



A. PROPUESTA PROYECTO DE INVESTIGACIÓN¹

1. TIPO DE PROYECTO:

Interno		Grupal	
Semilla	X	Multidisciplinario	

2. TIPO DE INVESTIGACIÓN:

Básica	X	Aplicada	
--------	---	----------	--

3. UNIDAD EJECUTORA (*Departamento, Instituto o Estructura de Investigación*):

Departamento de Informática y Ciencias de la Computación

4. LINEA(S) DE INVESTIGACIÓN Y/O LÍNEAS PRIORITARIAS DE INVESTIGACIÓN:

Ingeniería de software

5. TÍTULO DEL PROYECTO (*mínimo 10 palabras*):

Antipatrones en la producción de software: Estudio de caso aplicado en la Facultad de Ingeniería de Sistemas

6. RESUMEN (*máximo 200 palabras*)

Los patrones en ingeniería de software constituyen una colección de soluciones recurrentes que emergen frente a problemas comunes durante el desarrollo de software. Su contraparte, llamada antipatrones, hacen referencia al uso de un patrón cuya aplicación conlleva resultados negativos. A la fecha, pese a toda las aportaciones en el campo de la ingeniería de software, los problemas relacionados con el desarrollo de software se mantienen tales como: retrasos en tiempos de entrega, costos más allá de lo esperado o dificultad en la adopción de cambios, entre otros; siendo la reacción habitual de los profesionales, responsabilizar a las prácticas de ingeniería de software como los generadores de los problemas antes mencionados, dejando de lado los pilares fundamentales de la disciplina. Es por esto que el estudio de antipatrones está tomando realce debido a que permite comprender y entender las consecuencias y repercusiones de patrones considerados como "buenas prácticas", pero que, mal aplicadas, deterioran la calidad de un proyecto de desarrollo de software. El propósito de esta investigación apunta al estudio de los antipatrones en la ingeniería de software a través de un estudio cualitativo de los trabajos de titulación generados por los graduados de la Facultad de Sistemas de la EPN.

7. PALABRAS CLAVE (4-6)

Antipatrón, mala práctica, smell, ingeniería de software, catálogo de antipatrones.

¹ Nota: En caso de información adicional puede contactarse a direccion.investigacion@epn.edu.ec



8. OBJETIVOS

8.1. OBJETIVO GENERAL

Descubrir antipatrones en el ciclo de vida de desarrollo de software, a través de un estudio cualitativo de los trabajos de titulación generados por los graduados de la Facultad de Sistemas de la EPN.

8.2. OBJETIVOS ESPECÍFICOS

- Identificar los antipatrones más representativos en la literatura de software a través de una revisión sistemática.
- Descubrir los antipatrones presentes en proyectos de desarrollo de software realizados en 2024 de la carrera de ingeniería de software.
- Mapear los hallazgos de la literatura con las prácticas realizadas en los trabajos de titulación de la Facultad de Sistemas.

9. HIPÓTESIS (opcional)

No aplica

10. DETALLE DE LOS RESULTADOS ESPERADOS (con relación a los objetivos específicos)

- Listado de antipatrones obtenidos de la literatura más representativos en la industria del software (Artículo de revisión de literatura).
- Listado de antipatrones obtenidos de las prácticas en trabajos de titulación de la Facultad de Sistemas (Artículo de antipatrones presentes en trabajos de titulación)
- Análisis de antipatrones reflejados en la literatura y los encontrados en la academia (Representación cartográfica).

11. IMPACTO DE LA INVESTIGACIÓN

Seleccionar el o los Objetivos de Desarrollo Sostenible (ODS) en los que se enmarca el proyecto.

1. FIN DE LA POBREZA ()	2. HAMBRE CERO ()	3. SALUD Y BIENESTAR ()	4. EDUCACIÓN DE CALIDAD ()
5. IGUALDAD DE GÉNERO ()	6. AGUA LIMPIA Y SANEAMIENTO ()	7. ENERGÍA ASEQUIBLE Y NO CONTAMINANTE ()	8. TRABAJO DECENTE Y CRECIMIENTO ECONÓMICO ()
9. INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA (X)	10. REDUCCIÓN DE LAS DESIGUALDADES ()	11. CIUDADES Y COMUNIDADES SOSTENIBLES ()	12. PRODUCCIÓN Y CONSUMO RESPONSABLE ()
13. ACCIÓN EL CIMA ()	14. VIDA SUBMARINA ()	15. VIDA DE ECOSISTEMAS TERRESTRES ()	16. PAZ, JUSTICIA E INSTITUCIONES SOLIDAS ()
17. ALIANZA PARA LOGRAR LOS OBJETIVOS ()	18. NINGUN ODS ()	OBJETIVOS DE DESARROLLO SOSTENIBLE	

Este estudio tiene como principal aporte la investigación en un área fundamental del eje económico del país, pero que al momento supone problemas en capacidad tecnología y humana en los procesos de desarrollo de software de calidad. El software es un punto de apoyo de todos los sectores productivos de los países, aportando con mecanismos competitivos de diferenciación e innovación. La industria del software ecuatoriana no solo oferta sus servicios dentro de forma local, sino que la producción de software es solicitada también en el exterior. La investigación permitirá aportar con conocimiento sobre las prácticas para desarrollo de infraestructuras informáticas, que sin duda podrán mejorar el valor de los productos y servicios de software generados en la industria ecuatoriana.



ESCUELA POLITÉCNICA NACIONAL

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y VINCULACIÓN



Nota: Salto de página incluido por favor no retirarlo



12. ESTADO DEL ARTE

En la actualidad, la investigación en la disciplina de la ingeniería de software se ha concentrado en solucionar los problemas que implican el traslado de conceptos de negocios en aplicaciones de software, basándose en los fundamentos de la ingeniería. Según Brown (Brown et al., 1998), existe una falsa promesa en las soluciones planteadas en la disciplina de la ingeniería de software, ya que a pesar de que existe un vasto conocimiento en esta área, es difícil reunir los factores que garantizan el éxito de un proyecto. El mismo autor relaciona el posible fracaso de los proyectos de software a una incorrecta o inapropiada aplicación de una solución debido al "hype" provocado por una nueva técnica, herramienta, proceso, etc. Otros autores apuntan a que los problemas en esta área están relacionados a modelos de negocio, financiamiento, y la concepción del producto de software (Klotins et al., 2019).

Adicional a esto, el área de ingeniería de software ha sido estudiada bajo los lentes de los patrones en el ciclo de vida de desarrollo de software (Brown et al., 1998), (Neill & Laplante, 2005) (Chen & Jiang, 2017), (Karasneh et al., 2016), (Riaz & Williams, 2012), siendo éstos estrechamente relacionados a la calidad del producto de software, ya que desconocer y aplicar incorrectamente los patrones sin lugar a duda afecta el proceso de desarrollo de software, no obstante, se está dejando de lado la aplicación de nuevas soluciones que requieren de estudios empíricos que investiguen la relación entre contextos de problemas y solución; siendo el estudio de antipatrones en la ingeniería de software una forma de abordar estos análisis.

Estudiar a fondo los antipatrones implica un avance revolucionario que complementa al estudio de los patrones, ya que los antipatrones son definidos como soluciones a problemas que se repiten una y otra vez pero que tienen más consecuencias negativas que beneficios. Al igual que los patrones, los antipatrones se presentan no solo en los ámbitos de diseño de software, sino también en contextos de análisis, organizaciones, instruccionales, entre otras (Brown et al., 1998), (Jones, 2021).

Actualmente, el estudio de antipatrones se ha podido evidenciar en trabajos donde se catalogan a los antipatrones propios del área de gestión de procesos (Brada & Picha, 2019), otros tipos de antipatrones han sido abordados en la literatura, algunos enfocados en encontrar brechas de seguridad (Riaz & Williams, 2012), (Smith & Williams, 2002), otros trabajos centrados en estudiar patrones en la etapa de requerimientos (Laverdiere et al., 2006), otros estudios en la etapa de diseño (Halkidis et al., 2006), (Jones, 2021) y otros en las fases de emprendimiento de un software (Klotins et al., 2019). A pesar de que existen diversos estudios sobre antipatrones, el foco se ha concentrado en la catalogación y en técnicas de detección en distintas fases del desarrollo o sobre distintas tecnologías. Por ejemplo, (Alkharabsheh et al., 2019), en un mapeo sistemático describe el estado del arte sobre detección de antipatrones de diseño y muestra el impacto de este sobre distintos atributos de calidad del producto software. Brada (Brada & Picha, 2019) cataloga antipatrones y describe un esquema para ello; sin embargo, su estudio caracteriza antipatrones en la administración de proyectos de software. Siguiendo esta línea, estudios como el de Flores et al. (Flores et al., 2020a) intentar dar luz a los antipatrones relacionados a la administración de proyectos, haciendo énfasis a los proyectos ágiles y estableciendo una correlación a ciertas filosofías del agilismo como detonadores de antipatrones que afectan de forma práctica y financiera a un proyecto. Chouchen et al. (Chouchen et al., 2021) realizan un estudio enfocado sobre antipatrones en una moderna revisión de código; explorando las malas prácticas de los procesos de verificación de software y su contribución a la degradación del código, baja capacidad de integración y mantenibilidad del software a raíz de una cultura pobre de calidad. Cerny et al. (Cerny et al., 2023) realiza un estudio en donde describe antipatrones en el desarrollo de microservicios y establece técnicas para detectarlos, otros trabajos han realizado sistemas automatizados (Aras & Selçuk, 2016) basados en métricas y reglas para software orientado a objetos.



El análisis sobre antipatrones se ha incrementado y especializado en años recientes. A través de una exploración básica de literatura se puede observar que los estudios se centran más en la detección de antipatrones relacionadas al código (code smells). Seguidamente, otros estudios se centran en los antipatrones de las estructuras arquitectónicas y de diseño de los sistemas informáticos. Finalmente, un grupo de estudios se centran en la gestión de los proyectos de software.

El uso de los antipatrones sin duda complementa el conocimiento de los patrones de diseño; si no se comprende adecuadamente el propósito de estos, se puede llegar a lo que se conoce como el "uso destructivo de antipatrones" (Brown et al., 1998). Como resultado, en lugar de propiciar un cambio positivo, se puede generar ambientes donde no se aplique prácticas positivas a través de toma de decisiones llenas de controversias, tanto en los ámbitos organizacionales como técnicos. Adicionalmente a listar, clasificar y catalogar los antipatrones, éstos deben ser interpretados y analizados a través de competencias blandas y duras. La concientización es el primer paso para evitar la generación de ambientes disfuncionales (menos sofisticados, agradables y productivos). Con el estudio de patrones y antipatrones, es posible analizar la raíz de las conductas que provocan errores al aplicar determinada solución; así como los mecanismos o problemáticas de la toma de decisiones que impactan un contexto de solución que dan lugar a un "software disfuncional".

Con respecto al estudio de antipatrones en el contexto del desarrollo de software, el foco se ha concentrado en la catalogación y en técnicas de detección en distintas fases del desarrollo o sobre distintas tecnologías. Por ejemplo, (Alkharabsheh et al., 2019), en un mapeo sistemático describe el estado del arte sobre detección de antipatrones de diseño y muestra el impacto de este sobre distintos atributos de calidad del producto software. Brada (Brada & Picha, 2019) cataloga antipatrones y describe un esquema para ello; sin embargo, su estudio caracteriza antipatrones en la administración de proyectos de software. Siguiendo esta línea, estudios como el de (Flores et al., 2020a) y (Flores et al., 2020b) sigue la línea en clarificar los antipatrones relacionados a la administración de proyectos, haciendo énfasis a los proyectos ágiles y estableciendo una correlación a ciertas filosofías del agilismo como detonadores de antipatrones que afectan de forma práctica y financiera a un proyecto. Chouchen et al. (Chouchen et al., 2021) realizan un estudio enfocado sobre antipatrones en una moderna revisión de código; explorando las malas prácticas de los procesos de verificación de software y su contribución a la degradación del código, baja capacidad de integración y mantenibilidad del software a raíz de una cultura pobre de calidad. En (Cerny et al., 2023) se realiza un estudio en donde describe antipatrones en el desarrollo de microservicios y establece técnicas para detectarlos.

El incurrir en antipatrones en el software no solo dificultan el aseguramiento y el control de calidad, sino que genera una elevada deuda técnica. Conocer sobre antipatrones permite ampliar el enfoque que se puede dar a las distintas soluciones, sin embargo, también puede ser perjudicial sin un buen análisis. De acuerdo a (Jones, 2021) los problemas del proceso de desarrollo de software en la industria tienen su origen en los currículos académicos que carecen de la inclusión de tópicos fundamentales en ámbitos organizacionales y de toma de decisiones y que consecuentemente lleva a una falla en la evaluación de aplicación de patrones (Brown et al., 1998). La "vacuna" de la mala calidad de software puede darse a través de la comprensión de antipatrones. En este sentido, es necesario realizar estudios que permitan identificar los antipatrones desde su aparición en edades tempranas, como lo es en la academia; con el fin de atacarlos de una manera preventiva. De aquí, la motivación para este estudio.

Este estudio propone extraer los antipatrones del ciclo de vida de desarrollo de software que se presenta con más frecuencia para luego contrastarlos con los hallazgos del estudio cualitativo aplicado sobre trabajos de titulación de proyectos de software, con la finalidad de comprender la problemática y abordarla en los procesos de formación profesional.

Alkharabsheh, K., Crespo, Y., Manso, E., & Taboada, J. A. (2019). Software design smell detection: a systematic mapping study. *Software Quality Journal*, 27, 1069–1148.



- Aras, M. T., & Selçuk, Y. E. (2016). Metric and rule based automated detection of antipatterns in object-oriented software systems. *2016 7th International Conference on Computer Science and Information Technology (CSIT)*, 1–6.
<https://doi.org/10.1109/CSIT.2016.7549470>
- Brada, P., & Picha, P. (2019). Software process anti-patterns catalogue. *Proceedings of the 24th European Conference on Pattern Languages of Programs*, 1–10.
- Brown, W. H., Malveau, R. C., McCormick, H. W. S., & Mowbray, T. J. (1998). *AntiPatterns: refactoring software, architectures, and projects in crisis*. John Wiley & Sons, Inc.
- Cerny, T., Abdelfattah, A. S., Al Maruf, A., Janes, A., & Taibi, D. (2023). Catalog and detection techniques of microservice anti-patterns and bad smells: A tertiary study. *Journal of Systems and Software*, 206, 111829.
- Chen, B., & Jiang, Z. M. (2017). Characterizing and detecting anti-patterns in the logging code. *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 71–81.
- Chouchen, M., Ouni, A., Kula, R. G., Wang, D., Thongtanunam, P., Mkaouer, M. W., & Matsumoto, K. (2021). Anti-patterns in modern code review: Symptoms and prevalence. *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 531–535.
- Flores, G. C., Moreno, A. M., & Peters, L. (2020a). Agile and Software Project Management Antipatterns: Clarifying the Partnership. *IEEE Software*, 38(5), 39–47.
- Flores, G. C., Moreno, A. M., & Peters, L. (2020b). Agile and Software Project Management Antipatterns: Clarifying the Partnership. *IEEE Software*, 38(5), 39–47.
- Halkidis, S. T., Chatzigeorgiou, A., & Stephanides, G. (2006). A qualitative analysis of software security patterns. *Computers & Security*, 25(5), 379–392.
- Jones, C. (2021). *Software development patterns and antipatterns*. CRC Press.
- Karasneh, B., Chaudron, M. R. V., Khomh, F., & Guéhéneuc, Y.-G. (2016). Studying the relation between anti-patterns in design models and in source code. *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 1, 36–45.
- Klotins, E., Unterkalmsteiner, M., & Gorschek, T. (2019). Software Engineering Antipatterns in Start-Ups. *IEEE Software*, 36(02), 118–126.
<https://doi.org/10.1109/MS.2018.227105530>
- Laverdiere, M. A., Mourad, A., Hanna, A., & Debbabi, M. (2006). Security design patterns: Survey and evaluation. *2006 Canadian Conference on Electrical and Computer Engineering*, 1605–1608.
- Neill, C. J., & Laplante, P. A. (2005). *Antipatterns: identification, refactoring, and management*. CRC Press.
- Riaz, M., & Williams, L. (2012). Security requirements patterns: understanding the science behind the art of pattern writing. *2012 Second IEEE International Workshop on Requirements Patterns (RePa)*, 29–34.
- Smith, C. U., & Williams, L. G. (2002). New software performance antipatterns: More ways to shoot yourself in the foot. *Int. CMG Conference*, 667–674.



ESCUELA POLITÉCNICA NACIONAL
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y VINCULACIÓN



Nota: Salto de página incluido por favor no retirarlo



13. INVESTIGACIONES PREVIAS DEL EQUIPO

Flores, P., Rivas, Ismael, Torres, J. (2023), Difficulties in Object-Oriented Design and its relationship with with Abstraction: A Systematic Review of Literature, The 4th european symposium on software engineering. (Aceptado y por publicar).

V. Velepucha and **P. Flores**, "A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges," in *IEEE Access*, vol. 11, pp. 88339-88358, 2023, doi: 10.1109/ACCESS.2023.3305687.

Anchundia, C., Sanchez-Gordon, S., Gómez, A., & Fonseca, C. (2020). La paradoja de la mejora de reproducibilidad en la investigación empírica en Ingeniería de Software: Reflexiones derivadas de un estudio de caso. *Revista Ibérica de Sistemas e Tecnologías de Informação*(E31), 545–557.

Anchundia, C. (2020). Resources for reproducibility of experiments in empirical software engineering: Topics derived from a secondary study. *IEEE Access*, 8, 8992–9004.

Flores, P., Alvarez, M., & Torres, J. (2022). Identifying Difficulties of Software Modeling Through Class Diagrams: A Long-Term Comparative Analysis. *IEEE Access*, 10, 28895-28910.

Flores, P., Torres, J., & Fonseca-Delgado, R. (2019). Design decisions under object-oriented approach: A thematic analysis from the abstraction point of view. In *Proceedings of the 8th Computer Science Education Research Conference* (pp. 89–97).

Flores, P., Rodas, C., & Torres, J. (2022). Empirical study on the difficulties of software modeling through class diagrams. In *Proceedings of the Hawaii International Conference on System Sciences*.

Maldonado-Garcés, V., Araujo, E., Carrión, M., Santórum, M., Acosta-Vargas, P., López, D., Quijano, **M., Llanganate, L.** (2022). Designing a Serious Game to enhancement of musical skills of children using iPlus methodology. In: Tareq Ahram and Christianne Falcão (eds) *Human Factors in Virtual Environments and Game Design*. AHFE (2022) International Conference. AHFE Open Access, vol 50. AHFE International, USA.
<http://doi.org/10.54941/ahfe1002074>

H. Ordoñez, J. Azadobay, M. Morales, and C. Montenegro, "Web Automation of EPN's Electoral Voting Process Using Blind Signature Security Scheme", *LAJC*, vol. 9, no. 2, pp. 66-79, Jul. 2022.

Isacás-Ojeda, E., Intriago-Pazmiño, M., **Ordoñez-Calero**, H., Salazar-Jácome, E., Sánchez-Ocaña, W. (2018). Integrated Framework for the Civil Construction Projects Management by Mean PMBOK, ISO 21500 and ITIL V3. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds) *Trends and Advances in Information Systems and Technologies*. WorldCIST'18 2018. *Advances in Intelligent Systems and Computing*, vol 745. Springer, Cham.
https://doi.org/10.1007/978-3-319-77703-0_97

Rivera, R., Kotzias, P., Sudhodanan, A., & Caballero, J. (2019). Costly freeware: a systematic analysis of abuse in download portals. *IET Information Security*, 13(1), 27-35.

Kotzias, P., Matic, S., **Rivera, R.**, & Caballero, J. (2015, October). Certified PUP: abuse in authenticode code signing. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 465-478).



Nota: Salto de página incluido por favor no retirarlo

14. DESCRIPCIÓN DETALLADA DEL PROYECTO, INCLUIDO METODOLOGÍA

Con el propósito de realizar la investigación propuesta, se plantea ejecutar un proceso de investigación cualitativa para encontrar hallazgos relevantes sobre el tópico a estudiar. Consecuentemente se plantean las siguientes macro actividades:

A. Establecimiento del marco conceptual y preguntas de investigación

Se definirá el marco conceptual que dará forma al estudio. Incluye establecer el enfoque de investigación, preguntas a contestar, mecanismos y herramientas de análisis, criterios de selección, entre otras.

B. Revisión sistemática de literatura

Debido a la variedad y falta de estandarización sobre antipatrones de software, se plantea como primer acercamiento, el análisis respectivo de los antipatrones a través de un proceso de revisión de literatura sistemática (Kitchenham et al., 2015). Esto permitirá conocer las distintas clasificaciones propuestas, así como priorizar aquellos patrones recurrentes reportados por la literatura científica. En este proceso se aplica deberá desarrollar un protocolo para:

- La estrategia de búsqueda
- La selección de estudios primarios
- Extracción de datos
- Síntesis de datos

C. Estudio cualitativo

El estudio cualitativo se analizarán diversos casos para un estudio para determinar la ocurrencia de antipatrones en las prácticas de desarrollo (Yin, 2018). Se diseñara un estudio de caso para establece el proceso de investigación que cubrirá las siguientes etapas:

Preparación y recolección de datos

Se realizará la selección de un grupo de proyectos de titulación basado en criterios de selección. Para este proceso se realizará un [llamado para seleccionar proyectos de titulación relacionados al desarrollo de software que se estén ejecutando, a fin de establecer un procedimiento de seguimiento y documentación]

Análisis cualitativo

Para el análisis cualitativo se procederá a la elaboración, priorización, refinamiento y catalogación de códigos relevantes al marco conceptual establecido. Esto deberá incluir información asociada a los proyectos de desarrollo, al igual que el descubrimiento de antipatrones.

Síntesis de resultados

El propósito de este proceso será contar con los recursos de información suficientes para mapear los antipatrones de mayor relevancia con aquellos aspectos de las prácticas descubiertas en los proyectos de desarrollo de software. Con esto se podrá idear estrategias para contrarrestar la mala práctica presente en los futuros profesionales de la industria del software.

E. Difusión



La difusión contempla actividades como publicación de artículo en revista científica, participación en conferencia internacional, talleres o charlas dentro del grupo de investigación, presentación de resultados dentro de la carrera de Ingeniería de Software.

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

Yin, R. K. (2018). *Case study research and applications* (Vol. 6). Sage Thousand Oaks, CA.

15. DESCRIPCIÓN EQUIPOS DISPONIBLES

- Indicar los **equipos disponibles** que serán utilizados para la ejecución del proyecto. Para los equipos considerados en esta lista se podrá **incluir la adquisición de accesorios** necesarios para su funcionamiento y ejecución del proyecto.

Equipos		
Nombre del Equipo	Departamento	Ubicación del Equipo
PC de escritorio	DICCC	Facultad de Sistemas
Portátiles personales	DICC	Facultad de Sistemas

16. VINCULACIÓN CON POSGRADOS

No se tiene contemplada la vinculación con estudiantes de posgrado de la EPN.

17. JUSTIFICACIÓN DEL PRESUPUESTO PARA EJECUCIÓN DEL PROYECTO

Presupuesto del Proyecto	
17.1 Maquinaria y Equipo especializado requerido	
Justificación	Contribución a la consecución de los objetivos del proyecto
-	-
17.2 Equipo informático requerido	
Justificación	Contribución a la consecución de los objetivos del proyecto
Licenciamiento de la herramienta de análisis cualitativo	Contar con una herramienta informática y colaborativa aportará a aumentar la calidad de la investigación mejorando o automatizando tareas de catalogación, respaldo, integración, entre otras.
17.3 Insumos y reactivos requeridos	
Justificación	Contribución a la consecución de los objetivos del proyecto
-	-
17.4 Contratación de personal requerido	
Justificación	Contribución a la consecución de los objetivos del proyecto
Se requieren técnicos ayudantes para la realización de las actividades de seguimiento, documentación, y tareas de análisis cualitativo de la documentación	El aporte de este contingente contribuirá a los procesos de recolección de datos y análisis cualitativo. Otros procesos de la investigación al que aportarían incluyen el empaquetado de la investigación.



17.5 Análisis de laboratorio requeridos	
Justificación	Contribución a la consecución de los objetivos del proyecto
17.6 Investigadores colaboradores externos invitados	
Justificación	Contribución a la consecución de los objetivos del proyecto
17.7 Salidas de campo y de muestreo, ponencias nacionales, capacitaciones y/o visitas técnicas requeridas	
Justificación	Contribución a la consecución de los objetivos del proyecto
<i>Se espera hacer una difusión de los avances de investigación de forma local, especialmente sobre la revisión de literatura</i>	Aporta a la difusión y cumplimiento de los requisitos del proyecto de investigación.
17.8 Ponencias en el exterior, capacitaciones y/o visitas técnicas requeridas	
Justificación	Contribución a la consecución de los objetivos del proyecto
17.9 Pago de publicaciones	
Justificación	Contribución a la consecución de los objetivos del proyecto
<i>Se espera hacer una publicación de alto impacto sobre los hallazgos y trabajos futuros de la investigación.</i>	Aporta a la difusión y cumplimiento de los requisitos del proyecto de investigación

18. FONDOS ADICIONALES

No existen fondos de otros organismos



Nota: Salto de página incluido por favor no retirarlo

B. DATOS INFORMATIVOS

1. INFORMACIÓN DEL DIRECTOR, CODIRECTOR, COLABORADORES Y COLABORADORES TÉCNICOS

Apellidos y nombres	No. de Cédula	HSS*	Departamento	Rol	Cargo* *	Título de mayor nivel y mención.
Anchundia Valencia Carlos Eduardo	1308669496	16	DICC	Director	Profesor titular tiempo completo	Ph.D.
Flores Naranjo Pamela Catherine	1716270838	8	DICC	Codirectora	Profesor titular tiempo completo	Ph.D.
Ordoñez Calero Hernán David	1722511001	6	DICC	Colaborador	Profesor ocasional tiempo completo	Máster
Rivera Guevara Richard Paúl	1717160533	6	DICC	Colaborador	Profesor ocasional tiempo completo	Ph.D.
Luis Alejandro Llanganate Valencia	1722619523	0	-	Colaborador	Investigador externo	Ingeniero

***HSS =Horas Semana Semestre:** Es el número de horas que se dedica por semana a la investigación. Este número de horas se mantiene para todo el semestre

****Cargo** – Profesor titular tiempo completo, Profesor ocasional tiempo completo, estudiante de posgrado, Investigador externo, Personal administrativo.



Nota: Salto de página incluido por favor no retirarlo

C. INFORMACIÓN ADICIONAL (Opcional)

1. Información de presentación previa de la propuesta

Indique si la propuesta de proyecto fue presentada en convocatorias anteriores de la EPN	SI
Año de la Convocatoria	2023
Tipo de Proyecto	Semilla

2. Sugerencia de potenciales revisores

Nombre	Institución	Correo de contacto
Rodrigo Fonseca	ESPE	erfonseca@espe.edu.ec
Sonia Pamplona	UDIMA	sonia.pamplona@udima.es

- Los revisores sugeridos no deberán ser coautores de publicaciones científicas del equipo del proyecto en los últimos tres años
- La sugerencia de revisores no garantiza que la propuesta será evaluada por ellos
- Los revisores sugeridos deben ser personal externo a la EPN.

3. Exclusión de potenciales revisores

- En esta sección incluya los pares revisores que no desea sean considerados para la revisión de la propuesta.
- No es necesario incluir en esta lista personal de la EPN

Nombre	Institución
-	-



Nota: Salto de página incluido por favor no retirarlo

D. DECLARACIÓN FINAL

DECLARACIÓN DEL DIRECTOR DEL PROYECTO

El equipo de investigadores, representado por el Director del Proyecto declara lo siguiente:

- Que el presente proyecto es una creación original de mi autoría y del equipo de investigadores, y por tanto asumimos la completa responsabilidad legal en caso de que un tercero alegue la titularidad de los derechos intelectuales del proyecto, exonerando a la EPN de cualquier acción legal que se derive por esta causa.
- Que aceptamos conocer y cumplir con la normativa vigente para la gestión de proyectos y las Bases para la Convocatoria para la presentación de Proyectos de Investigación, Vinculación y Transferencia Tecnológica con financiamiento 2023.
- Que el presente proyecto no ha sido ejecutado, ni está actualmente siendo evaluado en ninguna convocatoria de otra institución pública o privada. El incumplimiento será causal para que el proyecto no sea tomado en consideración.
- Que si el proyecto genera algún producto o procedimiento susceptible de obtener derechos de propiedad intelectual, de los cuales se deriven beneficios, aceptamos que éstos serán compartidos entre los investigadores y la institución o las instituciones participantes en el proyecto, conforme a lo establecido en el COESC.
- Que el equipo de investigadores y/o instituciones participantes se comprometen a mantener la confidencialidad de la información si ésta podría ser susceptible de protección por patentes, y solicitar la valoración de propiedad intelectual respectiva previa a cualquier publicación o difusión.
- Que para el caso de derechos de autor otorgamos una licencia de uso exclusivo con fines académicos para la o las instituciones participantes en el proyecto.

Firma del Director del Proyecto
Nombre: Carlos Eduardo Anchundia Valencia, Ph.D.
C.I.: 1308669496



ESCUELA POLITÉCNICA NACIONAL
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y VINCULACIÓN





A. DECLARACIÓN DEL CODIRECTOR DEL PROYECTO

Declaro que conozco que la normativa de proyectos establece que, en caso de ausencia del Director del proyecto en ejecución, la Dirección de Investigación notificará el cambio de Director a favor del Codirector.

Firma del Codirector del Proyecto

Nombre: Pamela Catherine Flores Naranjo, Ph.D.

C.I.: 1716270838