ESCUELA POLITÉCNICA NACIONAL
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y VINCULACIÓN

1. DESCRIPCIÓN DETALLADA DEL PROYECTO, INCLUIDO METODOLOGÍA

Este proyecto tiene como objetivo principal el desarrollo de un artefacto para la detección de antipatrones a lo largo del ciclo de vida del desarrollo de software, el mismo que da continuidad al Proyecto Semilla PIS-23-28 que abordó la identificación y comprensión de antipatrones.

"Los antipatrones son soluciones comúnmente aplicadas a problemas recurrentes que, en lugar de resolverlos, generan consecuencias negativas en los procesos de desarrollo. Estos pueden surgir por falta de conocimiento, experiencia insuficiente o por la aplicación incorrecta de patrones válidos en contextos inapropiados" (Brown et al., 1998).

El proyecto se enfocará en estudiar los problemas recurrentes en el desarrollo de software que dan origen a los antipatrones, con el objetivo de comprender cómo surgen y en qué contextos aparecen. A partir de este análisis, se busca proponer un procedimiento que permita identificar de manera efectiva los antipatrones en cada fase del ciclo de vida del desarrollo de software, asegurando que dicha identificación se realice de forma sistemática y clara. Finalmente, el artefacto resultante será evaluado mediante un caso de estudio, con el fin de demostrar su utilidad práctica y validar cómo su aplicación puede contribuir significativamente a mejorar los procesos de desarrollo.

Metodología de investigación

El proyecto seguirá el enfoque de Design Science Research (DSR), dado que este método permite desarrollar soluciones prácticas mediante la creación y validación de artefactos que aborden problemas reales en contextos específicos (Dresch et al., 2015). En donde, la construcción y aplicación de un artefacto ayudará a resolver la problemática central, (Wieringa, 2014) indica que "El término artefacto es fundamental para la investigación científica del diseño y se utiliza para describir algo que es artificial o construido por humanos, en contraposición a algo que ocurre de forma natural" (p.106).

El DSR será implementado debido a su capacidad para combinar rigor científico con aplicabilidad práctica, facilitando la identificación de antipatrones en cada fase del ciclo de vida de desarrollo de software. La Figura 1 representa el proceso del DSR de una forma resumida.

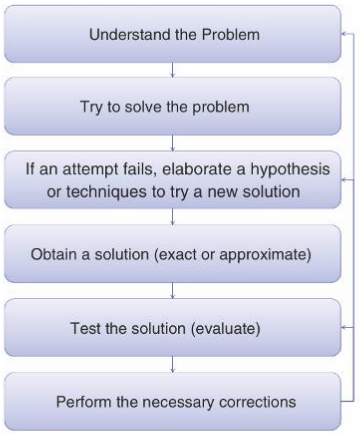

Figura 1. Pasos para realizar un proceso de investigación en DSR. Adaptado de (Dresch et al., 2015)

ESCUELA POLITÉCNICA NACIONAL
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y VINCULACIÓN

A. Diagnóstico

Se identificará la problemática y se establecerá la necesidad de investigar antipatrones en las distintas fases del ciclo de vida de desarrollo. Esta fase es clave para delimitar el alcance del estudio y justificar su relevancia. Se empleará una revisión sistemática de literatura utilizando el marco de Kitchenham para garantizar una base sólida de conocimiento y asegurar la validez de los hallazgos (Kitchenham et al., 2015).

- **Definición del marco conceptual**

Se identificarán los conceptos clave relacionados con los antipatrones y se definirán las fases del ciclo de vida de desarrollo de software a analizar.

- **Formulación de preguntas de investigación**

Se definirán preguntas para dirigir los esfuerzos del estudio y encaminar el proyecto hacia los resultados deseados.

- **Revisión sistemática de literatura con el marco de Kitchenham**

Se empleará una revisión sistemática de literatura utilizando el marco de Kitchenham para garantizar una base sólida de conocimiento y asegurar la validez de los hallazgos (Kitchenham et al., 2015).

  - **Estrategia:** Se establecerán cadenas de búsqueda y bases de datos relevantes. Posteriormente, se definirán criterios de inclusión y exclusión.
  - **Ejecución:** Se buscarán y extraerán estudios relevantes. Posteriormente, se eliminarán duplicados y se seleccionarán los estudios relevantes. Por último, se obtendrá un conjunto final de estudios seleccionados.

B. Planificación

Se diseñará un plan de acción que permita llevar a cabo la investigación y garantizar que cada fase del ciclo de vida sea adecuadamente cubierta. Esto es necesario para asegurar la viabilidad y efectividad del proyecto.

- **Análisis de la información recopilada.**

El análisis de la información recopilada es una etapa crítica para garantizar que los fundamentos teóricos y prácticos del proyecto estén alineados con los objetivos establecidos. Esta actividad implica la revisión detallada de estudios previos, artículos académicos, marcos metodológicos y casos prácticos relacionados con el ciclo de vida del desarrollo de software. Mediante este análisis, se busca identificar antipatrones relevantes y recurrentes que puedan ser consideradas en el diseño del artefacto. El análisis también permitirá detectar áreas donde la información es insuficiente o inconsistente, facilitando la identificación de temas que requieren mayor profundización.

- **Determinación de brechas en la información recopilada.**

El análisis de brechas será una actividad fundamental dentro del desarrollo del proyecto, ya que permitirá evidenciar los diversos enfoques utilizados en investigaciones previas y las problemáticas asociadas a los mismos. Esto permitirá determinar si la falta de un enfoque estandarizado en dichas investigaciones dificulta su validación práctica, lo que podría ocasionar que muchas de las propuestas académicas permanezcan en un estado teórico sin ser actualizadas ni mejoradas con el paso del tiempo, perdiendo relevancia ante los constantes cambios en la industria del software. Este análisis contribuirá a identificar vacíos de información y áreas que requieren una mayor atención para garantizar que los resultados del proyecto sean aplicables y útiles en contextos reales.

- **Diseño del cronograma:**
  - **Recursos y tiempos:** Identificación de recursos y tiempos necesarios.

ESCUELA POLITÉCNICA NACIONAL
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y VINCULACIÓN

- **Actividades e hitos para conseguir:** Planificación de actividades clave del proyecto y definición de hitos principales para evaluar avances.
- **Seguimiento de las actividades:** Implementación de revisiones periódicas para evaluar el progreso y realizar ajustes de forma óptima.

C. Construcción

Se desarrollará un artefacto que permitirá identificar los antipatrones en cada fase del ciclo de vida de desarrollo de software. Esta etapa es esencial para generar un instrumento práctico que facilite las aplicaciones del artefacto en fases posteriores. El proceso de construcción del artefacto incluye los siguientes pasos clave:

- **Definición del tipo de artefacto a construir**

En este paso se determina el tipo de artefacto que se va a desarrollar, definiendo su objetivo principal y el alcance que tendrá en la identificación de antipatrones. El artefacto puede tomar la forma de un modelo, una guía práctica, una herramienta automatizada o un marco conceptual, dependiendo de las necesidades del proyecto.

- **Desarrollo y documentación del artefacto**

Este paso implica la creación del artefacto en sí, asegurándose de que cumpla con los requisitos definidos previamente. Se utilizarán principios de diseño basados en el DSR para asegurar que el artefacto sea útil y efectivo en contextos reales.
La documentación del artefacto incluirá detalles sobre su estructura, propósito, funcionamiento y recomendaciones para su aplicación práctica.

D. Evaluación

Se evaluará la utilidad y aplicabilidad del artefacto en un caso de estudio. Esta fase es clave para asegurar la eficacia del artefacto en contextos prácticos.

- **Prueba del artefacto:** Se aplicará la guía y el artefacto en un caso de estudio.
- **Análisis de resultados:** Se compararán los resultados obtenidos con los objetivos propuestos.

E. Aprendizaje

Se extraerán lecciones aprendidas y se generará nuevo conocimiento sobre la identificación de antipatrones. Esta fase asegura que los hallazgos sean aprovechados para futuras investigaciones y aplicaciones.

- **Documentación de hallazgos:** Se registrarán los antipatrones detectados y las soluciones aplicadas.
- **Divulgación de resultados:** Se publicará un informe final y se compartirán los hallazgos con la comunidad académica y profesional.

Brown, W. J., Malveau, R. C., Mccormick, H. W., Thomas, I., Mowbray, J., Wiley, J., Brown, K., Malveau, C., Mccormick, K., & Mowbray, K. (1998). *AntiPatterns Refactoring Software, Architectures, and Projects in Crisis Executive Summary Overview*.

Dresch, A., Daniel, ·, Lacerda, P., Antônio, J., & Antunes, V. (2015). *Design Science Research A Method for Science and Technology Advancement*.

Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews* (Vol. 4). CRC press.

ESCUELA POLITÉCNICA NACIONAL
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y VINCULACIÓN

Wieringa, R. (2014). *Design Science Methodology: Principles and Practice Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE*. ACM Digital Library. https://doi.org/doi:10.1145/1810295.1810446