

PROYECTO DE INVESTIGACIÓN Y DESARROLLO

1. Título del Proyecto de I+D.

Propuesta de una PaaS con generación automática de servicios por exposición de aplicaciones legacy para ser consumidos por clientes SaaS. Definición y diseño de una plataforma que se sustente en los conceptos de SaaS, IaaS y Cloud Computing con transferencia a ámbitos estatales o municipales.

2. Departamento/Instituto de radicación:

Instituto de Ingeniería y Nuevas Tecnologías.

3. Línea de Investigación y Desarrollo de pertenencia:

(Marque con una cruz lo que corresponda)

Prioritaria	X	Complementaria	
-------------	---	----------------	--

Denominación:

- 6.- Aplicación de nuevas tecnologías en Ingeniería de Software.

4. Tipo de Proyecto:

(Marque con una cruz lo que corresponda)

Acreditable	X	Reconocimiento institucional	
-------------	---	------------------------------	--

5- Período de vigencia:

01/03/2023 al 31/12/2024

6. Justificación del Proyecto

(Máximo 1600 palabras. Desarrolle el objeto y problema del Proyecto así como el interés, la relevancia del Proyecto)

Algunas organizaciones cuentan con complejos sistemas, compuestos por centenares de miles de líneas de código, con una vida superior a los diez años, pudiendo llegar incluso a los veinte. Estos sistemas suelen representar el corazón del negocio de la organización y no es fácil sustituirlos por un sistema nuevo sin causar un fuerte impacto en la organización (Espinell et al., 2022) (Müller et al., 2000).

Estos sistemas, creados por diversos equipos de trabajo, sometidos a mantenimiento correctivo y perfectivo a lo largo de años, suelen contar con escasa documentación y de baja calidad. La arquitectura de los sistemas está basada en la tecnología existente en el momento de la concepción, que distará bastante de la tecnología actual (Tilley et al., 1995).

El mantenimiento de estos sistemas se vuelve tedioso y muy costoso (Espinel et al., 2022).

El desconocimiento de interfaces claras con estos sistemas hace muy difícil su integración con nuevos sistemas (Fanomezana et al., 2022).

Es necesario encontrar nuevos enfoques para modernizar estos sistemas, que, basándose en tecnología actual, permitan adaptarlos en tiempo y costo aceptables (Monteiro et al., 2022).

En (Favre, 2004) se sientan los fundamentos de la modernización de software dirigida por modelos -Model Driven Software Modernization (MDSM)-, basado en el paradigma MDE (Model Driven Engineering). Un proceso de modernización de software implica cambios sustanciales en un sistema informático existente (sistema de origen); no se trata de un proceso de mantenimiento del software (Pressman, 2010). Las técnicas empleadas son el metamodelado y las transformaciones entre modelos (Almonte et al., 2022).

La reingeniería de un sistema es un tipo de modernización disciplinado, cuyo fin es mejorar la calidad del sistema (Tilley et al., 1995) (Müller et al., 2000), manteniendo el valor estratégico del sistema existente (Castillo et al., 2021).

Un proceso de **modernización** consta de tres etapas (Hunold et al., 2008):

- 1) Ingeniería inversa;
- 2) Reestructuración;
- 3) Generación o ingeniería directa.

1) La ingeniería inversa se inicia con la inyección de modelos a partir de artefactos (como código o un esquema relacional) del sistema origen.

2) La reestructuración consiste en una sucesión de transformaciones modelo-a-modelo (M2M) que pueden elevar el nivel de abstracción de cada modelo obtenido en función de la semántica del Meta Modelo correspondiente. Cuando el nivel de abstracción llega al experto de negocio, éste puede incorporar nuevo conocimiento a los modelos generados, enriqueciendo su valor para la fase de generación.

3) La generación o ingeniería directa es otra cadena de transformaciones M2M que van disminuyendo el nivel de abstracción, pudiendo llegar hasta obtener código final en el sistema destino.

Un proceso de **migración** es un caso especial de reingeniería donde sólo se cambia la plataforma tecnológica, manteniendo la funcionalidad del sistema (Holley et al., 2004) (Zaragoza et al., 2022).

El proceso de modernización del software comprende el tratamiento de aplicaciones legacy, las cuales no son fáciles de mantener ni de sustituir. La mayoría de los métodos,

actualmente, están basados en el paradigma MDE y contemplan caminos de modernización como (Salvatierra et al., 2012) (Espinel et al., 2022):

- La ingeniería inversa que va de niveles bajos de abstracción que empiezan con las plataformas tecnológicas que dan origen al proceso: código y bases de datos.
- Ingeniería directa, que permite transformar modelos de alto nivel de abstracción en modelos de menor nivel, hasta llegar al código del nuevo sistema destino.

7. Estado actual del conocimiento sobre el tema.

(Máximo 2500 palabras. Desarrolle brevemente el marco teórico, los antecedentes y autores más relevantes que hayan tratado la problemática del Proyecto)

El crecimiento del mercado en diferentes ramas de negocios ha provocado que el desarrollo de las TIC's vaya de la mano de las organizaciones para el logro de sus objetivos (Microsoft Web Services, 2022).

Actualmente, las TIC's reflejan el modelo de negocio de una organización y proporcionan el apoyo a la funcionalidad de la misma (Zaragoza et al., 2022).

El deseo de las empresas por crecer y por ser diferentes ante sus competidores obligó a que las TIC's tengan una evolución inmediata. Las empresas, para formar parte del mercado, se dedican a solicitar el desarrollo de aplicaciones basadas en necesidades específicas como ser: contabilidad, liquidación de sueldos, facturación, tasación, notas de débito y crédito, sin tener en cuenta la infraestructura tecnológica de las aplicaciones que forman parte de su software corporativo (Microsoft Web Services, 2022).

Esto ha dado como resultado que las aplicaciones no puedan interactuar entre sí provocando que una operación esencial para la empresa, en lugar de durar instantes, se procese en días o semanas de transacciones. También produce que exista cierta redundancia de datos, por ejemplo, la CUIT de un cliente (Almonte et al., 2022).

El paradigma actual dentro del diseño de arquitecturas, estimula que las aplicaciones de una organización trabajen de forma integrada sin importar bajo qué plataforma están desarrolladas. Es un marco de trabajo que permite a las empresas centrarse en la resolución de sus objetivos sin tener en cuenta las infraestructuras con las que cuentan para llevar a cabo sus negocios (Monteiro et al., 2022).

La evolución física se ha centrado en el uso de redes, estas permiten que varias computadoras puedan trabajar compartiendo recursos informáticos. Los primeros programas fueron los Monolíticos y estaban preparados para ejecutarse en una estación de trabajo. Luego se impusieron los desarrollos basados en Cliente-Servidor y los N-tier. Por último nace Internet, como la herramienta que permite la conexión de millones de computadoras a través de una infinidad de recursos informáticos (Fanomezana et al., 2022). De la mano de Internet surgen los Servicios Web como metodología que promueve la Heterogeneidad e Interoperabilidad entre aplicaciones desarrolladas en diferentes plataformas (Microsoft Web Services, 2022).

Los Sistemas Monolíticos tienen vigencia entre los años 1950 y 1980. Fueron desarrollados en lenguajes como Cobol y Fortran. Estos sistemas tienen como

característica que todos los módulos del software se ejecuten en la misma computadora donde se encuentra la base de datos a la cual acceden.

Los desarrolladores de estos sistemas previeron que era necesario descomponer el software para evitar la complejidad. Gracias a ello nació la Programación Estructurada dividiendo en módulos todos los componentes del sistema (Fanomezana et al., 2022).

Alrededor de los años '80 surgen estos tipos de sistemas. Se definen por tener las aplicaciones que procesan los datos en una computadora Cliente y la base de datos en otra computadora Servidor. Se los denominó Sistemas Cliente-Servidor. Los mismos se comunican a través de conexiones de red. La separación es tanto lógica como física. La principal dificultad que presentaron este tipo de sistemas se basó en que las capas de las aplicaciones que exponían los datos a los usuarios se inmiscuían con las capas que procesaban a los mismos en la base de datos (Pressman, 2010).

Para solucionar la problemática que presentaban los Sistemas Cliente-Servidor, surgieron en los años '90 los Sistemas N-tier ó Multicapas. En este tipo de sistema se representó en diferentes capas a la Interfaz del Usuario, a la Lógica del Negocio y al Manejo de Base de Datos (Pressman, 2010).

Así, los sistemas legacy fueron, son y probablemente seguirán siendo el corazón de muchas organizaciones que automatizaron sus procesos de negocios hace décadas bajo tecnologías que actualmente son anticuadas pero siguen siendo útiles para los usuarios y de gran valor para las empresas.

Hoy en día, el acelerado avance tecnológico, la evolución de los paradigmas en relación a la arquitectura del software y el uso de plataformas hacen que la gran mayoría de los negocios y organizaciones basen sus procesos en sistemas informáticos mucho más evolucionados, lo que hace imprescindible que estos nuevos procesos convivan con los sistemas legacy (Microsoft Web Services, 2022).

Esta clase de sistemas (legacy) al igual que las organizaciones deben evolucionar y adaptarse a nuevos requerimientos.

Según Espinel (Espinel, 2022) las soluciones clásicas relacionadas son:

- el rediseño completo del sistema en una nueva tecnología;
- el wrapping del sistema legacy que provee una nueva interfaz para algún componente de éste, lo que permite mayor accesibilidad desde otras aplicaciones
- la migración o transformación que mueve al sistema legacy a un nuevo ambiente o plataforma más flexible, reteniendo la funcionalidad y los datos del sistema original.

Una de las técnicas para migrar sistemas legacy hacia nuevas tecnologías se logra realizando llamadas a las funcionalidades del sistema legacy existente mediante el uso del Wrapper (Zaragoza, 2022).

Mediante wrapper, es posible adaptar al sistema legacy, para que sea parte de una nueva generación de sistemas, sin los riesgos inherentes a la reescritura y los altos costos de nuevos desarrollos (Zaragoza, 2022). Es evidente, que para que esto no introduzca problemas a futuro, el estado de salud del sistema debe ser “sano” o debe ser factible pasarlo a dicho estado.

Wrapper implica una función contenedora del sistema legacy, a modo de subrutina, en una biblioteca de software cuyo propósito principal es llamar al sistema con poco o ningún cálculo adicional.

Los estudios de Zaragoza (Zaragoza, 2022) indican que los sistemas legacy son considerados potencialmente problemáticos por numerosos ingenieros de software por diversos motivos. Muchas organizaciones no han logrado los resultados esperados tras migrar, re implementar o integrar sus sistemas legacy.

Muchos de los sistemas de información todavía corren sobre plataformas legacy pero el deseo de reemplazar estos sistemas se ve limitado por varias restricciones tales como (Monteiro et al., 2022):

- Grandes inversiones acumuladas en el sistema legacy por un largo ciclo de vida.
- Disminución del personal técnicamente habilitado.
- Pérdida significativa de conocimiento sobre las tareas internas que trabajan estos sistemas.

Entre los muchos ejemplos de sistemas actuales que siguen funcionando bajo plataformas legacy, podemos mencionar:

- Aplicaciones gubernamentales.
- Sistemas contables.
- Sistemas bancarios.
- Aplicaciones Rapid Application Development (RAD) orientadas a interfaces gráficas.
- Sistemas empresariales basados en procesos de negocios.

Otro aspecto importante a tener en cuenta al momento de migrar sistemas legacy son las interfaces de usuario (Grechanik, 2007). Hace tiempo atrás era muy común el desarrollo RAD lo que generó el problema del acoplamiento entre la interfaz de usuario y la lógica de negocios.

8. Objetivos general y específicos

Objetivo Principal.

- Estudiar la posibilidad de la migración de aplicaciones legacy que expongan sus funcionalidades como servicios generados automáticamente en una plataforma que se sustente de tecnologías PaaS (Platform as a Service), SaaS (Software as a Service), IaaS (Internet as a Service) y Cloud Computing.

Objetivos Secundarios.

- Analizar el funcionamiento interno de las tecnologías de desarrollo impuestas por el mercado. Comparar y seleccionar la que por sus características resulte ser la más adecuada.
- Analizar los frameworks y librerías para exposición de metadatos y seleccionar de ellas la que mejor se adapte a la tecnología propuesta precedentemente.
- Modelar la estructura interna de una aplicación legacy. Definir posibles puntos de extensión.

- Definir un mecanismo de generación automática de servicios a partir de la estructura de una aplicación legacy.

Desarrollar una plataforma que permita exponer una aplicación legacy como servicio.

9. Hipótesis de la Investigación

(Máximo 500 palabras)

Hipótesis 1:

Es posible analizar las estructuras internas y los metadatos de las aplicaciones ejecutables existentes, abrirlas virtualmente, y localizar los puntos de extensión que permitan exponer automáticamente sus funcionalidades como servicios, e integrarlas con herramientas de terceros.

Hipótesis 2:

Es posible desarrollar una plataforma para generación automática de servicios que involucre tecnologías tales como PaaS, SaaS, IaaS y Cloud Computing, tal como se plantea en la Figura 1.

La IaaS permite a las empresas utilizar sistemas de funcionamiento, aplicaciones y almacenamiento basados en la web sin tener que comprar, administrar y brindar soporte a la infraestructura de nube subyacente.

SaaS permite a los usuarios conectarse a aplicaciones basadas en la nube a través de Internet y usarlas.

PaaS es un marco en el cual los desarrolladores pueden desarrollar o personalizar aplicaciones. Con PaaS, los desarrolladores pueden crear aplicaciones web rápidamente sin instalar ninguna herramienta o software .

Cloud computing es la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet.

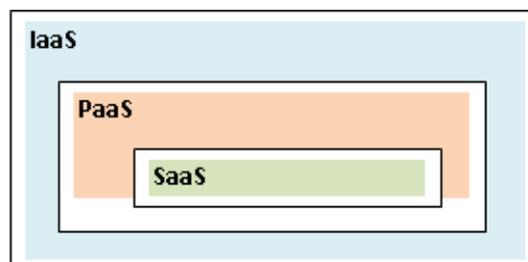


Figura 1: Integración con IaaS

10. Metodología a utilizar.

(Máximo 1600 palabras)

El proyecto está dividido en dos fases con cuatro etapas cada una de ellas.

La primera fase establecerá el marco teórico necesario y el estado del arte respecto de las arquitecturas orientadas a servicios (SOA) y de las metodologías existentes que posibiliten la generación automática de servicios a fin de determinar la posibilidad de exponer aplicaciones propietarias de la empresa a modo de servicios.

La segunda fase desarrolla e implementa el resultado de la primera fase con transferencia a estamentos estatales y/o municipales de la zona de influencia de la UNO.

Primera Fase

Etapa 1. Se relevarán las tecnologías existentes para desarrollo de software. Se seleccionará una de las metodologías relevando su funcionamiento interno y asegurando que posea las características adecuadas para el proyecto y un grado de inserción importante en el mercado.

Etapa 2. Se diseñará un protocolo para exposición de los metadatos que presente la metodología seleccionada en la Etapa 1 de esta fase. Asimismo se relevarán todos los frameworks y librerías existentes para el manejo de dichos metadatos.

Etapa 3. Diseño de un modelo para representar la estructura interna de una aplicación legacy. Se estudiarán diversas estrategias para encontrar puntos de extensión en la aplicación.

Etapa 4. Diseño de la metodología para la generación del modelo planteado en la Etapa 3 de la presente fase a partir de la exploración de los metadatos de una aplicación. La metodología que se proponga como resultante de la investigación contemplará la generación de un conjunto de servicios que expongan su funcionalidad mediante implementación de PaaS, SaaS, IaaS y Cloud Computing.

Segunda Fase:

Etapa 5. Se analizarán y documentarán los requerimientos que debe cumplir la plataforma que genere automáticamente los servicios de las aplicaciones legacy.

Etapa 6. Se realizará el diseño de la plataforma y la especificación técnica.

Etapa 7. Se desarrollará la plataforma en base a los requerimientos y las especificaciones técnicas definidas en las etapas anteriores.

Etapa 8. Se probará el uso de la plataforma exponiendo varias aplicaciones legacy como servicios.

11. Resultados Esperados

(Máximo 800 palabras)

La plataforma que surja como resultado de la presente investigación permitirá:

- Evitar reinvertir tiempo y dinero en volver a desarrollar aplicaciones existentes que necesiten integrarse con otras.
- Aprovechar las ventajas de los servicios sin necesidad de un desarrollo costoso y extenso.
- Exponer la funcionalidad de una aplicación legacy como servicio.

Se propone para el presente proyecto la transferencia específicamente a distintos estamentos estatales o municipales.

12. Antecedentes y funciones previstas del Grupo de Investigación en el área temática/disciplina

(Máximo 500 palabras)

El Director de este proyecto tiene categoría III como docente investigador del Ministerio de Educación de la Nación.

Ha integrado seis (6) proyectos acreditados del Programa de Incentivos a Docentes Investigadores del Ministerio de Educación, aprobados y financiados por la Universidad de Buenos Aires en el marco de la Programación UBACyT, relacionados al área de Inteligencia Artificial, Sistemas Inteligentes, Minería de Datos, Procesamiento Digital de imágenes y documentos degradados e Inclusión Social a partir de la Educación Tecnológica.

Ha revistado como Director de cuatro (4) proyectos de investigación acreditados del Programa de Incentivos a Docentes Investigadores del Ministerio de Educación, aprobados y financiados por la Universidad de Buenos Aires en el marco de la Programación UBACyT, relacionados al área de Sistemas Inteligentes Autónomos, Arquitectura de Software en contextos de datos masivos e Innovación Pedagógica vía TIC para la mejora de la calidad educativa.

Ha sido evaluador de proyectos para subsidio a emprendedores para el FONSOFT, dependiente del Ministerio de Ciencia y Tecnología de la Nación.

Es autor de numerosas publicaciones en diversos Congresos, Simposios, Eventos y Revistas nacionales e internacionales, con y sin referato.

El otro integrante de este proyecto tiene categoría V como docente investigador del Ministerio de Educación de la Nación.

Ha integrado proyectos de investigación en FIUBA y UNO.

Ha dirigido un proyecto de extensión en la UNO.

Ha actuado como evaluador y chair en diversas ediciones del CONAIISI.

Ha actuado como jurado en el "rally latinoamericano de innovación 2022, sede FIUBA"

Es autor de numerosas publicaciones en diversos Congresos, Simposios, Eventos y Revistas.

13. Transferencia de Resultados.

(Máximo 800 palabras. Detalle el objeto de la transferencia, su importancia, los destinatarios concretos o posibles y los procedimientos para concretarla)

Los derechos de la propiedad intelectual y/o la patente o registro de la plataforma que surja como resultado de la presente investigación quedará a nombre único y exclusivo de la Universidad Nacional del Oeste sin poder el Director ni integrantes de este proyecto hacer uso total o parcial de la plataforma sin la expresa autorización de dicha Casa de Altos Estudios.

El presente trabajo de investigación podrá articular con los distintos estamentos estatales y/o municipales del área de influencia de la Universidad Nacional del Oeste.

Este proyecto también es aplicable directamente a otras áreas tales como Videojuegos, Informática Industrial y aplicaciones web.

14. Viabilidad y Factibilidad Técnica

(Máximo 500 palabras)

El Proyecto se llevará a cabo en los laboratorios de la Universidad Nacional del Oeste. Se dispone del equipamiento de dichos laboratorios de investigación.

El resto de los recursos y requerimientos necesarios para llevar a cabo las tareas propuestas en la presente investigación se presentan en el ítem 19.

Los antecedentes académicos y de investigación del grupo otorgan la viabilidad y factibilidad técnica suficientes para el desarrollo de la investigación.

15. Aspectos Éticos.

(Si corresponde máximo 500 palabras)

Por sus características, el presente proyecto no está relacionado a aspectos éticos relevantes.

16. Aspectos de Seguridad Laboral, Ambiental y Bioseguridad requeridos

(Si corresponde máximo 500 palabras)

No corresponden al presente proyecto impactos relacionados con aspectos de seguridad laboral, ambiental o de salud.

17. Intervención de terceros

(Justifique la intervención de terceros y anexe los Convenios o Acuerdos específicos requeridos para su intervención)

No corresponde.

18. Cronograma de Actividades.

Detalle las actividades propuestas. Consigne separadamente cada actividad unitaria.

1^{er} Año

Actividad	Mes											
	1	2	3	4	5	6	7	8	9	10	11	12
Etapa 1.												
Actividad 1. Revisión sistemática de la bibliografía.	X	X										
Actividad 2. Se relevarán las tecnologías existentes para desarrollo de software. Incluye IaaS, SaaS, PaaS.		X	X									
Actividad 3. Selección del orden de prueba de las metodologías.			X									
Actividad 4. Selección de una metodología.			X									
Actividad 5. Relevamiento del funcionamiento interno de la metodología seleccionada, asegurando que posea las características adecuadas para el proyecto y un grado de inserción importante en el mercado.				X	X							
Etapa 2.												
Actividad 6. Análisis de un protocolo para exposición de metadatos.				X	X							
Actividad 7. Diseño del protocolo para exposición de los metadatos que presente la metodología seleccionada en la Etapa 1 de esta fase.						X	X					
Actividad 8. Relevamiento de todos los frameworks y librerías existentes para el manejo de los metadatos definidos en la Actividad 7 de esta misma Etapa.							X	X				
Etapa 3.												
Actividad 9. Se estudiarán diversas estrategias para encontrar puntos de extensión en las aplicaciones.								X				
Actividad 10. Análisis de las distintas estructuras									X			

requerimientos y las especificaciones técnicas definidas en las etapas anteriores.																				
Actividad 21. Implementación de la plataforma en un entorno de desarrollo.																			X	X
Etapa 8.																				
Actividad 22. Prueba de la plataforma exponiendo varias aplicaciones legacy como servicios. Testing.																				X
Actividad 23. Transferencia e implementación de la plataforma en un entorno de trabajo. Articulación con estamentos estatales y/o municipales.																				X

19. Presupuesto

Presupuesto del Primer año de ejecución

	Rubro	Descripción	Monto
1	Bienes de consumo	Gastos varios	30.000
2	Servicios no personales		
3	Servicios técnicos y profesionales		
4	Servicios comerciales y financieros	Acceso a servicios en la nube	30.000
5	Pasajes y viáticos		
6	Bienes de uso	Libros técnicos, suscripción a revistas científicas Software específico	30.000
7	Equipamiento	Notebook	480.000
Total 1° Año			\$570.000

Presupuesto del Segundo año de ejecución

	Rubro	Descripción	Monto
1	Bienes de consumo	Insumos de oficina	30.000
2	Servicios no personales	Mantenimiento de equipo	30.000
3	Servicios técnicos y profesionales	Asistencia a cursos Técnicos Específicos TICs	50.000
4	Servicios comerciales y financieros	Acceso a servicios en la nube	50.000
5	Pasajes y viáticos	Inscripción a Congresos Asistencia a Cursos y Congresos (Sólo pasajes y viáticos)	40.000 30.000
6	Bienes de uso		
7	Equipamiento		
Total 2° Año			\$230.000

Rubros

- Bienes de consumo: insumos de laboratorio, útiles de oficina, librería, fotocopias, etc.
- Servicios no personales: alquiler de equipos y mantenimiento, etc.
- Servicios técnicos y profesionales: traducciones, desgrabaciones, data-entry, etc.
- Servicios comerciales y financieros: imprenta, internet, transporte y almacenamiento, etc.
- Pasajes y viáticos en ámbito nacional, inscripciones a congresos nacionales o internacionales.
- Bienes de uso: libros, revistas, programas de computación, etc.
- Equipamiento

20. Referencias bibliográficas

(Consigne la bibliografía utilizada para la formulación del Proyecto)

Almonte, L., Guerra, E., Cantador, I. (2022). Recommender systems in model-driven engineering. *Softw Syst Model* 21, 249–280. <https://doi.org/10.1007/s10270-021-00905-x>

Barriga, A., Rutle, A., Heldal, R. (2020). Improving model repair through experience sharing. *J. Object Technol.* 19(2):13:1-21

Castillo, R., Guzmán, I. (2021). Knowledge Discovery Metamodel-ISO/IEC 19506: A standard to modernize legacy systems. *Research Gate*. University of Castilla-La Mancha 2021. Web. 25 Jul. 22

- Espinel, G. P., Carrillo Medina, J. L., Flores Calero, M. J., Urbietta, M. (2022). Software Configuration Management in Software Product Lines: Results of a Systematic Mapping Study. *IEEE Latin America Transactions*. vol. 20, no. 5, pp. 718-730, May 2022, doi: 10.1109/TLA.2022.9693556.
- Fanomezana M. L., Rapatsalahy A. M., Razafindrakoto N. R., Bădică C. (2022). Proposed Methodology for Designing a Microservice Architecture. *2022 23rd International Carpathian Control Conference (ICCC)*, 2022, pp. 303-308, doi: 10.1109/ICCC54292.2022.9805930.
- Favre, J. M. (2004). Foundations of model (driven)(reverse) engineering: Models. *Proceedings of the International Seminar on Language Engineering for Model-Driven Software Development*, Dagstuhl Seminar 04101.
- Grechanik, M., Conroy, K. (2007). Composing Integrated Systems Using GUI-Based Applications And Web Services, Web Services (ICWS) 2007 *IEEE International Conference on*, pp., 2007. Web. 30 May. 2022.
- Holley, K, Channabasavaiah, K. (2004) Migrating to a service-oriented architecture. IBM Global Services. IBM Corporation Software Group 2004. Web. 5 Jul. 2016.
- Hunold, S., Korch, M. (2008). Transformation of Legacy Software into Client/Server Applications through Pattern-Based Rearchitecturing. Research Gate. Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual *IEEE International*. Web. 15 May. 2022.
- Müller, H. A., Jahnke, J. H., Smith, D. B., Storey, M. A., Tilley, S. R., & Wong, K. (2000): Reverse engineering: A roadmap. In Proceedings of the Conference on the Future of Software Engineering (pp. 47-60). ACM.
- Monteiro, A., Vieira, G. (2022). Guiding legacy systems for evolution. PmatE: a case study of maintenance and engineering. *Journal of Information Systems Engineering and Management*, 7(1), 11689. <https://doi.org/10.55267/iadt.07.11689>
- Pressman, R.S. (2010). Ingeniería del software. Un enfoque práctico. McGraw-Hill, 7ªEd, ISBN: 978-607-15-0314-5.
- Salvatierra, G., Zunino, A. (2012). Towards a Computer Assisted Approach for Migrating Legacy Systems to SOA. *Research Gate*. International conference on Computational Science and Its Applications 2012. Web. 10 May. 2022.
- Tilley, S. R., Smith, D. (1995). Perspectives on legacy system reengineering. Technical report, Software Engineering Institute, Carnegie Mellon University.
- Zaragoza, P., Seriai, AD., Seriai, A., Shatnawi, A., Bouziane, HL., Derras, M. (2022). Materializing Microservice-oriented Architecture from Monolithic Object-oriented Source Code. In: Fill, HG., van Sinderen, M., Maciaszek, L.A. (eds) *Software Technologies*. ICSoft 2021. Communications in Computer and Information Science, vol 1622. Springer, Cham. https://doi.org/10.1007/978-3-031-11513-4_7

Otra Bibliografía utilizada:

"CodeTrigger." Herramienta de generación SOA. Web. 4 Jul 2022.
<<https://www.codetrigger.com>>.

"IBM SOA." Integrate legacy systems into your SOA initiative. Web. 4 Jul 2022.
<<https://www.ibm.com/developerworks/library/ws-soa-legacyapps/>>.

"Microsoft Web Services." Web Services. Microsoft, 2016. Web. 4 Jul 2022.
<<https://msdn.microsoft.com/en-us/library/ms950421.aspx>>.

" Migration from Legacy Systems." Migration from Legacy Systems. Web. 2 Jul 2022.
<<https://www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.50/2013/Topic_2_NZ_Legacy_Migration_Final.pdf>>.

" Transformation of Legacy Systems." Software Reengineering at the Architectural Level: Transformation of Legacy Systems. Web. 2 Jul 2022.
<<https://www.researchgate.net/publication/228943033_Software_Reengineering_at_the_Architectural_Level_Transformation_of_Legacy_Systems>>.