

Descripción del anteproyecto Licenciatura en Ingeniería de Software Facultad de Estadística e Informática	Junio 2024
--	-------------------

Xalapa, Veracruz, a 10 de Junio de 2024.

PROYECTO DE TITULACIÓN PARA REGISTRO.

Cuerpo Académico	Ingeniería y Tecnología de Software
Nombre del proyecto de Investigación VINCULACIÓN/PLADEA-FEI	
LGAC que alimenta	LGAC2. Tecnologías de software.
Línea de Investigación	
Duración Aproximada	6 meses
Modalidad de Trabajo Recepcional	Revisión Sistemática de la Literatura
Nombre del trabajo recepcional	Requisitos no funcionales en el desarrollo de software ágil, una revisión sistemática de la literatura
Requisitos	Ingeniería de requerimientos Diseño de software Lectura de documentos en inglés Capacidad de análisis
RESPONSABLE DEL TRABAJO RECEPCIONAL.	
Director	Ana Luz Polo Estrella
Codirector	
Alumnos Participantes	1

DESCRIPCIÓN DEL PROYECTO DE INVESTIGACIÓN

--

DESCRIPCIÓN DEL TRABAJO RECEPCIONAL.

La ingeniería de software es “un enfoque sistemático para el análisis, diseño, evaluación, implementación, prueba, mantenimiento y reingeniería de software, es decir, la aplicación de la ingeniería al software”(Laplante, 2007). Uno de los objetivos de la ingeniería de software es el desarrollo de software de calidad que cumpla con las necesidades de los

usuarios.

Para cumplir con este objetivo es que se pone especial énfasis en cada una de las fases del proceso de desarrollo de software, pero principalmente en la fase de requisitos de software, que es donde se identifican las necesidades de los usuarios. Los requisitos “son una especificación de lo que se debe implementar. Son descripciones de cómo debe comportarse el sistema, o de una propiedad o atributo del sistema. Pueden ser una limitación en el proceso de desarrollo del sistema”(Sommerville et al.,1997).

Los requisitos de software incluyen tres niveles distintos: requisitos del negocio, requisitos del usuario y requisitos funcionales. Además, cada sistema tiene una variedad de requisitos no funcionales (Wiegers & Beatty, 2013).

En los últimos años el desarrollo ágil ha ganado gran popularidad, diversos métodos y metodologías han surgido, por mencionar algunos SCRUM y Extreme Programming, (Domah, 2015). Una de las principales ventajas del desarrollo de software ágil es su énfasis en los requisitos de software funcionales, sin embargo, la literatura también sugiere que los requisitos no funcionales son ignorados o mal definidos (Maitini et al, 2015).

Ante la limitada atención que se ha puesto a los requisitos no funcionales en los procesos de desarrollo de software ágil, según la literatura, es que actualmente han surgido algunas técnicas y métodos que abordan los requisitos no funcionales en los procesos de desarrollo ágil, tal es el caso de la metodología NORMAP que identifica, vincular y modelar requisitos no funcionales con requisitos funcionales(Farid, 2012)

También existen esfuerzos en la literatura por identificar cuáles son las prácticas utilizadas en proyectos de desarrollo de software ágil, por mencionar uno, el estudio realizado por Jarzębowicz (2021), centrado en prácticas que permiten identificar, obtener y documentar requisitos no funcionales en proyectos industriales, enfocándose principalmente en una entrevista semi-estructurada a 10 profesionales experimentados.

Se hace necesario hacer una revisión sistemática de la literatura en la que se analice qué existe a la fecha 2024, en cuanto a técnicas, prácticas o metodologías que aborden a los requisitos no funcionales en los procesos de desarrollo de software ágil, ventajas, desventajas. Para que en un futuro se cuente con un compendio de técnicas, prácticas o metodologías que puedan utilizar los ingenieros de software en el proceso de desarrollo de software ágil.

RESULTADOS ESPERADOS.

- Reporte de la Revisión Sistemática de la Literatura
- Artículo de divulgación a nivel congreso nacional

BIBLIOGRAFÍA RECOMENDADA.

- W. M. Farid, "The NORMAP Methodology: Lightweight Engineering of Non-functional Requirements for Agile Processes," 2012 19th Asia-Pacific Software

Engineering Conference, Hong Kong, China, 2012, (pp. 322-325).

- R. R. Maiti and F. J. Mitropoulos,(2015) "Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development," SoutheastCon 2015, Fort Lauderdale, FL, USA, 2015, (pp. 1-8)
- D. Domah and F. J. Mitropoulos, (2015) "The NERV methodology: A lightweight process for addressing non-functional requirements in agile software development," In SoutheastCon 2015, Fort Lauderdale, FL, USA, 2015 (pp. 1-7)
- Wiegers & Beatty, Software Requirements, Tercera edición, Microsoft Press, 2013
- Richards, M., Ford N., Fundamentals of Software Architecture, An Engineering Approach, O'Reilly, 2020.
- Bass, C., Clements P., Kazman R., Software Architecture in Practice. Third edition, Addison-Wesley. 2013
- Interaction Design beyond human-computer interaction. Third Edition Wiley. 2011.
- Jarzębowicz and P. Weichbroth, (2021), "A Qualitative Study on Non-Functional Requirements in Agile Software Development," in IEEE Access, vol. 9, (pp. 40458-40475)
- Artículos científicos en temas requisitos de software y desarrollo ágil de la IEEE, ACM y Springer.

<hr/> M.I.S. Ana Luz Polo Estrella Nombre y Firma del Director del Trabajo	<hr/> Nombre y Firma del Codirector del Trabajo
Vo. Bo. <hr/> Dr. Ángel Juan Sánchez García Responsable de CA-ITS	Vo. Bo. <hr/> Dr. Jorge Octavio Ocharán Hernández Coordinación de Academia de Experiencia Recepcional

NOTAS:

- 1) Casos excepcionales serán evaluados por la Academia de ER.
- 2) Tratando de un CA externo a la Licenciatura en Ingeniería de Software, el proyecto deberá llevar el aval de los CA de la misma que se asocie con el tema.
- 3) El Vo. Bo. del Responsable de CA se obtiene en la reunión de cada CA, donde se presentan los temas del mismo para su aprobación.
- 4) El Vo. Bo. de la Coordinación de ER se obtiene en una reunión de la academia que se programa para ello.