

Sistemas inteligentes como herramienta para el seguimiento de proyectos ágiles: una revisión sistemática de la literatura

Nicolás Tortosa¹, Noelia Pinto¹, César Acuña¹, Gabriela Tomaselli¹

¹Centro de Investigación Aplicada a TICS (CInApTIC). Universidad Tecnológica Nacional. Facultad Regional Resistencia.
Resistencia, Chaco, Argentina (3500)

{nicotortosa, ns.pinto, csr.acn, gabriela.tomaselli}@gmail.com

***Abstract.** Con la evolución de la Ingeniería de Software adaptándose a las necesidades en cada contexto, ha cambiado, también, la manera en que se obtienen sus productos. Se destaca, por ejemplo, la implementación de enfoques inteligentes al desarrollo de software, potenciando las ventajas de los campos de Inteligencia Artificial y de Ingeniería de Software. Teniendo en cuenta esto, el objetivo del presente trabajo es clasificar la evidencia empírica existente respecto a experiencias de seguimiento de proyectos de software ágiles usando sistemas inteligentes. Se presentan, entonces, los resultados obtenidos luego de realizar una revisión sistemática de la literatura junto al análisis cuantitativo que permitió arribar a una serie de conclusiones y trabajos futuros.*

1. Introducción

La gestión del conocimiento, en procesos de desarrollo de software, es un campo de la ingeniería de software que se ha ido estudiando en la actualidad, con el fin de aplicarlo en la consecución de la mejora de los procesos de software y por ende, en la calidad del producto final [D. P. S. Montes et.al. 2008]. De hecho, se observa un crecimiento importante en la necesidad de adopción de herramientas de software que, por un lado, permitan la gestión de conocimiento para la innovación y mejora de productos y procesos, y, por el otro, favorezcan la toma de decisiones ejecutivas para la renovación y adaptación de las organizaciones [J. Rodríguez García et. al. 2017].

Además, asociado a la gestión de proyectos de software, se destaca la popularidad ganada por el uso de enfoques ágiles [J. Y. González et. al. 2016], pues ofrece a los equipos el control de requerimientos variables, la gestión efectiva y eficaz de los grupos de trabajo y el involucramiento y empoderamiento del cliente dentro del proyecto. Sin embargo, no siempre los equipos de desarrollo de software están preparados para afrontar la implementación y adopción rápida y eficiente de un enfoque ágil, debido en primer lugar a hábitos obtenidos de sus métodos tradicionales, y en segundo lugar, por la falta de un verdadero entendimiento y conocimiento de los valores, principios, prácticas y procesos en los cuales se basan los enfoques ágiles [S. Ilieva et. al. 2004] [J. A. Livermore 2008].

En respuesta a ello, se ha presentado anteriormente el framework AQF, con el objetivo de ofrecer a los equipos una alternativa que permita, no solo el seguimiento de sus proyectos, sino también sea posible evaluar la calidad cuando se opta por trabajar con procesos ágiles de desarrollo de software [N. S Pinto 2020]. Dicho framework se

denomina AQF (Agile Quality Framework) y su versión actual está compuesta por un modelo, QuAM (Quality Agile Model) y por una herramienta de software que brinda soporte a dicho modelo, QuAGI (Quality AGIle). A la fecha se han llevado adelante diversas experiencias de validación, que permitieron obtener resultados respecto a la implementación de QuAGI como herramienta de seguimiento de proyectos ágiles y seguir mejorando el framework AQF para lograr, de forma incremental, una mejor herramienta [N. Pinto et. al. 2017][N. Pinto et. al. 2018].

Sin embargo, con estas mismas experiencias se ha observado que resulta necesario enriquecer el framework de forma tal de ofrecer una nueva herramienta que permita liberar de trabajo de monitorización manual al grupo de administradores de proyecto y, al mismo tiempo, ofrezca soporte a la toma de decisiones de directivos de las empresas proporcionando recomendaciones automáticas de acuerdo al avance de sus actividades.

En función a lo antes expuesto, surge la propuesta de diseñar un enfoque inteligente que, formando parte del framework AQF, permita recomendar acciones al equipo de forma tal de mejorar los niveles de calidad del proceso ajustando los aspectos del proyecto que sean necesarios [N. S Pinto 2021]. Por tanto, el objetivo de este artículo es obtener evidencia empírica sobre experiencias de implementación de sistemas inteligentes en la gestión de proyectos de software ágiles. Para ello se realiza una revisión sistemática de la literatura donde, además se busca clasificar los trabajos de investigación recientes del área. De ésta forma también se persigue establecer si la comunidad científica ha planteado inquietudes similares a las que motivan a este artículo en los últimos años.

El resto del documento se estructura de la siguiente forma: la sección 2 describe el estado del arte en base a los detalles del mapeo sistemático de la literatura llevado a cabo; la sección 3 muestra los resultados de la investigación; en la sección 4 se discuten los hallazgos de la sección anterior y su relación con los objetivos generales del trabajo; por último, la sección 5 resume las conclusiones a las que se llegó durante la realización de este trabajo y se presentan futuras líneas de investigación.

2. Estado del Arte

Cuando en el año 2004, Kitchenham publica el informe técnico [B. Kitchenham 2004] sobre revisiones sistemáticas de literatura –basado en investigaciones previas aplicadas al dominio de medicina-, el empleo de las RSL en las diversas comunidades científicas de Ingeniería de Software (IS) ha aumentado su frecuencia de uso para obtener evidencias mayoritariamente de estudios primarios, y en menor medida de secundarios. Mayoritariamente los investigadores han reutilizado los procedimientos y guías propuestas por Kitchenham en 2004 [J. Biolchini et. al. 2005], las cuales fueron revisadas primero en 2005 por Biolchini et al. [S. Ilieva et. al. 2004], y luego actualizadas posteriormente [B. Kitchenham et. al. 2010].

La revisión sistemática de la literatura, en oposición a la técnica de mapeo sistemático de la literatura, es usada para descubrir, evaluar e identificar toda la evidencia presente en los artículos de investigación relevantes respecto a una pregunta específica de investigación [B. Kitchenham et. al. 2010]. La finalidad es afirmar que la revisión de la literatura es objetiva, estricta y auditable. No obstante, tiene como primordial problema el esfuerzo requerido para su ejecución.

De acuerdo al objetivo planteado en este trabajo, se decide llevar adelante una revisión sistemática de la literatura de manera tal de hallar investigaciones publicadas

recientemente, su relación con la adopción de enfoques inteligentes en proyectos de software y las experiencias de validación que se evidencian de estos sistemas. El resultado de este estudio presenta un punto de partida para la realización de futuras revisiones sistemáticas sobre las respuestas encontradas.

Para la realización del estudio se han llevado a cabo tres etapas principales: Planificación, Ejecución y Documentación. Las dos primeras etapas se describen en los siguientes apartados de la sección actual y la etapa de documentación se corresponde con la Sección 3.

2.1 Planificación

Se detallan a continuación las actividades llevadas a cabo en esta etapa de planificación de la revisión sistemática.

2.1.1 Preguntas de Investigación

Las preguntas de investigación se definen en función al paradigma de GQM (Goal-Question-Metric) [G. Caldiera et. al. 1994], estableciendo una meta a nivel conceptual para generar preguntas a nivel operacional que tendrán respuesta a través del proceso de revisión sistemática de la literatura. Por tanto, las preguntas de investigación que han guiado este estudio se muestra en la Tabla 1.

Tabla 1. Preguntas de Investigación

Preguntas de Investigación	Motivación
Q1. ¿Qué propuestas existen sobre sistemas inteligentes para el seguimiento de proyectos ágiles de software?	Clasificar las propuestas existentes desde el punto de vista de su contexto aplicable.
Q2. ¿Qué tipos de estudios empíricos han sido utilizados?	Determinar qué tipo de estudio empírico se ha utilizado para la validación.
Q3. ¿Qué tipos de sistemas inteligentes son los utilizados en las propuestas estudiadas?	Determinar cuáles son los tipos de enfoques más referenciados en las propuestas existentes
Q4. ¿Qué enfoques ágiles son los referenciados en las propuestas estudiadas?	Identificar enfoques o prácticas ágiles usadas en las propuestas.

2.1.2 Fuente de datos y estrategia de búsqueda

Para realizar el proceso de búsqueda se seleccionaron las siguientes fuentes digitales (Nombre[acrónimo]:website):

- Google Scholar
[GOOGLE]: <https://scholar.google.com.ar/>
- Springer Link
[SPRINGER]: <https://link.springer.com/>
- ACM Digital Library
[ACM]: <http://portal.acm.org/>
- IEEEExplore
[IEEEEX]: <http://ieeexplore.ieee.org/>

La inclusión de Google Scholar puede resultar redundante dentro del listado citado

anteriormente, pues indexa un gran número de fuentes de documentación técnica, tales como ACM, IEEE, entre otras, significativas en la Ingeniería de Software. Sin embargo, no puede excluirse porque tal como lo afirma Noruzi en [A. Noruzi 2005] se considera líder entre los motores de búsqueda para la comunidad de investigación académica y científica, incluyendo artículos de fuentes que no aparecen disponibles en otras bibliotecas científicas digitales.

Continuando con el protocolo para ejecutar el proceso de revisión sistemática, se deben definir un conjunto de términos para diseñar las cadenas de búsqueda a ejecutarse en cada fuente. Para ello, y a partir de las preguntas de investigación planteadas, se definieron 3 palabras claves tomando como base el método PICOC (Population, Intervention, Comparison, Outcome, Context) [B. Kitchenham 2007] tal como se describe en la Tabla 2.

Tabla 2. Términos del PICOC

Término	Definición
Población	Equipos de Desarrollo de Software
Intervención	Propuestas que han servido para mejorar la implementación de prácticas ágiles
Factor de Estudio	Beneficios de contar con sistemas inteligentes para la mejora de implementación de prácticas ágiles .
Resultado	Grado de evidencia empírica existente en el campo

De acuerdo con lo anterior, se realizaron búsquedas exploratorias utilizando diferentes palabras claves (keywords) y así identificar su relevancia dentro del resultado obtenido. El objetivo de estas búsquedas fue definir los términos claves que formarían parte de las cadenas de búsqueda a elaborarse a continuación.

Así, las palabras claves definidas, en función a las preguntas de investigación y a lo expuesto en la Tabla 2, han sido:

Agile Software Development – Intelligent System.

Resulta necesario aclarar que las palabras se describen en inglés por ser el idioma predominante en la literatura.

A partir del listado de términos expuesto anteriormente, y teniendo en cuenta el conjunto de librerías digitales tomadas como fuentes de datos, se procede a la obtención de diferentes cadenas de búsqueda en función a la sintaxis utilizada por cada fuente. Para ello se hace uso de operadores lógicos tales como AND y OR, con el fin de lograr la cadena de búsqueda acorde en cada buscador académico.

Esta combinación y generación de cadenas de búsqueda, de acuerdo a la sintaxis recomendada por cada fuente, se expone en la Tabla 3.

Tabla 3. Cadenas de Búsqueda

Fuente	Cadena de Búsqueda
M1. Google Scholar	"agile software development" and "intelligent system"
M2. Springer	(agile software development) AND (intelligent system)
M3. ACM Library	[All: "agile software development"] AND [[All: "intelligent software"] OR [All: or] OR [All: "intelligent system"]]

M4. IEEE Explore	("All Metadata":agile software development) AND ("All Metadata":intelligent system)
------------------	---

2.1.3 Criterios de Selección de los Estudios Primarios

Para una selección efectiva de estudios se definieron, como se recomienda en [B. Kitchenham et. al. 2010], un conjunto de criterios de inclusión y exclusión, con el fin de determinar cuáles deberían ser incluidos en el proceso de análisis y cuáles no.

Por un lado, se tuvieron en cuenta los estudios que cumplan con al menos uno de los siguientes criterios de inclusión:

- IC1. Artículos científicos publicados desde el 2015 a 2021
- IC2. Artículos científicos que incluyan los términos de las cadenas de búsqueda en sus palabras claves

Por otro lado, se obviaron de la revisión aquellos estudios duplicados y que cumplan con alguno de los siguientes criterios de exclusión:

- EC1. Artículos científicos publicados en formato de poster, resumen o presentación
- EC2. Artículos científicos centrados en temas fuera del dominio de la ingeniería de software
- EC3. Artículos científicos que representen Revisión Sistemática o Mapeo Sistemático

2.1.4 Valoración de Calidad

Un aspecto fundamental para obtener mejores resultados, es evaluar la calidad de los estudios para establecer cuáles son los más representativos e importantes en el ámbito de la problemática que se aborda.

La valoración de la calidad de los estudios individuales es necesaria para limitar los sesgos, formarse una idea más precisa de las potenciales comparaciones y guiar la interpretación de los resultados [F. Cascaes da Silva et. al. 2013].

Por ello, para realizar la valoración de calidad de los artículos seleccionados en este estudio se utilizó, una evaluación de tipo cuantitativa, basada en una serie de preguntas que se analizan obteniendo una calificación numérica. El procedimiento de calificación usado en la evaluación se basó en un sistema de puntuación de tres valores (-1, 0 y +1):

- VC1. ¿El trabajo contiene una descripción detallada del sistema inteligente?
- VC2. ¿El trabajo explica la aplicación en proyectos ágiles de software?
- VC3. ¿El trabajo describe el proceso experimental de validación?
- VC4. ¿El trabajo presenta los resultados de la validación empírica?

Las posibles respuestas a cada una de las 4 preguntas de valoración de calidad son: “Sí” (+1), “Parcialmente” (0) y “No” (-1).

2.1.5 Extracción de datos

Las guías de Petersen et al. [K. Petersen et. al. 2008] sugieren un método de clasificación denominado Keywording, el cual permite reducir los tiempos para la definición de un esquema de clasificación y asegurar que la extracción de datos contempla los artículos más relevantes obtenidos durante el proceso de revisión sistemática.

Por ello, se ha asignado, a cada una de las preguntas de investigación definidas anteriormente, una serie de posibles respuestas (tal y como se puede observar en la Tabla 4), lo que permitirá aplicar los mismos criterios de extracción de datos a todos los estudios y clasificarlos en función de dichas respuestas.

TABLA 4. Categorización de respuestas en la extracción de datos

Preguntas de Investigación	Posibles respuestas
Q1. ¿Qué propuestas existen sobre sistemas inteligentes para el seguimiento de proyectos ágiles de software?	a. Teóricas b. Aplicaciones prácticas c. Modelo
Q2. ¿Qué tipos de estudios empíricos se han sido utilizados?	a. Estudio de caso b. Experimentos c. Otros
Q3. ¿Qué tipos de sistemas inteligentes son los utilizados en las propuestas estudiadas?	a. AOSE b. Sistemas Multi-Agentes c. Redes bayesianas d. Machine Learning e. Algoritmos Genéticos f. Diseño de ontologías
Q4. ¿Qué enfoques ágiles son los referenciados en las propuestas estudiadas?	a. SCRUM b. XP c. LEAN KANBAN d. HÍBRIDOS e. PROPIOS

Cabe destacar que para el proceso de extracción de datos han participado un total de 2 investigadores, trabajando sobre las mismas fuentes de datos y plasmando resultados en una planilla colaborativa online.

Luego para el análisis y discusiones, se sumaron 2 investigadores al equipo inicial para enriquecer los resultados al momento de realizar el análisis cualitativo de los estudios primarios.

2.1.5 Ejecución

Durante la fase de ejecución se ha llevado adelante la aplicación del protocolo de revisión sistemática definido en la etapa anterior, tal como se enumera a continuación:

- *Selección de estudios potenciales:* se introdujo la cadena de búsqueda sobre cada una de las fuentes de datos.
- *Selección de estudios primarios:* sobre los estudios potenciales se aplicaron los criterios de inclusión y de exclusión, eliminando los trabajos duplicados.
- *Evaluación de calidad y clasificación de los estudios primarios:* para cada uno de los estudios primarios se realizó una evaluación de calidad y una clasificación en base a las diferentes respuestas de cada una de las preguntas de investigación.

A continuación, se exponen los resultados obtenidos durante la fase de Ejecución de la Revisión, que han sido documentados en forma de gráficos estadísticos para mayor comprensión.

3. Resultados

En esta sección, se presentan los resultados de la ejecución del protocolo de revisión, tal

como ha sido planificado. A fines de claridad, resulta necesario destacar que el estudio se realizó entre los meses de marzo y mayo del 2021.

El detalle cuantitativo de resultados obtenidos luego de la primera ejecución de la búsqueda se discrimina en la Tabla 5, la cual incluye en su primera columna la fuente de dato y luego, en la segunda columna se muestra la cantidad de artículos encontrados. Como se puede ver se han obtenido un total de 673 estudios iniciales.

TABLA 5. Primera ejecución de cadenas de búsqueda

Fuente de búsqueda	Estudios encontrados
[M1] Google Académico	269
[M2] Springer	299
[M3] ACM Library	12
[M4] IEEE Explore	93

Luego se procedió a la aplicación de los criterios de inclusión sobre la selección anterior de estudios, obteniéndose, como se muestra en la Tabla 5, un total de 104 estudios relevantes de los 673 analizados lo que representa aproximadamente un 16% de relevancia.

A posteriori se eliminaron los estudios duplicados, reduciéndose el conjunto de artículos válidos a un total de 92.

Y sobre este conjunto final se aplicaron los criterios de exclusión ya enumerados en la sección 2.3, lográndose un total definitivo de 15 estudios primarios a analizar tal como se resume en la Tabla 6.

TABLA 6. Estudios Primarios

Fuente de búsqueda	Criterios de Inclusión	Criterios de Exclusión
[M1] Google Académico	20	5
[M2] Springer	25	5
[M3] ACM Library	4	1
[M4] IEEE Explore	15	4
Total	64	15

Como se observa en la Figura 1, casi un 20% de los estudios relevantes han sido considerados como estudios primarios para la fase de valoración de calidad, por considerarse en condiciones de ser incluidos en este trabajo de revisión.

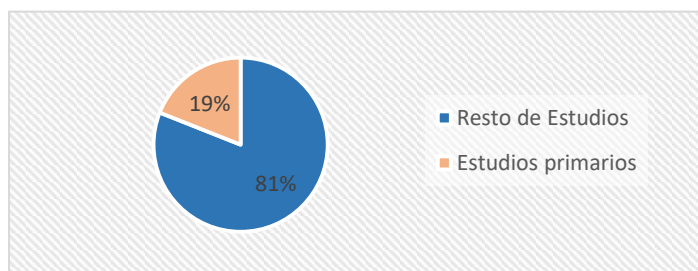


Figura 1. Porcentaje de estudios primarios obtenidos

En la Tabla 7, para mayor detalle, se presenta la lista de los 15 estudios primarios

considerados para la revisión sistemática y que fueron objeto de una valoración de calidad.

TABLA 7. Datos de los Estudios Primarios

Nro.	Título	Año
1	User and System Stories: An Agile Approach for Managing Requirements in AOSE	2021
2	User-story driven development of multi-agent systems: A process fragment for agile methods	2017
3	Ontology Based Multiagent Effort Estimation System for Scrum Agile Method	2017
4	Effort Prediction in Agile Software Development with Bayesian Network	2019
5	User Story based Information Visualization Type Recommendation System	2019
6	Predicting failures in agile software development through data analytics	2018
7	Empirical assessment of machine learning models for agile software development effort estimation using story points	2017
8	Feedforward and Feedbackward Approach-Based Estimation Model for Agile Software Development	2017
9	Chaos-Based Modified Morphological Genetic Algorithm for Effort Estimation in Agile Software Development	2020
10	Improving Agile Software Development with Domain Ontologies	2018
11	RefBot: intelligent software refactoring bot	2019
12	An Ontology-based Approach to Automate the Software Development Process	2018
13	A Framework For Transitioning Of Traditional Software Development Method To Distributed Agile Software Development	2019
14	The ontology-driven approach to intelligent support of requirements engineering in agile software development	2020
15	"An Intelligent Recommender and Decision Support System (IRDSS) for Effective Management of Software Projects"	2020

De acuerdo a lo especificado en la sección 2.4, se llevó a cabo el proceso de valoración de calidad, evaluando cada artículo en función a los criterios establecidos y obteniendo la calificación correspondiente.

A fin de mejorar la selección se estableció un umbral de calidad para los estudios primarios de forma tal que fueron descartados aquellos estudios con puntaje inferior a 3, es decir que hayan obtenido una valoración menor al 80% del mejor puntaje posible, que en este caso era 4.

El puntaje asignado a cada estudio primario por cada pregunta en función a la validación de la calidad y los estudios eliminados del conjunto (resaltados en rojo) puede verse en la Tabla 8.

TABLA 8. Resultados de valoración de calidad

Nro.	VC1	VC2	VC3	VC4	TOTAL
1	1	1	1	1	4
2	0	0	1	1	2
3	0	1	1	1	3
4	1	1	1	1	4
5	1	1	1	1	4
6	1	1	1	1	4
7	1	1	1	1	4
8	1	1	1	1	4
9	1	1	1	1	4
10	1	1	1	0	3
11	1	1	1	1	4
12	1	1	1	0	3
13	0	0	1	1	2
14	1	1	1	1	4
15	1	1	1	1	4

Como se observa, 13 estudios han logrado la valoración de calidad requerida para incluirlos en el análisis. Solo 2 estudios, han quedado fuera del análisis cuali-cuantitativo que se llevó adelante a posteriori.

A continuación, se incluye el análisis cuantitativo al que se arriba en este trabajo, seguido del análisis de amenazas que afectan al resultado final del estudio.

En la próxima sección, se profundiza a través de discusiones que caracterizan el análisis cualitativo llevado adelante en este proceso de revisión sistemática que se presenta.

3.1 Análisis Cuantitativo

Para resumir los datos extraídos de los artículos, se realiza una síntesis cuantitativa de la revisión sistemática, mediante el uso de gráficos y luego se realiza un análisis cualitativo de los artículos primarios obtenidos para dar respuesta a cada una de las preguntas de investigación.

En primer lugar, resulta necesario determinar el número actual de publicaciones y la tendencia en los últimos años respecto a enfoques inteligentes en agilidad. Así, la Figura 2 expone la cantidad de estudios primarios de acuerdo al año de publicación. Tal como se observa, hay un equilibrio respecto a trabajos desde el año 2017 a 2020. Asimismo, y de acuerdo a lo que ya se indicó anteriormente, el período en el que se ejecutó la revisión sistemática que aquí se presenta se extendió desde marzo a mayo del 2021, por lo que no todos los artículos de este año fueron recolectados.

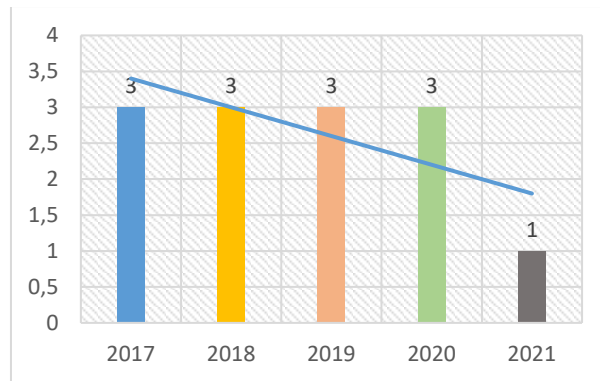


Figura 2. Cantidad de estudios primarios por año de publicación.

En segundo lugar, se describen los resultados obtenidos de acuerdo a las preguntas de investigación propuestas en la fase de planificación de este proceso de revisión sistemático de la literatura.

PREGUNTA Q1. ¿QUÉ PROPUESTAS EXISTEN SOBRE SISTEMAS INTELIGENTES PARA EL SEGUIMIENTO DE PROYECTOS ÁGILES DE SOFTWARE?

Para realizar el análisis desde el punto de vista de la pregunta de investigación Q1, se estableció una clasificación de las propuestas existentes, tal como se indica a continuación:

- *Teóricas*: Describen soluciones más generales en diferentes contextos, pero están restringidas a algún problema particular relacionado a proyectos ágiles de software
- *Aplicaciones prácticas*: Describen soluciones de aplicación práctica dentro del contexto de una organización en particular.
- *Modelo*: Describen arquitecturas de referencia que capturan el conocimiento de otras arquitecturas existentes y sirven de modelo para futuros diseños arquitectónicos.

Así, y como se observa en la Figura 3, aproximadamente el 43% de las propuestas hacen referencia a aplicaciones prácticas, desarrolladas e implementadas en diversos contextos. Respecto a propuestas teóricas, estas representan cerca del 36% de las evidencias consideradas en esta revisión sistemática de la literatura. Finalmente, un poco más del 20% hacen referencia a modelos, describiendo componentes que servirán de guía en el futuro para nuevas aplicaciones.

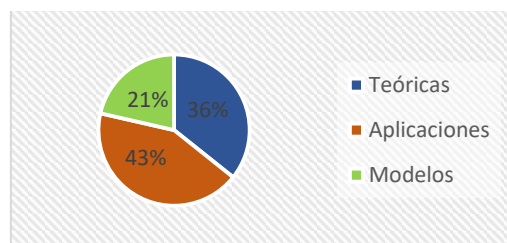


Figura 3. Clasificación de propuestas por tipo.

PREGUNTA Q2. ¿QUÉ TIPOS DE ESTUDIOS EMPÍRICOS HAN SIDO UTILIZADOS?

En relación con la evidencia empírica que existe sobre el uso de sistemas de agentes para

el seguimiento de proyectos de software cuyo desarrollo sea guiado por enfoques ágiles, se han considerado:

- *Estudios de Casos*: Los estudios de caso son un enfoque holístico que permite estudiar un fenómeno en su contexto real [P. Runeson and M. Höst 2008].
- *Experimentos*: El experimento controlado es el método donde existe mayor grado de control, permite estudiar relaciones causa-efecto, aislando variables con influencia significativa en un fenómeno determinado [S. Forrest et. al. 2008]. Se basa en ejemplos con datos ficticios que pueden estar relacionados a información histórica.
- *Otros*: Propuestas que no se encuadran en ninguno de los casos anteriores (diseño de ejemplos de validación, propuesta de experimentos, entre otros), que aún están en etapa temprana de validación.

Tal como se aprecia en la Figura 4, cerca de un 50% de los trabajos se han validado utilizando estudios de casos, es decir se han implementado en casos reales. Aproximadamente un 36% de los estudios ha diseñado una propuesta de validación a ser aplicada sobre casos ficticios, dando lugar a diversos experimentos. El resto, cerca del 14%, presenta propuestas de validación sin exponer aún los resultados finales de cada caso, algunas en curso y otras por realizarse.

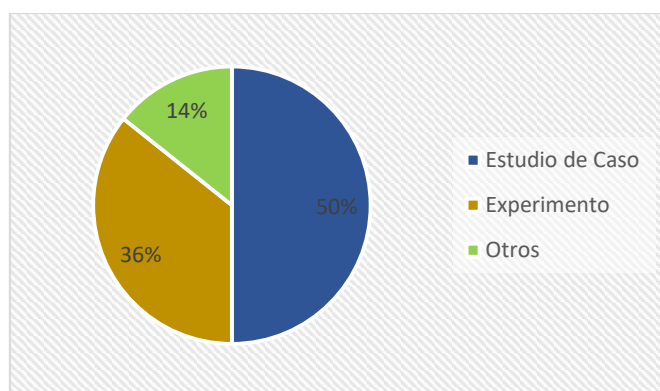


Figura 4. Clasificación por estudio empírico

PREGUNTA Q3. ¿QUÉ TIPOS DE ENFOQUES INTELIGENTES SON LOS UTILIZADOS EN LAS PROPUESTAS ESTUDIADAS?

Respecto al tipo de enfoque de sistema inteligente al que hace referencia cada uno de los 13 estudios analizados, se ha obtenido que la mayoría, cerca de un 35%, presenta propuestas a partir de sistemas multi-agentes. Seguido por estudios que abordan el desarrollo de propuestas usando Machine Learning, lo que representa aproximadamente el 30%, sobre el total de propuestas estudiadas. Cerca de un 14%, 2 de los 13 estudios, proponen diseños de ontologías previo al desarrollo de agentes sin hacer referencia ni a la arquitectura ni a la aplicación en sí misma. El resto de los estudios corresponden a propuestas que abordan soluciones usando el enfoque de AOSE (Agent Oriented Software Engineering), Redes Bayesianas y Algoritmos genéticos.

En la Figura 5, se representa gráficamente esta distribución por tipo de enfoque inteligente.

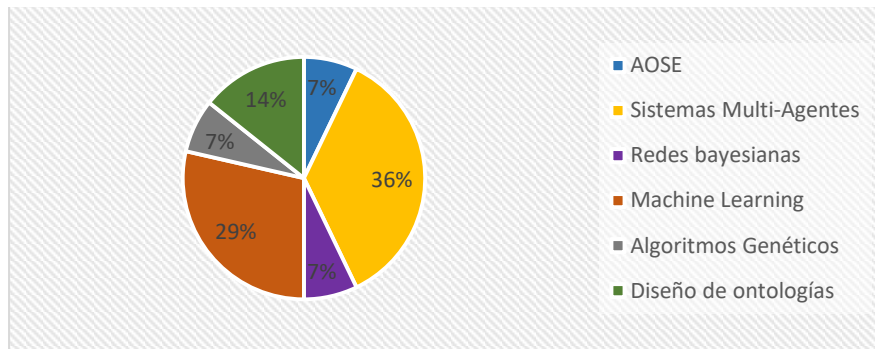


Figura 5. Distribución de estudios por tipo de enfoque.

PREGUNTA Q4. ¿QUÉ ENFOQUES ÁGILES SON LOS REFERENCIADOS EN LAS PROPUESTAS ESTUDIADAS?

Para el análisis de esta pregunta de investigación, resulta conveniente aclarar que varios estudios exponen resultados en base a más de 1 enfoque ágil, por lo tanto, se hace referencia también al total de propuestas en cada caso.

En consecuencia, como se aprecia en la Figura 6, 10 de los 13 estudios (lo que significa más de un 76% sobre el total) hacen referencia al enfoque SCRUM tanto para describir la propuesta como para cuestiones relacionadas a la validación posterior, convirtiéndose así, en el enfoque más utilizado en los estudios que se analizaron. En segundo lugar, se halla XP, con 5 estudios que utilizan este enfoque para describir su propuesta (lo que representa cerca de un 36%). En menor medida, se observan referencias de Lean Kanban (cerca del 14%), y 1 estudio presenta una propuesta híbrida, es decir que considera prácticas ágiles de diversos enfoques dando lugar a un enfoque adecuado a la realidad de la organización.

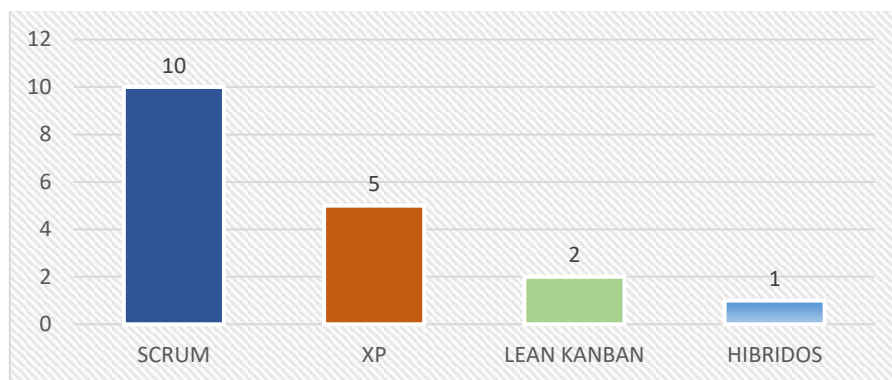


Figura 6. Clasificación por enfoques abordados

3.2 Análisis de Amenazas

Una revisión sistemática de la literatura y sus resultados pueden ser amenazado por la forma en que la investigación se lleva a cabo [K. Petersen, C. Gencel 2013]. Por lo general, la motivación para llevar a cabo estudios sistemáticos es que otros investigadores puedan reproducir los resultados obtenidos por el estudio inicial. En este trabajo se visualizan cuatro amenazas a la validez. las cuales se presentan a continuación.

VALIDEZ INTERNA

La validez interna hace referencia a la relación entre lo observado y la presentación de resultados [C. Wohlin et. al. 2012]. Esto asegura que las conclusiones obtenidas sean verdaderas. Esta amenaza se disminuye gracias a la definición de componentes recomendados por el protocolo, estableciendo criterios de inclusión y exclusión que permitan delimitar el espacio de investigación.

VALIDEZ EXTERNA

La validez externa hace referencia a la posibilidad de generalización de los resultados [C. Wohlin et. al. 2012]. Debido a que el trabajo representa un procedimiento sistemático, los resultados y las conclusiones generales que se obtienen presentan validez dentro del dominio de investigación abordado, permitiendo que los resultados del estudio puedan ser usados como un punto de partida para otras líneas de investigación.

VALIDEZ DE CONCLUSIÓN

La validez de conclusión observa la relación entre el proceso y el resultado final observado [C. Wohlin et. al. 2012]. Por ejemplo, un caso de amenaza a la validez de conclusión se puede dar en la extracción de datos. Para reducir el riesgo de esta amenaza se desarrolló un formulario de extracción de datos que asegura que los datos relevantes extraídos sean consistentes, y que puede observarse con mayor detalle en [“Resultados de la extracción de datos para el Mapeo Sistemático de la Literatura”. Disponible: <https://bit.ly/MSLAgentes>].

Luego, las clasificaciones utilizadas para el análisis de preguntas se realizaron a través de Keywording. Esta técnica, recomendada por Petersen et. Al en [K. Petersen et. al. 2008], permite reducir el tiempo necesario para desarrollar el esquema de clasificación y garantizar que el esquema tenga en cuenta los estudios existentes. Este procedimiento se lleva a cabo en 2 pasos: primero, se leen los resúmenes hallados en la búsqueda exploratoria inicial y se extraen palabras y conceptos claves que reflejen la contribución del artículo. Luego, cuando se ha elegido un conjunto final de palabras clave, se pueden agrupar y utilizar para formar las categorías que se usarán en cada pregunta de investigación para clasificar los resultados. Existen dominios de clasificación que han quedado fuera de este estudio, pero que se dejan para una posterior investigación, si fuera necesario.

VALIDEZ DEL CONSTRUCTO

La validez del constructo garantiza que el estudio esté relacionado con el problema de investigación y que las fuentes seleccionadas sean relevantes [C. Wohlin et. al. 2012]. Una posible amenaza a la validez del constructo radica en la diferencia durante el proceso de selección de estudios primarios. Esta amenaza se mitiga siguiendo las recomendaciones establecidas en el protocolo de investigación y ordenando las etapas. Otra amenaza puede estar asociada a que la búsqueda y selección de artículos la realice un único investigador. Por ello, en este trabajo, formaron parte de la búsqueda y selección de estudio 2 investigadores; sumándose luego, en la etapa de discusiones, otras 2 personas más.

4. Discusiones

De la búsqueda exploratoria realizada de manera preliminar, se observa una tendencia creciente en publicaciones de propuestas de sistemas inteligentes para el soporte al seguimiento de proyectos de software. Esto demuestra el interés del tema dentro de la ingeniería de software en los últimos años.

4.1 Análisis Cualitativo

A continuación, se presenta el análisis cualitativo de aspectos que surgen desde la definición de cada una de las preguntas de investigación presentadas en la sección II.

PREGUNTA Q1. ¿QUÉ PROPUESTAS EXISTEN SOBRE SISTEMAS INTELIGENTES PARA EL SEGUIMIENTO DE PROYECTOS ÁGILES DE SOFTWARE?

Realizando un análisis más profundo a partir de la pregunta de investigación Q1, que explora propuestas disponibles en la literatura sobre sistemas inteligentes para el seguimiento de proyectos de software, se observa una mayoría importante del tipo “Aplicaciones prácticas” enfocadas en la estimación del esfuerzo cuando el ciclo es guiado por enfoques ágiles. Algunas de estas propuestas se describen, brevemente, a continuación.

En referencia a esto último, el trabajo presentado en [M. Adnan y M. Afzal 2017] propone un enfoque que busca mejorar la estimación del esfuerzo del software y la gestión del conocimiento de los proyectos de software al centrarse en el proceso y las prácticas de Scrum utilizando el modelo de ontología en un sistema de estimación de agentes múltiples. La aplicación motiva además a guardar regularmente un conocimiento tácito significativo de situaciones únicas en forma de lecciones aprendidas durante el desarrollo del proyecto. En esta propuesta, los agentes del sistema acceden a la base de conocimientos existente y realizan de manera autónoma sus actividades de inferencia utilizando la lógica de descripción según los requisitos especificados por el Scrum master y le responden con una estimación adecuada en forma de tiempo, recursos y lecciones aprendidas para el éxito de proyectos futuros.

Otro ejemplo de aplicación práctica se expone en [L. I. U. Xu 2019], donde a partir del uso de técnicas de aprendizaje automático para modelar los tipos de visualización existentes con las historias de usuario correspondientes, se diseña e implementa un prototipo de sistema de recomendación ReVizy para predecir el tipo de visualización recomendado para nuevas historias de usuario.

En el caso del trabajo referenciado en [S. Dhir et. al. 2017], se presenta el desarrollo de un framework que describe la estimación de un desarrollo de software en las etapas iniciales y consta de enfoques de retroalimentación y retroalimentación en todo el ciclo de desarrollo, utilizando para ello técnicas basadas en Redes Bayesianas. El artículo concluye con la implementación de la técnica de estimación mediante la implementación de un pequeño proyecto.

Una propuesta diferente se presenta en [S. Bilgaiyan et. al. 2020], donde sus autores comparten un sistema inteligente basado en algoritmos genéticos para resolver el problema de estimación del esfuerzo de desarrollo de software (SDEE) en proyectos

ágiles. El trabajo se basa en la morfología matemática (MM) que consiste en una neurona híbrida-artificial (perceptrón Dilation Erosion (DEP)) extendido desde el concepto de teoría de la red completa (CLT), sus autores modifican los modelos existentes realizando adaptaciones al enfoque ágil para la estimación en proyectos de software.

En [M. Hamid et. al. 2020], se propone un sistema inteligente de recomendación y apoyo a la toma de decisiones (IRDSS) que da soporte al Scrum master para estimar mejor un próximo proyecto de software en términos de costo, tiempo y recomendaciones de recursos humanos. La aplicación toma como base las mejores prácticas recolectadas de casos exitosos y datos históricos guardados.

Otro aspecto, que se destaca entre las aplicaciones propuestas en el conjunto de estudios primarios seleccionados para la revisión sistemática, se relaciona con la necesidad de automatizar procesos de refactorización de código, favoreciendo la integración continua. Así, un único trabajo contribuye a esto, es el caso de [V. Alizadeh et. al. 2019] donde se propone un bot de refactorización de software inteligente, llamado RefBot. Integrado en el sistema de control de versiones (por ejemplo, GitHub), el bot supervisa continuamente el repositorio de software y reacciona frente a determinados eventos del ambiente.

PREGUNTA Q2. ¿QUÉ TIPOS DE ESTUDIOS EMPÍRICOS HAN SIDO UTILIZADOS?

Para el análisis cualitativo de esta pregunta se considera las propuestas de validación que se destacan en los diferentes estudios seleccionados para la revisión sistemática.

Un primer estudio a resaltar es el caso de [S. Rodriguez et. al. 2021], el cual introduce un concepto novedoso, System Story, un método que define los requisitos desde la perspectiva del sistema inteligente. Estas System Story son refinamientos de las User Story y proporcionan asignaciones más intuitivas a los conceptos de los agentes en el diseño y la implementación. La propuesta teórica se valida con una comparación basada en características de trabajos relacionados recientes, y una evaluación preliminar del usuario en una simulación de drones de un estudio de caso simple de búsqueda y rescate. Se ha utilizado un esquema de estudio de caso, recopilando resultados que sirvan para validar la propuesta desde la perspectiva de uso por parte de participantes reales.

En [L. D. Radu 2019], se propone un modelo para la predicción ágil de proyectos de desarrollo de software utilizando redes bayesianas. El modelo busca ayudar a los equipos ágiles a obtener una mejor estimación del esfuerzo del software. Para validar el modelo, se usan conjuntos de ejemplos que consideran dos factores en el cálculo: la calidad del trabajo en equipo y las características de las user stories. A partir de allí se exponen diversos resultados que varían conforme varía el conjunto de valores ejemplos. Esta técnica de validación permite hallar la versión más estable para el modelo, teniendo en cuenta el conocimiento experto recolectado de proyectos anteriores.

Otro estudio a destacar es el que se presenta en [F. A. Batarseh, y A. J. Gonzalez 2018], donde se describe un modelo para predecir fallas de software en los próximos sprints de un proyecto ágil. El modelo tiene por objetivo guiar el desarrollo de un sistema inteligente basado en técnicas de regresión y métodos predictivos, que permitan estimar

dónde y qué tipos de fallas del proyecto de software es probable que ocurran.

PREGUNTA Q3. ¿QUÉ TIPOS DE ENFOQUES INTELIGENTES SON LOS UTILIZADOS EN LAS PROPUESTAS ESTUDIADAS?

Como resultado, se han relevado diversos enfoques inteligentes para el desarrollo de propuestas. Más allá del análisis cuantitativo, que permite obtener un panorama de las técnicas más utilizadas en el desarrollo de sistemas inteligentes para seguimiento de proyectos ágiles, en esta sección se exponen casos destacables con foco en el objetivo que motivó este estudio.

Por ejemplo, el estudio presentado en [S. M. Satapathy y S. K. Rath 2017] presenta un enfoque que permite comparar el rendimiento de diferentes técnicas de aprendizaje automático, tales como el árbol de decisión, el aumento de gradiente estocástico y el algoritmo de random forecast, en la evaluación de la estimación de proyectos ágiles de software usando la técnica de puntos de historia. Un mismo modelo combina y evalúa constantemente la precisión en la estimación, comparando resultados finales al aplicarse diferentes enfoques de aprendizaje automático. Para validar la propuesta trabajaron con equipos ágiles de desarrollo de software.

En el caso del trabajo que se referencia en [P. L. De Souza 2018], la propuesta utiliza ontologías de dominio en el desarrollo ágil de software para reducir la ambigüedad causada por el uso del lenguaje natural como lenguaje ubicuo para reportar historias de usuarios. Para validar la propuesta, sus autores exponen resultados de haber usado el enfoque inteligente de ontologías de dominio para la especificación de requerimientos de un caso de estudio trabajando con SCRUM y dos de los roles: developers y product owners.

PREGUNTA Q4. ¿QUÉ ENFOQUES ÁGILES SON LOS REFERENCIADOS EN LAS PROPUESTAS ESTUDIADAS?

Se ha observado, de manera particular, al momento de analizar esta pregunta de investigación, no solo los enfoques ágiles más utilizados en las propuestas presentadas sino también aquéllos estudios que se centran, más bien, en determinadas prácticas ágiles independientes del enfoque.

Un caso, se representa en el estudio referenciado en [K. Athiththan et. al. 2018], el cual describe una solución llamada Sponto, una aplicación para automatizar la producción de historias de usuario en base a ontologías, ofreciendo plantillas tanto a desarrolladores como a la comunidad que desee reutilizarlas. Si bien no se focaliza en un único enfoque ágil, ni en todo el ciclo de desarrollo, esta propuesta se destaca por la posibilidad de reducir el tiempo empleado en recrear artefactos ya utilizados en proyectos de similares características. La herramienta no solo favorece la producción de historias de usuario, sino que las utiliza como entradas para obtener otros artefactos útiles para el desarrollo de software (por ejemplo, scripts de base de datos, diagramas de Business Process Model (BPM), fragmentos de código JAVA, entre otros).

Otra aplicación que considera también la elicitación de requerimientos en proyectos ágiles de software, es la presentada en [M. Murtazina y T. Avdeenko 2020],

donde sus autores proponen una ontología de dominio para dar soporte, en contextos de Scrum, al product owner durante el proceso de ingeniería de requisitos en la gestión ágil de proyectos de software. La fortaleza de la propuesta radica en que integra la ontología de soporte de información para el proceso de ingeniería de requisitos en la gestión ágil de proyectos y la ontología de dominio de aplicación del producto software que se está desarrollando.

4.2 Limitaciones

A partir del análisis de los resultados obtenidos durante la ejecución de la revisión sistemática de la literatura, se hallaron, además, dos aspectos que representan limitaciones desde las contribuciones propuestas en los estudios primarios y las cuales se exponen a continuación.

FALTA DE PROPUESTAS INTEGRALES

De los 13 estudios primarios considerados, ninguno expone propuestas que abarquen todo el proceso de desarrollo, sino que se focalizan en alguna etapa en particular (mayoritariamente durante la elicitación de requerimientos), representando esto una limitante si se desea automatizar tareas que se ejecuten durante todo el proyecto ágil que se lleve adelante. De hecho, la mayoría de los estudios, si bien plantean una ampliación de la contribución presentada, parecen no haber tenido continuidad en la investigación.

FALTA DE ANÁLISIS DEL IMPACTO DE LA PROPUESTA EN LA CALIDAD DEL PROCESO

Si bien, cerca del 90% de los estudios presenta resultados de validación de sus propuestas; en ningún caso, se expone un análisis de impacto del uso del sistema en la calidad del proceso de desarrollo en cuestión ni el feedback por parte de los participantes, en cuanto a la mejora (o no) respecto al seguimiento del proyecto ágil.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Este artículo presentó los resultados de una revisión sistemática con el objetivo de evaluar la literatura disponible sobre sistemas inteligentes como soporte al seguimiento de proyectos ágiles de software y su vigencia en ámbitos de investigación. Se definieron los criterios de investigación, se presentó el método utilizado para realizar la búsqueda, los criterios de selección de los estudios primarios y se realizó el reporte.

A partir del trabajo llevado a cabo, se encontró evidencia en trabajos previos que confirman que la comunidad científica ha planteado inquietudes similares a las que motivan este artículo. Se puede concluir, entonces, que la mayoría de las propuestas, se abocan de forma exclusiva a la automatización asociada a una práctica ágil durante el proceso de desarrollo de software: la obtención de historias de usuario o la estimación del proyecto. Derivado de esto, se observa un importante vacío de investigación relacionado a la ausencia de propuestas que abarquen el uso de sistemas inteligentes dando soporte al seguimiento del proyecto ágil en su totalidad.

Asimismo, si bien se valora la publicación de resultados de validación, resulta insuficiente para concluir respecto al impacto de su uso en la calidad del proceso y desde

la perspectiva de los usuarios de cada propuesta. Esto debido a que ningún estudio hace referencia a la evaluación de mejoras que impacten en el proceso al incorporar sistemas inteligentes en el seguimiento de proyectos ágiles.

Como trabajos futuros, se prevé a partir de los resultados obtenidos en este trabajo, iniciar con el diseño del enfoque y comportamiento de un Sistema Multi-agente que permita asistir a los usuarios del framework AQF de forma automática en la evaluación de calidad de sus procesos ágiles. Esto implicará, además, identificar aquéllas actividades en las que, el equipo abocado al proyecto, requiera recomendaciones automáticas que surjan del seguimiento de sus actividades, las cuales muchas veces son afectadas por acciones en segundo plano que pasan desapercibidas e impactan negativamente en los niveles de calidad del proceso de desarrollo asociado.

Resultaría interesante, además, ampliar esta revisión sistemática, analizando la literatura gris que se pueda encontrar relacionada con iniciativas que no se encuadran en este ámbito científico y sí están presentes en el ámbito de la industria del software.

Agradecimientos

Este trabajo se enmarca en las actividades preliminares relacionadas con el proyecto de investigación y desarrollo “I-QuAGI: Un enfoque inteligente para la evaluación de calidad de procesos de software ágiles” (PID SIPPBRE0008092), correspondiente al Centro de Investigación Aplicada a TIC (CInApTIC) de la Universidad Tecnológica Nacional, Facultad Regional Resistencia, de la provincia del Chaco, Argentina.

Referencias

- A. Noruzi, “The New Generation of Citation Indexes”. *Libri*, 55, pp. 170-180, 2005 [Online]. Available: <http://dx.doi.org/10.1515/LIBR.2005.170>.
- B. Kitchenham, “Guidelines for performing systematic literature reviews in software engineering”, *Engineering*, 2007.
- B. Kitchenham, “Procedures for Undertaking Systematic Reviews”, Joint TR, Comp. Science Dep., Keele University (TR/SE-0401) and National ICT Australia Ltd, 2004.
- B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, y Linkman, S. “Systematic literature reviews in software engineering—a tertiary study”. *Information and software technology*, vol. 52, pp. 792-805, 2010.
- C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell y A. Wesslén “Experimentation in Software Engineering”, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- D. P. S. Montes, L. A. C. Gaviria y O. H. Franco, “Gestión del conocimiento en procesos de desarrollo desoftware: un marco de trabajo para apoyar a las MiPyMEs”, *Scientia et Technica*, vol. 23, pp 76-83, 2018.
- F. A. Batarseh, y A. J. Gonzalez, “Predicting failures in agile software development through data analytics. *Software Quality Journal*”, pp. 49-66, 2018.
- F. Cascaes da Silva, B. A. Valdivia Arancibia, R. da Rosa Iop, P. J. B. Gutierrez Filho y R. da Silva, “Escalas y listas de evaluación de la calidad de estudios científicos”,

- Revista Cubana de Información en Ciencias de la Salud (ACIMED), pp. 295-312, 2013.
- G. Caldiera, V.R. Basili y H.D. Rombach, “The Goal Question Metric Approach”, *Encyclopedia of Software Engineering*, 1994, pp- 528–532.
- J. A. Livermore, “Factors that significantly impact the implementation of an agile software development methodology”, *Journal of Software* , pp 31-36, 2008.
- J. Biolchini, P.G. Mian, A.C.C. Natali y G. Travassos, “Systematic Review in Software Engineering”, *Technical Report RT-ES 679-05. PESC, COPPE/UFRJ*, 2005.
- J. Rodríguez García, M. Aguilar Romero y N. Raudales García, “Una mirada breve al software para gestión del conocimiento”, *I+D Tecnológico*”, vol. 13, pp 31-39, Jun. 2017
- J. Y. González, C. P. Calvache y O. S. Gómez, “Estado del Arte de la Utilización de Metodologías Ágiles y Otros Modelos en Pymes de Software”, *Informática-XVI Convención y Feria Internacional*, 2016.
- K. Athiththan, S. Rovinsan, S. Sathveegan, N. Gunasekaran, K. S. A. W. Gunawardena y D. Kasthurirathna, “An Ontology-based Approach to Automate the Software Development Process”, *IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*., 2018
- K. Petersen, C. Gencel, “Worldviews, research methods, and their relationship to validity in empirical software engineering research”, *Proceedings - Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, 2013.
- K. Petersen, R. Feldt, y S. Mujtaba, “Systematic mapping studies in software engineering,” in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, 2008, pp. 1-10
- L. D. Radu, “Effort Prediction in Agile Software Development with Bayesian Networks”, *ICSOFT*, pp. 238-245, 2019.
- L. I. U. Xu, “User Story based Information Visualization Type Recommendation System”, *International Journal of Information Engineering & Electronic Business*, 2019.
- M. Adnan y M. Afzal, “Ontology based multiagent effort estimation system for scrum agile method”, *IEEE Access*, vol. 5, 2017.
- M. Hamid, F. Zeshan, A. Ahmad, F. Ahmad, M. A. Hamza, Z. A. Khan y H. Aljuaid, “An Intelligent Recommender and Decision Support System (IRDSS) for Effective Management of Software Projects”, *IEEE*, 2020.
- M. Murtazina y T. Avdeenko, “The ontology-driven approach to intelligent support of requirements engineering in agile software development”, *International Conference on Information Technology and Nanotechnology (ITNT)*, (pp. 1-6). May. 2020.
- N. Pinto, C. Acuña, N. Tortosa y B. Cabas Geat, “Evaluating Quality in Agile Developments. A first validation experience with NEA Software SMEs.”, *XXIII Congreso Argentino de Ciencias de la Computación*, 2017.

- N. Pinto, N. Tortosa, B. Cabas Geat, L. Ibáñez y C. Acuña, “Validación de la reingeniería aplicada sobre la primera versión de Agile Quality Framework”, XIX Simposio Argentino de Ingeniería de Software (ASSE)-JAIIO 47, 2018.
- N. S. Pinto, “Framework para la evaluación de calidad de proyectos ágiles de software”, Doctoral dissertation, Universidad Nacional de La Plata, 2020.
- N. S. Pinto, C. Acuña, N. Tortosa y G. P. Tomaselli, “i-QuAGI: aproximación a un enfoque inteligente para la evaluación de calidad de procesos ágiles de software”, XXIII Workshop de Investigadores en Ciencias de la Computación, Chilecito, La Rioja, 2021.
- P. L. De Souza, A. F. do Prado, W. L. de Souza, S. M. dos Santos Forghieri Pereira y L. F. Pires, “Improving Agile Software Development with Domain Ontologies”, *Information Technology – New Generations*, pp. 267–274, 2018
- P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2008.
- “Resultados de la extracción de datos para el Mapeo Sistemático de la Literatura” [Online]. Disponible: <https://bit.ly/MSLAgentes>
- S. Bilgaiyan, P. K. Panigrahi, y S. Mishra, “Chaos-Based Modified Morphological Genetic Algorithm for Effort Estimation in Agile Software Development. In *A Journey Towards Bio-inspired Techniques in Software Engineering*”, Springer, pp. 89-102, 2020.
- S. Dhir, D. Kumar y V. B. Singh, “Feedforward and Feedbackward Approach-Based Estimation Model for Agile Software Development. In *Advances in Computer and Computational Sciences*”, Springer, pp. 73-80, 2017.
- S. Forrest, J. Singer y D. I. K. Sjøberg, “Guide to Advanced Empirical Software Engineering”, Springer, 2008
- S. Ilieva, P. Ivanov y E. Stefanova, “Analyses of an agile methodology implementation”, *EuromicroConference, Proceedings 30th*, pp. 326-333, 2004.
- S. M. Satapathy y S. K. Rath, “Empirical assessment of machine learning models for agile software development effort estimation using story points”, *Innovations in Systems and Software Engineering*, pp. 191-200, 2017.
- S. Rodriguez, J. Thangarajah y M. Winikoff, “User and System Stories: an agile approach for managing requirements in AOSE”, 2021.
- V. Alizadeh, M. A. Ouali, M. Kessentini y M. Chater, “RefBot: Intelligent Software Refactoring Bot”, 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019