torno a los ambientes de aprendizaje en el siglo XXI: una visión desde una región transfronteriza,» Revista Colombiana de Tecnologías de Avanzada, vol. 2, nº 42, pp. 150-157, 2023.

Alvarez Morales, E. L., Guillin Llanos, X. M., & Rodríguez Angulo, D. E. (2021). ANÁLISIS DE LOS EFECTOS QUE PRODUCE LA PRESENCIA DEL CADMIO EN EL CULTIVO DE CACAO (theobroma cacao). Ingeniería e Innovación, 9(2). <https://doi.org/10.21897/23460466.2723>

Arengas Acosta, J. M., Lopez Ramirez, M., & Guzman Cabrera, R. (2024). Impacto del preprocesamiento en la clasificación automática de textos usando aprendizaje supervisado y Reuters 21578. Revista Colombiana de Tecnologías de Avanzada, 1(43), 110–118.

<https://doi.org/10.24054/rcta.v1i43.2506>.

Arregocés, I., Ariza, M., Camargo, N., Díaz, J., & Gamarra, M. (2022). Integración de Scrum y RUP para el desarrollo de software de planes turísticos basado en preferencias de usuario. Revista ingeniería e Innovación. <https://doi.org/https://revistas.unicordoba.edu.co/index.php/rii/article/view/2974/5670>

Barroso Benítez, Y., Trujillo Casañola, Y., & Millet Lombida, Y. (2021). Marco de trabajo de evaluación de experiencia

de usuario en el desarrollo de software. Revista Cubana de Ciencias Informáticas, 15(3), 92–117. Editorial Ediciones Futuro. <https://www.redalyc.org/journal/3783/378369292006/html/>

Bastidas, J. V., & Vera, J. M. (2020). Biocombustible Sólido A Partir De Residuos Que Generan Los Procesos Agroindustriales Del Sector El Empalme. Ingeniería e Innovación, 8(22). <https://doi.org/10.21897/23460466.2333>

C. A. Mejía Rodríguez, M. A. Rincón Pinzón, L. M. Palmera Quintero, and L. M. Arévalo Vergel, "Aplicación de machine learning y metodología CRISP-DM para la clasificación precisa de severidad en casos de dengue," Revista Colombiana de Tecnologías de Avanzada (RCTA), vol. 1, no. 43, pp. 78–85, Mar. 2024. [Online]. Available: <https://doi.org/10.24054/rcta.v1i43.2822>.

Cando Carmilema, Á. V. (2025). Diseño de una arquitectura de microservicios para la integración y procesamiento de protocolos de comunicación industriales enfocado en alta disponibilidad [Tesis de maestría, Universidad Politécnica Salesiana].

Cardozo Rueda, K. S. (2022). Aplicación de redes neuronales artificiales para el pronóstico de precios de café. Revista Colombiana de Tecnologías de Avanzada, 1(39), 113–117. <https://doi.org/10.24054/rcta.v1i39.1403>.

Challa, K. K., Shaik, B. S., Gopalam, S. S., Chakravarthula, A., & Subramanyam, M. M. (2024). Agile Software Development. African Journal of Biomedical Research, 73–78. <https://doi.org/10.53555/ajbr.v27i2s.1258>

Cruz Diaz, C. T., & Ramos Pacheco, S. D. (2025). Diseño e implementación de microservicios en Java con Spring Boot para la gestión de transacciones bancarias en la empresa Socius Perú SAC, 2024 [Proyecto de grado, Universidad Peruana de Ciencias Aplicadas].

- Decimavilla Alarcón, D. C., & Marcillo Franco, P. F. (2025). Arquitectura de microservicios basada en contenedores para despliegue ágil de aplicaciones IoT en la nube. *Revista Episteme & Praxis*, 3(1), 35-49.
- Díaz, M.; Urdánigo, J.; Mercedes, A. y Muñoz, R. (2020). Cultura Ambiental en estudiantes de educación superior, 2020. *Revista Ingeniería e Innovación* ISSN: 2346-0474.
- Eigner, C., & Fahrnberger, G. (2025). Challenges in Scaling Agile Frameworks and Ways to Address Them with Scaled Agile Framework (SAFe) and Scrum of Scrums (SoS). *Communications in Computer and Information Science*, 2513 CCIS, 453–470. [https://doi.org/10.1007/978-3-031-94263-1\\_25](https://doi.org/10.1007/978-3-031-94263-1_25)
- Faiz, R. bin, Hamza, M., & Hussain, T. (2025). Investigating Challenges Faced by SCRUM Master while Practicing SCRUM Ceremonies in the Software Industry [Preprint]. *Research Square*. <https://doi.org/10.21203/rs.3.rs-7652158/v1>
- Freire, G. M. (2025). A distributed architecture of reactive microservices orchestrated by kubernetes — case study on load balancing in local cloud [Tesis de maestría, Universidade de São Paulo].
- García, D., Solórzano, C., Navarrete, Y., & Rojas, J. (2021). Características físicas, químicas y microbiológicas de la harina de banano morado (*Musa acuminata*) red dacca, producidos en los cantones Mocache, El Empalme y La Maná. *Ingeniería e Innovación*, 9(1), 1-12. <https://revistas.unicordoba.edu.co/index.php/rri/article/view/2418>
- Gomez, J. E. G., Pineda, J. J. G., & Riaño, V. L. H. (2026). Comparación de modelos de inteligencia artificial para la detección de cardiopatías usando imágenes de resonancia magnética. *REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA)*, 1(47), 34-45.
- Gomez Ortega, M. A. (2021). Desarrollo front-end de una aplicación empresarial para la gestión de órdenes [Tesis de maestría, Universidad de Antioquia]. <http://hdl.handle.net/10495/20140>
- Gómez, J. E. G., Cárdenas, S. R., & Ruiz, F. A. S. (2023). Sistema de identificación de pacientes basado en tecnología NFC y Blockchain. *Investigación e Innovación en Ingenierías*, 11(2), 1-15. DOI: <https://doi.org/10.17081/invinno.11.2.6671>.
- Gómez, J. G., Álvarez, D. S., & Ramírez, R. V. (2025, July). Sickle Cell Disease Patient Care System Using Artificial Intelligence. In *International Work-Conference on Bioinformatics and Biomedical Engineering* (pp. 230-241). Cham: Springer Nature Switzerland. [https://doi.org/10.1007/978-3-032-08452-1\\_19](https://doi.org/10.1007/978-3-032-08452-1_19).
- Gómez, J. G., Riaño, V. H., & Ramirez-Gonzalez, G. (2024). A context awareness system for clinical environments. *Electronics*, 13(15), 2999. <https://doi.org/10.3390/electronics13152999>.
- Gómez, J. G., Rosales, C. M., & Montaña, S. I. (2026). Sickle cell anemia prediction system in machine learning based on clinical data. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2026.3674938>.
- Gómez, K. Y., Caballero, L. A., y Maldonado, Y. Del C. (2021). Mejora De Un Proceso Productivo De Elaboración De Pan. *Revista De Ingeniería E Innovación*.
- H. Hernández Palma, D. J. Novoa, y D. Mendoza Cásseres, "Energía renovables y medidas de eficiencia energética aplicables a las instituciones prestadoras de salud en Colombia," *RCTA*, vol. 1, no. 41, pp. 123–131, may 2023, doi: 10.24054/rcta.v1i41.2557.
- Hamza, M. (2023). Effectiveness of Scrum Master in Agile Projects [Preprint]. *Research Square*.

<https://doi.org/10.20944/preprints202312.0046.v1>

Italo, E. G. (2020). Efecto de inclusión de cáscara de plátano en la degradabilidad in situ de ensilaje de maíz forrajero. *Ingeniería e Innovación*, 8(1).

<https://revistas.unicordoba.edu.co/index.php/rii/article/view/2327>

Lasso Cardona, L. A. ., Rincón Reyes, E. ., & Estrada Holguín, G. D. . (2020). Introducción a la evaluación de capacidades: una revisión teórica. *Revista Colombiana De Tecnologías De Avanzada (RCTA)*, 2(36), 34-43. <https://doi.org/10.24054/rcta.v2i36.18>.

Manisha, Khurana, M., & Kaur, K. (2021). Impact of Agile Scrum Methodology on Team's Productivity and Client Satisfaction - A Case Study. *Proceedings - 2021 3rd International Conference on Advances in Control and Networking, ICAC3N 2021*, 1686-1691. <https://doi.org/10.1109/ICAC3N53548.2021.9725505>

Matcha, S., & Kumar, P. A. (2025). Agile Methodologies in Practice: Evaluating the Effectiveness of Agile Practices in Large-Scale Software Projects. *Journal of Quantum Science and Technology*, 2(1), Feb(104-129).

<https://doi.org/10.63345/jqst.v2i1.224>

Miranda, O., Oyaga, R. F., Salas, A. R., Foris, Y. e Ibarquén, J. C. (2023). Impacto ambiental del botadero de residuos sólidos a cielo abierto en el corregimiento de Córdoba, departamento del Valle del Cauca. *Ingeniería e Innovación*, 11(1). <https://doi.org/10.21897/rii.3339>

Mosquera-Perdomo, A., Salazar Galindez, J., Ramírez-González, G., & Figueroa, C. (2023). Software for the extraction of bibliographic information registered in CvLAC and GrupLAC applied in the Department of Cauca. *Ingeniería E Innovación*, 11(2), 21. <https://doi.org/10.21897/rii.3464>

N. S. S. Carrero, N. M. A. Quintana, y L. M. S. Jaimes, "Lineamientos desde la industria 4.0 a la educación 4.0: caso tecnología IoT," *Revista Colombiana DE Tecnologías DE Avanzada (Rcta)*, vol. 1, no. 39, pp. 81-92, 2022. DOI: <https://doi.org/10.24054/rcta.v1i39.1379>.

Nikitashin, M., Kaluža, M., & Werber, B. (2024). Analysis of Methodologies and Tools for Software Development in Different Architectures. *2024 47th ICT and Electronics Convention, MIPRO 2024 - Proceedings, 1999-2007*. <https://doi.org/10.1109/MIPRO60963.2024.10569545>

Ortiz Alvarez, B. (2022). Herramienta para la gestión de actividades en los proyectos de software [Tesis de maestría, Universidad de Antioquia]. <https://hdl.handle.net/10495/31918>

Pashchenko, D. (2024). Excellence Practices in Scrum Paradigm in Software Development. *Journal of Data Science and Intelligent Systems*, 3(2), 79-86. <https://doi.org/10.47852/bonviewjdsis42023645>

Pettersson, M. (2025). Exploring factors that affect developer productivity and how they impact team performance and prioritization [Manuscrito en preparación].

Poveda Alzate, D. A. (2024). Cambio de Arquitectura Monolítica a Arquitectura basada en Microservicios y Serverless para el aplicativo inmobiliario Mobilia [Proyecto de grado, Universidad Piloto de Colombia].

Puche, M. B., Samper, O. M., & Martínez, R. F. (2023). Conciencia, concientización y educación ambiental: triada que se afianza en la primera infancia. *Ingeniería e Innovación*, 11-16. <https://doi.org/10.21897/rii.3416>

Ramírez García, J. de J. (2021). Estudio de la experiencia de usuario en los sistemas de gestión del aprendizaje. *IE Revista de Investigación Educativa de la REDIECH*, 12, e1358. Red de Investigadores Educativos Chihuahua A. C.

[https://doi.org/10.33010/ie\\_rie\\_rediech.v12i0.1358](https://doi.org/10.33010/ie_rie_rediech.v12i0.1358)

Román, G., Gómez, J. (2017). Intervención educativa apoyada en TIC en un proceso de enseñanza aprendizaje inclusivo. Ingeniería e innovación. Universidad de Córdoba. <https://revistas.unicordoba.edu.co/index.php/rii/article/view/1728>

Samadzadeh, M., Arab, A., & Valami, M. B. (2025). Agile Project Management for Information Technology.

Setiana, I., & Yunizar. (2025). The Effect of Scrum Implementation and Agile Organization on Project Productivity Mediated by Team Experience. Jurnal Ilmiah Manajemen Kesatuan, 13(5), 3725–3734.

<https://doi.org/10.37641/jimkes.v13i5.3798>

Vergara Pareja, C. M., Nino Vega, J. A., y Fernandez Morales, F. H. (2022). Fortalecimiento de la lectura crítica en inglés a estudiantes de grado quinto a través de un recurso educativo digital. Revista Colombiana de Tecnologías de Avanzada, 2(40), 160-170. <https://doi.org/10.24054/rcta.v2i40.2370>.