

Revisión de metodologías de desarrollo de propuestas tecnológicas mediante minería de texto: un caso de estudio

Review of technological proposal development methodologies using text mining: a case study

Nuvia Beltrán Robayo ¹[0000-0002-3335-576X], Irene Vásquez Villacis ¹[0000-0002-3335-576X]

¹ Universidad Agraria del Ecuador, Milagro, Ecuador. nbeltran, mivasquez @uagraria.edu.ec

CITA EN APA:

Beltrán Robayo, N., & Vásquez Villacis, I. (2025). Revisión de metodologías de desarrollo de propuestas tecnológicas mediante minería de texto: un caso de estudio. *Technology Rain Journal*, 4(2). Recuperado a partir de <https://technologyrain.com.ar/index.php/trj/article/view/101>

Recibido: 01 de septiembre del 2025
Aceptado: 30 de septiembre del 2025
Publicado: 15 de octubre del 2025

Technology Rain Journal
ISSN: 2953-464X

Resumen. Este estudio tiene como objetivo revisar las metodologías de desarrollo de propuestas tecnológicas aplicadas por los estudiantes de la carrera de Computación mediante la aplicación de técnicas de minería de texto. Para ello, se procedió a descargar las tesis de titulación relacionados a la Agromática correspondientes a los años 2022 a mayo del 2024. Posteriormente, se segmentó el contenido de cada tesis, focalizándose en la sección de metodología. A través de un proceso de extracción y análisis textual, se identificaron las metodologías de desarrollo de software empleadas, permitiendo así establecer un panorama sobre las estrategias metodológicas implementadas en los proyectos académicos. La utilización de técnicas de minería de texto permitió sistematizar y categorizar la información, facilitando el reconocimiento de patrones recurrentes y tendencias en el uso de metodologías de desarrollo. Los resultados muestran una clara predominancia de metodologías ágiles, en especial Extreme Programming (XP), mientras que modelos clásicos como el cascada aún mantienen presencia en contextos con requisitos bien definidos. La investigación evidencia que el uso de herramientas computacionales permite optimizar el análisis de grandes volúmenes de información, aportando objetividad en la clasificación metodológica.

Palabras clave: Agromática, Metodologías de desarrollo, minería de texto, propuestas tecnológicas



Los contenidos de este artículo están bajo una licencia de Creative Commons Attribution 4.0 International (CC BY 4.0)

Los autores conservan los derechos morales y patrimoniales de sus obras.

Abstract

This study aims to review the software development methodologies applied by Computer Science students in their technological project proposals through the implementation of text mining techniques. To this end, undergraduate theses related to Agromatics, published between 2022 and May 2024, were collected. The content of each thesis was segmented, focusing specifically on the methodology section. Through a process of text extraction and analysis, the software development methodologies employed were identified, allowing the establishment of an overview of the methodological strategies implemented in academic projects. The application of text mining techniques enabled the systematization and categorization of information, facilitating the recognition of recurring patterns and trends in the use of development methodologies. The findings reveal a clear predominance of agile methodologies, particularly Extreme Programming (XP), while traditional models such as the Waterfall approach still remain in use in contexts with well-defined requirements. The study highlights that the use of computational tools optimizes the analysis of large volumes of information, providing greater objectivity in methodological classification.

Keywords: Agromatics, Development Methodologies, Text Mining, Technological Proposals

1 INTRODUCCIÓN

En las instituciones de educación superior, los proyectos de titulación representan una de las instancias más importantes en la formación de los futuros profesionales, ya que los estudiantes tienen la oportunidad de aplicar de manera integrada los conocimientos adquiridos a lo largo de su carrera, desarrollando soluciones viables y bien fundamentadas. Sin embargo, establecer una metodología adecuada en ocasiones resulta complicada, ya que existen diferentes enfoques, debido a la existencia de múltiples enfoques, cada uno con sus ventajas, limitaciones y diferentes puntos de vista de su aplicación.

En el área de Computación es de relevancia seleccionar de forma correcta la metodología de desarrollo de propuestas tecnológicas, ya que esta influye directamente en la calidad, eficiencia y resultados del proyecto. Según Pérez (2023) el uso adecuado de metodologías de desarrollo de software no solo mejora la organización del trabajo, sino que también incrementa la probabilidad de éxito del producto final, al ofrecer una guía clara para las etapas de planificación, diseño, implementación, pruebas y mantenimiento del sistema.

En este sentido, resulta fundamental disponer de herramientas que permitan analizar de manera objetiva y sistemática la forma en que se estructuran las propuestas de titulación, con especial atención al componente metodológico. Para ello, se recurrirá al uso de técnicas de minería de texto, las cuales permiten extraer, procesar y categorizar información relevante a partir de documentos textuales. Esta herramienta facilitará la identificación de patrones recurrentes en la elección de metodologías de desarrollo empleadas por los estudiantes, aportando así una visión más clara y fundamentada sobre las prácticas metodológicas adoptadas en sus proyectos.

Por ende, este estudio tiene como objetivo revisar las metodologías de desarrollo de propuestas tecnológicas aplicadas por los estudiantes de la carrera de Computación, mediante la aplicación de técnicas de minería de texto. Para ello, se recopilarán tesis de titulación centradas en proyectos de Agromática, elaboradas entre 2022 y mayo de 2024. El análisis se enfocará en el apartado de metodología de cada documento, con el fin de identificar y categorizar las metodologías empleadas, permitiendo identificar las implementadas en el desarrollo de propuestas tecnológicas dentro del ámbito académico.

Minería de texto

Minería de texto es una técnica que permite descubrir información valiosa oculta dentro de grandes volúmenes de texto no estructurado, como documentos, artículos, informes o tesis. A través del uso de algoritmos y herramientas de procesamiento de lenguaje natural, esta técnica permite identificar

patrones, tendencias y relaciones que no siempre son evidentes a simple vista (Aleqabie et al., 2024). En este caso, los proyectos de titulación contienen una gran cantidad de información valiosa que, si se revisara manualmente, demandaría un tiempo considerable por lo cual la aplicación de técnicas de minería de texto permitió automatizar y sistematizar la extracción de datos, facilitando la identificación de metodologías de desarrollo de propuestas tecnológicas utilizadas por los estudiantes.

Metodologías de desarrollo de propuestas tecnológicas

Los estudiantes se deben familiarizar con las distintas metodologías de desarrollo de software, teniendo en cuenta que existen las metodologías tradicionales que son más estructuradas y tienen un proceso más detallado, lo que puede resultar favorable para los proyectos donde los requisitos están muy claros y casi no tienen necesidad de algún cambio; por otro lado, las metodologías ágiles que son más flexibles y rápidas, permitiendo los cambios que sean necesarios mientras que el proyecto vaya avanzando, estos son una buena opción para los proyectos donde los requisitos pueden cambiar rápidamente.

Al momento de desarrollar productos o soluciones dirigidas a un cliente o a un mercado específico, es fundamental considerar aspectos como los costos, la planificación, la complejidad del proyecto, los recursos humanos disponibles y las tecnologías empleadas. Todos estos elementos se integran dentro de una metodología de desarrollo, la cual proporciona una estructura organizada para gestionar el trabajo de manera eficiente (Merchán et al., 2024). Es importante que los estudiantes entiendan las ventajas y desventajas que tiene cada metodología para escoger la más adecuada para su proyecto de tesis.

Las metodologías de desarrollo de software ofrecen un marco sistemático para guiar el ciclo de vida de un proyecto, estructurando claramente sus fases esenciales. Según Klünder et al. (2021) estas metodologías suelen adaptarse al contexto del equipo y combinan prácticas y fases como gestión de requisitos, análisis, diseño, implementación, pruebas, calidad, administración de riesgos y mantenimiento. En particular, el modelo en cascada (o waterfall) sigue un flujo secuencial donde cada etapa como la recolección de requisitos, el diseño, la implementación, la verificación y el mantenimiento debe completarse antes de comenzar la siguiente.

Por otro lado, los enfoques iterativos e incrementales, como el modelo en V, el desarrollo incremental y los métodos ágiles (Agile, Scrum, FDD), promueven ciclos cortos con retroalimentación continua. Estos ciclos comprenden fases de planificación, análisis detallado, diseño, codificación, pruebas y retroalimentación o despliegue (Piotrovskyi, 2025).

Adicionalmente, el proceso espiral agrega un ciclo formal de evaluación de riesgos en cada iteración, aumentando el enfoque en la gestión de incertidumbre.

Metodologías tradicionales

Modelo Waterfall

El modelo Waterfall o en cascada es una metodología secuencial y planificada que divide el desarrollo en fases claramente delimitadas: requisitos, diseño, implementación, pruebas, despliegue y mantenimiento. Cada fase debe completarse antes de avanzar a la siguiente, lo que dota al proceso de transparencia, trazabilidad y control formal de entregables y facilita la documentación precisa (JW Wu, 2024). Sin embargo, su rigidez limita la adaptabilidad, pues los cambios posteriores son costosos y complejos, lo que lo hace adecuado exclusivamente cuando los requisitos son estables y bien definidos.

Las fases de esta metodología son:

- **Análisis:** donde se identifica, recolecta y documenta los requisitos del sistema, tanto funcionales como no funcionales, garantizando claridad y consenso entre cliente y equipo de desarrollo.
- **Diseño:** se realiza la traducción de requisitos en especificaciones técnicas, definiendo arquitectura, modelos de datos, interfaces y componentes necesarios para la construcción del sistema.
- **Implementación:** se centra en el desarrollo y codificación del sistema conforme al diseño establecido, aplicando estándares de programación e integrando módulos en un producto funcional inicial.
- **Verificación:** se evalúa el sistema mediante pruebas unitarias, de integración, sistema y aceptación, asegurando que cumple con los requisitos definidos y corrigiendo errores detectados.
- **Mantenimiento:** Corrección de fallos, actualización de funcionalidades y adaptación del sistema a nuevas necesidades o entornos, garantizando estabilidad y continuidad operativa.

Modelo Evolutivo

El modelo evolutivo de desarrollo de software se caracteriza por su enfoque incremental, en el cual el sistema se construye y mejora progresivamente a lo largo del tiempo. A diferencia de los enfoques lineales tradicionales, este modelo permite realizar entregas parciales del producto que van siendo ajustadas y refinadas en función de la retroalimentación del usuario final (González et al., 2022).

Desde las primeras etapas de implementación, el software se adapta continuamente a nuevas necesidades o condiciones, lo que garantiza una alineación más estrecha con los requerimientos reales. Esta constante evolución no solo mejora la calidad del sistema, sino que también permite reducir riesgos, ya que los errores o desajustes pueden ser identificados y corregidos en fases tempranas.

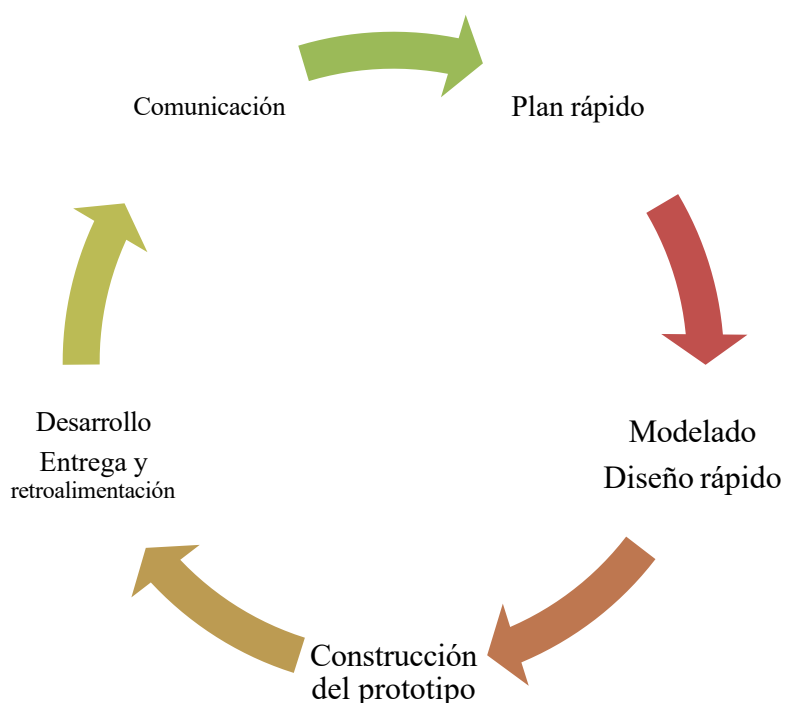


Figura 1. Fases del modelo evolutivo

Proceso Unificado de Rational (RUP)

El Proceso Unificado de Rational (RUP) es una metodología de ingeniería de software que proporciona un marco para distribuir tareas y responsabilidades dentro de un equipo de desarrollo (Barrientos M. et al., 2022).

Su finalidad es garantizar la creación de software de alta calidad que cumpla con las expectativas de los usuarios, asegurando la entrega dentro de plazos establecidos y respetando los límites presupuestarios previstos (Arregocés et al., 2022), entre las fases se destacan:

Inicio: En esta etapa se determina el alcance del proyecto, se identifican los participantes clave, se establecen los casos de uso iniciales y se evalúa la factibilidad del sistema. Su propósito es comprender claramente los requisitos y sentar una base sólida para el desarrollo posterior.

Elaboración: Esta fase implica un análisis más profundo de los requisitos, la definición de la arquitectura del sistema y la identificación de los principales riesgos. Busca construir una base técnica confiable y minimizar los riesgos asociados al proyecto.

Construcción: Durante esta etapa se desarrolla el sistema, incluyendo la programación, pruebas y corrección de errores. El objetivo es asegurar que el sistema cumpla con los requisitos establecidos y funcione correctamente.

Transición: En esta fase se entrega el sistema al cliente, realizando la instalación, capacitación y soporte necesarios. Se busca garantizar una implementación sin contratiempos y que el sistema satisfaga las expectativas del usuario.

Metodología Ágiles

El desarrollo ágil surge como una respuesta al entorno cambiante y altamente incierto de los proyectos de software modernos. Agile se concibe como la capacidad de crear y responder rápidamente al cambio, centrándose en la entrega temprana y continua de valor al cliente (Lutwama et al., 2024). Este enfoque promueve la adaptabilidad a lo largo del ciclo de vida del proyecto, evitando especificaciones rígidas al inicio .

En este marco, los requisitos y diseños emergen de manera iterativa, promoviendo interacciones frecuentes con el cliente para asegurar que el producto final cumpla con sus expectativas (Ghimire y Charters, 2022). Existen diferentes tipos de metodologías entre las que se tiene:

Tabla 1. Principales metodologías ágiles

Metodología	Definición	Fases principales
Extreme Programming (XP)	XP es una metodología ágil centrada en la simplicidad, comunicación continua y retroalimentación constante, diseñada especialmente para equipos pequeños y entornos de cambio rápido (Tetteh, 2024).	Planificación
		Codificación
		Diseño
		Pruebas
Agile Unified Process (AUP)	AUP adapta el Rational Unified Process con principios ágiles, enfatizando iteraciones, modelado ágil, TDD y gestión de cambios para mejorar productividad (Brisk Tech Solutions, 2025).	Planificación
		Elaboración
		Construcción
		Transición (iterativo)
Rational Unified Process (RUP)	RUP es un marco iterativo e incremental dirigido a casos de uso y centrado en la arquitectura, estructurado en cuatro fases (Barrientos et al., 2022).	Inicio
		Elaboración
		Construcción
		Transición
RAD (Rapid Application Development)	RAD es un enfoque iterativo y centrado en prototipos rápidos y retroalimentación constante de usuarios, ideal para requisitos en evolución (Microsoft, 2022).	Requisitos
		Diseño rápido
		Construcción de prototipos
		Implementación

Fuente: Elaboración propia

2. METODOLOGÍA

El presente estudio se desarrolló a través de un enfoque cualitativo-descriptivo con apoyo de técnicas computacionales de minería de texto. Su objetivo principal fue identificar y analizar las metodologías de desarrollo de software implementadas en las propuestas tecnológicas desarrolladas por estudiantes de la carrera de Computación, específicamente en el campo de la Agromática. Para alcanzar este propósito, se estructuró un procedimiento dividido en varias fases consecutivas, desde la recopilación de documentos hasta el análisis e interpretación de los patrones metodológicos detectados.

Fase 1: Recolección y selección del corpus documental

La primera fase consistió en la recopilación sistemática de trabajos de titulación disponibles en el repositorio digital institucional de la universidad. Se establecieron como criterios de inclusión que

las tesis seleccionadas correspondieran al período comprendido entre enero de 2022 y mayo de 2024, que fueran elaboradas por estudiantes de la carrera de Computación, y que tuvieran una orientación temática clara hacia el desarrollo de soluciones tecnológicas aplicadas al sector agropecuario. Para asegurar la coherencia del análisis, se excluyeron todos aquellos documentos que no contuvieran una sección metodológica explícita, así como trabajos inconclusos, con formatos alterados o que presentaran problemas de lectura digital.

id	tema
1	APLICACIÓN MÓVIL PARA EL MONITOREO Y CONTROL DE L...
2	APLICATIVO MÓVIL PARA EL MONITOREO DE LA TURBIDEZ ...
3	APLICATIVO WEB Y MÓVIL PARA CONTROL Y SEGUIMIENTO ...
4	AUTOMATIZACIÓN DE RIEGO CON TECNOLOGÍA DE SENSORIZ...
5	DESARROLLO DE SISTEMA WEB Y MOVIL PARA LA IDENTIFI...
6	DETECCIÓN DE LA PLAGA LANGOSTA MEDIANTE VISIÓN POR...
7	DETECCIÓN DE LA PLAGA SOGATA EN EL CULTIVO DE ARRO...
8	DETECCIÓN DE OBJETOS EN MOVIMIENTOEN EL CULTIVO DE...
9	DISEÑO DE PROTOTIPO DE MÁQUINA SECADORA DE CACAO, ...
10	IMPLEMENTACIÓN DE PLATAFORMA WEB Y MÓVIL "ALERTA M...
11	IMPLEMENTACIÓN DE SISTEMA WEB PARA GESTIÓN DE PROC...
12	IMPLEMENTACIÓN DE UN APLICATIVO WEB Y MÓVIL PARA E...
13	IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTRO...
14	IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA LA G...
15	IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN D...
16	SISTEMA PARA EL CONTROL DE DATOS DEL PROCESO DE PR...
17	SISTEMA PARA EL CONTROL DE LA GESTIÓN DE PRODUCCIÓ...
18	SISTEMA PARA EL CONTROL DE LOS PROCESOS OPERATIVOS...
19	SISTEMA WEB PARA ASISTENCIA TÉCNICA EN EL MANEJO A...
20	ESTUDIO COMPARATIVO DE ALGORITMOS DE APRENDIZAJE A...

Figura 2. Registro de documentos en base de datos

Cada tesis fue descargada en formato PDF, verificándose previamente que su contenido estuviera completo y que el texto fuera extraíble. Se organizó una base de datos con la información básica de cada documento, como título, autor, año, con el fin de mantener un control del corpus durante todo el proceso de análisis.

Fase 2: Procesamiento de textos y segmentación de contenido

Una vez descargadas las tesis, se procedió a su conversión a texto plano mediante el uso de la herramienta PDFMiner, seleccionada por su capacidad para conservar la estructura jerárquica de los documentos. Este paso fue fundamental para facilitar la posterior segmentación de los contenidos.

```
# =====
# Proyecto: Análisis de metodologías de tesis
# Fases: Extracción, preprocesamiento, análisis y agrupamiento
# =====
!python -m spacy download es_core_news_sm
# Librerías necesarias
import os
import re
import pandas as pd
from pdfminer.high_level import extract_text
import spacy
from sklearn.feature_extraction.text import CountVectorizer
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar modelo de spaCy para procesamiento de texto
nlp = spacy.load("es_core_news_sm") # Modelo en español
```

Figura 3. Librerías necesarias para el proyecto

El siguiente paso consistió en la segmentación automática de los textos, centrando el análisis exclusivamente en la sección metodológica de cada trabajo. Para ello, se emplearon expresiones regulares diseñadas para localizar encabezados frecuentes como “3.2 Metodología” “Marco metodológico” “Diseño metodológico”. El texto delimitado entre estos encabezados y el siguiente apartado (como “Resultados”) fue extraído y almacenado en una base separada, generando un corpus metodológico listo para su análisis.

```
# -----
# Fase 2: Extracción sección metodológica
# -----
def extraer_metodologia(texto):
    # Buscar encabezados típicos
    patrones_inicio = r"(3.2 Metodología|Marco metodológico|Diseño metodológico)"
    patrones_fin = r"(4. RESULTADOS)"
    inicio = re.search(patrones_inicio, texto, re.IGNORECASE)
    fin = re.search(patrones_fin, texto, re.IGNORECASE)
    if inicio:
        start = inicio.end()
        end = fin.start() if fin else len(texto)
        return texto[start:end]
    return ""

df_tesis['metodologia'] = df_tesis['texto_completo'].apply(extraer_metodologia)

# -----

df_tesis.head()
```

	archivo	texto_completo	metodologia
0	TESIS1.pdf	1 \n\nUNIVERSIDAD AGRARIA DEL ECUADOR \n\nFACU... \n\nLa metodología Extreme Programming (XP) e...	

Figura 4. Extracción de Metodología en Dataframe

Fase 3: Limpieza y preparación del texto

El corpus metodológico fue sometido a un proceso de preprocesamiento lingüístico utilizando Python como lenguaje base, junto con las bibliotecas NLTK, spaCy y Scikit-learn. La limpieza incluyó varios pasos: eliminación de caracteres especiales, signos de puntuación, números no significativos, y palabras vacías (stopwords). Asimismo, se implementó una etapa de lematización para reducir las palabras a su forma base, garantizando así mayor consistencia en la identificación de patrones lingüísticos.

Una vez depurado el texto, se organizó en una estructura de datos tipo matriz, donde cada fila correspondía al contenido metodológico de una tesis y cada columna representaba una característica textual relevante (frecuencia de términos clave, presencia de ciertas frases, etc.).

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words = set(stopwords.words('spanish'))

def limpiar_texto(texto):
    texto = texto.lower()
    texto = re.sub(r'\d+', '', texto) # Eliminar números
    texto = re.sub(r'[^\w\s]', '', texto) # Eliminar puntuación
    doc = nlp(texto)
    lemas = [token.lemma for token in doc if token.text not in stop_words and token.is_alpha]
    return " ".join(lemas)

df_tesis['metodologia_limpia'] = df_tesis['metodologia'].apply(limpiar_texto)
df_tesis.head()
```

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

	archivo	texto_completo	metodologia	metodologia_limpia
0	TESIS1.pdf	1 \n\nUNIVERSIDAD AGRARIA DEL ECUADOR \n\nFACU...	\n\nLa metodologia Extreme Programming (XP) e...	metodologia extreme programming xp excelente e...

Figura 5. Limpieza del texto

Fase 4: Extracción de metodologías mediante minería de texto

El análisis de las metodologías de desarrollo de software se realizó combinando dos técnicas complementarias. En primer lugar, se efectuó un análisis de frecuencia de términos, identificando las menciones más recurrentes asociadas a enfoques metodológicos comunes, como *Scrum*, *Cascada*, *XP*, *RUP*, entre otros. Esta extracción se logró mediante la creación de un diccionario léxico de metodologías, el cual incluyó sinónimos y variantes frecuentes (por ejemplo, “Desarrollo en cascada” o “modelo en cascada”).

```

# Fase 4: Extracción de metodologías
# -----
metodologías = {
    # Ágiles
    "XP": ["xp", "extreme programming", "retroalimentación continua", "pruebas frecuentes"],
    "Scrum": ["scrum", "sprint", "iteraciones cortas", "daily meeting", "product backlog"],
    "Kanban": ["kanban", "tablero kanban", "flujo continuo", "visualización de tareas"],
    "AUP": ["aup", "agile unified process", "iterativo", "modelado ágil"],
    "DSDM": ["dsdm", "dynamic systems development method", "entregas incrementales", "prototipos"],
    "Lean Software Development": ["lean", "desarrollo lean", "eliminación de desperdicio", "entrega rápida"],

    # Tradicionales
    "RUP": ["rup", "rational unified process", "casos de uso", "iterativo incremental"],
    "Modelo de Prototipo": ["prototipo", "rapid prototyping", "entregas parciales", "validación temprana", "desarrollo prototipo"],
    "Modelo de Cascada": ["cascada", "waterfall", "etapas secuenciales", "flujo lineal"],
    "Spiral": ["espiral", "spiral", "gestión de riesgos", "iterativo con evaluación de riesgos"],

    # Híbridos
    "RAD": ["rad", "rapid application development", "prototipos rápidos", "retroalimentación constante"],
    "BMP": ["bmp", "business model process", "modelo de negocio", "proceso de negocio"],

    # Otros
    "Feature Driven Development (FDD)": ["fdd", "feature driven development", "modelado por características", "entregas iterativas"],
    "Crystal": ["crystal", "metodología crystal", "equipos pequeños", "adaptativa"],
}

def identificar_metodologías(texto):
    encontradas = []
    for mtd, patrones in metodologías.items():
        for p in patrones:
            if p in texto:

```

Figura 6. Extracción de metodologías

En segundo lugar, se aplicó un modelo de clasificación semántica basado en patrones léxicos. Este modelo consistió en reglas programadas que buscaban secuencias de palabras asociadas a características particulares de cada metodología (por ejemplo: “iteraciones cortas” para *Scrum*, o “etapas secuenciales” para *Cascada*). La identificación automática de estos patrones permitió clasificar cada tesis según la metodología principal utilizada, así como detectar menciones a metodologías secundarias o híbridas.

Fase 5: Sistematización y análisis descriptivo

Las metodologías encontradas en cada proyecto de titulación se almacenaron en una base de datos, el tema la metodología, y las palabras más utilizadas.

id	tema	metodologia	palabras_clave
1	APLICACIÓN MÓVIL PARA EL MONITOREO Y CONTROL DE L...	desarrollo prototipo	suelo, sistema, riego, aplicación, monitoreo, móvil, hum...
2	APLICATIVO MÓVIL PARA EL MONITOREO DE LA TURBIDEZ ...	desarrollo prototipo	sistema, cultivo, monitoreo, móvil, nivel
3	APLICATIVO WEB Y MÓVIL PARA CONTROL Y SEGUIMIENTO ...	AUP	seguimiento, agrotécnico, producción, móvil, web, contr...
4	AUTOMATIZACIÓN DE RIEGO CON TECNOLOGÍA DE SENSORIZ...	RUP	riego, cultivo, sensor, sensorización, automatización, ...
5	DESARROLLO DE SISTEMA WEB Y MOVIL PARA LA IDENTIFI...	RUP	sistema, enfermedad, cultivo, visión, web
6	DETECCIÓN DE LA PLAGA LANGOSTA MEDIANTE VISIÓN POR...	RUP	detección, cultivo, visión
7	DETECCIÓN DE LA PLAGA SOGATA EN EL CULTIVO DE ARRO...	XP	detección, cultivo, visión
8	DETECCIÓN DE OBJETOS EN MOVIMIENTO EN EL CULTIVO DE...	desarrollo prototipo	sistema, cámara, movimiento, cultivo, detección
9	DISEÑO DE PROTOTIPO DE MÁQUINA SECADORA DE CACAO, ...	RAD	prototipo, implementación, diseño
10	IMPLEMENTACIÓN DE PLATAFORMA WEB Y MÓVIL "ALERTA M...	XP	automatización, web, implementación, móvil
11	IMPLEMENTACIÓN DE SISTEMA WEB PARA GESTIÓN DE PROC...	XP	sistema, gestión, implementación, producción, web
12	IMPLEMENTACIÓN DE UN APLICATIVO WEB Y MÓVIL PARA E...	XP	implementación, producción, móvil, web, control
13	IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTRO...	desarrollo de prototipo	sistema, cultivo, monitoreo, implementación, control
14	IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA LA G...	XP	producción, sistema, gestión, implementación
15	IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN D...	XP	sistema, gestión, cultivo, implementación, web
16	SISTEMA PARA EL CONTROL DE DATOS DEL PROCESO DE PR...	XP	producción, sistema, control
17	SISTEMA PARA EL CONTROL DE LA GESTIÓN DE PRODUCCIÓN...	RUP	producción, sistema, gestión, control
18	SISTEMA PARA EL CONTROL DE LOS PROCESOS OPERATIVOS...	cascada	producción, sistema, cultivo, control
19	SISTEMA WEB PARA ASISTENCIA TÉCNICA EN EL MANEJO A...	XP	sistema, cultivo, web
20	ESTUDIO COMPARATIVO DE ALGORITMOS DE APRENDIZAJE A...	Desconocida	algoritmo, comparativo, aprendizaje

Figura 7. Resultados con metodologías

Finalmente, se construyó una matriz de frecuencia que relacionaba cada tesis con las metodologías identificadas, permitiendo detectar tendencias generales y patrones de uso. Esta matriz fue la base para generar interpretaciones sobre las estrategias metodológicas preferidas por los estudiantes y su alineación con los estándares contemporáneos del desarrollo de software.

No se aplicaron pruebas estadísticas inferenciales, dado que el propósito del estudio fue descriptivo y exploratorio. El análisis se limitó a la observación de patrones y la identificación de regularidades sin establecer relaciones causales ni generalizaciones poblacionales.

3. RESULTADOS Y DISCUSIÓN

El análisis de las metodologías de desarrollo identificadas en las propuestas tecnológicas, realizado sobre un total de 49 documentos de titulación, revela una marcada preferencia por Extreme Programming (XP), que representa el 54 % del total de los proyectos analizados. Esta metodología, caracterizada por su enfoque en la simplicidad, la comunicación constante y la retroalimentación continua, se posiciona como la más utilizada, reflejando una tendencia hacia enfoques ágiles y adaptativos. En segundo lugar, el Modelo de Prototipo alcanza un 13 %, evidenciando la importancia que los estudiantes otorgan a la validación temprana de ideas mediante prototipos funcionales. El Modelo de Cascada, de corte tradicional y secuencial, registra un 8 %, manteniendo una presencia moderada, generalmente en proyectos con requisitos bien definidos. El Agile Unified Process (AUP) concentra un 6 %, mientras que el Rational Unified Process (RUP), más estructurado

y orientado a casos de uso, se encuentra en el 5 % de los casos. Por su parte, el Rapid Application Development (RAD) y el Business Model Process (BMP) presentan porcentajes más bajos, con un 2 % cada uno, indicando una menor adopción de estos enfoques en el contexto académico analizado. Estos resultados muestran una clara inclinación hacia metodologías ágiles, sin dejar de lado modelos tradicionales o híbridos, adaptados a las necesidades de cada proyecto, como se puede constatar en la tabla adjunta.

Tabla 2. Frecuencia y porcentaje de archivos encontrados

Metodología	Cantidad de documentos	Porcentaje
XP	26	54%
RUP	3	5%
Modelo de Prototipo	6	13%
Modelo de Cascada	4	8%
AUP	3	6%
RAD	1	2%
BMP	1	2%
Total	49	100%

Los resultados obtenidos en este estudio evidencian una clara preferencia por las metodologías ágiles, especialmente Extreme Programming (XP), en los proyectos de titulación de estudiantes de Computación, representando el 54 % de los casos analizados. Esta tendencia se alinea con investigaciones recientes en América Latina que destacan la adopción de enfoques ágiles en el ámbito educativo. Por ejemplo, un estudio realizado en la región 7 del Ecuador identificó a Scrum como la metodología ágil más utilizada en proyectos de desarrollo de software (Armijos et al., 2024), resaltando la importancia de la capacitación y la adaptación organizacional para su implementación efectiva.

La adopción de metodologías ágiles en la educación superior ha mostrado beneficios significativos en el desarrollo de competencias técnicas y transversales en los estudiantes (De la Cruz, 2024). Investigaciones en Colombia han evidenciado que la implementación de metodologías ágiles, como Scrum y Kanban, en programas de Ingeniería de Sistemas, ha promovido una mayor interacción entre docentes y estudiantes, facilitando un aprendizaje más dinámico y colaborativo (Castro et al., 2020). Asimismo, en Ecuador, se ha observado que la combinación de metodologías ágiles con el desarrollo de habilidades blandas, como la comunicación y la gestión del tiempo, ha enriquecido la formación integral de los estudiantes (Guerrero, 2024).

El uso de técnicas de minería de texto para identificar las metodologías empleadas en los proyectos de titulación ha permitido una visión más objetiva y sistemática de las prácticas metodológicas adoptadas por los estudiantes (Charitopoulos et al., 2025). Este enfoque ha facilitado la clasificación de las metodologías en principales, secundarias e híbridas, proporcionando una base sólida para futuras investigaciones y mejoras en los procesos educativos. Es relevante destacar que, aunque las metodologías ágiles predominan en los proyectos analizados, los modelos tradicionales, como el modelo en cascada, aún se aplican en proyectos con requisitos bien definidos y menos susceptibles a cambios. Este hallazgo sugiere que la elección de la metodología depende de las características específicas de cada proyecto, como la estabilidad de los requisitos y la complejidad del desarrollo.

Por ende, es importante incorporar las metodologías ágiles en la formación académica de los estudiantes de Computación, no solo para el desarrollo de software, sino también para el fortalecimiento de competencias transversales esenciales en el entorno profesional actual. La implementación de enfoques ágiles en los procesos educativos puede contribuir significativamente a la mejora de la calidad educativa y a la preparación de los estudiantes para enfrentar los desafíos del mercado laboral.

4. CONCLUSIONES

La preferencia por metodologías ágiles refleja una adaptación de los estudiantes a entornos dinámicos, donde los proyectos requieren iteraciones rápidas y retroalimentación constante, evidenciando una cultura de aprendizaje orientada a la experimentación y mejora continua.

El análisis mediante minería de texto demuestra que las herramientas computacionales pueden optimizar la evaluación de patrones metodológicos en grandes volúmenes de información académica, ofreciendo una perspectiva más objetiva sobre las tendencias en el desarrollo de proyectos tecnológicos.

La coexistencia de metodologías tradicionales y ágiles indica que los estudiantes desarrollan un criterio crítico para seleccionar la estrategia metodológica más adecuada según la complejidad, riesgos y objetivos del proyecto, lo que sugiere un proceso de toma de decisiones informado y contextualizado.

Los resultados del estudio subrayan la necesidad de fortalecer la integración de metodologías híbridas en la formación académica, promoviendo que los estudiantes sean capaces de combinar enfoques y herramientas de manera creativa, potenciando la innovación y la eficacia en sus proyectos de titulación.

CONFLICTO DE INTERESES

Los Autores declaran que no existe conflicto de intereses, o lo que corresponda.

CONTRIBUCIÓN DE AUTORÍA

En concordancia con la taxonomía establecida internacionalmente para la asignación de créditos a autores de artículos científicos (<https://credit.niso.org/>). Los autores declaran sus contribuciones en la siguiente matriz:

	Beltrán N.	Vásquez L.
Participar activamente en:		
Conceptualización	X	X
Análisis formal	X	X
Adquisición de fondos		X
Investigación	X	X
Metodología	X	
Administración del proyecto		X
Recursos	X	X
Redacción –borrador original	X	
Redacción –revisión y edición		X
La discusión de los resultados	X	X
Revisión y aprobación de la versión final del trabajo.	X	X

REFERENCIAS BIBLIOGRÁFICAS

Aleqabie, H. J., Sfoq, M. S., Albeer, R. A., & Abd, E. H. (2024). A Review Of TextMining Techniques: Trends, and Applications In Various Domains. *IJCSM*, 5(1). <https://doi.org/https://doi.org/10.52866/ijcsm.2024.05.01.009>

Armijos, L., Vélez, C., & Lojan, E. (2024). Estudio de la adopción de metodologías ágiles en proyectos de desarrollo de software en la región 7 del Ecuador. *Revista Espacios*, 45(4), 73-84. <https://doi.org/10.48082/espacios-a24v45n04p06>

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>

Barrientos, M., Zacca, G., Castro, M., Álvarez, D., Vidal, M., & Valdés, D. (2022). Metodología para el desarrollo del sistema Web para la gestión de los programas de maestría del Instituto “Pedro Kourí”. *Revista Cubana de Informática Médica*, 14(2). <http://scielo.sld.cu/pdf/rcim/v14n2/1684-1859-rcim-14-02-e537.pdf>

Barrientos, M., Zacca, G., Castro, P., Álvarez, D., Vidal, M., & Valdés, D. (2022). Metodología para el desarrollo del sistema Web para la gestión de los programas de maestría del Instituto “Pedro Kourí”. *Revista Cubana de Informática Médica*, 142. <https://doi.org/http://scielo.sld.cu/pdf/rcim/v14n2/1684-1859-rcim-14-02-e537.pdf>

Brisk Tech Solutions. (11 de Febrero de 2025). *Agile Unified Process – Methodology & Best Practices [2025]*. Brisk Tech Solutions: <https://brisktechsol.com/agile-unified-process/>

Castro, F., García, D., & Henao, C. (2020). *Metodologías ágiles aplicadas en la educación superior*. Modalidad: Reporte de práctica.

https://chat.iztacala.unam.mx/sites/chat.iztacala.unam.mx/files/2020-04/Metodolog%C3%ADas%20%C3%A1giles.pdf?utm_source=chatgpt.com

- Charitopoulos, A., Rangoussi, M., Metdas, D., & Koulouriotis, D. (2025). Text mining technologies applied to free-text answers of students in e-assessment. *Discover Computing*, 28(5). <https://doi.org/https://doi.org/10.1007/s10791-024-09496-9>
- De la Cruz, S. (2024). Implementación de Metodologías Ágiles en Proyectos Tecnológicos Educativos de Nivel Superior. *LATAM*, 5(5), 3942. <https://doi.org/https://doi.org/10.56712/latam.v5i5.2902>
- Ghimire, D., & Charters, S. (2022). The Impact of Agile Development Practices on Project Outcomes. *MDPI*, 1(3). <https://doi.org/https://doi.org/10.3390/software1030012>
- González, Y., MClenan, J., & Abrego, A. (2022). Los proyectos y las metodologías / Modelos de procesos Ciclos de vida de desarrollo de software. *Semilla Científica*, 3(3). <https://doi.org/ISSN: 2710-7574 ISSN Electrónico: L2710-7574>
- Guerrero, M. (2024). *Influencia de las metodologías ágiles como complemento en la formación y desarrollo de habilidades blandas en los estudiantes de la residencia intercultural universitaria Don Bosco*. Trabajo de titulación, Universidad Politécnica Salesiana.
- JW Wu, J. (2024). An Exploratory Study of V-Model in Building ML-Enabled Software: A Systems Engineering Perspective. *2024 IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI (CAIN)*. <https://doi.org/https://arxiv.org/abs/2308.05381>
- Klunder, J., Busch, M., Dehn, N., Karras, & Oliver. (2021). Towards Shaping the Software Lifecycle with Methods and Practices. *Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE)*.
- Lutwama, P., Dlulane, M., Pillay, T., Hassan, F., & Grobbelaar, S. (2024). Agile: advantages, disadvantages, enablers, and barriers. *Agile: advantages, disadvantages, enablers, and barriers*, 4, 35. <https://doi.org/https://doi.org/10.7166/35-4-3058>
- Merchán, N., Palma, E., & Poma, D. (2024). Comparación de metodologías ágiles para el desarrollo de software. *Investigar*, 8(1), 5052-5074. <https://doi.org/https://doi.org/10.56048/MQR20225.8.1.2024.4985-5052-5074>
- Microsoft. (2022). *Work with Rapid Application Development*. <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-rad-publishing>
- Pérez, S. (2023). *Administración de proyectos tecnológicos y recursos humanos por medio de metodologías convencionales y ágiles*. Tesis, Universidad Autónoma de México. <https://ru.dgb.unam.mx/bitstream/20.500.14330/TES01000850355/3/0850355.pdf>
- Piotrovskiy, Y. (Diciembre de 2025). *The Most Common Software Development Methodologies*. Yojji: <https://yojji.io/blog/the-most-common-software-development-methodologies>

Tetteh, S. G. (2024). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *Asian Journal of Research in Computer Science* , 17(5), 30-42.
<https://doi.org/https://doi.org/10.9734/ajrcos/2024/v17i5436>.