

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	Métodos y Prácticas de Desarrollo de Software Emergentes
LGAC que alimenta	L1. Gestión, modelado y desarrollo de software L2. Tecnologías de software
Línea de Investigación	
Duración Aproximada	18 meses
Modalidad de Trabajo Recepcional	Tesis
Nombre del trabajo recepcional	Evaluación del Rendimiento y Eficacia de EVOSUITE en la Generación Automática de Suites de Pruebas para Aplicaciones Java
Requisitos	Pruebas de Software; Programación; Desarrollo de Software

RESPONSABLE DEL TRABAJO RECEPCIONAL.

Director	Dr. Saúl Domínguez-Isidro
Codirector	MCC. Juan Carlos Pérez Arriaga
Alumnos Participantes	1

DESCRIPCIÓN DEL PROYECTO DE INVESTIGACIÓN

Los métodos de desarrollo son enfoques estructurados que guían el proceso completo del desarrollo de software, proporcionando un marco de trabajo para la planificación, ejecución y gestión de proyectos de Software. Por su parte, las prácticas de desarrollo son acciones específicas aplicadas por los equipos de desarrollo para lograr objetivos concretos en cada etapa del ciclo de vida del desarrollo de software. No obstante, debido al dinamismo de la sociedad, las necesidades y requisitos tecnológicos se ha originado una evolución constante del desarrollo de software. Los métodos y prácticas tradicionales de desarrollo, aunque sólidos, a menudo enfrentan desafíos para adaptarse a las demandas cambiantes del mercado, las expectativas de los usuarios y los avances tecnológicos. En este contexto, surge la necesidad de explorar y comprender a fondo los métodos y prácticas emergentes en el desarrollo de software. Con este

proyecto de investigación se propone examinar estos nuevos enfoques y su impacto en las etapas del ciclo de vida de desarrollo del software. Además de permitir a los interesados en el desarrollo de software mantenerse al tanto de las tendencias y herramientas en el área de Ingeniería de Software, se busca comprender el uso y adopción de prácticas emergentes, permitiendo la innovación y adaptación a los cambios del mercado. Finalmente, se busca contribuir al avance del conocimiento en Ingeniería de Software facilitando la creación de modelos, guías y enfoques que puedan beneficiar a la comunidad de desarrollo de software.

DESCRIPCIÓN DEL TRABAJO RECEPCIONAL.

Las pruebas unitarias son una de las técnicas más eficaces para detectar errores en fases tempranas del desarrollo de software; contribuyendo significativamente en el aseguramiento de la calidad del código (Bakharev, 2024). No obstante, la creación manual de estas pruebas puede ser costosa en términos de tiempo y es propensa a errores, lo que ha llevado al desarrollo de herramientas de generación automática de pruebas (Testsigma, 2024). EVOSUITE es una de estas herramientas, diseñada particularmente para generar suites de pruebas automáticas para aplicaciones Java (Fraser y Arcuri, 2011). A diferencia de los casos de prueba, que son unidades individuales de pruebas específicas, una suite de prueba es una colección organizada de estos casos, diseñada para ejecutar múltiples pruebas de manera sistemática y coherente (John, 2023). Esta propuesta se centra en la evaluación del rendimiento y la eficacia de EVOSUITE, además de proporcionar una guía práctica para su implementación y uso en proyectos desarrollados en Java.

En la literatura existen trabajos en los que se analizan las capacidades de herramientas como EVOSUITE (Almasi et al., 2017; Fan 2019) destacando su capacidad para mejorar la cobertura de código y detectar errores ocultos. Por otro lado, también existen trabajos en los que se exploran sus ventajas y limitaciones en comparación con otras herramientas (). Sin embargo, pocos estudios recientes han realizado una evaluación comparativa exhaustiva entre EVOSUITE y otras herramientas que deriven en una guía práctica detallada para su uso. Es por ello, que este trabajo tiene como objetivo **evaluar el rendimiento y la eficacia de EVOSUITE en la generación automática de suites de pruebas para aplicaciones Java mediante un análisis comparativo con otras herramientas de generación automática de pruebas disponibles en el mercado**. Se busca que los resultados de este trabajo recepcional contribuyan al avance del conocimiento en el campo de la generación automática de pruebas, proporcionando datos y análisis que pueden ser utilizados en futuros estudios y desarrollos.

Para alcanzar el objetivo planteado, se propone desarrollar el trabajo en diferentes fases: i) se realizará una Revisión Sistemática de la Literatura (RSL) para identificar las herramientas de generación automática de pruebas más relevantes y sus características; ii) se seleccionará un conjunto de proyectos Java de código abierto para realizar pruebas; iii) se utilizará EVOSUITE para generar suites de pruebas para cada proyecto seleccionado, y se realizarán pruebas similares con herramientas seleccionadas de la RSL. Los resultados se analizarán en términos de cobertura de código, detección de errores, y uso de recursos. iv) se desarrollará una guía práctica detallada basada en los resultados obtenidos, y se presentarán las recomendaciones para la adopción de EVOSUITE en el desarrollo de software.

RESULTADOS ESPERADOS.

- Reporte de RSL
- Documento de Tesis
- Borrador de artículo para congreso o journal

BIBLIOGRAFÍA RECOMENDADA.

Almasi, M. M., Hemmati, H., Fraser, G., Arcuri, A., & Benefelds, J. (2017). An Industrial Evaluation of Unit Test Generation: Finding Real Faults in a Financial Application. 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), 263–272. doi:10.1109/ICSE-SEIP.2017.27

Bakharev, N. (2024, February 14). *Unit testing: Definition, examples, and critical best practices*. Bright Security. <https://brightsec.com/blog/unit-testing/>

Fan, Z. (2019). A Systematic Evaluation of Problematic Tests Generated by EvoSuite. 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 165–167. doi:10.1109/ICSE-Companion.2019.00068

Fraser, G., & Arcuri, A. (2011). EvoSuite: automatic test suite generation for object-oriented software. Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering, 416–419. Presented at the Szeged, Hungary. doi:10.1145/2025113.2025179

John, S. (2023, November 16). *Understanding test suite & test case: Examples and best practices*. Next-Generation Mobile Apps and Cross Browser Testing Cloud. <https://www.lambdatest.com/learning-hub/test-suite>

Tang, Y., Liu, Z., Zhou, Z., & Luo, X. (2024). ChatGPT vs SBST: A Comparative Assessment of Unit Test Suite Generation. IEEE Transactions on Software Engineering, 1–19. doi:10.1109/TSE.2024.3382365

Testsigma (2024, May 27). *Top 7 challenges in manual testing*. Testsigma Blog. <https://testsigma.com/blog/is-manual-testing-becoming-a-bottleneck-in-continuous-delivery/>

Dr. Saúl Domínguez-Isidro
Director del Trabajo

Codirector del Trabajo

Vo. Bo.

Vo. Bo.

Dr. Ángel Juan Sánchez García
Responsable de CA-ITS

Dr. Jorge Octavio Ocharán Hernández
Coordinación de Academia de
Experiencia Receptonal

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.