

INGENIERÍA DE SOFTWARE, SISTEMAS DE INFORMACIÓN Y MÉTODOS FORMALES: DESAFIOS Y APLICACIONES EN PANAMÁ

SOFTWARE ENGINEERING, INFORMATION SYSTEMS AND FORMAL METHODS: CHALLENGES AND APPLICATIONS IN PANAMA

Gordon Graell, Roberto Daniel



 Roberto Daniel Gordon Graell

roberto.gordon@up.ac.pa

Universidad de Panamá, Facultad de Informática, Electrónica y Comunicación del Centro Regional Universitario de Panamá Oeste., Panamá

Revista Saberes APUDEP

Universidad de Panamá, Panamá

ISSN-e: 2644-3805

Periodicidad: Semestral

vol. 6, núm. 1, 2023

revistasaberesapudep@gmail.com

Recepción: 15 Mayo 2022

Aprobación: 13 Agosto 2022

URL: <http://portal.amelica.org/ameli/journal/223/2233750010/>

Resumen: La ingeniería del software como disciplina profesional está en constante avance y transformación. Es una práctica que se hace disruptiva en la medida en que los productos resultantes obligan nuevas investigaciones y desarrollos. Como técnica, el desarrollo de los sistemas de información cuenta con variadas metodologías que se adaptan a las necesidades del desarrollador/usuario o del cliente final/usuario. Una de esas técnicas se denomina Metodologías Formales, cuya principal característica es que se fundamenta en el proceso lógico y el conocimiento de la ciencia matemática para el diseño de modelos teóricos previo la construcción de los algoritmos. Los métodos que ofrece son los preferidos en ambientes muy técnicos y ecosistemas digitales especializados que requieren de respuestas exactas, a eventos en tiempo real, sin la presencia humana, que solucionan problemas de sistemas y aportan confiabilidad en los resultados. El artículo que se presenta es un análisis descriptivo de la utilización de los métodos formales en varios campos del conocimiento humano y áreas de producción, que pueden ser aplicados en Panamá. Es una investigación documental cuya principal conclusión es que la metodología formal para el desarrollo de ecosistemas digitales científico técnicos, más que una opción, es la solución adecuada.

Palabras clave: Diseño, ingeniería, métodos formales, sistemas de información.

Abstract: Software engineering as a professional discipline is constantly advancing and transforming. It is a practice that becomes disruptive to the extent that the resulting products force new research and development. As a technique, the development of information systems has several methodologies that adapt to the needs of the developer/user or the end customer/user. One of these techniques is called Formal Methodologies, whose main characteristic is that it is based on the logical process and the knowledge of mathematical science for the design of theoretical models prior to the construction of algorithms. The methods it offers are preferred in highly technical environments and specialized digital ecosystems that require accurate responses to events in real time, without human presence, that solve system problems and provide reliability in the results. The article presented is a descriptive analysis of the use of formal methods in various fields of human knowledge and production areas, which can be applied in Panama. It is a

documentary research whose main conclusion is that the formal methodology for the development of scientific-technical digital ecosystems, more than an option, is the appropriate solution.

Keywords: Design, engineering, formal methods, information systems.

INTRODUCCIÓN

La ciencia de la informática es el diseño de sistemas de información. La sistematización lógica del dato seleccionado para integrar un ecosistema cuyo resultado es una herramienta de uso computacional para un usuario determinado sea este humano o artificial. Su fundamento es la teoría de los sistemas y, dentro de este, las diversas metodologías existentes para el diseño y desarrollo de algoritmos (Carbo et al, 2021)

Como disciplina, la ingeniería de software evoluciona al ritmo y con las pautas que evolucionan sus desarrollos. Estos, a su vez, se retroalimentan de las formas y funciones de aplicaciones, herramientas, hardware y, en general, todo lo que representa una tecnología de información. En la medida en que los sistemas de información son más complejos, presentan problemas más complejos y requieren de respuestas más acertadas y plausibles. Esa es la historia de la ingeniería del software, que progresó década a década (Piattini, 2016).

La historia de los métodos de construcción de los sistemas de información es, fundamentalmente, ensayo y error. Es la evolución desde métodos perceptivos, que, en un principio, no se diferenciaban del desarrollo de hardware hasta las diferentes metodologías que se aplicaron desde que la actividad computacional se hizo masiva allá por la década de los sesenta del siglo pasado. En la actualidad el papel de las metodologías de desarrollo de software para crear, o mantener, un producto de calidad para cumplir los requisitos particulares de los usuarios es fundamental e indispensable de la ingeniería de software (Zumba y León, 2018).

La definición de metodología de inmediato refiere a un proceso organizado, trae implícita varios conceptos a saber: Orden, secuencia (que responde a una lógica), objetivo y resultados esperados. La literatura científica del tema muestra diferentes metodologías, cada una con sus particularidades. Lo que las diferencia es el proceso que asumen para el desarrollo de los programas que se aplican a diferentes elementos como objeto eje del como son: La función que realizará el objeto, la obsolescencia

Las clásicas, llamadas también modelos de proceso prescriptivo, dan una estructura útil al diseño del sistema con un mapa razonablemente eficaz para los equipos de software. Entre ellas se encuentran: Ciclo de Vida o Cascada, incremental, evolutivo, espiral, prototipos y desarrollo basado en componentes. Como “proceso prescriptivo” refieren a un proceso finito hasta un cierto nivel, de igual forma suponen un proceso detrás de otro, para que se complete un paso, previamente debe completarse el que lo antecede por lo que son escalonados (León et al, 2021).

Las orientadas a objetos se basan en técnicas para el desarrollo del software orientado al objeto. Pueden ser: Dirigidas por los datos (data-driven, por su definición en inglés), que se ocupan de la estructura de los objetos, en este caso el dato, tiene tres subdivisiones:

- Dirigidas al dato: Fusión y Técnica de modelado de objetos (OMT por sus siglas en inglés)
- Dirigidas por las responsabilidades (responsability-driven por su definición en inglés): Facilidad de gestión de objetos (OMF por sus siglas en inglés) y Sistema de gestión de objetos (OMS, por sus siglas en inglés)
- Proceso de unificado de desarrollo de software (USDP, por sus siglas en inglés)

Las ágiles suponen que las empresas/usuarios utilizarán el software en todas las operaciones empresariales que sea posible, por lo que los programas se desarrollarán para responder oportunamente y satisfactoriamente a

las necesidades del usuario, en el que se centra su desarrollo. Se encuentran: Programación extrema (XP), Mobile-D para móviles, Scrum, Crystal, Gestión evolutiva de proyectos (Evo, por sus siglas en inglés), Desarrollo basado en características (FDD, por sus siglas en inglés), Desarrollo de software adaptativo (ASD, por sus siglas en inglés), Desarrollo esbelto (LD, por sus siglas en inglés) (Maida y Pacienza, 2015).

Las Metodologías para la web relacionadas con la utilización de principios científicos (hipótesis, prueba, resultado), de ingeniería y gestión, y con enfoques sistemáticos y disciplinados de desarrollo y metas. Entre ellas están: Ingeniería web, diseño de aplicaciones (webapps) y método de diseño de hipermedios orientados a objetos (MDHOO, por sus siglas en inglés) (Molina et al, 2018).

Una metodología formal, es el uso de lógica y matemáticas como técnica aplicada en todos los pasos del diseño de un software específico. Que permita el diseño, el desarrollo, la verificación y validación de un programa en un fin particular. Se determina desde la validez o no validez del algoritmo para realizar la función por lo que prescinde del contenido empírico del razonamiento para considerar sólo la estructura y comprobar los argumentos utilizados para probar todos los aspectos en el diseño e implementación de un sistema de información (Suaza y Lora, 2020).

Los métodos formales en la Ingeniería de Software cumplen funciones básicas de seguridad y especificación de las tareas de los sistemas como son:

- Verificar y aprobar las políticas de los requisitos del sistema de información como propiedades de seguridad, como confidencialidad o integridad de datos.
- Especificar el comportamiento del sistema a través de una descripción matemática basada en su comportamiento por medio de tablas de estado o lógica matemática como forma de establecer sus propiedades críticas.
- Hacer pruebas de correspondencia entre la especificación y los requisitos y establecer y conservar las propiedades de las políticas de los requisitos.
- Hacer pruebas de correspondencia entre el código fuente y la especificación en los componentes críticos del sistema.
- Hacer pruebas de correspondencia entre el código máquina y el código fuente, de considerarse necesario. (Serna, 2021)

Suaza y Lora, (2020) consideran que la importancia de los métodos formales está en su rigor y lógica de desarrollo.

La importancia de los métodos formales yace en sistematizar e introducir rigor en todas las fases de desarrollo del software, en especial en la Ingeniería de Requisitos, con lo que sería posible evitar que se pasen por alto problemas críticos. (p. 122)

Los autores señalan que las metodologías formales estandarizan el proceso de trabajo de un proyecto debido a que son una base coherente para cohesionar las actividades relacionadas, que son muchas, gracias a los mecanismos de descripción, que son precisos y no dan espacio para interpretaciones diferentes o ambigüedades. Los métodos se fundamentan en enunciados bien definidos en lógica matemática, específicos, claros y únicos, con reglas de inferencia para cada paso y que deben permitir la verificación y comprobación mediante procesos mecánicos.

Aunque existen variadas metodologías que, según sus principios epistemológicos, pueden clasificarse de formales, la que más los representa, y se utiliza, son los fundamentados en la red de Petri (RdP). Lleva el nombre de su diseñador, Carl Petri, quien la presentó en la década de los 60 del siglo XX.

Es un método formal que consiste en la representación matemática, y gráfica, de un sistema completo a través de sus eventos discretos. Es un grafo de flujo, dirigido y bipartido, que se orienta con dos tipos de nodos: Lugares, representados mediante circunferencias; y Transiciones (representadas por segmentos rectos verticales, que se conectan por medio de arcos o flechas. (Vega, Lao et al, 2020)

Para Osorio (2021) básicamente son una herramienta para una representación de los encadenamientos del sistema de información conveniente para el diseñador, pero que, no necesariamente aporta a la simplificación computacional en la solución del problema de la secuencia óptima de disparos de transiciones (OTFSP, por sus siglas en inglés) es un problema combinatorio de secuenciación que puede generar muchas alternativas de solución para cada evento singular.

Este artículo se centra en el análisis de ambientes y ecosistemas computacionales para los que, el desarrollo de software con precisión la matemática que aportan los métodos formales, son un requerimiento, más que una opción. Es una muestra descriptiva de los campos del conocimiento hacia los que deben orientarse la formación de ingenieros y desarrolladores de software de Panamá que garantice el avance progresivo de la especialidad y sus aplicaciones profesionales.

MATERIALES Y MÉTODOS

Se realizó una revisión documental, de enfoque cualitativo de sobre las características de los software e investigaciones científicas, basados en los métodos formales de diseño de la ingeniería del software, en la comunidad científica de habla hispana con trabajos publicados: Tesis de grado, artículos científicos, contenidos de cátedras y páginas web.

Con apoyo de la barra de búsqueda general de Google y los buscadores académicos Google Scholar, Redalyc y Scielo, se realizó una búsqueda sistematizada entre los años 2018 al 2022, incluidos, con la combinación sin conectores booleanos, de palabras claves: Métodos formales, red de Petri, sistemas de información, sistemas computacionales y diseño de sistemas informáticos. Se encontraron más de 6000 documentos en campos diversos como son: Ingeniería de sistemas e industrial, investigación médica, investigación social en estadísticas de salud, reflexión científica, educación científica e informática.

Ante la imposibilidad de un estudio cualitativo de gran envergadura se escogió una muestra, a través de la lectura de resúmenes, con la selección de 15 posibles trabajos de estructura y divulgación científica, de los cuales se seleccionaron 4 cuya lectura de contenido cumple criterios de inclusión para el estudio a saber: Estructura de investigación científica, diseño de investigación con métodos formales, bibliografía científica y conclusiones acordes al objetivo de trabajo.

RESULTADOS

Pinto et al (2018) realizaron una revisión sistemática de literatura (RSL), en documentos científicos en idioma inglés, aplicada a la seguridad en requerimientos de software de sistemas críticos ferroviarios como principal criterio de calidad.

“[...] centrándose en la seguridad, uno de los atributos de calidad más importantes en estos tipos de sistemas, ya que de ellos pueden depender en gran medida la viabilidad económica y funcional del sistema, la conservación de su entorno e incluso la preservación de vidas humanas.” (p. 37)

Entre sus resultados destacan que un 53.1% de los desarrolladores prefirieron métodos formales, un 34.4% métodos estructurados y un 12.5% métodos mixtos. La lectura permite afirmar que un 65.6% se decantó por la utilización de los métodos formales. Aunque también afirman que su implementación se dio, casi siempre, parcialmente e integrando otras metodologías y herramientas para lograr una comprensión más completa del problema. Su uso es muy recomendado por los estándares de calidad a cumplir.

Los problemas del transporte de datos en los sistemas de información es una eventualidad recurrente a la que se enfrenta el diseño del software. En las medidas en que crecen las redes de información, se multiplican los usuarios o se requiere de ecosistemas digitales específicos y muy especializados, se presentan los problemas de

latencia (Chanchí et al, 2020). Es el retraso en la entrega del dato, en el tiempo computacional del algoritmo, que puede incluir hasta la pérdida de este.

Osorio (2021) presenta un trabajo de ingeniería en el cual utiliza las redes de Petri y los algoritmos genéticos como herramientas para problemas de ruteo, de transporte de datos. Una definición sintética de algoritmos genéticos es que es el que se construye aleatoriamente desde una población, de datos, inicial que proveerá una cadena de “generaciones” algorítmicas que, en la medida en que aparecen, seleccionará la más adecuada, “fuerte” como generación sobreviviente.

Se empieza pues, con una población inicial predeterminada, comúnmente generada aleatoriamente, a partir de la cual se inicia el proceso de evolución iterativo del algoritmo. Cada iteración produce una generación nueva con nuevos cromosomas que son descendencia de la anterior, y consiste en evaluar a los individuos de la población usando la función de fitness, también denominada función de adaptación, a partir de la cual se definirán los mejores individuos para la evolución, con el objetivo de que esos mejores cromosomas tengan mayores probabilidades de sobrevivir [...] (p. 15)

Es interesante la adaptación de la jerga científica biológica en el problema computacional. Su principal conclusión es que la aplicación de las redes de Petri, para dar sentido a la generación algorítmica de la función del sistema de información en el transporte de datos, hace que la integración sea una herramienta de modelamiento en la resolución de problemas combinatorios, logrando una representación matemática previa de cada evento singular, comprobable por medios mecánicos, que se traduce en la minimización de errores, o posibilidad de error de traducción de acciones, al usuario. En relación con la metodología formal Osorio (2021) señala la calidad de su adaptabilidad, “[...] la naturaleza tiempo dependiente del problema se aborda adecuadamente a través de la representación en RdP del problema, a la cual no es necesario hacerle ningún cambio, así como tampoco a la lógica de simulación de la red.” (p. 72)

Un sistema embebido es un circuito cerrado que cumple una función muy específica. La mayoría son reactivos, Se basa en microprocesador, o microcontrolador, que se diseña para pocas, o una sola función, frecuentemente en un sistema de información en tiempo real.

Micolini et al (2018) presentan sus características como: “Estos sistemas tienen que cumplir requerimientos no funcionales específicos dado que interactúan con variables y eventos del propio sistema y del mundo exterior, donde los datos y eventos son heterogéneos y no determinísticos.” (p.1) que se requieren para productos modulares, sencillos, verificables y con posibilidad de mantenimiento.

Los autores integran una RdP a un monitor como modelo lógico, del sistema ejecutable, en sistemas reactivos guiados por eventos que permita: Desacoplar la lógica, la sincronización y ejecución de las acciones y determinar el orden lógico de atención de eventos, la política que decide la ejecución ante conflictos, la cantidad de hilos necesarios y la responsabilidad de los hilos además de lograr mantenibilidad y flexibilidad (Micolini et al, 2018)

La mayoría de los sistemas embebidos son en tiempo real, en los que no se puede predecir la ocurrencia de un evento. Realizan control, supervisión y procesamiento de señales, como los sistemas de tráfico. Se utilizan en protocolos de comunicación e interfaces hombre-máquina. Son procesos incompletos que interactúan con el entorno para producir un cierto comportamiento en el sistema. Pueden estar sujetos a requisitos rigurosos de tiempo real y, frecuentemente, ejecutar varios procesos en paralelo.

Se basaron en RdP sincronizadas, o no autónomas, para modelar sistemas cuyos disparos, modos que las pueden accionar, son sincronizados con el evento externo. Sincronizan el cambio de estado exterior (al sistema) con el evento interno (del sistema). Las RdP, tanto transaccional como de tiempo y el uso de guardas, les permitieron graficar los eventos con exactitud los eventos singulares como paso fundamental del diseño del sistema embebido que se requiera.

Los ambientes robóticos son una marca de la era que vive la humanidad. El Internet de las Cosas (IoT) la conectividad maquina a máquina (M2M) que se encuentra en la industria 4.0, desde la complejidad productiva grandes establecimientos manufactureros hasta sistemas sencillos de

almacenamiento automático. Determina el inicio de la cuarta revolución industrial con la digitalización de procesos productivos e hiperconectividad (Martínez et al, 2020).

Mahulea et al (2021) publica un documento de trabajo con una estrategia de planificación de trayectorias de robots móviles, basada en el uso de modelos definidos con redes de Petri, para los que no funcional otras metodologías para determinar de antemano la secuencia de configuraciones que necesita cada robot, que les permita su función fundamental, moverse en trayectorias libres de colisión en un ambiente con obstáculos y otros robots interactuantes.

La trayectoria se define como el camino a recorrer desde una posición inicial ' hasta una final, en la que la visita programada de una serie de configuraciones planificadas, en un entorno que integra un conjunto de obstáculos como impedimentos estáticos, que remite a un mapa y una posición objetivo que el robot alcanza desde una ubicación inicial.

El diseño del sistema de información requirió de una arquitectura en capas de navegación, típica de un robot móvil, a saber: La capa más alta es la planificación de trayectorias; la siguiente el controlador de movimientos para que el robot siga la trayectoria con la mayor precisión posible a la trayectoria de referencia de la capa de planificación; la tercera son controladores de bajo nivel para que las acciones generadas sean aplicadas por los actuadores del robot y la cuarta es de la localización, que determina su posición actual. (Mahulea et al, 2021).

La RdP fue utilizada para programar las condiciones del movimiento coordinado cuando interactúan muchos robots móviles en un espacio determinado asumiendo que, cada unidad, cumple con los requisitos aportados por las capas. caracterizaron el movimiento por medio de ejecuciones/transiciones en de un sistema discreto. En conjunto con la teoría de conjuntos y diversas fórmulas matemáticas se creó un mapa del espacio dividido en celdas. De manera general La RdP presenta: Un conjunto finito de lugares P; Un conjunto finito de transiciones T, de posibilidad de movimiento; Matriz de post- incidencia definiendo los arcos de transiciones de movimiento; Matriz de pre- incidencia definiendo los arcos de lugares a transiciones de movimiento; El marcado inicial que corresponde al número de robots que se encuentra inicialmente en la celda inicial y El conjunto de los símbolos de salidas (Mahulea et al, 2021).

Mahulea et al (2021) consideran que los modelos de RdP permiten obtener modelos escalables, de igual funcionamiento, por lo que, al agregar robots, la estructura del modelo no cambia.

La computación biológica, la biología computacional y la bioinformática son ciencias en las que el dato de origen biológico se complementa con su representación digital a través de técnicas y operaciones matemáticas. Se recurre a métodos analíticos, modelos matemáticos y simulación digital. Sus programas, equipos y herramientas digitales están diseñadas para la secuenciación lógica de datos de origen genético, como cadenas de proteínas, en algoritmos muy exactos que permiten predecir comportamientos y funciones (Campos et al, 2021).

Clavijo et al (2018) presentan un trabajo centrado en la evaluación de un modelo de “[...] una red génica simple para la síntesis de una proteína, analizando la red de Petri producto del trabajo de Goss y Peccoud en 1998 [...]” (p.9) que supone la posibilidad de predicción de la activación de genes para la secuenciación de proteínas. Para el análisis, desde la visión estocástica, primero se esquematizó un modelo biológico, por medio de una red de Petri, con un modelo génico simple para la síntesis de una proteína, en la cual los procesos de activación-inactivación, del gen, se representan por las reacciones de las ecuaciones matemáticas determinadas.

El modelo biológico de la red génica simple de síntesis proporciona las tasas de los parámetros correspondientes a cada una de las reacciones implicadas en el modelo y las condiciones iniciales para cada especie de la red. Con estos datos se crea un modelo determinístico a través de sistemas de ecuaciones diferenciales que representan la interacciones entre las especies. La investigación es un modelo de diseño matemático aplicado a la genética de manera sistemática para cada uno de los pasos que realizan.

Las redes de Petri son las más utilizadas, en el campo técnico científico, puesto que, la especificidad de su diseño, con pocos elementos y limitadas combinaciones de uso permite una mayor exactitud previa a la construcción del algoritmo.

Los métodos formales se integran con otras soluciones a la disposición del desarrollador. Los campos de un diseño que no requieren de la exactitud del lenguaje pueden ser abordadas con otros métodos. Lo importante de las metodologías de diseño de sistemas de información es que son complementarias, no excluyentes de uso combinado y

La complejidad científica de muchos de los trabajos de investigación que requieren de los métodos formales también obliga a la colaboración investigativa, muchas veces multidisciplinaria. Son pocos los trabajos que se asumen de manera individual ya que el método obliga conocimientos profundos tanto de la ciencia en cuestión como de la ciencia exacta de la matemática. Los resultados concretos de las investigaciones de métodos formales muchas veces solucionan problemas complejos y se traducen en softwares y sistemas de mucha complejidad.

CONCLUSIÓN

Existen muchas y muy variadas metodologías para el diseño, arquitectura y construcción de sistemas de información y eso permite al diseñador de programas utilizar a que mejor convenga a sus fines. En aquellos ambientes donde los procesos requieren de la fiabilidad exacta del dato y su manejo, las metodologías formales son las más recomendadas.

Estas metodologías formales se utilizan en ambientes altamente técnicos en los que los datos tienen criterios de inclusión/exclusión muy bien definidos en algoritmos secuenciales y eventos recurrentes, pero no predichos, para procesos muy específicos, en tiempo real, cuyo encadenamiento, de la singularidad de los eventos, permita un proceso lógico, de alternativas reales ajustadas al diseño algorítmico. Los resultados obtenidos deben cumplir altos estándares de calidad.

Los sistemas embebidos, los programas de instrumental y equipos complejos de salud, seguridad, producción, etc., requieren de este tipo de metodología que, tal vez menos atractivas que las agiles, no están pensadas para la facilidad de manejo del usuario, sino para que los programas y equipos cumplan una función determinada y recurrente.

Para los ingenieros y desarrolladores de programas es fundamental la formación en estas metodologías pues siempre se enfrenta al desafío de proporcionar soluciones informáticas de calidad dentro de rangos de tiempo y costos predeterminados por el usuario/cliente un cierto coste y tiempo, siendo responsable de construir productos de calidad. El campo de las disciplinas de los sistemas de información es una gran oportunidad de desarrollo profesional y científico para Panamá.

La Ingeniería del Software evoluciona en la medida en que apuntala ciencias y áreas del conocimiento humano. Está en adaptación progresiva a las nuevas formas de computación y especialidades de las cuales los sistemas de información son fundamento como la bioinformática.

Un problema recurrente para los investigadores de Panamá es que no se encuentra literatura científica de origen panameño, sobre el tema, en los buscadores académicos y/o científicos de acceso público. Es un desafío, en el tiempo, para las academias e instituciones encargadas del avance técnico científico del país.

REFERENCIAS BIBLIOGRÁFICAS

- Campos, R., Flores, A., Molina, J., Mora, C., Rodríguez, C., Gatica, A., y Guzmán, C. (2021). *Avances de la bioinformática en Costa Rica: vista retrospectiva y perspectivas*. Revista de Biología Tropical, 69(4), 1204-1223: h <https://www.scielo.sa.cr/pdf/rbt/v69n4/0034-7744-rbt-69-04-1204.pdf>

- Carbo, C., Carbo, R., Yunga, D., y Villamar, C. (2021). *Diseño de prototipos de software. Fundamentos epistémico-metodológicos para su elaboración*. Dominio de las Ciencias, 7(6), 1520-1531: <https://dominiodelasciencias.com/ojs/index.php/es/article/view/2407/5303>
- Chanchí, G., Durán, D., y Campo, W. (2020). *Herramienta para el análisis de latencia y pérdida de paquetes en redes haciendo uso de algoritmos de clasificación soportados en árboles de decisión*. Revista Espacios, 41(29), 1-17 : <http://www.revistaespacios.com/a20v41n29/a20v41n29p01.pdf>
- Clavijo, D., Mejía, B., Rojas, L., y Sáenz, L. (2018). *Análisis estocástico de un sistema génico simple para la síntesis de una proteína implementando los métodos de Gillespie*. Revista Cuarzo, 24(1), 7-16: <https://revistas.juanncorpas.edu.co/index.php/cuarzo/article/view/350/364>
- León, A., Acosta, J., y Díaz, R. (2021). *Aplicación de la metodología incremental en el desarrollo de sistemas de información*. Revista Universidad y Sociedad, 13(5), 175-182: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202021000500175
- Mahulea, C., González, R., Montijano, E., y Silva, M. (2021). *Planificación de trayectorias en sistemas multirobot utilizando redes de Petri. Resultados y problemas abiertos*. Revista Iberoamericana de Automática e Informática Industrial, 18, 19-31: <https://riunet.upv.es/bitstream/handle/10251/160489/Mahulea%3bGonz%C3%A1lez%3bMontijano%20-20Planificaci%C3%B3n%20de%20trayectorias%20en%20sistemas%20multirobot%20utilizando%20de....pdf?sequence=1&isAllowed=y>
- Maida, E., y Pacienzia, J. (2015). *Metodologías de desarrollo de software*. [Tesis de Licenciatura en Sistemas y Computación] Universidad Católica Argentina: <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>
- Martínez, R., Palma, A., y Velásquez, A. (2020). *Revolución tecnológica e inclusión social: reflexiones sobre desafíos y oportunidades para la política social en América Latina*. Naciones Unidas, serie Políticas Sociales, N° 233 (: http://repositorio.cepal.org/bitstream/handle/11362/45901/1/S2000401_es.pdf
- Micolini, O., Ventre, L., y Ludemann, M. (2018). *Metodología de diseño y desarrollo de Sistemas Embebidos y Reactivos Basados en Redes de Petri*. IEEE ARGENCON 2018, Universidad Nacional de Tucumán: https://www.researchgate.net/publication/328812770_Metodologia_de_diseño_y_desarrollo_de_Sistemas_Embebidos_y_Reactivos_Basados_en_Redes_de_Petri
- Molina, J., Zea, M., Contento, M., y García, F. (2018). *Comparación de metodologías en aplicaciones web*. 3c Tecnología: glosas de innovación aplicadas a la pyme, 7(1), 1-19: Comparación de metodologías en aplicaciones web
- Osorio, C. (2021). *Integración de algoritmos genéticos y redes de petri como propuesta metodológica para la solución del time-dependent traveling salesman problem*. [Tesis de maestría en Ingeniería Industrial] Universidad del Valle : <https://bibliotecadigital.univalle.edu.co/handle/10893/21143>
- Piattini, M. (2016). *Evolución de la Ingeniería del Software y la formación de profesionales*. Revista institucional de la facultad de informática, UNPL (4), 15-17: http://sedici.unlp.edu.ar/bitstream/handle/10915/57358/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- Pinto, C., Irrazabal, E., y Sambrana, I. (2018). *Revisión Sistemática de la Literatura: aplicación de seguridad a requerimientos software de sistemas críticos ferroviarios*. XIX Simposio Argentino de Ingeniería de Software, 37-48: http://sedici.unlp.edu.ar/bitstream/handle/10915/70854/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- Serna, E. (2021). *Métodos Formales, Ingeniería de Requisitos y Pruebas del Software*. Medellín : Editorial Instituto Antioqueño de Investigación.
- Suaza, J., y Lora, G. (2020). *Proceso y evolución de los métodos formales en la ingeniería de requisitos*. Revista Ingenierías, Universidad de Medellín, 19(37), 119-136: <http://www.scielo.org.co/pdf/rium/v19n37/2248-4094-rium-19-37-119.pdf>
- Vega, L., Lao, Y., Marrero, F., y Pérez, M. (2020). *Redes de Petri: una herramienta para la validación de procedimientos*. Ciencias Holguín, 26(2), 1-16: <https://www.redalyc.org/journal/1815/181563169001/181563169001.pdf>

Zumba, J., y León, C. (2018). *Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software*. INNOVA Research Journal, 3(10), 20-33.: <https://dialnet.unirioja.es/servlet/articulo?codigo=6777227>