

## Comparison of agile methodologies for software development

### Comparación de metodologías ágiles para el desarrollo de software

#### Autores:

Merchán-Narváez, Nancy Jhanina  
UNIVERSIDAD CATÓLICA DE CUENCA  
Estudiante de la Unidad Académica de Informática,  
Ciencias de la Computación, e Innovación  
Tecnológica  
Cuenca – Ecuador



[nancy.merchan@est.ucacue.edu.ec](mailto:nancy.merchan@est.ucacue.edu.ec)



<https://orcid.org/0009-0004-2412-5139>

Palma-Peralta, Elena Elizabeth  
UNIVERSIDAD CATÓLICA DE CUENCA  
Estudiante de la Unidad Académica de Informática,  
Ciencias de la Computación, e Innovación  
Tecnológica  
Cuenca – Ecuador



[elena.palma@est.ucacue.edu.ec](mailto:elena.palma@est.ucacue.edu.ec)



<https://orcid.org/0009-0002-3931-6451>

Poma-Japón, Diana Ximena  
UNIVERSIDAD CATÓLICA DE CUENCA  
Docente de la Unidad Académica de Informática,  
Ciencias de la Computación, e Innovación  
Tecnológica  
Cuenca – Ecuador



[dpomaj@ucacue.edu.ec](mailto:dpomaj@ucacue.edu.ec)



<https://orcid.org/0000-0001-9231-1655>

Fechas de recepción: 15-FEB-2024 aceptación: 15-MAR-2024 publicación: 15-MAR-2024



<https://orcid.org/0000-0002-8695-5005>

<http://mqrinvestigar.com/>



## Resumen

Las metodologías ágiles son más adaptativas que predictivas, se centran en las personas no en los procesos. Las tendencias de innovación ofrecen la oportunidad de contrastar diversas metodologías ágiles en la búsqueda del camino, que conduzca hacia la eficacia y el éxito en el emocionante viaje del desarrollo de software, lo cual representa un desafío continuo. En la industria del desarrollo, el enfoque habitual se ha reemplazado progresivamente por metodologías ágiles, esto ha permitido afrontar eficazmente los desafíos de la gestión de proyectos y su entrega. No obstante, existe una variedad de metodologías ágiles, además, es importante percibir sus diferencias, y el cómo se adaptan a los contextos de desarrollo. Por lo que el presente artículo explora diversas divulgaciones sobre metodologías ágiles, sus bases y principios; especifica estándares para determinar su importancia; define y expone detalladamente las más destacadas. De ello se deduce que los métodos ágiles: XP, Scrum, Kanban y DSDM tienen diferencias significativas, pero comparten un enfoque común en la adaptabilidad, la excelencia del producto y el agrado del cliente. Por lo tanto, se puede destacar que ninguna metodología analizada, aborda de manera concreta la eliminación de desperdicios y la entrega continua de valor al cliente. Razón por la cual se propone una nueva metodología que combina principios ágiles y conceptos de Lean para garantizar la eficiencia, optimizar procesos y ofrecer productos de calidad que cumplan con las expectativas del cliente.

**Palabras Clave:** Comparación; Software; Métodos; Metodologías Ágiles

## Abstract

Agile methodologies are more adaptive than predictive, they focus on people rather than processes. Innovation trends offer the opportunity to contrast various agile methodologies in the search for the path that leads to efficiency and success in the exciting journey of software development, which represents a continuous challenge. In the development industry, the business-as-usual approach has been progressively replaced by agile methodologies, which has effectively addressed the challenges of project management and delivery. However, there is a variety of agile methodologies, and it is important to understand their differences, and how they adapt to development contexts. Therefore, this article explores various disclosures on agile methodologies, their bases and principles; specifies standards to determine their importance; defines and exposes in detail the most outstanding ones. It is deduced that the agile methods: XP, Scrum, Kanban and DSDM have significant differences, but share a common focus on adaptability, product excellence and customer delight. Therefore, it can be noted that none of the analyzed methodologies concretely addresses the elimination of waste and the continuous delivery of value to the customer. For this reason, a new methodology which combines agile principles and Lean concepts to ensure efficiency, optimize processes and deliver quality products that meet customer expectations is proposed.

**Keywords:** Comparison; Software; Methods; Agile Methodologies

## Introducción

Actualmente, la exploración académica en el desarrollo de software es influyente, ya que surgen nuevas directrices para mejorar la elaboración de software de primera calidad. Los responsables de cumplir con este objetivo son los métodos de desarrollo siendo evidente los cambios a lo largo del tiempo. Hoy existen metodologías que facilitan el desarrollo de software y algunas están estructuradas en fundamentos ágiles, utilizando procedimientos eficientes, etapas repetitivas, además se orientan en la colaboración, flexibilidad, escalabilidad y entrega continua de valor; permitiendo adaptarse rápidamente a los cambios del mercado y satisfacer ágilmente las necesidades de los clientes.

Senapathi y Srinivasan (2012), mencionan que cuantas más prácticas ágiles adopte una organización, mejor será el resultado, y a su vez hacer uso de factores de innovación y agilidad, factores sociológicos, tecnológicos y organizadores, mismos que inciden en el uso positivo de las metodologías ágiles. Lo expuesto se correlaciona directamente con el estudio realizado por Langley (2018), en donde un 71% de organizaciones han incrementado la competencia a través de la agilidad.

En naciones latinoamericanas como Perú, Chile, Brasil, México, Argentina y Colombia se han realizado varias investigaciones, las cuales manifiestan el avance de madurez de la aplicabilidad de las metodologías ágiles en esta región, en donde un 85% de las empresas se sitúan en fases medias en la adopción de estos métodos ya que estos brindan beneficios estratégicos y operativos, como una mayor innovación y competitividad dentro de las compañías (Lazzeretti, 2019).

Y en lo que concierne a Ecuador, la Dirección Nacional de Comunicación de la Policía Nacional (DINACOM) planteó como objetivo central desarrollar productos de software, usando metodologías ágiles. Al examinar las diferentes metodologías de desarrollo de software, concluyó que SCRUM se adecuaba de manera óptima a su método de trabajo, específicamente en la documentación del desarrollo de software. El caso más evidente se dio en la integración del sistema “SIIPNE 3W” (Loaiza, 2019).

Haciendo referencia a lo antes mencionado, es primordial resaltar que la gestión de desarrollo es eficaz e indispensable en la tecnología actual, donde la rapidez y la adaptabilidad son fundamentales para alcanzar el éxito empresarial. Por lo cual esta indagación nace de la necesidad de optimizar la eficacia y calidad de los procedimientos para desarrollar software, adoptando metodologías ágiles. Generando una mayor satisfacción por parte del cliente, proporcionando un software funcional más flexible y reduciendo los costos operativos. Además, este enfoque posibilita que las compañías conserven su competitividad en un entorno de mercado cada vez más sofisticado.



Sin embargo, la variedad de métodos ágiles disponibles y la falta de pautas claras para su elección generan un problema importante en la industria del software. Por ello, el propósito de esta investigación consiste en llevar a cabo una comparación entre metodologías ágiles, donde se ha decidido tomar en cuenta las más utilizadas: Scrum, Kanban, Extreme Programming (XP) y Dynamic System Development Methods (DSDM). Para cumplir con este objetivo se planteó los siguientes ítems de investigación:

- Características principales de las metodologías ya mencionadas.
- Casos propicios donde se aplican las metodologías ágiles.
- Ventajas y desventajas que ofrecen estas metodologías durante su utilización.

Esto posibilitará construir un fundamento sólido para tomar decisiones acertadas al momento de implementar métodos ágiles.

Este artículo contiene las siguientes secciones: sección dos, se presentan los conceptos vinculados a la investigación que permitirán comprender los resultados obtenidos, sección tres, se examinan los trabajos más destacados que aportan al desarrollo de esta investigación, sección cuatro, se describe la metodología empleada para efectuar la comparación de las metodologías ágiles, sección cinco se exponen resultados y conclusiones derivados de la investigación.

## Conceptos relacionados

**Manifiesto Ágil:** Es una divulgación de valores y principios (Satisfacción del cliente, adaptación al cambio, entregas frecuentes, trabajo en equipo, personal motivado, comunicación directa, software funcional, continuidad, excelencia técnica, simplicidad, equipos autoorganizados, mejora continua), que establece nuevas formas de desarrollar software, esto surgió en el año 2001 como respuesta a los métodos formales tradicionales que se utilizaban en la industria en ese momento (Durán, 2023).

**Ingeniería de Software:** Estudio práctico sistemático, disciplinado y valorizable mediante el cual se logrará analizar las necesidades del usuario de manera rentable, confiable y certificada en máquinas reales, obteniendo información necesaria que permite desarrollar, operar, brindar mantenimiento de software y analizar la documentación relacionada (López, 2018).

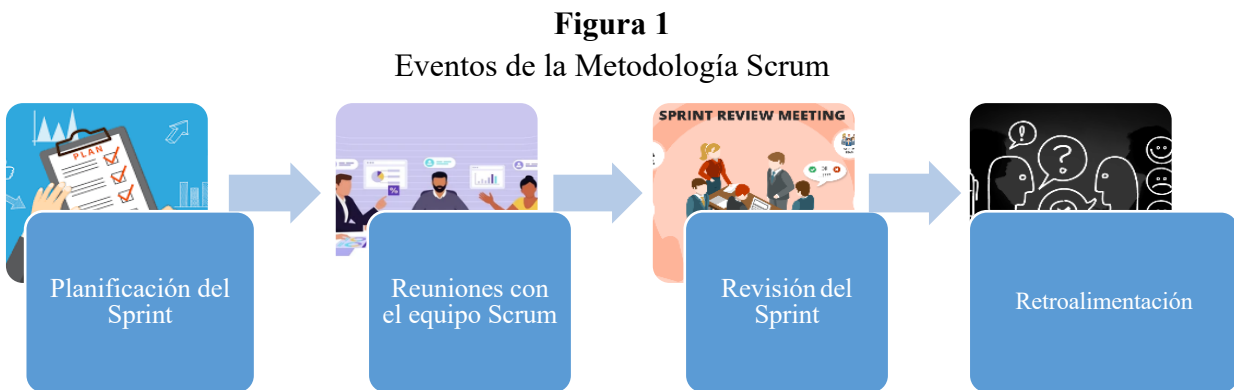
**Metodología:** Grupo de procesos, habilidades, procedimientos y documentación de ayuda, para que los desarrolladores logren la implementación de nuevos sistemas de información. Una metodología se divide en sub-fases, las cuales son utilizadas como guía para el desarrollador, y pueda seleccionar la tecnología más apropiada en cada momento de ejecución del proyecto y en su planificación, gestión, control y evaluación de este (Angelica, 2015).

**SCRUM:** Marco de trabajo ágil para conseguir una cooperación efectiva en el equipo en diferentes proyectos, además utiliza reglas, artefactos y permite definir roles que crean la estructura indispensable para obtener la funcionalidad correcta. Scrum adopta un enfoque incremental basado en la teoría de gestión de procesos. Dicha teoría se basa en la transparencia, verificación y adaptación; la transparencia asegura la claridad durante el proceso de cosas que pueden influir en el resultado; la verificación permite descubrir desviaciones no deseadas del proceso; y la adaptación, efectúa arreglos apropiados para minimizar su impacto (Navarro et al., 2013). Además, Molina et al. (2018) mencionan que Scrum puntualiza tres roles: Scrum Master, Product Owner y equipo de desarrollo.

**Scrum Master:** Persona que se encarga de dirigir y guiar al grupo para que se dé cumplimiento con las normas y procesos de la metodología.

**Product Owner:** Representa a los accionistas y usuarios que hacen uso del software, además comunica la visión del proyecto al equipo, valida los beneficios en historias para incluirlas al Product Backlog y las prioriza de forma frecuente.

**Equipo de desarrollo:** Está integrado por profesionales responsables de transformar la nómina de requerimientos o Product Backlog en funcionalidad del programa. Seguidamente, en la Figura 1 se describe los eventos de Scrum:



Fuente: Elaboración Propia

**Planificación del Sprint:** La planificación representa el inicio y la etapa más significativa del método Scrum para definir cómo llevar a cabo un trabajo, facilitando la designación de trabajos a cada miembro del equipo, además permite determinar el tiempo de entrega.

**Reunión con el equipo Scrum:** Consiste en mantener reuniones cortas diarias, donde se exponen los inconvenientes que aparecen en la ejecución de las tareas en curso, buscando activamente soluciones.

**Revisión del Sprint:** Fase que permite analizar los resultados obtenidos mediante una reunión, donde el cliente y los miembros del equipo son partícipes, por si algo necesita modificarse.

**Retroalimentación:** Aquí se mantiene una reunión final con los miembros del equipo y se analiza todos los aspectos del proyecto de inicio a fin, con el objetivo de adquirir el conocimiento necesario y así evitar errores en futuros proyectos (Merizalde, 2018).

**KANBAN:** Método visual para la gestión de proyectos que se ejecuta mediante la utilización de tablero Kanban. Estos tableros permiten a los equipos observar de forma gráfica su flujo y carga de trabajo, además se encuentran organizados en columnas que indican distintas fases que conlleva el proceso laboral, tales como: “pendiente”, “en progreso” y “completado”. Las tareas específicas se representan mediante tarjetas que progresan a lo largo de las columnas hasta su finalización. Este enfoque se fundamenta en la transparencia y representación visual del proceso de trabajo, permitiendo a los equipos detectar cuellos de botella y mejorar continuamente el proceso (Martins, 2024).

En la Figura 2, se detalla las fases de Kanban:

**Figura 2**  
Fases de la Metodología Kanban



Fuente: Elaboración Propia

Además, Rodríguez (2023) define a las fases de la siguiente manera:

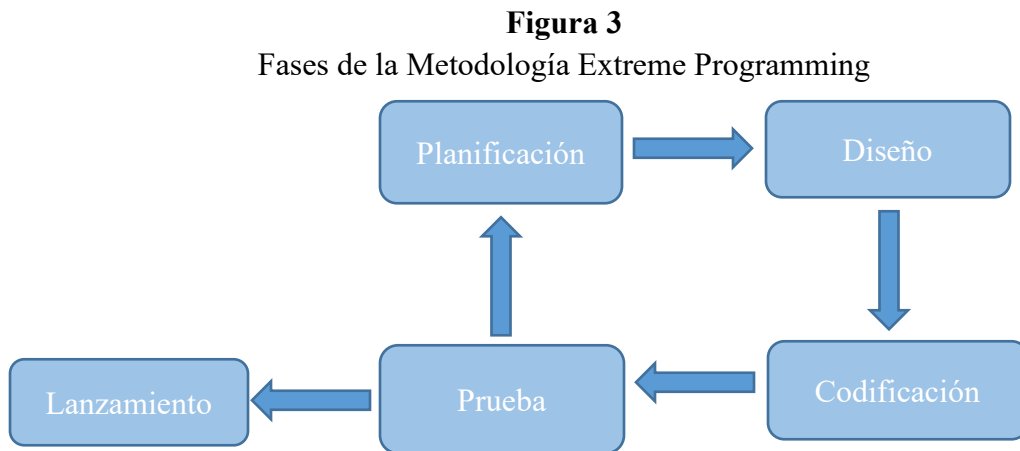
**Inicio del proyecto:** Se establecen los objetivos del proyecto, se identifican tareas importantes y se asignan los recursos requeridos.

**Diseño del tablero Kanban:** Se elabora el tablero, el cual ayuda a visualizar las tareas y su estado, determinando la estructura donde se incluyen las columnas y los indicadores visuales que representan cada tarea.

**Implementación del Kanban:** Se pone en marcha el tablero diseñado, el cual será utilizado en la organización. Además, se asignan tareas a los miembros del equipo y se establecen los límites de trabajo en curso.

**Seguimiento y mejora continua:** En esta última fase se realiza un seguimiento constante del flujo de trabajo, identificando posibles cuellos de botella o problemas en el proceso, que permiten mejorar la eficiencia y la productividad de manera continua.

**Extreme Programming (XP):** Pone énfasis en la relevancia de perfeccionar las interacciones entre personas como un elemento fundamental para lograr el éxito a la hora de desarrollar software. XP promueve la colaboración, enseñanza y un entorno laboral positivo. Fundamenta su enfoque en la retroalimentación entre desarrolladores y cliente, la comunicación flexible entre los involucrados, la implementación de soluciones de manera sencilla y la disposición para realizar cambios. Es particularmente apropiado para proyectos con requerimientos indefinidos, altamente variables y con un riesgo técnico considerable (Penadés & Letelier, 2006). A continuación, se describe cada etapa que nos brinda XP de acuerdo con Carrasco et al. (2019), en la Figura 3 se encuentran detalladas de manera específica:



Fuente: Elaboración Propia

**Planificación:** Se detallan historias de usuario y criterios de adaptación, permitiendo llevar un plan de iteración.

**Diseño:** Se diseñan las CRC (Tarjetas de clase, responsabilidad y colaboración) definiendo los prototipos a utilizar.

**Codificación:** En esta fase, la codificación es base para la continuidad del proyecto, por lo que la mayoría de las veces se codifica en pareja, logrando crear un código universal, que otros programadores consigan trabajar e inferir la dirección que está siendo seguido en el trabajo actual.

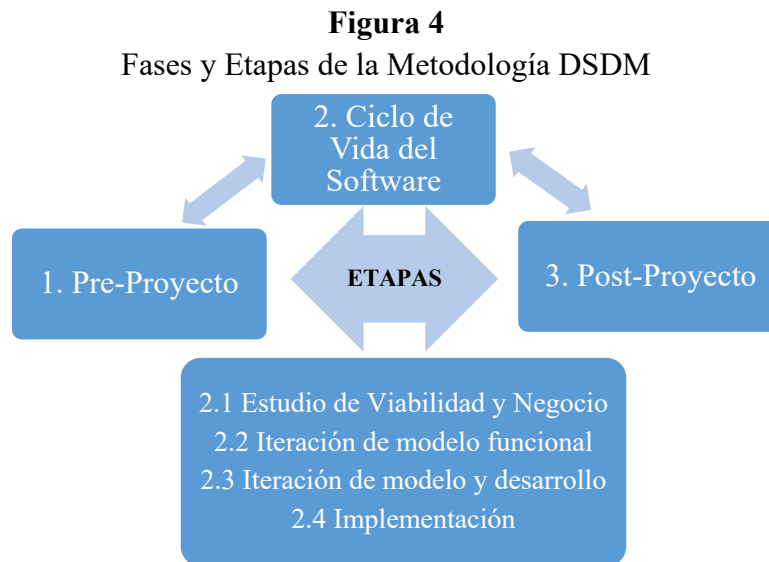
**Prueba:** Se deben realizar pruebas automáticas consecutivamente, esto nos permitirá ir validando el proyecto.

**Lanzamiento:** Si llegamos a esta fase, significa que logramos satisfacer las solicitudes del cliente, consiguiendo cumplir todas las historias de usuario tal cual el cliente lo requirió.

**Dynamic System Development Methods (DSDM):** Método ágil para el desarrollo de software a corto plazo, orientado a realizar entregas rápidas y flexibles con un marco iterativo e incremental que permite adaptarse a los cambios según la necesidad del proyecto. Además, promueve una estrecha colaboración entre equipos de desarrollo y usuarios finales con el propósito de brindar soluciones que atiendan las necesidades de los clientes de manera puntual (Cortés, 2017).



En la Figura 4, se enumeran las fases y etapas de DSDM como se menciona en (Wikiversidad, 2018).



Fuente: Elaboración Propia

**Pre-Proyecto:** Se originan varias propuestas de proyectos candidatos, de los cuales se elige uno. También se efectúa una valoración de fondos necesarios para el desarrollo del proyecto, tomando la decisión de lanzar o no el proyecto.

**Ciclo de vida del Software:** Tiene como propósito construir el proyecto durante la fase de pre-proyecto. El cual se divide en las siguientes etapas:

#### **Estudio de Viabilidad y Negocio**

**Estudio de Viabilidad:** Etapa en la que se llevan a cabo juntas para identificar las ideas que no están relacionadas directamente al proyecto, pero sí con la administración de este, además, sus soluciones facilitarán la orientación deseada para el desarrollo de proyectos estableciendo costos, y adaptabilidad para este modelo DSDM, amenazas existentes, originando así un documento viable.

**Estudio de Negocio:** Aquí se desarrolla un análisis flexible, el cual se ejecuta exclusivamente solo si se decidió desarrollar usando la metodología DSDM. También se definen las características del proyecto, y necesidades de los usuarios. Al finalizar se establecerá una lista de requisitos, que será priorizada en función de su necesidad para el desarrollo.

**2.2 Iteración de Modelo Funcional:** Durante esta fase, los requisitos detallados en fases previas son recopilados y transformados en modelos funcionales. Por lo que se desarrollan prototipos funcionales de estos requisitos, brindando al usuario una visión de cómo funcionará la aplicación. Estos prototipos son probados por varios conjuntos de usuarios para evaluar y verificar el nivel de excelencia del modelo.

**2.3 Iteración de Diseño y Desarrollo:** Su tarea principal consiste en tomar patrones funcionales derivados de la etapa antepuesta, y perfeccionarlos para satisfacer plenamente las exigencias del consumidor, mediante la creación de prototipos de diseño.

**2.4 Implementación:** En esta etapa los clientes comprueban que los modelos cumplan con sus expectativas para proceder con su ejecución; además se realiza la preparación de usuarios futuros para asegurar su competencia en el uso del sistema.

**Post-Proyecto:** Permite probar la funcionalidad y eficiencia del producto, además se realizan labores de mantenimiento y mejoras en el software, si es necesario, se promueve en 6 meses tras la entrega al cliente.

**Metodología Lean:** Se inició en la industria manufacturera, y ha sido exitosamente adecuada para su aplicación en el desarrollo de software. Además, se basa en los principios del sistema de producción de Toyota, con el objetivo de optimizar la entrega de valor al cliente, al mismo tiempo que se reduce al mínimo el desperdicio de recursos (Yépez et al., 2017).

### Principios Fundamentales

**Identificar el Valor:** Se centra en entender y precisar lo que el cliente considera naturalmente valioso. Esto implica descifrar sus necesidades, aspiraciones y perspectivas en relación con el producto que se está creando.

**Mapeo del Flujo de Valor:** Examinar y representar el proceso del flujo de trabajo esencial para proporcionar el valor al cliente, desde la idea inicial a la entrega final del proyecto. Este análisis facilitará identificar áreas de desperdicio y oportunidades para optimizar el procedimiento.

**Crear Flujo Continuo:** Busca eliminar obstáculos y limitaciones en el proceso de trabajo para que las tareas se desarrollen de forma continua y eficiente mediante el sistema.

**Establecer Pull (Tirar):** Crear sistemas que admitan a los equipos de trabajo gestionar las tareas de acuerdo con la demanda, en lugar de acumular excesivamente el trabajo. Esto contribuye a reducir la sobreproducción garantizando satisfacer las necesidades reales del cliente.

**Buscar la Perfección:** Iniciar una cultura de avance continuo, con el objetivo permanente de eliminar el desperdicio, optimizar procesos y así elevar la calidad del producto.

En el ámbito del desarrollo de software, la metodología Lean se dedica a eliminar actividades que no añaden valor al cliente, tales como repetición de esfuerzos, espera, defectos y procesamiento excesivo. Para lograrlo se implementan prácticas continuas y se optimiza el trabajo no iniciado. El objetivo final es producir software de calidad de modo eficiente, reduciendo los plazos de entrega y mejorando la satisfacción del cliente (Yépez et al., 2017).

### Trabajos Relacionados

En 2019, se realizó una investigación para comparar metodologías ágiles mediante análisis de documentos existentes, para ofrecer una orientación durante la elección de la metodología para el desarrollo más conveniente para escenarios particulares. Los resultados obtenidos mostraron que las metodologías ágiles se distinguen por tener un enfoque en la entrega rápida de software funcional, la capacidad de adaptarse a los requisitos cambiantes y la colaboración con los clientes; estas características han generado un interés considerable en las empresas que buscan cumplir sus metas de desarrollo de software rápido y funcional. Adicionalmente los autores concluyeron que

es importante seleccionar adecuadamente la metodología ágil a utilizarse para afirmar el éxito del proyecto y, también para maximizar las probabilidades de entregar un software de buena calidad que cumpla con los requisitos del usuario final (Casierra et al., 2019).

Más tarde Bautista (2020), desarrolló un artículo con el objetivo principal de establecer una comparativa entre metodologías ágiles más utilizadas dentro del desarrollo web: Extreme Programming y Scrum. Para llevar a cabo esta comparación se realizó una investigación exhaustiva de documentación disponible en fuentes más aceptadas como ScienceDirect, IEEE Xplore Digital Library y Scielo. En los resultados obtenidos se estableció que las metodologías ágiles son adecuadas para manejar los requisitos especiales y en constante cambio exigido por las aplicaciones web en un entorno de avances tecnológicos. En este artículo se concluyó que Scrum es la metodología más adecuada y flexible a la adaptación al cambio en comparación con XP.

Por otra parte, en 2021 la Universidad de Guayaquil efectuó una investigación para establecer qué metodología ágil es la más adecuada para desarrollar software, por la quiebra durante el uso de los modelos tradicionales, ya que establecía un déficit en varios momentos al facilitar información al cliente. Razón por la que se planteó realizar el estudio de cuatro modelos ágiles, Extreme Programming (XP), Scrum, Dynamic System Development Methods (DSDM) y Kanban, exploración que se realizó por medio de compilación de información, logrando así adquirir sus características y los aspectos más relevante acerca de cada una de las metodologías. Al analizar estas cuatro metodologías que fueron el ente de estudio, se obtuvo que Scrum es el modelo ágil más utilizado, con mayor afluencia en el ámbito profesional a diferencia de los modelos de Kanban, XP y DSDM.

Esto debido a que Scrum destaca características fundamentales como: los roles, trabajos de iteración en tiempos establecidos, maneja el equipo Scrum, tareas basadas en Sprints, a diferencia de Kanban ya que su desventaja es que no utiliza roles, se debe efectuar los trabajos continuamente, no tiene ciclos, permite realizar modificaciones, este modelo se centra en la programación de productos, donde el cliente es la parte integral del equipo de trabajo. El estudio de estas metodologías permitió identificar características relevantes de cada una, también facilitó entender cómo se adaptan en el trabajo según la necesidad del cliente o condiciones del proyecto a desarrollarse sin importar las circunstancias de su entorno (Flores et al., 2021).

Después, Yánez (2022) realizó un estudio comparativo de herramientas empleadas en los métodos ágiles para la implementación de prácticas de desarrollo encaminadas en el margen de excelencia de desarrollo, donde el objetivo principal fue analizar la trascendencia de la calidad del software en las industrias y cómo las metodologías ágiles influyen en este aspecto. Además, se buscó comprender el aseguramiento de calidad de software, especialmente en el marco de enfoque de metodologías ágiles como Scrum, Extreme Programming y Kanban siendo las más populares. Los resultados obtenidos en este estudio fueron: Scrum se aplica preferentemente en proyectos que

requieren resultados a corto plazo y además fomenta una colaboración efectiva en un equipo de trabajo sólido; Kanban por otro lado, se basa en un sistema de tablero que designa tareas a los miembros del equipo de manera individual y Extreme Programming ofrece flexibilidad al momento de desarrollar software. Los autores concluyeron que es importante usar metodologías ágiles para desarrollar software, ya que proporcionan una forma de trabajo rápida y confiable que conduce a la excelente calidad del software.

## Material y Métodos

El desarrollo del artículo se fundamenta principalmente en la investigación bibliográfica, analítica y comparativa con respecto a las metodologías ágiles (Snyder, 2019). Esta metodología brinda una base teórica sólida, además permite obtener un análisis detallado que ayuda a identificar las diferencias clave entre las metodologías. Con el propósito de llevar a cabo el siguiente estudio se establecieron cuatro pasos centrados en comparar las

**Exploración bibliográfica y recopilación de datos:** Realizar una investigación de la literatura existente sobre metodologías ágiles de desarrollo de software y recopilar información sobre las características, ventajas y desventajas de cada una de las metodologías ágiles seleccionadas.

**Justificación de las metodologías:** Analizar a detalle cada una de las metodologías seleccionadas, considerando sus características específicas y su idoneidad para abordar diferentes tipos de proyectos.

**Planteamiento de un marco comparativo:** Diseñar un marco con criterios para evaluar las metodologías ágiles.

**Construcción de una propuesta:** Elaborar una propuesta centrada en los hallazgos y el análisis comparativo. Además, se propone una nueva metodología tomando como base Lean y los principios ágiles.

## Resultados y Discusión

### Exploración bibliográfica y recopilación de datos

Se realizó una revisión de la literatura existente sobre metodologías ágiles de desarrollo de software. Además, se recopiló información detallada de las metodologías seleccionadas de la Tabla 1, de las que podemos observar sus características en la Tabla 2 y sus ventajas y desventajas en la Tabla 3.

**Tabla 1**

Identificación de metodologías y sus acrónimos

Nombre de Metodología	Acrónimo	Identificador
Extreme Programming	XP	M1
Scrum	-	M2
KANBAN	-	M3
Dynamic System Development Methods	DSDM	M4

Fuente: Elaboración propia

En la Tabla 1 se detallan los nombres de las metodologías seleccionadas con sus respectivos acrónimos, además de su identificador, con el objetivo de facilitar la comparación entre estas cuatro metodologías.

**Tabla 2**

Características de las metodologías ágiles

Metodología	Características
M1	<ul style="list-style-type: none"> <li>• Desarrollo de productos de calidad mediante un enfoque iterativo e incremental.</li> <li>• Enfoque en la calidad del software.</li> <li>• Pruebas unitarias.</li> <li>• Integración continua.</li> <li>• Programación en parejas.</li> <li>• Propiedad colectiva del código.</li> <li>• Diseño simple.</li> </ul>
M2	<ul style="list-style-type: none"> <li>• Desarrollo iterativo e incremental de productos de calidad.</li> <li>• Establece roles definidos.</li> <li>• Colaboración en equipo.</li> <li>• Entrega de progresos sustanciales dentro del plazo establecido.</li> <li>• Fomento de la interacción.</li> <li>• Aseguramiento de una participación.</li> <li>• Responsabilidad.</li> </ul>
M3	<ul style="list-style-type: none"> <li>• Enfoque en la mejora continua.</li> <li>• Visualización y gestión del flujo de trabajo.</li> <li>• Limitación del trabajo en progreso.</li> <li>• Mejora del proceso de entrega</li> </ul>
M4	<ul style="list-style-type: none"> <li>• Desarrollo iterativo e incremental de productos de calidad.</li> <li>• Enfoque en la entrega temprana de productos.</li> <li>• Colaboración del usuario.</li> </ul>

- Enfoque en la viabilidad del negocio.
- Gestión del riesgo.
- Enfoque en la calidad del software.

Fuente: Elaboración propia

En la Tabla 2, se realiza un análisis basado en las características principales de las metodologías, y se observa que a pesar de ser diferentes métodos comparten una característica en común. Se enfocan en el desarrollo iterativo e incremental de productos de alta calidad, pero presentan diferencias en cuanto a características específicas. XP se destaca por su atención a la calidad del software, y por hacer uso de prácticas de desarrollo que contribuyen para obtener un software más eficiente, escalable y confiable. Por otro lado, Scrum se centra en definir roles, trabajar en equipo para lograr avances significativos en un tiempo determinado, garantizando el compromiso y participación. Kanban se centra en la mejora continua, la visualización y gestión de flujos de trabajo, la reducción del trabajo en curso, y la mejora de los procesos de entrega. DSDM se enfoca en la entrega rápida de productos, la colaboración de los usuarios, la rentabilidad empresarial, la gestión de riesgos y la calidad del software.

**Tabla 3**

Ventajas y Desventajas de las metodologías ágiles

ID	Ventajas	Desventajas
M1	<p><b>-Adaptabilidad:</b> Es fácilmente adaptable a los requisitos que desee el cliente, permitiendo obtener respuesta inmediata a nuevas necesidades.</p> <p><b>-Calidad del código:</b> XP se caracteriza por su énfasis en la calidad del código.</p> <p><b>-Mejora continua:</b> Incita a la programación estructurada, y se centra en la eficacia de sus métodos.</p> <p><b>-Entregas frecuentes:</b> Permite entregas continuas del proyecto, facilitando obtener comentarios rápidos y así lograr adaptarse al cambio.</p>	<p><b>-XP no es apta para todos los proyectos:</b> No es la mejor opción para proyectos grandes y con requisitos altamente estables.</p> <p><b>-Dependencia de comunicación:</b> Falta de comunicación efectiva puede afectar de forma negativa la implementación exitosa de XP.</p> <p><b>-Imposible predecir errores antes de iniciar su desarrollo.</b></p>
M2	<p><b>-Flexibilidad:</b> Facilita adaptarse a los cambios en los requisitos del proyecto durante su desarrollo.</p> <p><b>-Entrega Reiterativa:</b> Permite obtener retroalimentación temprana del cliente.</p> <p><b>-Satisfacción del cliente:</b> Integra al cliente de forma directa en el desarrollo del proyecto, atendiendo a sus necesidades.</p> <p><b>-Transparencia:</b> Ofrece una visibilidad clara del avance del proyecto mediante artefactos, como es el tablero Scrum y reuniones frecuentes.</p>	<p><b>-Complejidad:</b> Esta metodología puede ser compleja al inicio, si el equipo de trabajo no está familiarizado.</p> <p><b>- Dependencia de Scrum Master:</b> Los procesos son dependientes del gestor de proyectos.</p> <p><b>-Resistencia al Cambio:</b> Puede generar incomodidad en los integrantes debido a que se nieguen a adaptar esta metodología.</p>



- Optimiza procesos al dividir tareas complejas en pequeñas partes.

**-Colaboración:** Fomenta la comunicación y colaboración entre los integrantes del grupo de trabajo.

**-Flexibilidad:** Flexible y adaptable a entornos con requisitos cambiantes.

**-Responsabilidad:** Responsable con los tiempos de entrega priorizando entregas constantes del producto al cliente.

**M3** **-Mejora continua:** Permiten mejorar de forma continua al facilitar los ajustes de tiempo.

**-Colaboración mejorada:** Fomenta la colaboración entre los miembros del equipo, enfocándose en la comunicación y solución de problemas.

**-Calidad:** Tiene procesos que se enfocan directamente en la calidad del producto.

**-Dependencia:** Los procesos dependen del gestor del proyecto.

**-Falta de estructura predeterminada:** La ausencia de reglas estrictas puede llevar a resultados negativos como equipos de trabajo indisciplinados.

**-No tiene guía para quienes están iniciándose:** Puede ser menos intuitivo para los equipos que están iniciando en el mundo Kanban debido a su falta de guías claras.

**-Entrega Rápida:** Centrada en la entrega rápida, lo que permite a los clientes obtener beneficios tangibles de forma temprana durante el proceso de desarrollo.

**-Flexible a los cambios:** Es adaptable a los cambios en los requisitos, los que facilita ajustarse a nuevas prioridades sin problema.

**M4** **-Mejora Continua:** Fomenta la retroalimentación continua y la mejora constante, esto contribuye a la evolución positiva del proceso de desarrollo del proyecto.

**-Colaboración activa del cliente:** Se involucra directamente con los clientes y usuarios durante el desarrollo del proyecto, asegurando que el producto final cumpla con las expectativas y satisfaga las necesidades.

**-Complejo para grandes proyectos:** La aplicabilidad en proyectos de gran escala puede volverse complejo, esto podría afectar la eficiencia de la metodología.

**-Requiere personal experimentado:** Requiere de equipos y líderes experimentados en la metodología, debido a que puede ser un desafío para entornos nuevos o con personas iniciales.

**-Dependencia de la participación del cliente:** La participación del cliente en el proyecto puede convertirse en un desafío si los usuarios finales no se encuentran comprometidos lo suficiente.

Fuente: Elaboración propia

En la Tabla 3, se describen las ventajas y desventajas de cada metodología ágil como: XP, Scrum, Kanban y DSDM, cada una tiene un enfoque diferente para contribuir al desarrollo de software. XP tiene ventajas como: adaptabilidad, calidad del código, mejora continua y entrega frecuente. Sin embargo, tiene desventajas como no ser adecuado para todos los proyectos, depender de la comunicación y dificultar la predicción de errores antes de que comience la programación. Por



otro lado, la metodología Scrum ofrece agilidad, entrega iterativa, satisfacción del cliente, transparencia, optimización y colaboración, pero puede resultar compleja desde el principio, tiene dependencia del Scrum Master y puede provocar resistencia al cambio. Kanban se destaca por su flexibilidad, responsabilidad, mejora continua, mejor colaboración y énfasis en la calidad, pero tiene desventajas como la dependencia del director del proyecto y la falta de una organización de estimación definida. DSDM se centra en la entrega rápida, la flexibilidad para cambiar, la mejora continua y la colaboración activa con los clientes, pero puede ser complejo para proyectos grandes y exigentes, requiere personal experimentado y depende de la contribución del cliente.

### Justificación de las metodologías

Cada metodología se justificó considerando sus fortalezas y capacidades específicas, y así proveer una base sólida para la comparación y evaluación posterior. En la Tabla 4 se destaca la idoneidad de cada metodología para ciertos escenarios. Esta justificación ayuda a identificar la metodología adecuada para cada tipo de proyecto, brindando una guía para su selección y aplicación.

**Tabla 4**  
Justificación para utilizar las siguientes metodologías ágiles

M1	M2	M3	M4
Metodología ideal para proyectos cambiantes porque garantiza una respuesta rápida a las necesidades de los clientes. Su enfoque se basa en la calidad del código, además fomenta la mejora continua, lo que le hace imprescindible en proyectos dinámicos que requieren entregas frecuentes y rápidas.	Destacada por su flexibilidad y capacidad de adaptación a los requisitos cambiantes del proyecto. Se centra en la entrega iterativa, satisfacción del cliente y colaboración en equipo, lo que le convierte en una gran opción para proyectos que requieren transparencia, optimización y participación del cliente.	Proporciona flexibilidad y enfatiza la responsabilidad a la hora de entregar su proyecto. Permite priorizar entregas continuas. Además, favorece su continuidad y logra la colaboración en equipo, enfocado de manera primordial en la comunicación y resolución de problemas.	Sobresale por su rapidez de entrega, flexibilidad en relación con los cambios y cooperación proactiva con los clientes. Es adecuado para proyectos que buscan beneficios reales en una etapa temprana y deben ajustarse constantemente en función de una retroalimentación perseverante.

Fuente: Elaboración propia

### Planteamiento de un marco comparativo

En esta sección se establece un cuadro comparativo como se aprecia en la Tabla 5, utilizando diversos criterios como: escalabilidad, funcionalidad y rapidez, con el propósito de detallar el alcance que abarca cada metodología, en base a los criterios mencionados anteriormente.



**Tabla 5**

Cuadro comparativo de las metodologías

Metodología	Escalabilidad	Funcionalidad	Rapidez
<b>Ágil</b>			
M1	Gran aceptación de adaptabilidad a las necesidades cambiantes.	Su prioridad se basa en la calidad del código y pruebas unitarias.	Se enfoca en entregas diarias y en retroalimentaciones rápidas.
M2	Adaptable para proyectos de todos los tamaños.	Asigna roles y procesos claros para lograr una funcionalidad efectiva.	Orientado a iteraciones específicas y entregas repetidas de productos.
M3	Flexibilidad y adaptabilidad ambientes y necesidades diferentes.	Permite visualizar el flujo y la limitación del avance de trabajo.	Mejora continua y gestión eficaz del trabajo.
M4	Se enfoca en la entrega rápida e intervención del cliente.	Evaluación de la posibilidad de negocio y gestión de riesgos.	Itera ciclos de desarrollo para adaptarse a los cambios que el cliente requiere.

Fuente: Elaboración propia

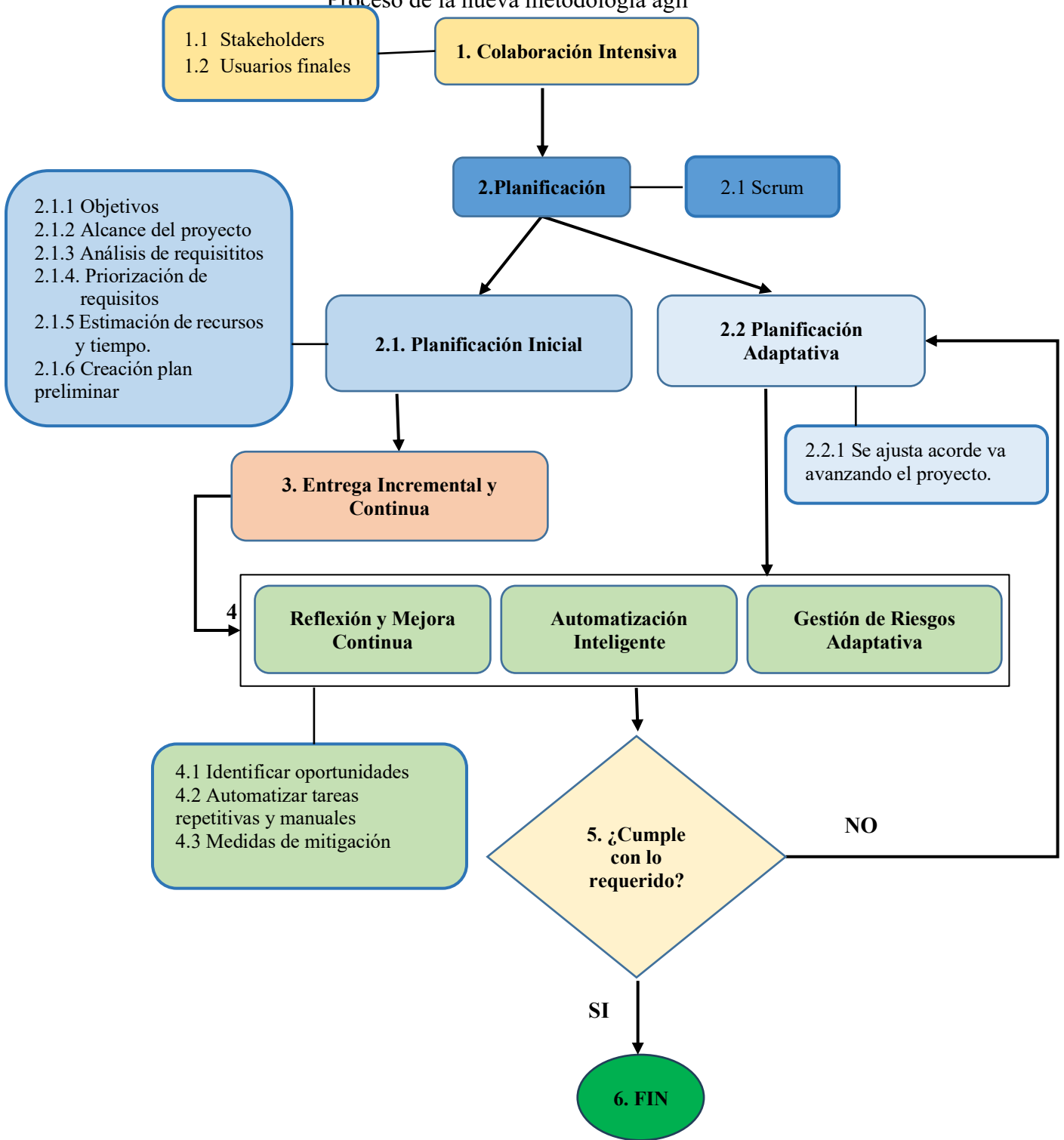
En la Tabla 5, se establecen comparaciones entre las metodologías, en base a la escalabilidad, funcionalidad y rapidez. XP se caracteriza por su capacidad de adaptación y su énfasis en la excelencia del código, mientras que Scrum pone énfasis en roles bien definidos y entregas de trabajo en iteraciones. Por otro lado, Kanban proporciona flexibilidad y una representación visual del flujo de trabajo, mientras que DSDM se enfoca en la entrega rápida y la intervención del usuario.

### Construcción de una propuesta

Esta nueva metodología propuesta, fusiona los principios de Lean con las prácticas ágiles, creando un enfoque que elimina el desperdicio, maximiza la eficiencia, y acelera la entrega de valor. En la Figura 5 se puede apreciar el proceso a seguir en esta metodología:

**Figura 5**

Proceso de la nueva metodología ágil



Fuente: Elaboración propia



El proceso de esta metodología inicia con la colaboración intensiva de todos los integrantes del equipo, los cuales incluyen a los stakeholders y usuarios finales, en conjunto se realiza la planificación tomando como base la metodología Scrum, misma que ayudará a verificar que se cumplan las demás actividades de forma eficiente. Esta planificación se divide en dos: inicial y adaptativa, en la inicial se establecen los objetivos, el alcance del proyecto, se efectúa un análisis y la priorización de requisitos, además se estima recursos, tiempo creando un plan, por otra parte, la adaptativa se ajusta acorde va avanzado el proyecto. Después de la planificación inicial, el equipo comienza a trabajar, en la entrega incremental y continua del producto, a medida que el equipo avanza en la entrega, dedica tiempo a la reflexión y mejora continua, también durante el proceso de desarrollo el equipo puede identificar oportunidades para automatizar tareas repetitivas y manuales, a su vez está atento a los riesgos potenciales para tomar medidas de mitigación de modo proactivo. Si el proceso se cumple eficazmente, se procede a la entrega del producto final, caso contrario, si surgen cambios en los requisitos del cliente o en las prioridades del proyecto, desviaciones significativas en los plazos o recursos asignados, entre otros, retorna a la planificación adaptativa, misma que facilitará realizar los ajustes necesarios, de manera que una vez realizado todos los cambios requeridos, continúa el proceso nuevamente para asegurar la entrega del proyecto con éxito, y cumpla con las expectativas de los usuarios finales.

Seguidamente, se detalla cada actividad dentro del proceso:

1. **Colaboración Intensiva:** Enfatiza la contribución de los miembros del equipo, los stakeholders y los usuarios finales, llevando a cabo encuentros periódicos de coordinación, retroalimentación y planificación, promoviendo la comunicación abierta y transparente.
2. **Planificación:** Este marco de planificación incluye elementos de la metodología ágil Scrum, que ayudará a conseguir que las actividades se efectúen eficazmente, ya que brinda una estructura sólida para la planificación, seguimiento y gestión del trabajo en equipo.
  - 2.1 **Planificación Inicial:** En este punto se realizan varias actividades, estas son:
    - 2.1.1 **Definición de objetivos:** Implica definir claramente los objetivos que debe alcanzar el proyecto. Esto incluye especificar los resultados esperados, los beneficios deseados y las metas cuantitativas que se debe lograr para considerar el proyecto como exitoso.
    - 2.1.2 **Alcance del proyecto:** Identificar las actividades específicas requeridas para lograr estos objetivos, y así definir lo que se incluye o no del proyecto, las limitaciones del trabajo a realizar, así como los entregables esperados y los resultados reales.
    - 2.1.3 **Análisis de requisitos:** Consiste en examinar cuidadosamente todos los requisitos del proyecto para comprender su naturaleza y alcance. Lograr identificar y entender las necesidades y expectativas del usuario, tales como requerimientos funcionales y no funcionales del producto.

**2.1.4 Priorización de Requisitos:** Después del análisis, se categoriza y prioriza los requisitos según su relevancia y urgencia. Esto permite determinar qué requisitos son críticos para el éxito del proyecto y cuáles pueden abordarse en etapas posteriores. La priorización ayuda a establecer una dirección clara para el desarrollo del producto, asegurando que se satisfagan primero las necesidades más importantes del cliente.

**2.1.5 Estimación de recursos y tiempo:** Evaluar los recursos necesarios, como, por ejemplo: personal, tiempo y presupuesto, para llevar a cabo el proyecto de manera efectiva. Esto admite establecer expectativas realistas y planificar el trabajo de modo adecuada.

**2.1.6 Creación de un plan:** Elaborar un plan inicial que detalle las actividades importantes, los criterios fundamentales y el cronograma general del proyecto.

**2.2 Planificación Adaptativa:** Se ajusta acorde va avanzando el proyecto y se adquiere más información, es decir, se realizan reuniones de planificación donde el equipo revisa y ajusta sus objetivos y prioridades según sea necesario.

3. **Entrega Incremental y Continua:** Destaca la entrega paulatina y constante en el desarrollo del proyecto, implica dividir el trabajo en partes manejables y entregarlas frecuentemente, facilitando retroalimentación temprana y haciendo ajustes necesarios.

4. **Reflexión y Mejora Continua:** Después de cada iteración o entrega, el equipo reflexiona sobre los aspectos positivos, los desafíos y las oportunidades de mejora. Esta práctica ayuda al equipo a identificar áreas para optimizar e implementar acciones correctivas.

**Automatización Inteligente:** Automatizar tareas repetitivas y manuales mediante herramientas y tecnologías permitiendo al equipo trabajar de forma más eficiente. Esto abarca la automatización de pruebas, integración continua, despliegue continuo, entre otros.

**Gestión de Riesgos Adaptativa:** Identificar y abordar los riesgos de manera proactiva durante todo el proyecto. Esto implica evaluar regularmente los riesgos, implementar medidas de mitigación y revisar continuamente el plan para garantizar un progreso seguro y efectivo del proyecto.

## Características Principales

**Eliminación de Desperdicios:** Esta metodología busca identificar y eliminar cualquier actividad que no aporte valor al cliente o al proyecto. Esto incluye tareas que no son necesarias, tiempos de espera o procesos innecesarios complicados, optimizando así tiempo y recursos del equipo.

**Valor al Cliente:** Prioriza la entrega de valor al cliente en cada etapa del desarrollo. Enfocándose en las funcionalidades y características del producto que le interesa al cliente, en lugar de perder tiempo y recursos en aspectos poco relevantes. Al entregar valor de manera

rápida y constante, se obtiene una retroalimentación temprana y se pueden realizar ajustes según las necesidades del cliente.

**Optimización del Flujo de Trabajo:** Se basa en la visualización de flujos de trabajo, prohibiciones constantes y avance continuo del proceso. Visualizar el proceso de trabajo que permite al equipo determinar limitaciones y áreas de mejora. Además, limita el trabajo en curso, ayuda a mantener un flujo constante, evita la sobrecarga del equipo, asegurando que el proceso se ajuste y optimice continuamente para maximizar la eficiencia.

## Beneficios

**Maximización de la Eficiencia:** Eliminar desperdicios y optimizar el flujo de trabajo, asegura que el equipo utilice sus recursos de manera más eficiente, obteniendo un proceso de desarrollo más ágil y rentable.

**Entrega Continua de Valor al Cliente:** Priorizar el valor para el cliente en todas las etapas del proceso, garantiza que el producto final satisfaga las necesidades y deseos de sus clientes, logrando así una mayor eficiencia y relación más sólida a largo plazo.

**Mejora de la Calidad del Producto:** Al centrarse en funciones y características que agregan valor para el cliente, asegura que el resultado final sea de alta calidad y cumpla con los más altos estándares.

**Reducción de Tiempos de Entrega:** Optimizar los flujos de trabajo y eliminar el desperdicio, los equipos logran desarrollar y entregar software de manera más rápida y eficiente, reduciendo el tiempo de entrega y acelerando el tiempo de comercialización del producto.

## Validación de la propuesta

La metodología pasó por un proceso de validación realizada por expertos en la industria de desarrollo de software. Este proceso incluyó una revisión exhaustiva y evaluación detallada de la metodología mediante varias sesiones de trabajo. Los comentarios y sugerencias de estos expertos se tomaron en consideración y se utilizaron para mejorar y perfeccionar esta metodología, asegurando así su eficacia y aplicabilidad en diferentes contextos de desarrollo de software.

## Conclusiones

En el análisis comparativo realizado, se lograron identificar las diferencias más relevantes entre las metodologías seleccionadas, las cuales, pese a basarse en principios ágiles, adoptan diversos enfoques para abordar los desafíos que plantean otras metodologías. En este hallazgo no se menciona que dichas metodologías sean erróneas, sino que demuestran su utilidad significativa en contextos específicos para los que se diseñaron.

Las metodologías ágiles como XP, Scrum, Kanban y DSDM, han sido reconocidas por su capacidad para adaptarse a entornos cambiantes y facilitar la entrega continua de productos de calidad. Además, se destacan por tener un enfoque en la adaptabilidad, calidad del producto, cumplimiento de las expectativas del cliente, colaboración y la mejora continua, convirtiéndose en opciones altamente significativas en el desarrollo de software.

Finalmente, tras los hallazgos y comparaciones realizadas se puede destacar que ninguna metodología ágil analizada ha abordado completamente la eliminación de desperdicios, maximización de la eficiencia y la entrega constante de valor al cliente, por lo que se propuso una nueva metodología ágil que fusiona los principios ágiles con los conceptos de Lean, logrando un enfoque completo que garantice la eliminación de actividades innecesarias, optimizando procesos y así una entrega continua de productos de alta calidad acorde a las expectativas del cliente.

### Referencias bibliográficas

- Angelica. (2015). *Metodología de desarrollo*. <http://angelicahb.blogspot.com/>
- Bautista, E. (2020). Metodologías ágiles Scrum y XP empleadas para el desarrollo de páginas web, bajo el modelo MVC, con el lenguaje de desarrollo PHP, y el framework Laravel. <https://repositorio.upeu.edu.pe/handle/20.500.12840/3251>
- Carrasco, M., Ocampo, W., Ulloa, L., & Azcona, J. (2019). Metodología híbrida de desarrollo de software combinando XP y ScRUM. Dialnet. <https://dialnet.unirioja.es/servlet/articulo?codigo=8605609>
- Casierra, J., Quiñonez, X., Herrera, L., & Mera, J. (2019). Análisis comparativo de metodologías ágiles de desarrollo de software: una revisión bibliográfica. ResearchGate. <https://n9.cl/5zxh5>
- Cortés, J. (2017). Comparativo, metodología ágil Crystal y arquitectura dirigida por modelos MDA. <https://repositorio.utp.edu.co/items/d7dcd3f3-4aa5-431c-9433-faa36bc162de>
- Durán, A. (2023, 23 febrero). Manifiesto «agile», ¿qué es? BBVA NOTICIAS. <https://n9.cl/zmm7v>
- Flores, F., Sanhueza, V., Valdés, H., & Reyes, L. (2021). Metodologías ágiles: un análisis de los desafíos organizacionales para su implementación. Redalyc, 43(1). <https://www.redalyc.org/journal/5043/504371975003/>
- Langley, M. (2018). Success in Disruptive Times - Expanding the Value Delivery Landscape to Address the High Cost of Low Performance - PMI. <https://n9.cl/scz6l>
- Lazzeretti, A. (2019, 21 junio). El 85% de las compañías en LATAM se suman a la nueva tendencia de Metodologías Ágiles. Talento y Empresa. <https://goo.su/m3HxhMV>
- Loaiza, J. (2019). Implementación de la metodología Scrum, en el desarrollo del software de la Dirección Nacional de Comunicaciones en la sección de desarrollo de software de la Policía Nacional del Ecuador. <https://repositorio.uisrael.edu.ec/xmlui/handle/47000/2164>



- López, A. (2018). Estudio comparativo de metodologías tradicionales y ágiles para proyectos de desarrollo de software. Universidad de Valladolid. <https://uvadoc.uva.es/handle/10324/32875>
- Martins, J. (2024, 19 enero). ¿Qué es la metodología Kanban y cómo funciona? [2024] • Asana. Asana. <https://asana.com/es/resources/what-is-kanban>
- Merizalde, M. (2018). Aplicación de la metodología SCRUM en la gestión y desarrollo de proyectos. Caso de estudio: empresas consultoras de software de Guayaquil [Tesis de Posgrado]. En Dspace. <http://biblioteca.uteg.edu.ec:8080/handle/123456789/239>
- Molina, B., Vite, H., & Dávila, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. Espirales Revista Multidisciplinaria de Investigación, 2(17). <https://doi.org/10.31876/re.v2i17.269>
- Navarro, A., Fernández, J., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. Redalyc.org. <https://www.redalyc.org/articulo.oa?id=496250736004>
- Penadés, C., & Letelier, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Dialnet, 5(26). <https://dialnet.unirioja.es/servlet/articulo?codigo=1983605>
- Rodríguez, E. (2023, 20 septiembre). ¿Qué es y para qué sirve el Kanban? Definición, Fases y Pasos. Canal Innova. <https://n9.cl/uku8h>
- Senapathi, M., & Srinivasan, A. (2012). Understanding post-adoptive agile usage: An exploratory cross-case analysis. Journal Of Systems And Software, 85(6), 1255-1268. <https://doi.org/10.1016/j.jss.2012.02.025>
- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. Journal Of Business Research, 104, 333-339. <https://doi.org/10.1016/j.jbusres.2019.07.039>
- Wikiversidad. (2018). Dynamic Systems Development Method. [https://es.wikiversity.org/wiki/Dynamic\\_Systems\\_Development\\_Method](https://es.wikiversity.org/wiki/Dynamic_Systems_Development_Method)
- Yáñez, C. (2022). Estudio comparativo de las herramientas de metodologías ágiles para el aplicar buenas prácticas de desarrollo en la calidad de software. [Universidad Técnica de Babahoyo]. <https://n9.cl/1f4mp>
- Yépez, M. P. S., Villamarín, G. A. F., & Bocanegra-Herrera, C. C. (2017). Modelo metodológico de implementación de lean manufacturing1. Journal Of Systems And Software, 85. <https://n9.cl/50ke34>

**Conflicto de intereses:**

Los autores declaran que no existe conflicto de interés posible.

**Financiamiento:**

No existió asistencia financiera de partes externas al presente artículo.

**Agradecimiento:**

N/A

**Nota:**

El artículo no es producto de una publicación anterior.

