

Algorithms and Metrics in Effort Estimation and Impact on DevOps Project Management: A Systematic Mapping of the Literature

Algoritmos y métricas en la estimación del esfuerzo y su impacto en la gestión de proyectos DevOps: un mapeo sistemático de la literatura

Iliana Lizbeth Alvarado Lara¹
d23ce054@cenidet.tecnm.mx
Noé Alejandro Castro Sánchez¹
Blanca Dina Valenzuela Robles¹
René Santaolaya Salgado¹
Gabriel González Serna¹

¹*Ciencias Computacionales* Tecnológico Nacional de México/CENIDET Cuernavaca, Morelos, México

Resumen — En proyectos de desarrollo de software, la estimación del esfuerzo desempeña un papel crítico en el éxito o fracaso de éstos, ya que está intrínsecamente ligada al alcance, el costo y el tiempo. A pesar de numerosos esfuerzos por mejorar la precisión de las estimaciones, la mayoría de la literatura existente se centra en proyectos de desarrollo de software tradicionales, dejando un vacío en cuanto a la estimación del esfuerzo para la parte operativa en proyectos DevOps. Esta investigación se enfoca en identificar los algoritmos/métodos, métricas y procesos de validación que actualmente se han implementado para la estimación del esfuerzo. Dando como resultado diversas técnicas, incluyendo algoritmos basados en aprendizaje automático, estimación del tamaño de software, listas de verificación, técnicas de estimación ágil y marcos de trabajo. Estas técnicas han evaluado la precisión de las estimaciones utilizando una variedad de métricas, tales como la media de la magnitud del error relativo, mediana de la magnitud del error relativo, predicción en el nivel n , precisión estandarizada, error medio absoluto, error cuadrático, residuo medio absoluto, magnitud del error relativo, mediana del sesgo del error relativo equilibrado, error relativo, error absoluto, error cuadrático relativo raíz, error cuadrático medio raíz, precisión, error cuadrático medio, coeficiente de variación, media y desviación estándar. Estos estudios se han validado mediante encuestas, casos de estudio, historias de usuario, entrevistas, factores y proyectos extraídos de repositorios de datos como Cocomo81, Nasa 93, Maxwell, China y ISBSG. Los resultados de esta investigación tienen el potencial de aportar nuevos enfoques para mejorar el proceso de estimación en proyectos DevOps, llenando un vacío importante en la literatura actual sobre este tema.

Palabras clave — Estimación del esfuerzo, gestión de proyectos, DevOps, alcance, costo, tiempo.

Abstract — In software development projects, effort estimation is critical in their success or failure, as it is intrinsically linked to scope, cost, and time. Despite numerous efforts to improve estimation accuracy, most existing literature focuses on traditional software development projects, leaving a gap in effort estimation for the operational aspect of DevOps projects. This research aims to identify the algorithms/methods, metrics, and validation processes currently implemented for effort estimation, resulting in various techniques, including machine learning-based algorithms, software size estimation, checklists, agile estimation techniques, and frameworks. These techniques have assessed estimation accuracy using a variety of metrics such as mean relative magnitude error, median relative magnitude error, prediction at n level, standardized accuracy, mean absolute error, squared error, mean absolute residual, relative magnitude error, median balanced relative error bias, relative error, absolute error, square root relative squared error, square root mean squared error, accuracy, mean squared error, coefficient of variation, mean, and standard deviation. These studies have been validated through surveys, case studies, user stories, interviews, factors, and projects extracted from data repositories such as Cocomo81, Nasa 93, Maxwell, China, and ISBSG. The results of this research have the potential to provide new approaches to enhance the estimation process in DevOps projects, filling a significant gap in the current literature on this topic.

Keywords — Effort estimation, project management, DevOps, scope, cost, time.

1. INTRODUCCIÓN

La estimación, según McConnell, (2006), consiste en proyectar la duración y/o el costo de un proyecto. Esta definición adquiere importancia en el ámbito del desarrollo de software, donde la comunidad científica ha dedicado más de tres décadas a perfeccionar estas estimaciones; su relevancia es respaldada en el informe del "CHAOS Report 2020" (*The Standish Group - View News*, s. f.).

En Grimstad et al., (2006), Pincirolí, (2023) y en la Guía del PMBOK, (2017) se enfatiza que las estimaciones del esfuerzo son problemas graves en la industria del software. Cuando son demasiado pesimistas pueden perderse oportunidades de negocio, mientras que un exceso de optimismo puede ir seguido de pérdidas económicas significativas. Algunos autores argumentan que existen elementos que impactan y aumentan la complejidad de los proyectos, entre ellos se encuentran el alcance, costo y tiempo, también conocidos como la triple restricción y que dichos elementos forman parte de la estimación del esfuerzo para el desarrollo de software. Por lo tanto, es necesario definir al menos las referencias del proyecto en cuanto a los elementos mencionados, de modo que la ejecución del proyecto pueda ser medido y comparado con esas referencias y que se pueda gestionar el desempeño.

A través de los años se han propuesto una serie de cuestiones fundamentales que son cruciales para mejorar las estimaciones del esfuerzo de proyectos ágiles/DevOps. Es de suma importancia integrar cada uno de los diversos enfoques de solución, factores, métricas de precisión, algoritmos/métodos, valores de precisión y procesos de validación propuestos con el fin de comprender y mejorar el proceso de estimaciones del esfuerzo de proyectos ágiles/DevOps (Valenzuela et al., 2023).

Con el objetivo de abordar estas cuestiones fundamentales y proporcionar una comprensión integral de la estimación del esfuerzo en el contexto de proyectos ágiles/DevOps, el presente artículo sigue la siguiente estructura: en la segunda sección, se presenta un mapeo sistemático de la literatura sobre la estimación del esfuerzo en proyectos de software ágiles/DevOps, seguido por la tercera sección que presenta las respuestas a las preguntas de investigación enunciadas en la sección 2.2 de esta investigación y la cuarta sección presenta las conclusiones derivadas de ésta.

2. OBJETIVO Y MÉTODO DE LA INVESTIGACIÓN

2.1. OBJETIVO DE LA INVESTIGACIÓN

Esta investigación tiene como objetivo proporcionar una comprensión integral de la estimación del esfuerzo en el desarrollo de software ágil/DevOps y establecer una base sólida para proponer nuevos enfoques que mejoren este proceso crucial en el área de software.

2.2. METODOLOGÍA DE LA INVESTIGACIÓN

En este trabajo de investigación se siguió el mapeo sistemático propuesto por Kitchenham, el cual está diseñado para dar una visión general de un área de investigación mediante la clasificación y recuento de contribuciones en relación con las categorías de dicha clasificación (Kitchenham & Charters, 2007). Es importante destacar que un mapeo y una revisión bibliográfica sistemática comparten algunos puntos en común por ejemplo en la selección de estudios (Petersen et al., 2015). La Figura 1 muestra las actividades que se realizaron durante el proceso de esta investigación siguiendo las pautas sugeridas por Kitchenham.

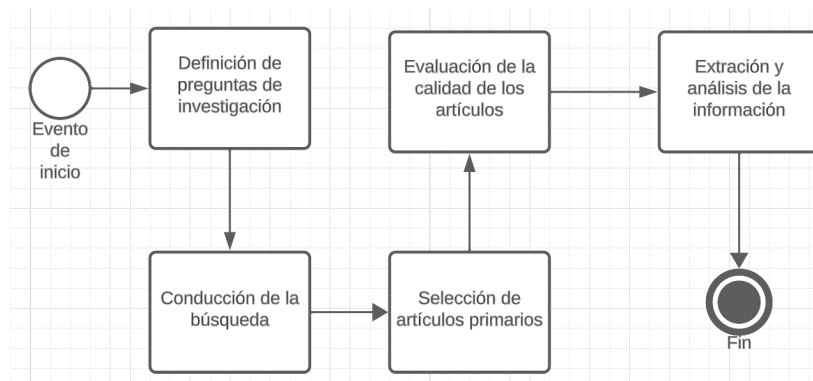


Figura 1. Proceso para el mapeo sistemático adaptado de [7].

2.2.1. Definición de preguntas de investigación

Se definieron tres preguntas de investigación para abordar el objetivo principal del estudio.

- PI1: ¿Qué algoritmos/métodos se han implementado para la estimación del esfuerzo en proyectos de software?
- PI2: ¿Qué métricas se han implementado para la estimación del esfuerzo en proyectos de software y qué precisión presentan los algoritmos implementados en la estimación del esfuerzo en proyectos de software?
- PI3: ¿Qué proceso de validación se ha implementado para cada algoritmo/método identificado?

2.2.2. Conducción de la búsqueda

El proceso de búsqueda de los estudios relevantes fue llevado a cabo mediante la siguiente cadena de búsqueda: (Devops OR Development OR Operation) AND ("triple constraint" OR "Effort estimation" OR "Predicting Effort" OR "Estimating Effort"). Esta cadena fue adaptada y ejecutada en Scopus y Google Académico. Para la realización de esta investigación, en la búsqueda se consideraron artículos dentro de una ventana de tiempo de seis años, es decir, de enero del 2018 a mayo del 2023. Se implementó el diagrama de PRISMA (Page et al., 2021), para representar el flujo del proceso de filtrado de artículos a través de las diferentes fases de una revisión sistemática y/o mapeo, la cual indica la cantidad de registros identificados, incluidos y excluidos, tal como se muestra en la Figura 2. La búsqueda fue realizada en las bases de datos de Google Académico y Scopus, lo que permitió encontrar las técnicas o métodos que se han utilizado para estimar el esfuerzo en el desarrollo de software con base en metodologías ágiles. Se obtuvieron 900 artículos para su revisión, de los cuales 810 son provenientes de Google Académico y 90 artículos de Scopus. En la primera fase se excluyeron 16 artículos que no cumplían con los criterios iniciales. En la siguiente fase se seleccionaron 848 artículos para la revisión de títulos y resúmenes de los cuales se excluyeron un total de 796 artículos. Del resto de artículos se hizo una revisión completa y de estos 52 artículos, 19 fueron excluidos por ser estudios secundarios. Finalmente, se incluyeron solamente 33 artículos que cumplieron con los criterios de inclusión y que responden a las preguntas de investigación.

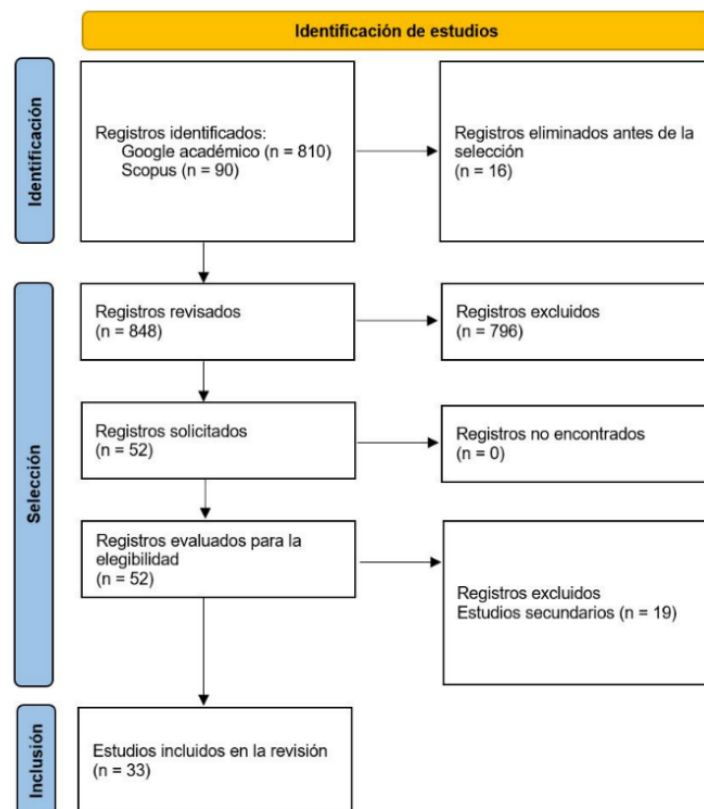


Figura 2. Flujograma del proceso de filtrado de artículos basado en PRISMA.

2.2.3. Selección de estudios primarios

Una vez obtenidos los estudios primarios potencialmente relevantes, se evalúa su pertinencia real. La **Figura 3. Criterios de inclusión y exclusión** muestra los criterios de inclusión y exclusión evaluados para recuperar la lista final.

Figura 3. Criterios de inclusión y exclusión	
Criterios de inclusión	Criterios de exclusión
Publicaciones por año: 2018-2023	Tipo de trabajo: libros, capítulos de libros, tesis, material didáctico.
Idioma: inglés	Tipo de búsqueda: revisión de la literatura, reportes, encuestas, estudios comparativos.
Título o Resumen: Contiene la cadena de búsqueda	Opiniones o artículos de debate.
Tipo de literatura: Estudios primarios	Artículos disponibles sólo en presentaciones o en resúmenes. Artículos no relevantes para las preguntas de investigación.

2.2.4. Evaluación de la calidad de los artículos

La selección de artículos se basó en una revisión de cuatro niveles:

- CI1: Revisión del título
- CI2: Revisión del resumen, introducción y conclusiones
- CI3: El texto completo del artículo debe estar disponible.
- CI4: Revisión sobre el texto completo para determinar si el estudio cumple con los criterios de inclusión.

Así mismo, se listan los criterios de evaluación que permiten validar la calidad de los estudios seleccionados:

- a) El artículo se enfoca en investigar factores, técnicas, estándares, buenas prácticas, modelos predictivos, y/o métodos para estimar el esfuerzo para proyectos de software ágil y/o DevOps.
- b) El artículo ofrece una descripción clara del problema de investigación.
- c) El artículo sigue un proceso de investigación estructurado y fundamentado.
- d) El artículo expone de manera clara y detallada los resultados obtenidos.
- e) El artículo ha sido publicado en una revista, conferencia o congreso relevante.
- f) El artículo ha sido citado por otros autores.
- g) El artículo describe claramente trabajos futuros o alternativas de investigación

La Figura 4 muestra el listado de artículos que fueron seleccionados a partir de los criterios de inclusión, exclusión y calidad para su análisis en esta investigación. Dichos estudios abarcan desde técnicas de aprendizaje automático hasta enfoques basados en métodos tradicionales de estimación, lo que refleja la diversidad de enfoques utilizados en la comunidad de desarrollo de software.

Figura 4. Artículos relacionados con la estimación de esfuerzo en el desarrollo de proyectos de Software		
Artículo	Referencia	Artículo
1	Abusaeed et al., (2023)	A fuzzy AHP-based approach for prioritization of cost overhead factors in agile software development
2	Adnan et al, (2019)	Ontology-oriented software effort estimation system for e-commerce applications based on extreme programming and scrum methodologies
3	Alshammari, (2022)	Cost estimate in scrum project with the decision-based effort estimation technique
4	Angara et al, (2020)	DevOps Project Management Tools for Sprint Planning, Estimation and Execution Maturity
5	Angara et al, 2018	Towards benchmarking user stories estimation with COSMIC function points: A case example of participant observation

6	Bhaskaran et al 2019	A hybrid effort estimation technique for agile software development (HEETAD)
7	Bilgaiyan et al, (2019)	Effort estimation in agile software development using experimental validation of neural network models
8	Butt et al, (2022)	A software-based cost estimation technique in scrum using a developer's expertise
9	Choetkiertikul et al, 2019	A deep learning model for estimating story points
10	Dantas et al, (2019)	An effort estimation support tool for agile software development: an empirical evaluation
11	Dave, (2021)	An efficient framework for cost and effort estimation of scrum projects
12	Gandomani et al, 2019	Planning poker in cost estimation in agile methods: averaging vs. consensus
13	Hacaloglu & Demirors, 2019	Measureability of functional size in agile software projects: Multiple case studies with COSMIC FSM
14	Kaushik et al., (2020)	A comparative analysis on effort estimation for agile and non-agile software projects using DBN-ALO
15	Khuat & Le, 2018	A novel hybrid ABC-PSO algorithm for effort estimation of software projects using agile methodologies
16	Mallidi & Sharma, (2021)	Study on agile story point estimation techniques and challenges
17	Meedeniya & Thennakoon, (2021)	Impact factors and best practices to improve effort estimation strategies and practices in DevOps
18	Prasada Rao et al., 2018	An agile effort estimation based on story points using machine learning techniques
19	Premalatha & Srikrishna, (2019)	Effort estimation in agile software development using evolutionary cost- sensitive deep belief network
20	Prykhodko & Prykhodko 2019	A multiple non-linear regression model to estimate the agile testing efforts for small web projects
21	Radu, (2019)	Effort prediction in agile software development with bayesian networks
22	Raslan & Ramadan, (2018)	An enhanced framework for effort estimation of agile projects
23	Rola & Kuchta, (2019)	Application of fuzzy sets to the expert estimation of scrum-based projects
24	Scott & Pfahl, 2018	Using developers' features to estimate story points
25	Sharma & Chaudhary, (2020)	Linear regression model for agile software development effort estimation
26	Sharma & Chaudhary, (2021)	Analysis of software effort estimation based on story point and lines of code using machine learning
27	Tanveer et al, (2019)	An evaluation of effort estimation supported by change impact analysis in agile software development
28	Tripathi, (2021)	Evaluation of the feasibility of parametric estimation in DevOps continuous planning
29	Usman et al., (2018)	Developing and using checklists to improve software effort estimation: a multi-case study
30	Vera-Rivera et al 2020	Automatic generation of sprint planning in agile software development, assignment of user stories to developers using genetic algorithms
31	Vetrò et al., (2018)	Combining data analytics with team feedback to improve the estimation process in agile software development
32	Vyas & Hemrajani, (2021)	Predicting effort of agile software projects using linear regression, ridge regression and logistic regression
33	Zakrani et al, 2018	Support vector regression based on grid-search method for agile software effort prediction

2.2.5.Extracción y análisis de la información

En la Figura 5 se muestra de manera detallada la clasificación conforme a los métodos y/o algoritmos utilizados en cada estudio seleccionado en el marco de esta investigación, así como las métricas implementadas para medir la precisión del método y/o algoritmo. Así mismo, se observa en la columna cuatro el resultado de la aplicación de dichas métricas para medir la precisión de las estimaciones realizadas con los diferentes algoritmos o métodos utilizados; esto ofrece una visión general de cómo se desempeñan las diferentes estrategias para la estimación del esfuerzo en el desarrollo de proyectos de software. Finalmente, en la columna cinco se muestran los procesos de validación que se han utilizado para cada algoritmo o método identificado, destacando que las diversas estrategias utilizadas abarcan desde proyectos, encuestas, casos de estudio, entrevistas hasta conjuntos de datos.

Figura 5. Algoritmos, métricas, precisión y proceso de validación identificados.

Estudio	Algoritmo/método	Métrica	Precisión	Proceso de validación
(Zakrani et al., 2018)	(SVR-RBF) Support Vector Regression with RBF kernel	MMRE	0.062	Conjunto de datos (21 proyectos)
		MdMRE	0.0426	
		Pred	(0.25%) 100	
	(SVM-RBF-GS) SVR-RBF optimized by Grid Search	MMRE	0.164	
		MdMRE	0.0893	
		Pred	(0.25%) 80.952	
(Angara et al., 2018)	COSMIC-FP	No reportado	No reportado	Encuestas y conjunto de datos (49 encuestas 11 proyectos)
(Hacaloğlu & Demirörs, 2019)	COSMIC FSM	No reportado	No reportado	Conjunto de datos (4 proyectos)
(Scott & Pfahl, 2018)	SVM (Support Vector Machines)	SA	46.334 - 93.152	(Conjunto de datos) 8 proyectos
		MAE	0.366 - 3.453	
(Khuat & Le, 2018)	ABC (artificial bee colony)	R2	0.9734	(Conjunto de datos) 21 proyectos
		MAR	3.12	
		MMRE	5.69	
		MdMRE	3.33	
		Pred	(0.08%) 66.67	
	PSO (particle swarm optimization)	R2	0.9732	
		MAR	3.15	
		MMRE	5.84	
		MdMRE	5.18	
		Pred	(0.08%) 61.9	
	ABC-PSO (integración de algoritmos)	R2	0.9626	
		MAR	3.66	
		MMRE	6.69	
		MdMRE	7.14	
(Prasada Rao et al., 2018)	Adaptive Neuro-Fuzzy Modeling	Pred	(0.08%) 61.9	Conjunto de datos (2 proyectos)
		MMRE		
		Costo	3.9587	
		Tiempo	19.5922	
		Costo	5.8700	
		Tiempo	3.2042	
		Costo	16.2868	
		Tiempo	6.6430	
		MMRE		
		Costo	3.9079	
		Tiempo	8.4277	
		Costo	4.8335	
	Generalized Regression Neural Network	Tiempo	2.7864	
		Costo	9.9604	
		Tiempo	8.0909	
		MMRE		
		Pred	57.1429	
		Costo	761905	
	Radial Basis Function Networks	Tiempo	76.905	
		Costo	76.1905	
		Tiempo	76.1905	
		Costo	76.1905	
		Tiempo	76.1905	
		Costo	76.1905	
		RE		
		Media	0.09	
(Gandomani et al., 2019)	Planning Poker	Consenso	0.03	Conjunto de datos (29 proyectos)
		AE		

Estudio	Algoritmo/método	Métrica	Precisión	Proceso de validación
		Media	42.5	
		Consenso	16.5	
(Choetkiertikul et al., 2019)	Deep learning: long short-term memory (LSTM) and recurrent highway network (RHN)	SA MAE	9.16 - 69.62 0.66 - 10.90	Conjunto de datos (16 proyectos)
(Vera-Rivera et al., 2020)	Algoritmos genéticos	No reportado	No reportado	2 casos de estudio
(Bhaskaran et al., 2019)	Puntos de función e historias de usuario	No reportado	No reportado	No reportado
(Prykhodko & Prykhodko, 2019)	Regresión lineal	R2 MMRE Pred	0.9847 0.0565 (0.25) 1.0	Conjunto de datos (40 proyectos)
	Regresión no lineal múltiple	R2 MMRE Pred	0.9818 0.0443 (0.25) 1.0	
(Dantas et al., 2019)	Aprendizaje automático: árboles de decisión	MRE	10.12 - 58.12	26 backlogs con 24 profesionales
(Tanveer et al., 2019)	Gradient Boosted Trees (GBT).	MAE MMRE	0.88 1.03	345 historias de usuario
	Planning Poker	Pred MAE MMRE Pred	(0.25) 0.5 0.72 1.55 (0.25) 0.5	
	Juicio de expertos	MAE MMRE Pred	0.22 0.5 (0.25) 75	
(Rola & Kuchta, 2019)	Números difusos	No reportado	No reportado	Conjunto de datos (2 proyectos)
(Bilgaiyan et al., 2018)	ANN-feedforward back-propagation neural	MSE MMRE Pred	0.052 0.1349 (0.25) 95.2301	Conjunto de datos (21 proyectos)
	Elman neural network	MSE MMRE Pred	0.056 0.1480 (0.25) 94.8659	
(Angara et al., 2020)	Machine learning: Naive Bayes Classifier	Coefficient e de variación Media Desviación estándar	14% - 49% 1.67 - 2.67 0.41 - 0.98	Conjunto de datos (21 proyectos)
(Sharma & Chaudhary, 2022)	Redes neuronales	MSE R2	6.220651 17.03561	Conjunto de datos (21 proyectos)
	Algoritmos genéticos	MSE R2	6.742859 21.46326	
(Kaushik & Tayal, 2019)	Deep belief network (DBN) con Antlion optimization (ALO)	MMRE MdmRE Pred	0.02 0.02 (0.25) 98	Conjunto de datos (4 conjuntos de datos ágiles (Cocomo 81, Nasa 93, Maxwell y China) 3 conjuntos de datos no ágiles (Zia, Company Dataset 1 y Company Dataset 2))
(Vyas & Hemrajani, 2021)	Regresión lineal	RMSE MMRE Pred	16.86 0.15 (0.25) 71.42	Conjunto de datos (21 proyectos)
	Regresión logística	RMSE MMRE Pred	14.06 0.19 (0.25) 71.42	

<i>Estudio</i>	<i>Algoritmo/método</i>	<i>Métrica</i>	<i>Precisión</i>	<i>Proceso de validación</i>
	Regresión Ridge	RMSE MMRE Pred	7.75 0.13 (0.25) 85.71	
(Dave, 2021)	Sistemas difusos Marco de trabajo	No reportado	No reportado	Conjunto de datos (ISBSG data sets Atlassian JIRA)
(Raslan et al., 2018)	Lógica difusa	MER MMER Pred	0.22 - 0.24 0.25 (0.25) 80	Conjunto de datos (10 proyectos)
(Tripathi, 2021)	Marco de trabajo	No reportado	No reportado	No reportado
(Alshammari, 2022)	Métodos difusos	MER	0.22	Encuestas
(Butt et al., 2022)	Estadística descriptiva	R2 Tiempo Costo	 0.092 0.016	30 cuestionarios
(Abusaeed et al., 2023)	Proceso de jerarquía analítica difusa Marco de trabajo	No reportado	No reportado	Conjunto de datos (154 proyectos)
(Vetrò et al., 2018)	Regresión lineal Coeficiente de correlación de Spearman	R2	0.00 - 0.02	Conjunto de datos (4 proyectos)
(Usman et al., 2018)	Listas de verificación	BRE	-0.05	No reportado
(Adnan et al., 2019)	Modelos ontológicos	Accuracy MMRE Pred	85% 0.091 (0.25) 0.85	Conjunto de datos (20 proyectos)
	Planning Pocker	Accuracy MMRE Pred	75% 0.131 (0.25) 0.75	
	Delphi	Accuracy MMRE Pred	55% 0.231 (0.25) 0.55	
(Mallidi & Sharma, 2021)	Planning Pocker Talla de camiseta Votación por puntos Sistema de cubos Grande / Incierto / Pequeño Método de Ordenación Dividir hasta el tamaño máximo o menos	No reportado	No reportado	No reportado
(Premalatha & Srikrishna, 2019)	Deep belief network (DBN)	RRSE RAE MSE RMSE Accuracy MAE MMRE Pred	17.21% 9.23% 0.0232 0.054 99.487% 0.023 6.1 (0.25) 100, (0.10) 100	Conjunto de datos (160 tareas (dataset 1: 40 tareas, dataset 2: 50 tareas, dataset 3: 70 tareas))
(Radu, 2019)	Redes bayesianas	No reportado	No reportado	No reportado
(Meedeniya & Thennakoon, 2021)	Modelo conceptual	R2	22%	25 entrevistas
(Sharma & Chaudhary, 2020)	Regresión lineal	R2 MMRE	0.9476 0.099	Conjunto de datos (21 proyectos)
	Árboles de decisión	MMRE	0.38	
	Potenciación del gradiente	MMRE	0.16	

Estudio	Algoritmo/método	Métrica	Precisión	Proceso de validación
	Bosque aleatorio	MMRE	0.25	

3. RESULTADOS

Como respuesta a las preguntas de investigación planteadas, a continuación, se muestran los resultados obtenidos del análisis realizado a los 33 trabajos seleccionados para el mapeo sistemático:

PI1: ¿Qué algoritmos/métodos se han implementado para la estimación del esfuerzo en proyectos de software?

A lo largo del tiempo, se han desarrollado taxonomías que facilitan la clasificación de los métodos de estimación del esfuerzo, lo que contribuye significativamente a una mejor comprensión del campo de investigación y permite deducir diversas implicaciones prácticas y valiosas. En este contexto destacan algunos autores ampliamente reconocidos por sus propuestas de taxonomías, que abordan diversas técnicas de estimación del esfuerzo, incluyendo modelos algorítmicos, juicio de expertos, analogías, contexto de la estimación y técnicas predictoras del esfuerzo (Yang et al., 2023), (Boehm, 1984), (Mendes, 1d. C.) y (Usman et al., 2017). De acuerdo con esta clasificación se agruparon los estudios de este análisis, en la que se listan los diversos algoritmos/métodos que se han propuesto por la comunidad científica para estimar el esfuerzo en el desarrollo de proyectos de software bajo metodologías ágiles (Zakrani et al., 2018), (Angara et al., 2018), (Hacaloğlu & Demirörs, 2019), (Scott & Pfahl, 2018), (Khuat & Le, 2018), (Prasada Rao et al., 2018), (Gandomani et al., 2019), (Choetkiertikul et al., 2019), (Vera-Rivera et al., 2020), (Bhaskaran et al., 2019), (Prykhodko & Prykhodko, 2019), (Dantas et al., 2019), (Tanveer et al., 2019), (Rola & Kuchta, 2019), (Bilgaiyan et al., 2018), (Angara et al., 2020), (Sharma & Chaudhary, 2022), (Kaushik & Tayal, 2019), (Vyas & Hemrajani, 2021), (Dave, 2021), (Raslan et al., 2018), (Tripathi, 2021), (Butt et al., 2022), (Abusaeed et al., 2023), (Vetrò et al., 2018), (Usman et al., 2018), (Adnan et al., 2019), (Mallidi & Sharma, 2021), (Premalatha & Srikrishna, 2019), (Radu, 2019), (Meedeniya & Thennakoon, 2021), (Sharma & Chaudhary, 2020).

PI2: ¿Qué métricas se han implementado para la estimación del esfuerzo en proyectos de software y qué precisión presentan los algoritmos implementados en la estimación del esfuerzo en proyectos de software?

En el marco de esta investigación sobre la estimación del esfuerzo en proyectos de desarrollo de software, es esencial evaluar y medir la precisión de los métodos utilizados. Para alcanzar una comprensión completa de la efectividad de estas técnicas, se han implementado diversas métricas diseñadas específicamente para cuantificar la precisión en la estimación del esfuerzo. Los resultados a esta pregunta se presentan en la Tabla III columna tres, los trabajos del [8]-[40] incluyen magnitud del error relativo, mediana de la magnitud del error relativo, predicción en el nivel n, precisión estandarizada, error medio absoluto, error cuadrático, residuo medio absoluto, magnitud del error relativo, mediana del sesgo del error relativo equilibrado, error relativo, error absoluto, error cuadrático relativo raíz, error cuadrático medio raíz, precisión, error cuadrático medio, coeficiente de variación, media y desviación estándar (Zakrani et al., 2018), (Scott & Pfahl, 2018), (Khuat & Le, 2018), (Prasada Rao et al., 2018), (Gandomani et al., 2019) y (Choetkiertikul et al., 2019), (Prykhodko & Prykhodko, 2019), (Dantas et al., 2019), (Tanveer et al., 2019) (Bilgaiyan et al., 2018), (Angara et al., 2020), (Sharma & Chaudhary, 2022), (Kaushik & Tayal, 2019), (Vyas & Hemrajani, 2021).

PI3: ¿Qué proceso de validación se ha implementado para cada algoritmo/método identificado?

Uno de los pilares fundamentales en cualquier investigación científica es la validación rigurosa de los métodos y resultados. En el contexto de esta investigación centrada en la estimación del esfuerzo en proyectos de desarrollo de software, se ha dado énfasis particular a la evaluación de los procesos de validación implementados en cada uno de los estudios considerados. Los procesos de validación aplicados en cada estudio se presentan en la Tabla III columna cinco, en donde se observa que los estudios fueron validados con conjuntos de datos, encuestas, casos de estudio y cuestionarios (Zakrani et al., 2018), (Angara et al., 2018), (Hacaloğlu & Demirörs, 2019), (Scott & Pfahl, 2018), (Khuat & Le, 2018), (Prasada Rao et al., 2018), (Gandomani et al., 2019), (Choetkiertikul et al., 2019), (Vera-Rivera et al., 2020), (Bhaskaran et al., 2019), (Prykhodko & Prykhodko, 2019), (Dantas et al., 2019), (Tanveer et al., 2019), (Rola & Kuchta, 2019), (Bilgaiyan et al., 2018), (Angara et al., 2020), (Sharma & Chaudhary, 2022), (Kaushik & Tayal, 2019), (Vyas & Hemrajani, 2021), (Dave, 2021), (Raslan et al., 2018), (Tripathi, 2021), (Butt et al., 2022), (Abusaeed et al., 2023), (Vetrò et al., 2018), (Usman et al., 2018), (Adnan et al., 2019), (Mallidi & Sharma, 2021), (Premalatha & Srikrishna, 2019), (Radu, 2019), (Meedeniya & Thennakoon, 2021), (Sharma & Chaudhary, 2020).

En la Figura 6 se destaca la frecuencia con la que se han aplicado las métricas para medir la precisión de las estimaciones, proporcionando una evaluación objetiva de la adecuación de las estimaciones a los resultados reales del proyecto.

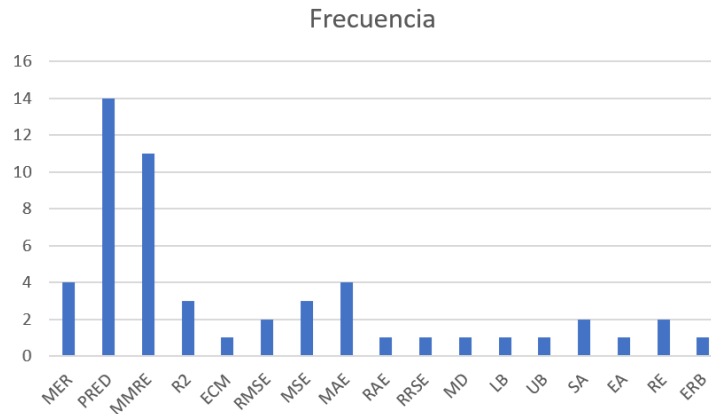


Figura 6.. Frecuencia de uso de las métricas implementadas en las investigaciones

4. DISCUSIÓN

De acuerdo con el análisis realizado en este mapeo sistemático de la literatura los procesos de validación tienen diferente número de proyectos, por lo que es difícil generalizar la validez de estos procesos.

Es importante señalar que Bhaskaran et al., (2019), Tripathi, (2021), Usman et al., (2018), Mallidi & Sharma, (2021) y Radu, (2019) no han reportado explícitamente su método de validación, lo que podría limitar la capacidad de generalizar los hallazgos en este mapeo, o bien puede sugerir la posibilidad de sesgo en la implementación del método para medir la precisión de las estimaciones.

Solo Kaushik & Tayal, (2019) proporciona información sobre el conjunto de datos utilizado para validar la investigación, que incluye Cocomo81, Nasa 93, Maxwell, China y ISBS. La falta de información sobre los criterios de los proyectos en los demás estudios dificulta la reproducción de las investigaciones y complica la comparación de los valores de precisión de diversos métodos o algoritmos.

La precisión (PRED) y la magnitud media del error relativo (MMRE) son las métricas más utilizadas en la evaluación de la precisión de las estimaciones en relación con los resultados reales. Siendo Deep Belief Network (DBN) con Antlion Optimization (ALO) el algoritmo implementado con la mejor precisión, arrojando un valor de MMRE=0.02; por otro lado, el método más frecuentemente sugerido para llevar a cabo la estimación del esfuerzo en entornos ágiles es el Planning Poker. Llegar a una conclusión definitiva acerca de la idoneidad de estos algoritmos o métodos para la estimación en entornos DevOps se presenta como un desafío significativo. Esto se debe a la falta de revelación de información detallada sobre las características fundamentales, tales como complejidad, tamaño o naturaleza de los proyectos, en los estudios existentes

5. CONCLUSIONES

En este estudio sobre la estimación del esfuerzo en proyectos de desarrollo de software ágiles/DevOps, se han abordado preguntas fundamentales que arrojan luz sobre la complejidad de este proceso esencial en la ingeniería de software. A través de un análisis detallado de los 33 estudios primarios seleccionados, de los cuales el 25% (8 artículos) son artículos de conferencias y el 75% (24 artículos) son revistas internacionales, esto indica la calidad de la investigación. Se obtuvieron valiosas respuestas que contribuyen al conocimiento y la mejora de la estimación del esfuerzo en este campo. Se identificó la diversidad de algoritmos que han sido implementados en la estimación del esfuerzo para el desarrollo de proyectos de software bajo metodologías ágiles, esto sugiere a los investigadores y profesionales utilizar diversas estrategias para abordar este desafío. También se identificaron las métricas que han sido implementadas para medir la precisión entre el valor estimado y el valor real del esfuerzo para el desarrollo de proyectos de software, además de las métricas que han sido utilizadas con mayor frecuencia por la comunidad científica. Al identificar los algoritmos que han sido implementados permitirá a la comunidad científica mejorar la precisión en la planificación, ayudando a establecer expectativas realistas sobre los tiempos estimados.

Este estudio aporta una comprensión profunda de la estimación del esfuerzo en el desarrollo de software ágil y proporciona una base sólida para mejorar las prácticas de estimación. La identificación de factores influyentes, la selección de métodos adecuados y la aplicación de métricas de precisión y procesos de validación son esenciales para avanzar hacia estimaciones más precisas y efectivas en el campo de la ingeniería de software. Se enfatiza que

la mayoría de la literatura existente se centra en proyectos de desarrollo de software tradicionales y ágiles dejando una gran brecha para el enfoque DevOps.

La precisión en la estimación del esfuerzo, respaldada por algoritmos confiables y métodos ágiles impacta directamente en la capacidad de la gestión de proyectos DevOps para cumplir con los objetivos de entrega rápida y eficiente. Este estudio da la pauta para realizar estimaciones continuas en un entorno ágil y dinámico como DevOps, adaptando o modificando los algoritmos o métodos basados en evidencias que arrojan una mejor o mayor precisión.

6. RECONOCIMIENTOS

Esta investigación fue apoyada por el Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) con la beca No. CVU: 328123. Los autores reconocen al Tecnológico Nacional de México el apoyo prestado para la realización del presente trabajo.

7. REFERENCIAS

- Abusaeed, S., Khan, S. U. R., & Mashkoor, A. (2023). A Fuzzy AHP-based approach for prioritization of cost overhead factors in agile software development. *Applied Soft Computing*, 133, 109977. <https://doi.org/10.1016/j.asoc.2022.109977>
- Adnan, M., Afzal, M., & Asif, K. (2019). Ontology-Oriented Software Effort Estimation System for E-commerce Applications Based on Extreme Programming and Scrum Methodologies. *The Computer Journal*, 62, 1605-1624. <https://doi.org/10.1093/comjnl/bxy141>
- Alshammari, F. (2022). Cost estimate in scrum project with the decision-based effort estimation technique. *Soft Computing*, 26, 1-13. <https://doi.org/10.1007/s00500-022-07352-w>
- Angara, J., Prasad, S., & Sridevi, G. (2018). Towards Benchmarking User Stories Estimation with COSMIC Function Points-A Case Example of Participant Observation. *International Journal of Electrical and Computer Engineering (IJECE)*, 8(5), Article 5. <https://doi.org/10.11591/ijece.v8i5.pp3076-3083>
- Angara, J., Prasad, S., & Sridevi, G. (2020). DevOps Project Management Tools for Sprint Planning, Estimation and Execution Maturity. *Cybernetics and Information Technologies*, 20(2), 79-92. <https://doi.org/10.2478/cait-2020-0018>
- Bhaskaran, N. A., Jayaraj, Dr. V., & Professor, School of Computer Science Engineering & Applications, Bharathidasan University Tiruchirappalli, India,. (2019). A Hybrid Effort Estimation Technique for Agile Software Development (HEETAD). *International Journal of Engineering and Advanced Technology*, 9(1), 1078-1087. <https://doi.org/10.35940/ijeat.A9480.109119>
- Bilgaiyan, S., Mishra, S., & Das, M. N. (2018). Effort estimation in agile software development using experimental validation of neural network models. *International Journal of Information Technology*, 11. <https://doi.org/10.1007/s41870-018-0131-2>
- Boehm, B. W. (1984). Software Engineering Economics. *IEEE Transactions on Software Engineering*, SE-10(1), 4-21. <https://doi.org/10.1109/TSE.1984.5010193>
- Butt, S. A., Khalid, A., Ercan, T., Ariza-Colpas, P. P., Melisa, A.-C., Piñeres-Espitia, G., De-La-Hoz-Franco, E., Melo, M. A. P., & Ortega, R. M. (2022). A software-based cost estimation technique in scrum using a developer's expertise. *Advances in Engineering Software*, 171, 103159. <https://doi.org/10.1016/j.advengsoft.2022.103159>
- Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T., Ghose, A., & Menzies, T. (2019). A Deep Learning Model for Estimating Story Points. *IEEE Transactions on Software Engineering*, 45(7), 637-656. <https://doi.org/10.1109/TSE.2018.2792473>
- Dantas, E., Costa, A., Vinicius, M., Perkusich, M., Almeida, H., & Perkusich, A. (2019). An Effort Estimation Support Tool for Agile Software Development: An Empirical Evaluation. 82-87. <https://doi.org/10.18293/SEKE2019-141>
- Dave, C. V. (2021). An Efficient Framework for Cost and Effort Estimation of Scrum Projects. *International Journal for Research in Applied Science and Engineering Technology*, 9(11), 1478-1487. <https://doi.org/10.22214/ijraset.2021.39030>
- Gandomani, T. J., Faraji, H., & Radnejad, M. (2019). Planning Poker in cost estimation in Agile methods: Averaging Vs. Consensus. *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 066-071. <https://doi.org/10.1109/KBEI.2019.8734960>
- Grimstad, S., Jørgensen, M., & Moløkken-Østfold, K. (2006). Software effort estimation terminology: The tower of Babel. *Information and Software Technology*, 48(4), 302-310. <https://doi.org/10.1016/j.infsof.2005.04.004>
- Guía del PMBOK. (2017). GUÍA DE LOS FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS (PMBOK). *Project Management Institute, Inc.*

- Hacaloğlu, T., & Demirörs, O. (2019). *Measurability of functional size in agile software projects: Multiple case studies with COSMIC FSM*. IEEE. <https://doi.org/10.1109/SEAA.2019.00041>
- Kaushik, A., & Tayal, D. (2019). A Comparative Analysis on Effort Estimation for Agile and Non-agile Software Projects Using DBN-ALO. *Arabian Journal for Science and Engineering*, 45. <https://doi.org/10.1007/s13369-019-04250-6>
- Khuat, T. T., & Le, M. H. (2018). A Novel Hybrid ABC-PSO Algorithm for Effort Estimation of Software Projects Using Agile Methodologies. *Journal of Intelligent Systems*, 27(3), 489-506. <https://doi.org/10.1515/jisys-2016-0294>
- Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2.
- Mallidi, R. K., & Sharma, M. (2021). Study on Agile Story Point Estimation Techniques and Challenges. *International Journal of Computer Applications*, 174(13), 9-14. <https://www.ijcaonline.org/archives/volume174/number13/31736-2021921014>
- McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*.
- Meedeniya, D., & Thennakoon, H. (2021, agosto 12). *Impact Factors and Best Practices to Improve Effort Estimation Strategies and Practices in DevOps*. <https://doi.org/10.1145/3484399.3484401>
- Mendes, E. (1d. C.). Cost Estimation Techniques for Web Projects. En <https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-135-3>. IGI Global. <https://www.igi-global.com/book/cost-estimation-techniques-web-projects/www.igi-global.com/book/cost-estimation-techniques-web-projects/207>
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Alonso-Fernández, S. (2021). Declaración PRISMA 2020: Una guía actualizada para la publicación de revisiones sistemáticas. *Revista Española de Cardiología*, 74(9), 790-799. <https://doi.org/10.1016/j.recsep.2021.06.016>
- Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1-18. <https://doi.org/10.1016/j.infsof.2015.03.007>
- Pincirol, F. (2023). *Strategies for agile software development based on technical and environmental complexity factors*. XXVIII Congreso Argentino de Ciencias de la Computación (CACIC) (La Rioja, 3 al 6 de octubre de 2022). <http://sedici.unlp.edu.ar/handle/10915/149423>
- Prasada Rao, Ch., Siva Kumar, P., Rama Sree, S., & Devi, J. (2018). An Agile Effort Estimation Based on Story Points Using Machine Learning Techniques. En V. Bhateja, J. M. R. S. Tavares, B. P. Rani, V. K. Prasad, & K. S. Raju (Eds.), *Proceedings of the Second International Conference on Computational Intelligence and Informatics* (pp. 209-219). Springer. https://doi.org/10.1007/978-981-10-8228-3_20
- Premalatha, H., & Srikrishna, C. (2019). Effort Estimation in Agile Software Development using Evolutionary CostSensitive Deep Belief Network. *International Journal of Intelligent Engineering and Systems*, 12, 261-269. <https://doi.org/10.22266/ijies2019.0430.25>
- Prykhodko, N. V., & Prykhodko, S. B. (2019). A MULTIPLE NON-LINEAR REGRESSION MODEL TO ESTIMATE THE AGILE TESTING EFFORTS FOR SMALL WEB PROJECTS. *Radio Electronics, Computer Science, Control*, 2, Article 2. <https://doi.org/10.15588/1607-3274-2019-2-17>
- Radu, L.-D. (2019). Effort Prediction in Agile Software Development with Bayesian Networks: *Proceedings of the 14th International Conference on Software Technologies*, 238-245. <https://doi.org/10.5220/0007842802380245>
- Raslan, A., Darwish, N., & Cairo University. (2018). An Enhanced Framework for Effort Estimation of Agile Projects. *International Journal of Intelligent Engineering and Systems*, 11(3), 205-214. <https://doi.org/10.22266/ijies2018.0630.22>
- Rola & Kuchta. (2019). Application of Fuzzy Sets to the Expert Estimation of Scrum-Based Projects. *Symmetry*, 11, 1032. <https://doi.org/10.3390/sym11081032>
- Scott, E., & Pfahl, D. (2018). Using developers' features to estimate story points. *Proceedings of the 2018 International Conference on Software and System Process*, 106-110. <https://doi.org/10.1145/3202710.3203160>
- Sharma, A., & Chaudhary, N. (2020). Linear Regression Model for Agile Software Development Effort Estimation. *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 1-4. <https://doi.org/10.1109/ICRAIE51050.2020.9358309>
- Sharma, A., & Chaudhary, N. (2022). Analysis of Software Effort Estimation Based on Story Point and Lines of Code using Machine Learning. *International Journal of Computing and Digital Systems*, 12(1), 131-140. <https://doi.org/10.12785/ijcds/1201012>

- Tanveer, B., Vollmer, A. M., Braun, S., & Ali, N. (2019). An evaluation of effort estimation supported by change impact analysis in agile software development. *Journal of Software: Evolution and Process*, 31. <https://doi.org/10.1002/smr.2165>
- The Standish Group—View News. (s. f.). Recuperado 15 de enero de 2024, de <https://www.standishgroup.com/news/45>
- Tripathi, D. D. R. (2021). Evaluation of the Feasibility of Parametric Estimation in Devops Continuous Planning. *International Journal for Research in Applied Science and Engineering Technology*, 9(9), 542. https://www.academia.edu/52397035/Evaluation_of_the_Feasibility_of_Parametric_Estimation_in_Devops_Continuous_Planning
- Usman, M., Börstler, J., & Petersen, K. (2017). An Effort Estimation Taxonomy for Agile Software Development. *International Journal of Software Engineering and Knowledge Engineering*, 27, 641-674. <https://doi.org/10.1142/S0218194017500243>
- Usman, M., Petersen, K., Börstler, J., & Santos Neto, P. (2018). Developing and using checklists to improve software effort estimation: A multi-case study. *Journal of Systems and Software*, 146, 286-309. <https://doi.org/10.1016/j.jss.2018.09.054>
- Valenzuela, B., Alvarado, L., Santaolaya, R., & Reyes, H. (2023). Identification of methods, approaches, and factors in effort estimation for DevOps projects: A systematic literature mapping. *2023 Mexican International Conference on Computer Science (ENC)*.
- Vera-Rivera, F., Barbosa-Mora, J., & Gaona, M. (2020). Generación automática de la planificación de la entrega en desarrollo de software ágil, asignación de historias de usuario a los desarrolladores usando algoritmos genéticos. *Aibi revista de investigación, administración e ingeniería*, 8, 29-38. <https://doi.org/10.15649/2346030X.735>
- Vetrò, A., Dürre, R., Conoscenti, M., Fernández, D. M., & Jørgensen, M. (2018). Combining Data Analytics with Team Feedback to Improve the Estimation Process in Agile Software Development. *Foundations of Computing and Decision Sciences*, 43(4), 305-334. <https://doi.org/10.1515/fcds-2018-0016>
- Vyas, M., & Hemrajani, N. (2021). *PREDICTING EFFORT OF AGILE SOFTWARE PROJECTS USING LINEAR REGRESSION, RIDGE REGRESSION AND LOGISTIC REGRESSION*. 13(2).
- Yang, X., Liu, J., & Zhang, D. (2023). A Comprehensive Taxonomy for Prediction Models in Software Engineering. *Information*, 14(2), Article 2. <https://doi.org/10.3390/info14020111>
- Zakrani, A., Najm, A., & Marzak, A. (2018). Support Vector Regression Based on Grid-Search Method for Agile Software Effort Prediction. *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, 1-6. <https://doi.org/10.1109/CIST.2018.8596370>

BIOGRAFÍA DE LOS AUTORES



M.C. Iliana Lizbeth Alvarado Lara recibió el grado de Maestra en Ciencias en Ciencias de la Computación en el Tecnológico Nacional de México campus Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) México. Actualmente es estudiante de doctorado en el CENIDET. Sus áreas de investigación son Gestión de proyectos, Estimación de proyectos de software, Ingeniería de software y Automatización de procesos de software.



Dra. Blanca Dina Valenzuela Robles, profesora del Tecnológico Nacional de México campus Centro Nacional de Investigación y Desarrollo Tecnológico CENIDET. Obtuvo el grado de Maestría y Doctorado en Ciencias de la Computación en el Tecnológico Nacional de México campus Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) México. Área de interés es la Ingeniería de Software, específicamente, e-learning y servicios web, el análisis y diseño de metodologías de reuso de software, patrones de diseño, metodologías para el desarrollo de proyectos de software, gestión de proyectos de software, arquitecturas de software, DevOps.



Dr. René Santaolaya Salgado, es profesor del Tecnológico Nacional de México/Centro Nacional de Investigación y Desarrollo Tecnológico, ubicado en Cuernavaca, Morelos, México. Sus áreas de investigación incluyen “Modelos de Procesos de Software”, “Arquitectura de Software”, “Reuso de Software”, “Servicios Web”, “Microservicios” y “Computación en la Nube”. El profesor Santaolaya recibió su grado de Doctor en Ciencias de la Computación con especialidad en Ingeniería de Software del Instituto Politécnico Nacional/Centro de Investigación en Computación. El es miembro senior del IEEE y ha sido analista, desarrollador, consultor, y líder del Cuerpo Académico de Ingeniería de Software de su Centro de Trabajo.



Dr. Juan Gabriel González Serna, doctor en ciencias computacionales por el CIC-IPN, miembro del Sistema Nacional de Investigadores nivel II (SNI-II) del CONAHCyT, sus áreas de investigación son Interacción Humano Computadora, Evaluación UX, computación afectiva y análisis de emociones.



Dr. Noé Alejandro Castro Sánchez, realizó sus estudios de maestría y doctorado en el Centro de Investigación en Computación del Instituto Politécnico Nacional, especializándose en Procesamiento de Lenguaje Natural. Es miembro del Sistema Nacional de Investigadores, vocal de la Sociedad Mexicana de Inteligencia Artificial, miembro de la Asociación Latinoamericana de Tecnologías del Lenguaje y docente del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET). Sus trabajos e intereses de investigación se insertan en el área de Procesamiento de Lenguaje Natural e incluyen la minería de opiniones y análisis de sentimientos, el humor computacional y el análisis de contenido, aplicados al sector salud, educación y seguridad.



Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.