

## Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

### Exploratory study on the implementation of two software development methodologies in a school context

Diana Concepción Mex Alvarez <sup>1</sup>

Juan Miguel Durán Lugo <sup>2</sup>

Denice Ivet González Góngora <sup>3</sup>

Zurisaddai Alejandro Mayo <sup>4</sup>

#### RESUMEN

Actualmente las instituciones educativas de educación superior tienen el reto de recrear escenarios reales del campo laboral, por lo que los estudiantes de la Licenciatura como Ingeniero en Sistemas Computacionales, requieren generar las competencias necesarias en el desarrollo de metodologías para la creación de software. En este trabajo expone los hallazgos de un estudio exploratorio sobre la selección, implementación y reflexión de una metodología tradicional y una ágil en equipos de desarrollo de software de proyectos escolares. Para ello se construyó un instrumento con indicadores relevantes, obtenidos de una revisión sistemática de literatura. En los resultados obtenidos se observa que los proyectos desarrollados bajo la metodología tradicional con el modelo de Cascada presentan un desempeño predominantemente intermedio, donde la autoevaluación crítica y la producción de evidencias o artefactos se mantuvieron en valores reducidos, lo que refleja limitaciones en la retroalimentación y en la formalización del proceso. Los proyectos gestionados con la metodología ágil bajo el marco de trabajo Scrum, alcanzaron mejores resultados en la mayoría de los criterios evaluados, no obstante, se identificaron áreas de mejora en la comparación con Cascada y en la generación de evidencias o documentación, en donde los puntajes se concentraron en niveles medios.

**PALABRAS CLAVES:** Metodologías de desarrollo, software, cascada, scrum, instrumento.

**Fecha de recepción:** 26 de agosto, 2025.

**Fecha de aceptación:** 30 de septiembre, 2025.

<sup>1</sup> Universidad Autónoma de Campeche, [diancmex@uacam.mx](mailto:diancmex@uacam.mx), <https://orcid.org/0000-0001-9419-7868>

<sup>2</sup> Universidad Autónoma de Campeche, [jumduan@uacam.mx](mailto:jumduan@uacam.mx), <https://orcid.org/0000-0001-7179-4301>

<sup>3</sup> Dirección General de Educación Tecnológica Industrial, [denice.ivet.ce82@dgeti.sems.gob.mx](mailto:denice.ivet.ce82@dgeti.sems.gob.mx), <https://orcid.org/0009-0000-1057-1796>

<sup>4</sup> Dirección General de Educación Tecnológica Industrial, [zurisaddai.alejandro.cb9@dgeti.sems.gob.mx](mailto:zurisaddai.alejandro.cb9@dgeti.sems.gob.mx), <https://orcid.org/0009-0007-6451-4536>

## Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

### ABSTRACT

Currently, higher education institutions have the challenge of recreating real scenarios of the labor field, so students of the Bachelor's Degree in Computer Systems Engineering, need to generate the necessary skills in the development of methodologies for the creation of software. This paper presents the findings of an exploratory study on the selection, implementation and reflection of a traditional and an agile methodology in software development teams of school projects. For this purpose, an instrument was constructed with relevant indicators, obtained from a systematic literature review. The results obtained show that the projects developed under the traditional methodology with the Waterfall model present a predominantly intermediate performance, where critical self-evaluation and the production of evidence or artifacts remained at low values, reflecting limitations in the feedback and formalization of the process. The projects managed with the agile methodology under the Scrum framework achieved better results in most of the evaluated criteria, however, areas for improvement were identified in the comparison with Waterfall and in the generation of evidence or documentation, where the scores were concentrated in medium levels.

**KEYWORDS:** Development methodologies, software, waterfall, scrum, tool.

### INTRODUCCIÓN

El modelo de cascada es forma parte de las metodologías tradicionales, que dividen el proyecto en fases basadas en el tipo de trabajo como: toma de requerimientos, diseño de la arquitectura, desarrollo del software, pruebas y despliegue (Bibik, 2018).

Uno de los problemas con estas metodologías es que se podría llevar demasiado tiempo, meses o incluso años en obtener resultados finales.

Esto conlleva que a veces se descubra hasta en la etapa final que el diseño original era incorrecto, además que muchas cosas pueden ir cambiando en el transcurso.

En 1986 Alfred Spector, entonces presidente de Transarc Corporation, fue coautor de un documento que compara la construcción de puentes con el desarrollo del software. Considerando que la metodología de cascada se generó en la industria de la construcción, Alfred propuso una premisa, la cual es que los puentes normalmente se construyen a tiempo o dentro del presupuesto, debido al detalle extremo del diseño, el cual está congelado y el contratista tiene poca flexibilidad de cambiar las especificaciones. Por otro lado, el software nunca llega a tiempo o dentro del presupuesto, con el supuesto que el diseño requiere cierta flexibilidad para cambiar las especificaciones. (Standish Group, 1994)

En consecuencia, desde la década de los 90, han surgido nuevas metodologías (Carvajal, 2008) y a fines de los 90 trasciende una corriente caracterizada por la desaprobación de los supuestos beneficios de las metodologías, que dan lugar a la era postmetodológica, de donde surgen las metodologías ágiles al ser las más adecuadas para proyectos pequeños donde el entorno del sistema es muy cambiante y se exige reducir al máximo los tiempos de desarrollo manteniendo una alta calidad. (Garcés & Egas, 2015)

Las metodologías de desarrollo ágil procuran realizar los procesos de software de acuerdo con las prácticas que le han dado resultados al grupo. En las metodologías pesadas se desarrolla de acuerdo con las normas sugeridas por los estándares de desarrollo.

## Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

En las metodologías ágiles por la misma capacidad de reacción son más adaptables a los cambios, por el contrario, en las metodologías pesadas por el nivel de formalidad en la fase de requerimientos son más resistentes al cambio. Por su capacidad de adaptación el proceso se hace menos controlado que en las metodologías tradicionales que ejercen mayor control en el proceso por su nivel de formalización. (Rivas et al, 2015).

Respecto a la documentación, las metodologías ágiles no se hacen énfasis en la documentación excesiva, sino en los Artefactos que se generan a diferencia de las metodologías pesadas, en el caso de los equipos de trabajo, se generan con un bajo número de participantes y perfiles, por el contrario, en las metodologías pesadas se sugieren varios roles que proporcionan las normas.

Las metodologías ágiles, debido a su flexibilidad; se basan en la adaptación y la inspección continua por lo que generan marcos de trabajo que definen qué, quién, cuándo y por qué, pero no con la rigurosidad de un modelo, a eso le llamamos Marcos de trabajo como Scrum, Crystal Methodologies, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Feature-Driven Development (FDD) y Kanban. (Molina, 2013)

Para los fines de esta investigación, se eligió la metodología Scrum, la cual tiene iteraciones cortas como llamadas Sprints, tiene roles en el equipo, eventos y artefactos. La ventaja principal es que puede incorporar lo mejor tanto de XP como de Kanban según la situación y necesidades. En la tabla 1, se muestra la Comparación de Scrum y otras metodologías.

**Tabla 1. Comparación de Scrum y otras metodologías.**

Scrum	Otras
Estructura de reuniones formales, roles e iteraciones bien definidos.	<b>Kanban:</b> Sin reuniones formales, roles e iteraciones indefinidos.
La mayor prioridad es la productividad lo que lleva a la satisfacción del cliente y es más flexible.	<b>XP:</b> Menor flexibilidad y prioridad a la productividad.
Comunicación efectiva entre miembros del equipo, Menor complejidad involucrada.	<b>FDD:</b> Menor comunicación, procesos involucrados más complejos.
Mejor comunicación entre los miembros del equipo.	<b>DSDM:</b> Menor comunicación entre los miembros del equipo.
Procedimientos seguidos son fáciles y complejos.	<b>ASD:</b> Complejidad en la estructura de los procesos.
Los requerimientos de usuario definen estrictamente el desarrollo y planificación, mejor trazabilidad.	<b>Crystal:</b> Menor consideración con los requerimientos de usuario, dificultad de dar seguimiento al trabajo realizado.

Fuente: Srivastava et al., 2017.

En conclusión, en Scrum se da prioridad a los individuos y las interacciones más que a los procesos y a las herramientas, a los sistemas funcionando antes que, a la documentación detallada, a la colaboración con el cliente antes que la negociación de contratos, a la respuesta al cambio antes que seguir el plan. (Srivastava et al, 2017)

Considerando que cada metodología debe ser capaz de adaptarse a las circunstancias del proyecto, así como al tipo de sistema empleado, tiempos de desarrollo y recursos, se dice que no existe una metodología universal que afronte cualquier tipo de proyecto de desarrollo de software.

## Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

A lo largo de la historia, las metodologías tradicionales con un gran esfuerzo han intentado adaptarse a las diferentes situaciones de los proyectos los cuales presentan requisitos que van cambiando a lo largo del desarrollo del proyecto.

En este trabajo expone los hallazgos de un estudio exploratorio sobre la selección, implementación y reflexión de una metodología tradicional y una ágil en equipos de desarrollo de software de proyectos escolares.

### JUSTIFICACIÓN

La elección de una metodología de desarrollo de software constituye un elemento central en la gestión de proyectos, ya que influye directamente en la organización de las actividades, el cumplimiento de objetivos y la calidad de los resultados. (Yepes González et al, 2015)

En el ámbito académico, los proyectos de software se configuran como espacios formativos donde los estudiantes adquieren no solo competencias técnicas, sino también habilidades de planificación, comunicación y trabajo colaborativo. Bajo este contexto, resulta pertinente llevar a cabo un estudio exploratorio que examine la selección, implementación y reflexión en torno al uso de una metodología tradicional, como el modelo en cascada, y una metodología ágil, como Scrum, dentro de equipos de desarrollo en proyectos escolares. El carácter exploratorio de la investigación permitirá analizar de qué manera los estudiantes interpretan y adaptan metodologías profesionales a entornos educativos con recursos, tiempos y objetivos delimitados. (Orjuela & Rojas, 2008).

### METODOLOGÍA

El estudio es un diseño no experimental, de tipo cuantitativo con alcance exploratorio, ya que no se manipularán variables y los datos se recolectarán en un solo momento en el tiempo. Este diseño es adecuado para poder abordar los fenómenos que aún no han sido ampliamente estudiados en este contexto. El alcance de este trabajo no abarca el ofrecer soluciones finales ni conclusivas acerca de los problemas existentes en el área, sino proporcionar un análisis descriptivo que sirva como base para futuras investigaciones y la implementación de estrategias de mejora.

A través del análisis de los datos obtenidos, se podrán identificar aspectos clave que contribuyan a comprender la forma en la que aplican las metodologías y con ello fortalecer el conocimiento y las prácticas las metodologías tradicionales y ágiles.

#### Construcción del instrumento.

Para obtener la información relevante sobre el empleo de las metodologías tradicionales y ágiles, se empleó la revisión sistemática de literatura con la finalidad de recabar los principales indicadores y con ello generar nuestro propio instrumento. (Biolchini, Gomes, Cruz, y Travassos, 2005)".

#### (1) Enfoque de la pregunta:

Está investigación se llevó a cabo para analizar el conjunto de indicadores para conocer el empleo de metodologías tradicionales y ágiles en contextos escolares.

#### (2) Amplitud y calidad de la pregunta

##### (a) Problema:

Actualmente las instituciones educativas de educación superior tienen el reto de recrear escenarios reales del campo laboral, por lo que los estudiantes de la Licenciatura como Ingeniero en Sistemas

## Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

Computacionales, requieren generar las competencias necesarias en el desarrollo de metodologías tradicionales y ágiles para la creación de software.

### (b) Pregunta:

¿Cuáles son los indicadores más relevantes para conocer el empleo de metodologías tradicionales y ágiles en contextos escolares?

### (c) Población:

Publicaciones y estudios previos sobre el empleo de metodologías tradicionales y ágiles en contextos escolares

### (3) Identificación y selección de fuentes.

#### (a) Definición de los criterios de la selección de fuentes:

- Utilizar mecanismos de búsqueda con palabras claves.
- Publicaciones recomendadas por otros autores.
- Publicaciones disponibles en sitios web.

#### (b) Idioma de los estudios:

- Español
- Inglés

### (4) Métodos de búsqueda de fuentes

Para llevar a cabo esta revisión sistemática, se realizó una investigación empleando bases de datos de artículos y resúmenes de literatura arbitrada para seleccionar los documentos que contengan información relevante sobre el tema.

#### (a).- Cadena de búsqueda

Mediante la combinación de las palabras identificadas, se aplicaron los conectores lógicos “AND”, “OR” y “NOT” para formar una cadena básica de búsqueda general (ver tabla 2).

**Tabla 2 - Cadena de búsqueda.**

<b>Palabras clave con operadores lógicos</b>
<i>(“Metodologías tradicionales” OR “Metodologías ágiles” OR “Cascada” OR “Scrum”) AND (“Nivel de empleo” OR “Apropiación” OR “Indicadores” OR “Evaluación”) AND (“contexto escolar” OR “estudiantes” OR “alumnado” OR “educación superior”)</i>

Fuente: Elaboración propia

#### (b).- Lista de fuentes

- Google Académico
- Redalyc
- SciELO
- Dialnet

#### (c).- Definición de criterios de inclusión (CI) de estudios y de exclusión (CE) de estudios

En la definición de criterios de inclusión (CI) de estudios y de exclusión (CE) de estudios, se identificaron 4 y 2, respectivamente, los cuales se muestran en la tabla 3.

Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

**Tabla 1 - Definición de criterios de inclusión y exclusión de estudios.**

<b>Criterio</b>	<b>Descripción</b>
CI1	<i>Incluye estudios sobre el empleo de metodologías tradicionales y ágiles en contextos escolares</i>
CI2	<i>Incluye publicaciones que contengan términos clave relacionados al empleo de metodologías tradicionales y ágiles en contextos escolares</i>
CI3	<i>Incluye estudios con metodologías aplicada en poblaciones estudiantiles.</i>
CI4	<i>Incluye investigaciones revisadas y de acceso público.</i>
CI5	<i>Incluye estudios que presentan análisis cuantitativos sobre el empleo de metodologías tradicionales y ágiles en contextos escolares</i>
CE1	<i>Excluye estudios que no aborden directamente sobre el empleo de metodologías tradicionales y ágiles en contextos escolares.</i>
CE2	<i>Excluye todas las publicaciones duplicadas.</i>
CE3	<i>Excluye estudios que no presenten una metodología clara o carecen de datos verificables.</i>
CE4	<i>Excluye documentos que no puedan ser visualizados de manera directa o parcialmente.</i>

Fuente: Elaboración propia

### (5) Ejecución de la Extracción

El conjunto de documentos recopilados fue organizado utilizando plantillas que facilitan la extracción de datos relevantes. La evaluación se realizó analizando las ideas principales, siguiendo criterios específicos para incluir o descartar información. En algunos casos, ciertos artículos no abordaban el tema con suficiente profundidad, por lo que se procedió a examinar e interpretar la introducción para obtener una mejor comprensión del enfoque de la investigación. Esto ayudó a determinar si el contenido de los artículos estaba relacionado con el tema de estudio.

La información extraída de las publicaciones consideradas como fuentes primarias fue registrada en las plantillas correspondiente. En la tabla 4, se ejemplifica cómo se utilizó en el análisis de uno de los estudios seleccionados.

Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

**Tabla 4 – Ejemplo de almacenamiento de la información.**

<b>Título</b>	Modelo de Evaluación de Metodologías para el Desarrollo de Software
<b>Año</b>	2006
<b>Autores</b>	Méndez Nava Elvia Margarita
<b>Resumen</b>	El objetivo general del trabajo es diseñar un modelo de evaluación de metodologías para el desarrollo de software. Se plantean tres objetivos específicos: Identificar variables para la evaluación de metodologías aplicadas al desarrollo de software, diseñar el modelo de evaluación de metodologías y desarrollar la matriz de evaluación de metodologías.

Fuente: Elaboración propia

**(6) Construcción del instrumento.**

De acuerdo con la información extraída, el instrumento resultante se compuso de 19 indicadores, clasificados en 4 secciones de acuerdo con la finalidad de la pregunta. Posteriormente por cada indicador se desarrolló un reactivo de opción múltiple donde se establecieron cinco escalas de Likert con diferentes niveles: bajo, regular, suficiente, bueno y excelente En la tabla 5, se presentan los 19 indicadores con sus identificadores (Id\_indicador), clasificados por sección.

**Tabla 5 – Ejemplo de almacenamiento de la información.**

	Id indicador	Indicador
Sección 1: Comprensión del Modelo de Proceso	A	Identificación clara del modelo utilizado (nombre, tipo, origen).
	B	Explicación adecuada de los fundamentos y fases del modelo.
	C	Descripción crítica de ventajas y desventajas.
	D	Comparación con al menos otro modelo de proceso
	E	Comprensión de cómo el modelo se ajusta al ciclo de vida del software.
Sección 2: Justificación de la Elección del Modelo	F	Análisis del contexto del proyecto para justificar la elección.
	G	Relación entre el modelo y el tamaño/alcance del software.
	H	Consideración del grado de claridad de los requisitos.
	I	Ajuste del modelo a los recursos y tiempo disponibles.
Sección 3: Aplicación Práctica del Modelo	J	Implementación coherente de las fases del modelo.
	K	Gestión adecuada de requerimientos.
	L	Planificación, estimación y seguimiento del trabajo.
	M	Gestión del cambio a lo largo del desarrollo.
	N	Asignación de roles según el modelo.



**Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar**

	O	Generación de evidencias o artefactos (documentos, tableros, actas, etc.)
Sección 4: Reflexión y Mejora	P	Autoevaluación crítica del proceso seguido
	Q	Identificación de lecciones aprendidas.
	R	Propuestas de mejora para el modelo aplicado.
	S	Adaptabilidad del modelo según el desarrollo del proyecto.

Fuente: Elaboración propia

El instrumento resultante se digitalizó en la herramienta de Google Forms.

**Figura 1. Portada de instrumento digitalizado.**

Fuente: Elaboración propia

### Definición de la población

La población para estudiar se compuso por 65 estudiantes de la licenciatura de Ingeniero en Sistemas Computacionales de una Institución de Educación Superior del Estado de Campeche, organizados en trece equipos de cinco integrantes.

Todos los estudiantes cursan el mismo ciclo escolar y recibieron la capacitación sobre las diferentes metodologías para el desarrollo de software. De acuerdo con su afinidad, cinco equipos eligieron emplear el modelo de cascada y ocho la metodología ágil de scrum para el desarrollo de un proyecto de software, el cual desarrollaron en tres meses.

### Aplicación del instrumento.

Una vez liberado el proyecto de software, un grupo de tres investigadores, aplicaron la técnica de entrevista a los integrantes de cada equipo de trabajo, para posteriormente verter los resultados en el instrumento por cada equipo, los cuales se presentan en la tabla 5.



Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

**Tabla 6 – Valores obtenidos en la aplicación de la encuesta.**

Equipo	Indicador																		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	3	3	3	2	4	3	4	4	3	5	5	5	4	4	4	5	5	3	5
2	3	3	3	3	4	3	4	4	3	4	3	5	4	4	3	4	4	4	3
3	3	3	3	3	4	3	3	3	3	3	4	4	4	4	4	4	3	4	3
4	4	4	3	2	4	4	4	4	5	5	4	5	4	3	3	3	3	3	5
5	3	3	3	2	3	2	3	3	4	5	5	5	5	4	3	5	4	4	4
6	3	3	3	3	4	3	3	3	3	3	4	4	4	4	4	4	3	4	3
7	3	3	3	2	3	3	3	3	4	5	5	5	4	3	3	4	4	4	4
8	3	3	3	2	3	2	3	3	4	5	5	5	5	4	3	5	4	4	4

Fuente: Elaboración propia

### Evaluación de la confiabilidad

La consistencia interna del instrumento se evaluó mediante el coeficiente alfa de Cronbach de acuerdo con los valores obtenidos en la aplicación de la encuesta:

Siendo que:

$$\alpha = \frac{k}{k-1} \left( 1 - \frac{\sum \sigma_x^2}{\sigma^2_x} \right)$$

Donde:

$$\alpha = \frac{19}{19-1} \left( 1 - \frac{10.982}{66.99} \right)$$

El resultado es de 0.88, considerado “Muy Alto” de acuerdo con los criterios de George y Mallery (2024).

## DISCUSIÓN DE RESULTADOS

### Análisis de los datos.

Una vez recolectados los datos del cuestionario, se procesaron en Excel, obteniendo los valores estadísticos, presentados en la tabla 7.

**Tabla 7 – Valores estadísticos, ordenados por media.**

Indicador	$\mu$	$\sigma$	Nivel mínimo	Nivel máximo	Indicadores relevantes (niveles 4-5)
L	4.75	0.463	4	5	8
J	4.375	0.916	3	5	6
K	4.375	0.744	3	5	7
P	4.25	0.707	3	5	7
M	4.25	0.463	4	5	8

**Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar**

S	3.875	0.835	3	5	5
N	3.75	0.463	3	4	6
Q	3.75	0.707	3	5	5
R	3.75	0.463	3	4	6
I	3.625	0.744	3	5	4
E	3.625	0.518	3	4	5
G	3.375	0.518	3	4	3
H	3.375	0.518	3	4	3
O	3.375	0.518	3	4	3
B	3.125	0.354	3	4	1
A	3.125	0.354	3	4	1
C	3	0.000	3	3	0
F	2.875	0.641	2	4	1
D	2.375	0.518	2	3	0

**Fuente: Elaboración propia**

La media ( $\mu$ ) nos presenta el nivel promedio de la frecuencia por cada indicador, por lo tanto, los valores altos denotan mayor presencia del rango. El indicador L, es el que presenta una mayor media con un valor de 4.75, es decir, que los participantes expresan un acuerdo casi total. Por otro lado, los indicadores K, J, P y M, tienen una media alta con valores que oscilan entre 4.25 y 4.38. Los indicadores con una media moderada-alta son S, N, Q, R, I, E (3.63-3.88) y G, H, O (3.38) al tener una media moderada, muestran neutralidad de opiniones. Por último, los indicadores A, B, F, C y D, con una media baja-moderada, nos presentan que los grupos no tienen afinidad con ellos.

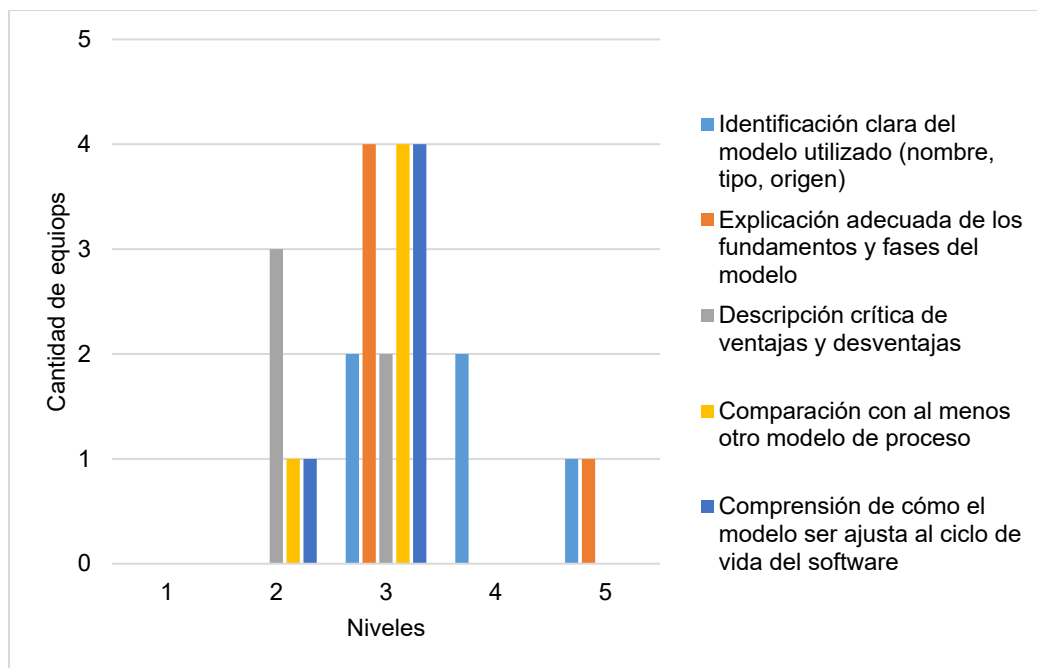
A través de la desviación estándar ( $\sigma$ ), obtuvimos que tan dispersas están las respuestas. El rango de las desviaciones va de 2.38 a 4.75, por lo que hay una dispersión notable, por lo tanto, valoraciones variadas.

En las columnas nivel mínimo y nivel máximo, presentan las escalas de Likert de mayor y menor valor por cada indicador, destacando que el valor más bajo que fue elegido fue el 2. Por último, en la columna de indicadores relevantes, se muestran cuántas veces fueron elegidos los niveles 4 o 5, en cada indicador, destacando L y M con 8, K y P con 7, así como J y N con 6.

A través de un histograma, en las figuras 2, 3, 4 y 5 se presentan los resultados de los cinco equipos que implementaron la metodología tradicional de cascada, donde el eje x representa cada uno de los niveles de la escala de Likert y la cantidad de equipos que obtuvieron ese nivel está representado en el eje de las y.

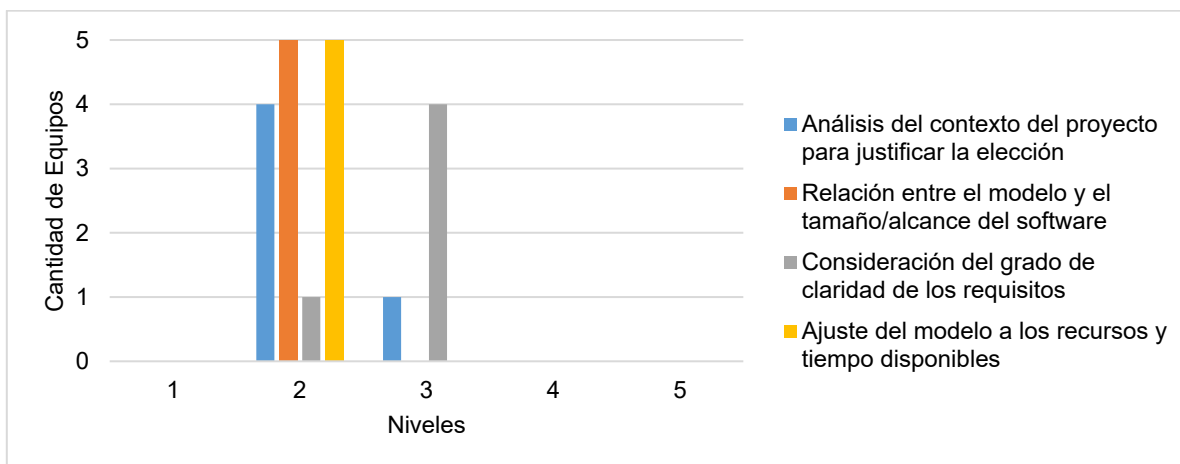
La figura 2, correspondiente a la categoría “Comprensión del Modelo de Proceso”, destaca con los indicadores “Explicación adecuada de los fundamentos y fases del modelo”, “Comparación con al menos otro modelo de proceso” y “Comprensión de cómo el modelo se ajusta al ciclo de vida del software” los cuatro equipos tuvieron un nivel 3 (suficiente), siendo el más alto.

Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar



**Figura 2. Comprensión de la metodología**  
 Fuente: Elaboración propia

Respecto a la categoría “Justificación de la Elección del Modelo”, en la figura 3 podemos observar que los indicadores “Ajuste del modelo a los recursos y tiempo disponibles” y “Relación entre el modelo y el tamaño/alcance del software” los cinco equipos tuvieron un nivel 2 (regular). Otro indicador a destacar es el “Análisis del contexto del proyecto para justificar la elección” donde cuatro equipos obtuvieron el nivel 2 (regular). Es notorio que ningún equipo tuvo niveles de bueno o excelente.

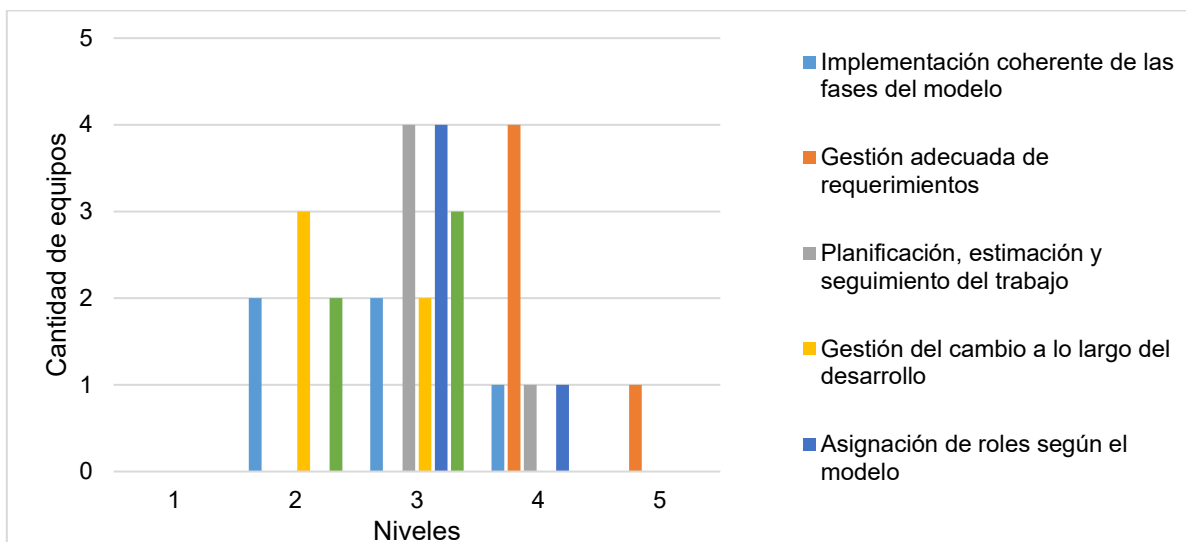


**Figura 3. Justificación de la Elección de la Metodología**  
 Fuente: Elaboración propia

En la categoría “Aplicación Práctica de la Metodología” observamos de acuerdo con la figura 4 que los indicadores “Planificación, estimación y seguimiento del trabajo” y “Asignación de roles según

## Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

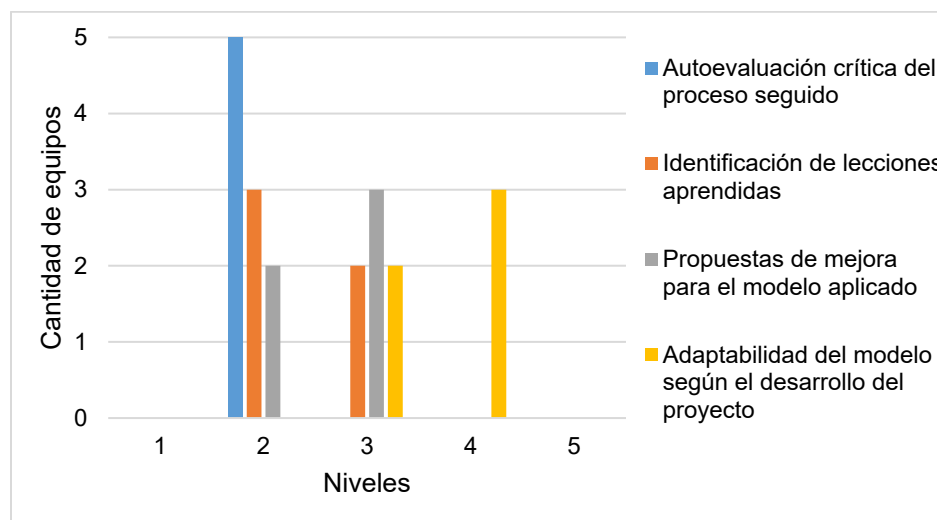
modelo” cuatro de los cinco equipos obtuvieron el nivel 3 (suficiente) y en el indicador “Gestión adecuada de requerimientos” cuatro equipos alcanzaron el nivel 4 (bueno) y un equipo el nivel 5 (excelente).



**Figura 4. Aplicación Práctica de la Metodología**

Fuente: Elaboración propia

En la categoría “Reflexión y mejora” que hace referencia la figura 5 que el indicador “Autoevaluación crítica del proceso seguido” obtuvo los 5 equipos con el nivel 2 (regular) y el indicador “Adaptabilidad del modelo según el desarrollo del proyecto” tuvo un equipo con nivel 4 (bueno)



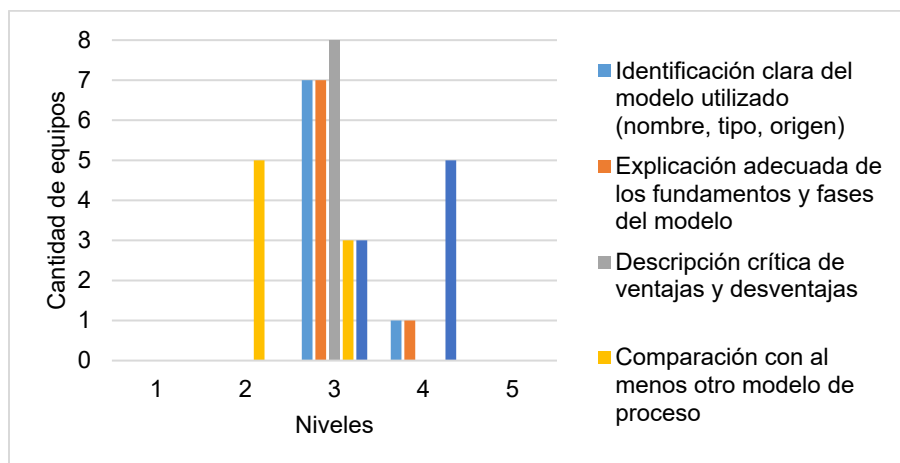
**Figura 5. Reflexión y mejora.**

Fuente: Elaboración propia

En las figuras 6, 7, 8 y 9 se encuentran los resultados de los ocho equipos que implementaron la metodología ágil Scrum.

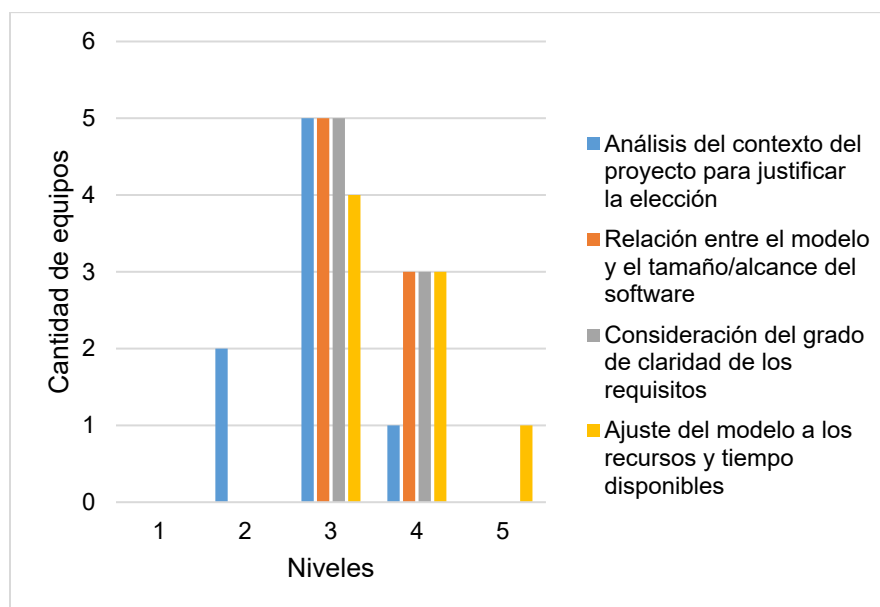
## Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar

Para la metodología ágil Scrum, en la figura 6, observamos que la categoría “Comprensión del Modelo de Proceso”, en el indicador “Descripción crítica de ventajas y desventajas” los ocho equipos obtuvieron el nivel 3 (suficiente). Respecto a los indicadores “Explicación adecuada de los fundamentos y fases del modelo” e “Identificación clara del modelo utilizado (nombre, tipo, origen)”, siete equipos obtuvieron el nivel 3 (suficiente). Por su parte el indicador “Comparación con al menos otro modelo de proceso”, obtuvo el nivel 4 (bueno) en cinco de sus equipos.



**Figura 6. Comprensión de la Metodología**  
 Fuente: Elaboración propia

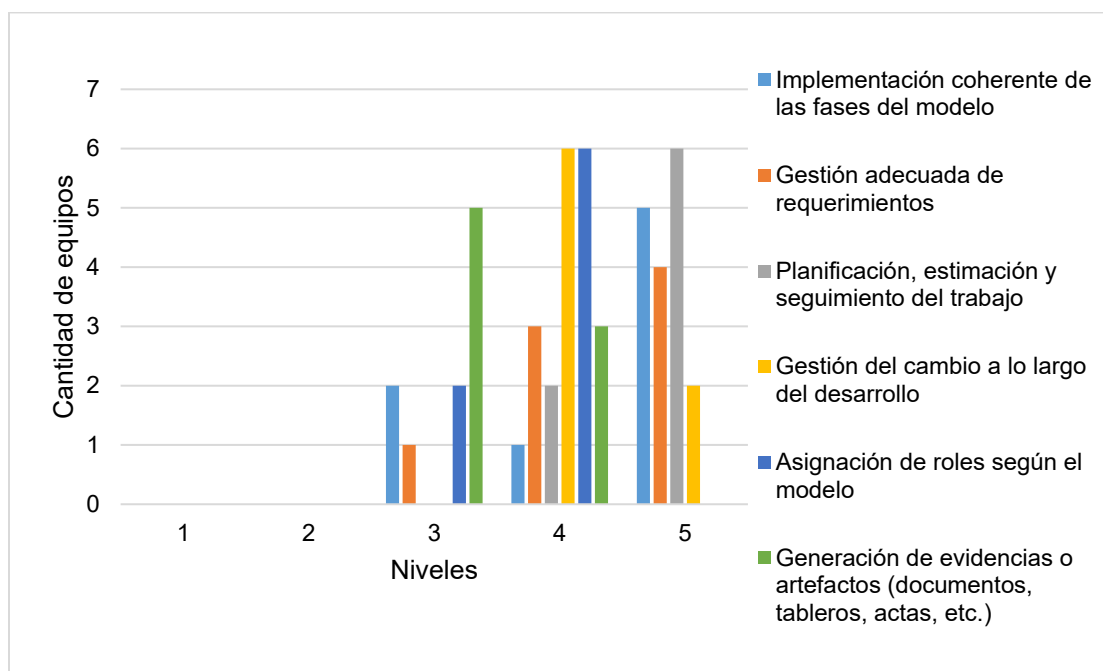
En la figura 7, observamos que la categoría “Justificación de la Elección del Modelo” el indicador “Ajuste del modelo a los recursos y tiempo disponibles” es el que tuvo un equipo con el nivel más alto.



**Figura 7. Justificación de la Metodología**  
 Fuente: Elaboración propia

**Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar**

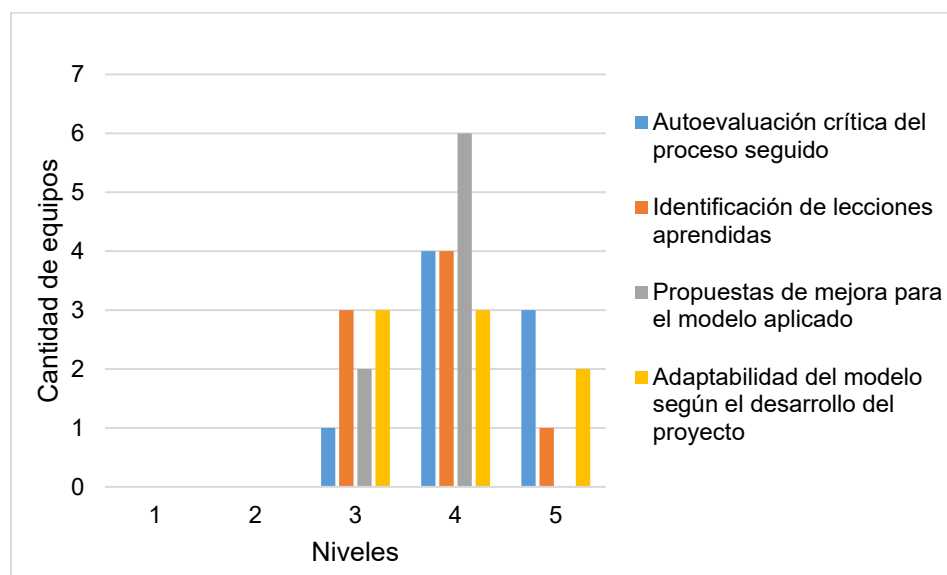
En la categoría “Aplicación práctica del modelo”, cuatro de los seis indicadores en total, tuvieron equipos con nivel de excelencia.



**Figura 8. Aplicación práctica de la Metodología**

Fuente: Elaboración propia

En la categoría “Reflexión y mejora”, tres de los cuatro indicadores que lo conforman, tuvieron equipos con nivel de excelencia.



**Figura 9. Reflexión y mejora**

Fuente: Elaboración propia

## CONCLUSIÓN Y RECOMENDACIONES

En primer lugar, se observa que los proyectos desarrollados bajo la metodología tradicional (modelo de Cascada) presentan un desempeño predominantemente intermedio. La mayoría de las respuestas se concentraron en valores de 2 y 3, lo cual refleja un nivel moderado de cumplimiento de los criterios de evaluación. Se identificaron fortalezas en aspectos como la gestión de requerimientos, donde varios proyectos alcanzaron calificaciones altas. No obstante, se evidenciaron debilidades significativas en criterios como la relación entre el modelo y el tamaño del software, el ajuste a los recursos y tiempo disponibles, y la gestión del cambio, en los cuales los puntajes se ubicaron en su mayoría en niveles bajos. Asimismo, la autoevaluación crítica y la producción de evidencias o artefactos se mantuvieron en valores reducidos, lo que refleja limitaciones en la retroalimentación y en la formalización del proceso.

En contraste, los proyectos gestionados con la metodología ágil (marco de trabajo Scrum) alcanzaron mejores resultados en la mayoría de los criterios evaluados. La distribución de respuestas se concentró en niveles altos (4 y 5), lo que muestra un desempeño favorable. Destacan especialmente la planificación, estimación y seguimiento del trabajo, la implementación coherente de las fases, la gestión del cambio y la autoevaluación crítica, donde la mayoría de los proyectos lograron calificaciones sobresalientes. No obstante, se identificaron áreas de mejora en la comparación con Cascada y en la generación de evidencias o documentación, en donde los puntajes se concentraron en niveles medios.

Al realizar la comparación entre ambos modelos, se evidencia una clara diferencia, mientras que Cascada muestra consistencia únicamente en la gestión de requerimientos y rigidez frente a cambios o contextos dinámicos, Scrum se posiciona como una metodología flexible, con resultados superiores en adaptabilidad, seguimiento, retroalimentación y aprendizaje.

## REFERENCIAS

- Bibik, I. (2018). From Waterfall to Agile. En How to kill the Scrum monster (pp. 1–3). Apress.  
[https://doi.org/10.1007/978-1-4842-3884-0\\_1](https://doi.org/10.1007/978-1-4842-3884-0_1)
- Biolchini, J., Gomes, P., Cruz, A., & Orta, G. (2005). Systematic review in software engineering (RT-ES679/05). Rio de Janeiro: Systems Engineering and Computer Science Department, COPPE/UFRJ.
- Garcés, L., & Egas, L. (2015). Evolución de las metodologías de desarrollo de la ingeniería de software en el proceso de la ingeniería de sistemas. Revista Científica y Tecnológica UPSE, 1(3), 15–22.
- George, D., & Mallery, P. (2024). IBM SPSS statistics 29 step by step: A simple guide and reference. Routledge.
- Méndez Nava, E. M. (2006). [Tesis de maestría, Universidad Católica Andrés Bello]. Biblioteca UCAB. <http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAQ7365.pdf>
- Molina, S. G. R. (2013). Metodologías ágiles enfocadas al modelado de requerimientos. Informes Científicos Técnicos – UNPA, 5(1), 1–29. <https://doi.org/10.22305/ict-unpa.v5i1.58>



**Estudio exploratorio sobre la implementación de dos metodologías de desarrollo de software en un contexto escolar**

- Rivas, C., Corona, V., Gutiérrez, J., & Hernández, L. (2015). Metodologías actuales de desarrollo de software. *Revista de Tecnología e Innovación*, 2(5), 980–986.
- Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). SCRUM model for agile methodology. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 864–869). IEEE. <https://doi.org/10.1109/CCAA.2017.8229928>
- The Standish Group. (1994). The CHAOS Report. [http://www.standishgroup.com/sample\\_research\\_files/chaos\\_report\\_1994.pdf](http://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf)
- Orjuela, A. D., & Rojas, M. (2008). Las metodologías de desarrollo ágil como una oportunidad para la ingeniería del software educativo. *Revista Avances en Sistemas e Informática*, 5(2), 159–171.
- Yepes González, J. D., Pardo Calvache, C. J., & Gómez, O. S. (2015). Revisión sistemática acerca de la implementación de metodologías ágiles y otros modelos en micro, pequeñas y medianas empresas de software. *Revista Tecnológica – ESPOL*, 28(5), 45–54.