





AvPI: Extended Analysis of Average Parallelism in Systems Modeled with S³PR Petri Nets under Deadlock-Prevention Policies

Luis O. Ventre^{1,2,3} , Orlando Micolini^{1,2} , Mauricio Ludemann^{1,2} , Gabriel Valenzuela^{1,2} 

¹ Universidad Nacional de Córdoba, Córdoba, Argentina

² Laboratorio de Arquitectura de Computadoras, FCEfYN

³ luis.ventre@unc.edu.ar

Abstract. This article presents an extended formal and experimental analysis of the Average Parallelism Index (AvPI), a metric designed to quantify the effective parallelism of S³PR Petri nets under different deadlock-prevention policies. Building on our previous work, the extended version incorporates a complete formalization of the method, including the algebraic characterization of transition invariants, the identification of action-related places, and the mathematical definition of the AvPI over the projected reachability space. Using this formal foundation, we provide a deeper evaluation of three representative configurations—original, siphon-based control, and selective control—highlighting structural trade-offs between safety, concurrency, and resource utilization. The results show that over-restrictive policies, although deadlock-free, can drastically reduce concurrency, while selective control preserves both liveness and operational efficiency. The extended discussion clarifies the role of AvPI as a quantitative complement to classical qualitative properties (boundedness, liveness), enabling a more precise comparison of alternative control strategies. Finally, we outline the methodological limitations of the current approach and discuss its potential extension toward timed Petri nets and early-stage design flows for embedded systems.

Keywords: S³PR Petri nets, embedded systems, transition invariants, deadlock prevention, AvPI, and parallelism quantification.

AvPI: Análisis extendido del paralelismo promedio en sistemas modelados con redes de Petri S³PR bajo políticas de prevención de interbloques

Resumen. Este artículo presenta un análisis formal y experimental ampliado del Average Parallelism Index (AvPI), una métrica destinada a cuantificar el paralelismo efectivo en redes de Petri S³PR bajo diferentes políticas de prevención de interbloques. Sobre la base del trabajo previamente presentado, esta versión extendida incorpora una formalización completa del método, incluyendo la caracterización algebraica de los invariantes de transición, la identificación de plazas de acción y la definición matemática del AvPI sobre la

Received May 2026; Accepted June 2026; Published July 2026



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

proyección del espacio de estados alcanzables. A partir de este marco formal, se profundiza la evaluación de tres configuraciones representativas —red original, control basado en sifones y control selectivo— poniendo de manifiesto los compromisos estructurales entre seguridad, concurrencia y utilización de recursos. Los resultados muestran que políticas excesivamente restrictivas, aunque libres de interbloqueos, pueden degradar severamente el paralelismo del sistema, mientras que estrategias de control selectivo permiten preservar tanto la vivacidad como la eficiencia operativa. La discusión ampliada posiciona al AvPI como un complemento cuantitativo de las propiedades cualitativas clásicas, y la sección final presenta las limitaciones del enfoque actual junto con su potencial extensión hacia redes temporizadas y flujos de diseño temprano para sistemas embebidos.

Palabras clave: Redes de Petri S³PR, sistemas embebidos, invariantes de transición, prevención de bloqueos, AvPI y cuantificación del paralelismo.

1 Introducción

El término "paralelismo", según la Real Academia Española (RAE), denota la cualidad de elementos que mantienen equidistancia constante sin cruzarse, extendiéndose metafóricamente a correspondencias entre entidades disímiles (De la Lengua Española, 2001). En sistemas embebidos, este concepto adquiere especificidad técnica: se refiere a la ejecución concurrente de múltiples tareas para optimizar eficiencia y rendimiento, contrastando con enfoques secuenciales (Culler, Singh, & Gupta, 1999). Esta capacidad resulta crucial en aplicaciones que exigen respuestas en tiempo real y gestión eficiente de recursos limitados, como procesamiento y energía (Tanenbaum, 2016), especialmente ante la creciente complejidad de dominios tales como el Internet de las Cosas (IoT), vehículos autónomos y sistemas de control industrial (Wolf, 2010).

Para modelar dicha concurrencia, las Redes de Petri (RdP) —en particular, los Sistemas de Procesos Secuenciales Simples con Recursos (S³PR)— ofrecen un marco formal que describe interacciones sincronizadas y competencia por recursos (Murata, 1989). No obstante, el diseño de estas redes conlleva riesgos sustanciales: políticas de disparo de transiciones y restricciones estructurales pueden generar **deadlocks** (bloqueos irreversibles) o limitar drásticamente el paralelismo (Abdul-Hussin, 2015), comprometiendo la capacidad de ejecución simultánea. Por ende, resulta imperativo cuantificar el impacto de tales políticas en el índice de paralelismo comparativo promedio (AvPI), métrica esencial que refleja el número medio de tareas procesables concurrentemente (Ezpeleta, Tricas, Garcia-Valles, & Colom, 2002; Tricas, Garcia-Valles, Colom, & Ezpeleta, 2005).

En este contexto, se presenta un algoritmo innovador para determinar el AvPI en sistemas embebidos modelados con RdP, continuando investigaciones previas (Li, Wu, & Zhou, 2011; Ventre & Micolini, 2022). Este desarrollo complementa metodologías existentes para evaluar responsabilidad y cantidad de hilos (Ventre & Micolini, 2022), conformando un conjunto integral de herramientas de diseño temprano. La propuesta no solo optimiza la asignación de recursos y previene configuraciones erróneas derivadas de aproximaciones manuales (Navet & Simonot-Lion, 2017), sino que también impulsa la confiabilidad en aplicaciones críticas mediante la automatización de estimaciones de paralelismo.

2 Tesis Principal

Este trabajo aborda una brecha crítica en el análisis cuantitativo de sistemas embebidos modelados mediante RdP S^3PR , centrándose en la cuantificación objetiva del grado de paralelismo promedio (Average Parallelism Index , AvPI) y la evaluación del impacto de **políticas de disparo** y **criterios de desbloqueo** en esta métrica. La cuestión central se formula así: ¿Cómo puede medirse cuantitativamente el paralelismo efectivo en sistemas embebidos bajo restricciones de control, y de qué modo las políticas de disparo (estáticas/dinámicas) o los mecanismos de desbloqueo (ej. control basado en lugares clave) afectan dicho paralelismo? La respuesta a este interrogante permitirá resolver compensaciones fundamentales entre prevención de **deadlocks** y eficiencia concurrente (Ezpeleta et al., 1995; Marwedel, 2021), un desafío no resuelto en la síntesis de supervisores. Para ello, se postulan tres objetivos articulados:

1. Desarrollar el AvPI como métrica que cuantifique el número promedio de acciones ejecutables en paralelo a lo largo del espacio de estados alcanzables. Su fundamento teórico integra análisis estructural de RdP (Murata, 1989) con métodos probabilísticos adaptados a espacios de estado complejos.

2. Validar la aplicabilidad del AvPI para evaluar configuraciones bajo políticas de disparo heterogéneas y estrategias de desbloqueo, identificando sobre-restricciones que, pese a garantizar ausencia de deadlocks (H. Liu, Xing, Wu, Zhou, & Zou, 2014), limiten injustificadamente la concurrencia.

3. Implementar un algoritmo innovador que automatice el cálculo del AvPI, complementando herramientas existentes para la evaluación de responsabilidad y cantidad de hilos (Ventre & Micolini, 2022). Este aporte consolida un marco metodológico integral para la toma de decisiones en etapas tempranas de diseño, alineado con enfoques de co-diseño hardware/software operativa (Smiri, Dimerad, & El Ayeb, 2024) y mitigando errores derivados de aproximaciones manuales (Navet & Simonot-Lion, 2017).

La propuesta se enmarca en la línea de investigación del autor (Eng. Ventre & Phd. Micolini, 2021; Ventre & Micolini, 2022), ofreciendo no solo un avance teórico en el modelado de la concurrencia, sino también un instrumento práctico para optimizar asignación de recursos, prevenir bloqueos y maximizar la eficiencia en sistemas embebidos multi-hilo de alta criticidad (ej. vehículos autónomos, IoT industrial).

El presente trabajo aborda esta brecha mediante un algoritmo para calcular explícitamente el AvPI en redes S^3PR , fundamentado en técnicas de álgebra lineal y teoría de grafos aplicadas al espacio de estados alcanzables. A diferencia de métodos basados en trazas específicas o métricas estáticas (ej. paralelismo máximo), este enfoque evalúa sistemáticamente el potencial de concurrencia a lo largo de todos los estados posibles, independientemente de ejecuciones particulares. Su innovación radica en integrar:

- Evaluación comparativa de políticas de disparo: Identificación de sobre-restricciones mediante contraste entre configuraciones.
- Criterios de desbloqueo: Análisis de mecanismos como control basado en lugares clave.
- Aplicabilidad temprana: Implementación en etapas iniciales de diseño mediante procesadores de Petri especializados.

Esta metodología optimiza la concurrencia promedio sin comprometer la corrección funcional, proporcionando un instrumento para maximizar la utilización de recursos en sistemas de misión crítica. Su relevancia trasciende el avance teórico al

consolidar un marco cuantitativo para el co-diseño hardware/software, reduciendo errores de configuración manual y mejorando la confiabilidad sistémica.

En función de estos objetivos y de las limitaciones identificadas en la literatura especializada, este trabajo propone un enfoque integrador para abordar dicha brecha metodológica, estableciendo así las bases conceptuales y técnicas que guían el desarrollo del algoritmo presentado en las secciones siguientes.

2.1 Motivación y Relevancia

El diseño de sistemas embebidos exige modelado riguroso del comportamiento dinámico para garantizar eficiencia y confiabilidad operativa. Las RdP S³PR constituyen un marco formal ampliamente adoptado para representar sistemas concurrentes con competencia por recursos limitados (Murata, 1989). No obstante, un desafío crítico radica en la aparición de deadlocks (bloqueos irreversibles), estados donde ninguna transición puede dispararse, comprometiendo la funcionalidad del sistema (H. Liu et al., 2014). Para mitigar este riesgo, se implementan políticas de disparo (estáticas/dinámicas) y restricciones estructurales que, paradójicamente, pueden inducir sobre-restricciones, limitando drásticamente el paralelismo y reduciendo la eficiencia del sistema (Roszkowska, 2004). Esta tensión entre prevención de bloqueos y optimización concurrente motiva la necesidad de métricas cuantitativas que evalúen objetivamente el impacto de tales políticas en el paralelismo efectivo, evitando diseños sub-óptimos donde los recursos se subutilizan (Marwedel, 2021).

2.2 Estado del arte

El análisis del paralelismo en sistemas concurrentes modelados mediante RdP ha evolucionado de forma significativa durante las últimas décadas, impulsado principalmente por la necesidad de garantizar confiabilidad y eficiencia en sistemas embebidos cada vez más complejos. La literatura registra múltiples avances en verificación estructural, prevención de deadlocks y evaluación de desempeño, aunque la mayoría de los enfoques se concentran en propiedades cualitativas antes que en métricas cuantitativas que permitan comparar configuraciones alternativas de un mismo sistema.

Dentro de las primeras contribuciones relevantes, los trabajos clásicos se enfocan en el análisis formal del comportamiento dinámico y estructural de las RdP, introduciendo conceptos como vivacidad, boundedness y técnicas algebraicas para el estudio de invariantes (Murata, 1989). A partir de estos fundamentos, numerosas investigaciones desarrollaron estrategias para evitar deadlocks mediante restricciones estructurales o supervisores basados en el control de recursos, particularmente en sistemas S³PR. Entre estos se destacan las políticas de prevención de bloqueos en sistemas flexibles de manufactura, tanto a través de técnicas basadas en lugares de control como mediante criterios estructurales para garantizar ausencia de interbloqueos (Ezpeleta et al., 1995; Hu et al., 2012; Liu et al., 2014). Estos enfoques consolidaron un marco robusto para garantizar la corrección funcional, pero no ofrecen mecanismos para comparar el impacto de dichas políticas en el nivel de concurrencia disponible.

En paralelo, otros trabajos exploraron extensiones semánticas orientadas a capturar comportamientos concurrentes no deterministas. Tal es el caso de las NT-Petri nets, orientadas al modelado de programas concurrentes con disparo alternativo y agrupamiento de transiciones en work clusters que operan como hilos independientes, siempre que no compitan por recursos conflictivos (Rawson & Rawson, 2022). A

diferencia de los modelos tradicionales, este formalismo enfatiza la representación explícita del paralelismo y su relación con las decisiones de scheduling, aunque no aporta una métrica global equivalente a un índice promedio de concurrencia sobre el grafo de alcanzabilidad.

Otra línea de investigación complementaria se centra en medir atributos internos de modelos concurrentes desde una perspectiva estructural. En este ámbito, (Morasca, 1999) propone un enfoque basado en propiedades para evaluar características como tamaño, complejidad y acoplamiento en especificaciones concurrentes representadas mediante RdP. Si bien estos indicadores no cuantifican directamente paralelismo, constituyen herramientas relevantes para caracterizar cómo la estructura del modelo condiciona su capacidad de ejecutar acciones simultáneas.

También se observan avances en la integración entre RdP y técnicas de planificación en sistemas embebidos modernos. Por ejemplo, diversos trabajos aplican simulación y ejecución guiada para analizar concurrencia en sistemas reales, particularmente cuando la explosión del espacio de estados hace impracticable el análisis exhaustivo. Aunque estos métodos permiten estimar comportamientos observados, carecen de reproducibilidad completa y dependen del recorrido específico explorado, lo que limita la comparación sistemática entre diferentes políticas de control.

En el contexto de redes temporizadas, investigaciones recientes han abordado el desempeño mediante tiempos de disparo, scheduling dinámico y estrategias de aprendizaje profundo con arquitecturas tipo grafo (L. Hu et al., 2020). Estos enfoques analizan la relación entre recursos temporales y ejecución paralela, pero su complejidad computacional y dependencia de parámetros temporales los alejan del tipo de análisis estructural temprano que se requiere durante las fases iniciales del diseño de sistemas embebidos.

Finalmente, es importante señalar que una parte relevante de la literatura se orienta a garantizar propiedades cualitativas tradicionales —como vivacidad, acotación y ausencia de deadlocks— y al diseño de políticas de control (Roszkowska, 2004; Smiri et al., 2024; Zhou et al., 2023), sin ofrecer métricas cuantitativas que permitan evaluar las compensaciones entre seguridad y eficiencia. Este vacío es particularmente crítico en sistemas S³PR, donde pequeñas modificaciones en recursos o políticas de disparo pueden afectar drásticamente el nivel de paralelismo.

En síntesis, a pesar de los múltiples avances en modelado concurrente, supervisión y evaluación estructural, persiste una brecha metodológica en la cuantificación temprana del paralelismo global. La mayoría de los enfoques disponibles se basan en simulaciones específicas o en métricas estáticas como el paralelismo máximo, lo que dificulta comparar de forma objetiva el impacto de distintas políticas de control sobre el comportamiento completo del sistema. El presente trabajo aborda esta limitación mediante un algoritmo capaz de calcular explícitamente un índice de paralelismo promedio (AvPI) sobre todos los estados alcanzables de una red S³PR, permitiendo así una evaluación sistemática y reproducible de la concurrencia inherente a configuraciones alternativas y estrategias de desbloqueo.

2.3 Alcance: Limitaciones y Línea de Investigación Futura

El algoritmo propuesto presenta una limitación inherente a su marco conceptual actual: está diseñado para RdP no temporizadas (Popova-Zeugmann, 2013), lo que implica que no considera el impacto de la dinámica temporal (ej., retardos de disparo o duraciones asociadas a transiciones) en el cálculo del índice de paralelismo promedio (AvPI). Esta restricción afecta particularmente su aplicabilidad en sistemas embebidos donde la temporalidad es un factor crítico para la eficiencia

operativa, como en sistemas de control cíclico o entornos con restricciones de latencia estricta (Zhou et al., 2023). No obstante, esta limitación no invalida su utilidad en la evaluación de la estructura lógica de concurrencia durante fases preliminares de diseño, donde el análisis precede a la definición de características temporales y permite optimizar la coordinación estática de recursos (Marwedel, 2021).

Para superar esta barrera, se propone una línea de investigación futura centrada en extender el algoritmo a RdP temporizadas. La estrategia fundamental implica:

1. Identificar sistemáticamente los marcados que involucren transiciones con retardos asociados.

2. Implementar un mecanismo de ponderación basado en parámetros temporales clave, como:

- Tiempos de disparo de transiciones.
- Frecuencia de activación en trazas de ejecución.
- Impacto de cuellos de botella temporales en la concurrencia efectiva.

Esta adaptación permitiría que el AvPI refleje no solo el paralelismo potencial intrínseco, sino también la degradación inducida por la temporalidad, ofreciendo así una métrica holística del comportamiento dinámico. El enfoque se alinearía con metodologías establecidas para redes temporizadas (L. Hu et al., 2020), donde la integración de álgebra lineal y modelos de Markov podría cuantificar compensaciones entre concurrencia y restricciones temporales.

Esta extensión ampliaría sustancialmente la aplicabilidad del algoritmo, habilitando su uso en dominios como sistemas de tiempo real críticos y optimizando el co-diseño hardware/software en entornos con requisitos temporales estrictos.

3 Algoritmo propuesto

3.1 Conceptos estructurales

La metodología propuesta en esta investigación se estructura en las siguientes fases interrelacionadas que permiten evaluar sistemáticamente el AvPI en sistemas modelados con RdP S³PR. La primera fase parte del sistema modelado con una RdP S³PR que puede presentar riesgo de deadlock, siguiendo los criterios estructurales establecidos en (Ezpeleta et al., 1995).

Sea una **RdP** definida por la tupla (Murata, 1989):

$$N = (P, T, Pre, Post) \quad (Ec. 3.1)$$

Dónde:

- P es el conjunto de plazas
 - Formalmente $P = \{p_1, p_2, \dots, p_m\}$
- T es el conjunto de transiciones
 - Formalmente $T = \{t_1, t_2, \dots, t_n\}$
- Pre es la matriz de incidencia de entrada en donde se cumple que
 - Si $Pre(p,t)=k$, entonces la transición t consume k tokens de la plaza p al dispararse.
 - Si $Pre(p,t)=0$, la plaza no es una entrada de esa transición.
- $Post$ es la matriz de incidencia de salida en donde se cumple que

- Si $Post(p,t)=k$, entonces la transición t agrega k tokens a la plaza p .
- Si $Post(p,t)=0$, la plaza no es una salida de esa transición.

Una **RdP S³PR** es una subclase de las RdP orientada a modelar sistemas de procesos secuenciales que comparten un conjunto de recursos. Su definición se construye de forma modular para capturar de manera nativa las semánticas fundamentales de los sistemas concurrentes: la ejecución paralela de múltiples procesos y la competencia por recursos compartidos, lo que las hace particularmente adecuadas y altamente utilizadas en la industria para modelar y analizar diferentes clases de sistemas entre los cuales se encuentran los sistemas embebidos, caracterizados típicamente por ser sistemas reactivos, concurrentes y con recursos limitados (Ezpeleta et al., 1995).

El principal **componente básico** de esta subclase es la Red de Proceso Simple (Simple Sequential Process Net, **S²P**). Un **S²P** es una RdP que modela el ciclo de vida de un único proceso. Se define como una tupla:

$$S^2P = (P \cup \{p^0\}, T, F) \quad (Ec. 3.2)$$

Dónde:

- P es un conjunto finito de **plazas**
- $P \neq \emptyset$
- P^0 es llamada plaza de idle del sistema.
- S^2P es una máquina de estados fuertemente conexa y viva.
- F es la la relación de incidencia total $F \subseteq (P \times T) \cup (T \times P)$
- Todos los circuitos o caminos del sistema *contienen la plaza P^0* .

Luego es posible definir un proceso secuencial simple con recursos **S²PR**. Esto es debido a que la interacción con restantes procesos en un sistema de mayor magnitud va a implicar compartir recursos. Se asume que en el estado de idle el sistema tiene interacción por lo que no hay consumo de recursos en ese estado.

Se define una **S²PR** como el acoplamiento de una **S²P** con los recursos compartidos del sistema:

$$S^2PR = (P \cup \{p^0\} \cup P_R, T, F) \quad (Ec. 3.3)$$

De manera que:

- La subred conformada por $X = P \cup \{p^0\} \cup T$ es un **S²P**
- $P_R \neq \emptyset$
- $(P \cup \{p^0\}) \cap P_R = \emptyset$
Sea $R = \{r_1, r_2, \dots, r_m\}$ un conjunto finito de **recursos** que serán compartidos por los procesos.
- $\forall p \in P, \forall t \in *p, \forall t' \in p^* \cap P_R = t' \cap P_R = \{r_p\}$

De este modo, es posible definir un **Sistema de Procesos Secuenciales Simples con Recursos (S³PR)** como la composición recursiva de múltiples **S²PR** que compiten por el mismo conjunto de recursos.

Se define como una tupla:

$$S^3PR = (P, T, F, M_0) \quad (Ec. 3.4)$$

Dónde:

- P es la unión disjunta de las plazas de actividad de todos los procesos del sistema, las plazas idle de todos los procesos y las plazas de recursos compartidos.
- T es la unión de todas las transiciones de todos los procesos.
- F es la unión de todos los arcos de los procesos y todos los arcos de recursos.
- $M_0: P \rightarrow \mathbb{N}$ es el **marcado inicial** que debe satisfacer:
 1. $M_0(p_i^0) = 1$ para toda plaza de idle $\in P^0$ (cada proceso comienza en su estado de reposo).
 2. $M_0(p) = 0$ para toda plaza de actividad p .
 3. $M_0(r) \geq 1$ para toda plaza de recurso r (cada recurso tiene al menos una unidad disponible inicialmente).

Así una **RdP S³PR** presenta una estructura modular compuesta que se construye a partir de procesos secuenciales independientes en donde la única interacción entre los distintos procesos se da a través de las plazas de recursos. A su vez cada proceso individual es fuertemente conexo y sigue un ciclo bien definido que comienza en el estado de idle y luego adquiere y libera recursos avanzando hasta completar su propósito y regresar al estado de idle. La estructura completa induce posibles conflictos y deadlocks derivados exclusivamente de la asignación de recursos, por lo que las S³PR constituyen un modelo ampliamente utilizado para analizar y supervisar procesos concurrentes con recursos limitados.

Un invariante de transición es un vector:

$$x \in \mathbb{N}^{\{T\}} \quad (Ec. 3.5)$$

Que satisface:

$$(Post - Pre)x = 0 \quad (Ec. 3.6)$$

Cada invariante representa un ciclo del sistema, modelando una secuencia completa de transiciones que retorna al mismo marcado.

3.2 Construcción del grafo de alcanzabilidad

Mediante software especializado Tina (Toulouse, 2025), Pipe (Computing, 2002/3) o similar se determinan los marcados estables, es decir, todos los estados posibles sin llegar a la repetición de ciclos infinitos.

Identificación de invariantes de transición (IT)

Se deben identificar los IT de la red, cada invariante representa una secuencia mínima que lleva la red desde un marcado inicial hasta retornar al mismo marcado, lo que significa que el proceso ha cumplido un ciclo de acuerdo a las especificaciones, avanzando por las distintas etapas/estados del proceso compartiendo recurso y realizando acciones.

Obtención de plazas asociadas al IT

Para cada IT se debe obtener el conjunto de plazas asociadas a ese IT.

$$PI_i = \bigcup_{\forall t \in Inv} t \cup \bigcup_{\forall t \in Inv} t \bullet \quad (Ec. 3.7)$$

Dónde: PI_i representa el conjunto de plazas asociadas al iésimo IT.

Filtrado de plazas

Determinar las plazas relacionadas a acciones de cada IT. Para esto es necesario eliminar del conjunto PI_i , las plazas que son restricciones, recursos e idle(buffer):

$$PA_i = PI_i - \{PI_{restricciones_i} \cup PI_{recursos_i} \cup PI_{idle_i}\} \quad (Ec. 3.8)$$

Dónde: PA_i representa el conjunto de plazas de acciones asociadas al iésimo IT.

Luego necesitamos definir el conjunto PA , de todas las plazas de acción de la red, el cual es el equivalente a la unión de todas las plazas de acción de cada IT PA_i .

Estados Alcanzables

Del árbol de alcanzabilidad de la RdP, se debe obtener MA , el cual es el conjunto de todos los marcados posibles del conjunto de plazas PA .

Determinación del AvPI

Se define el AvPI como:

$$AvPI = \sum_{p \in PA} \underline{M}(p) \quad (Ec. 3.9)$$

Donde dado el conjunto de marcados MA

- Se debe operar sumando la cantidad de tokens de cada plaza en todos los estados y dividirla por la cantidad de estados estables del sistema.
- En esta instancia se obtendrá un promedio de tokens por plaza de acción en todos los marcados del sistema.
- Con este valor promedio para cada plaza de acción, sumando los promedios de tokens se obtendrá un índice promedio que determina el grado de paralelismo potencial de acciones del sistema modelado.

La complejidad del cálculo del AvPI está dominada por la generación del grafo de alcanzabilidad, cuyo costo es $O(|R(N)|)$. La identificación de invariantes es polinómica en $|T|$, mientras que la proyección sobre las plazas relevantes mantiene el costo lineal en $|P|$. En conjunto, el método resulta adecuado para redes S3PR de tamaño moderado y permite un análisis exhaustivo en etapas tempranas de diseño.

Caso de Aplicación y resultados

A continuación se detalla un caso de aplicación en donde con la finalidad de validar la aplicación de la metodología se parte de un sistema modelado con una red

con deadlock, ver Fig. 1. Es importante mencionar que la red ha sido levemente complejizada a partir de la red expuesta en (Zhao & Hou, 2013).

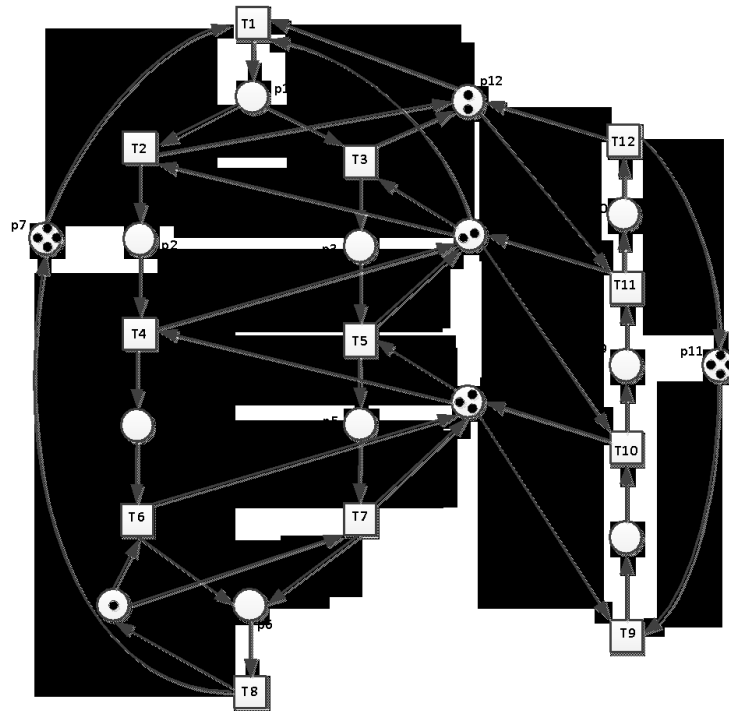


Fig. 1 Red de Petri original con deadlock

Se procede a desbloquear la misma con dos políticas de restricción distintas basadas respectivamente en el control de sifones mínimos y en políticas de priorización de transiciones, lo que permite contrastar diferentes enfoques de prevención de bloqueos ver

$IT_1 = \{T_9, T_{10}, T_{11}, T_{12}\}$ y Fig. 3. Estas propuestas de desbloqueo se limitan a solo el agregado de una plaza de restricción, a saber P_{16} , difiriendo en sus arcos de conexión en la red y su marcado inicial de tokens.

Luego se aplica la metodología propuesta a ambas redes y se analizan los resultados del AvPI para cada caso. Mediante herramientas de análisis formal como Petrinator y PIPE, se construye el grafo de alcanzabilidad completo para cada variante del sistema. Este proceso considera únicamente los marcados estables, excluyendo aquellos estados transitorios que no contribuyen al comportamiento cíclico estable del sistema. La generación del espacio de estados se realiza hasta alcanzar saturación, evitando la repetición de ciclos infinitos mediante técnicas de poda automática.

Luego la metodología se centra en la identificación de IT's, que representan conjuntos mínimos de transiciones capaces de ejecutarse en ciclo completo (Murata, 1989). Este análisis estructural, realizado mediante métodos algebraicos sobre la matriz de incidencia, permite determinar los componentes fundamentales que contribuyen al paralelismo potencial del sistema, proporcionando una base matemática para la posterior cuantificación del índice.

Los IT de la red son:

$$IT_1 = \{T_9, T_{10}, T_{11}, T_{12}\}$$

$$IT_2 = \{T_1, T_2, T_4, T_6, T_8\}$$

$$IT_3 = \{T_1, T_3, T_5, T_7, T_8\}$$

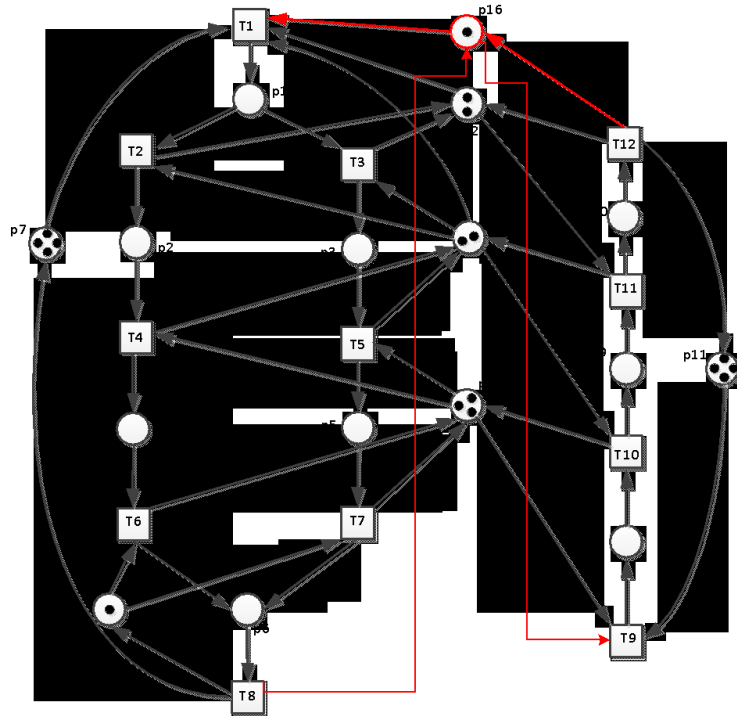


Fig. 2 Red de Petri solución propuesta 1

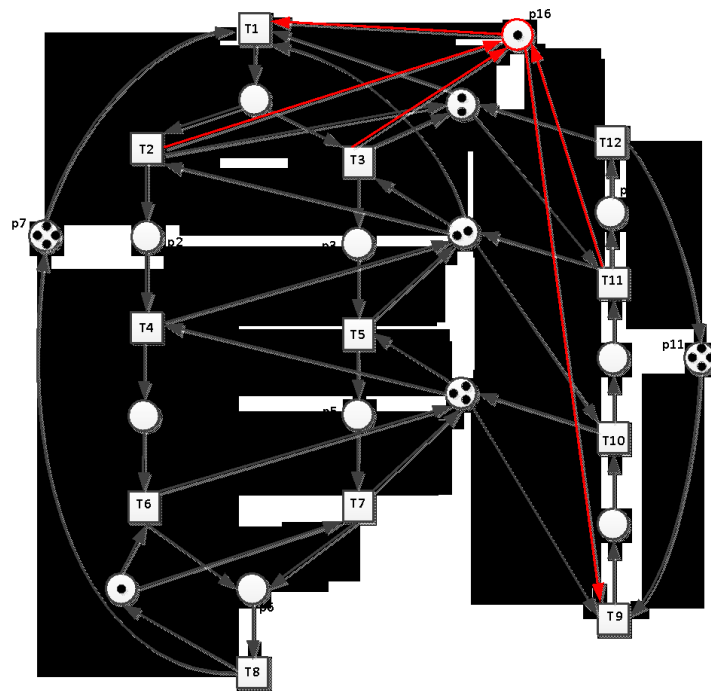


Fig. 3 Red de Petri solución propuesta 2

A continuación, se implementa el algoritmo que calcula el AvPI mediante un proceso en tres etapas. Inicialmente, se filtran las plazas correspondientes a recursos compartidos y buffers (idle places).

Plazas identificadas como Idle/buffer:

$$idle = \{P_7, P_{11}\}$$

Plazas identificadas como recursos /resources

$$resource\ restriction = \{P_{12}, P_{13}, P_{14}, P_{15}\}$$

Luego de identificar las plazas antes mencionadas y excluyéndolas del sistema se obtienen las plazas de acción de la red

$$PA = \{P_1, P_2, P_3, P_4, P_5, P_6, P_8, P_9, P_{10}\}$$

Posteriormente, se calcula el valor promedio de tokens en cada una de estas plazas de acción considerando todos los estados alcanzables, aplicando técnicas de muestreo estadístico para redes complejas. Finalmente, la suma ponderada de estos promedios genera el AvPI, que cuantifica numéricamente la capacidad concurrente del sistema.

Al realizar el análisis con el grafo de marcados para la red original de la Fig. 1, se obtienen por un lado 1477 estados estables alcanzables y los siguientes promedios de tokens de todos esos estados para cada plaza:

Tabla 1. Promedio de tokens de estados alcanzables en red original

Plaza	Promedio de tokens
-------	--------------------

P_1	0.5223
P_2	0.3938
P_3	0.3938
P_4	0.5792
P_5	0.5792
P_6	0.4222
P_8	0.8627
P_9	0.5480
P_{10}	0.6441

Grado promedio de paralelismo

Operando obtenemos el AvPI de la red original ($c/deadlock$): 4.9452

Luego se realiza el análisis para la red de la solución 1 de la $IT_1 = \{T_9, T_{10}, T_{11}, T_{12}\}$. Del análisis y la aplicación de la metodología es posible inferir que los invariantes de transición se mantienen así como también las plazas de acción y como es esperado solo se agrega una plaza de restricción P_{16} .

Al realizar el análisis de manera análoga se obtiene:

Tabla 2. Promedio de tokens de estados alcanzables en red solución 1

Plaza	Promedio de tokens
P_1	0.1000
P_2	0.1000
P_3	0.1000
P_4	0.1000
P_5	0.1000
P_6	0.1000
P_8	0.1000
P_9	0.1000
P_{10}	0.1000

3.3 Grado promedio de paralelismo

Operando obtenemos el AvPI de la red Solución 1: 0.90

Luego se realiza el análisis para la red de la solución 2 de la Fig. 3. Del análisis y la aplicación de la metodología es posible inferir que los IT's se mantienen así como también las plazas de acción y restricciones.

Al realizar el análisis con el grafo de marcados se obtienen solo 1017 estados estables alcanzables y los siguientes promedios de tokens de todos esos estados:

Tabla 3. Promedio de tokens de estados alcanzables en red solución 2

Plaza	Promedio de tokens
P_1	0.2790
P_2	0.4273
P_3	0.4273

P_4	0.6729
P_5	0.6729
P_6	0.4204
P_8	0.5246
P_9	0.3890
P_{10}	0.7819

3.4 Grado promedio de paralelismo

Operando obtenemos el AvPI de la red Solución 2: 4.5953

La fase final del estudio consiste en el análisis comparativo de los resultados obtenidos para las diferentes configuraciones del sistema. Este contraste multidimensional evalúa no solo la efectividad en la prevención de deadlocks, sino también el impacto en la concurrencia global, considerando métricas adicionales como el grado máximo de paralelismo y la densidad del grafo de alcanzabilidad. La comparación sistemática entre la red original (con deadlock) y las redes modificadas permite validar empíricamente las ventajas y limitaciones de cada política de control, aportando criterios objetivos para la toma de decisiones en el diseño de sistemas embebidos.

El marco analítico de esta investigación integra dos enfoques complementarios para evaluar rigurosamente las configuraciones del sistema. El análisis comparativo se fundamenta en el examen sistemático de métricas clave, particularmente el número de estados alcanzables y el AvPI. Esta aproximación permite cuantificar el trade-off entre las restricciones impuestas por las políticas de desbloqueo y la capacidad concurrente preservada, un aspecto crítico en el diseño de sistemas embebidos eficientes. La comparación se extiende más allá de valores absolutos, incorporando análisis de sensibilidad para identificar patrones de comportamiento en diferentes escenarios operativos.

La validación estructural constituye el segundo pilar metodológico, enfocándose en la preservación de las propiedades fundamentales del sistema mediante el examen de IT's. Esta verificación sigue los principios de conservación de comportamiento propuestos en (H. Hu, Zhou, Li, & Tang, 2012), garantizando que las modificaciones introducidas para prevenir deadlocks no comprometan la funcionalidad básica de la red. El proceso incluye tanto la verificación algebraica de invariantes lineales (Murata, 1989) como la evaluación de propiedades de vivacidad y boundedness, empleando para ello herramientas formales de análisis estructural implementadas en plataformas como PIPE (Computing, 2002/3) y Tina (Toulouse, 2025). Esta doble capa de validación asegura que las soluciones propuestas no solo mejoren la confiabilidad del sistema, sino que mantengan su corrección funcional frente a especificaciones de diseño originales.

4 Técnicas de Análisis

El marco metodológico integra dos enfoques complementarios para evaluar configuraciones de sistemas embebidos modelados con RdP S³PR. El análisis comparativo examina métricas clave —número de estados alcanzables e índice de paralelismo promedio (AvPI)— siguiendo metodologías establecidas para sistemas discretos (Christos Cassandras 2010). Esta aproximación cuantifica el trade-off entre restricciones de control y capacidad concurrente, incorporando análisis de sensibilidad para patrones de comportamiento en escenarios operativos heterogéneos

(Smiri et al., 2024). Paralelamente, la validación estructural verifica la preservación de propiedades fundamentales mediante invariantes de transición, garantizando que las modificaciones para prevenir deadlocks no comprometan la funcionalidad básica (H. Liu et al., 2014). Este proceso combina verificación algebraica de invariantes lineales (Murata, 1989) con evaluación de vivacidad y boundedness (Y. Liu, Li, & Li, 2023), utilizando herramientas como PIPE y Tina para asegurar corrección funcional frente a especificaciones de diseño.

4.1 Resultados Experimentales

La aplicación metodológica revela características críticas del comportamiento en redes S^3PR :

- Configuración original : Exhibe alto paralelismo ($AvPI = 4.9452$ en 1477 estados alcanzables), pero tiende a deadlocks , invalidando su aplicabilidad práctica (Ezpeleta et al., 1995).
- Enfoque restrictivo (control de sifones) : Elimina deadlocks pero reduce drásticamente los estados alcanzables (10) y el $AvPI$ (0.9), evidenciando sobre-restricción que limita la concurrencia (H. Liu et al., 2014) .
- Control selectivo : Mantiene un equilibrio óptimo, con $AvPI = 4.5953$ y 1017 estados alcanzables sin bloqueos, preservando eficiencia operativa (Smiri et al., 2024).

Este contraste demuestra empíricamente el trade-off inherente al diseño de sistemas concurrentes: soluciones maximalmente restrictivas garantizan seguridad pero generan ineficiencias, mientras enfoques balanceados optimizan paralelismo sin sacrificar confiabilidad.

4.2 Discusión e Implicaciones

La evaluación comparativa de las configuraciones analizadas permite comprender con mayor profundidad las compensaciones fundamentales entre corrección funcional y eficiencia concurrente en redes S^3PR . La red original presenta el valor más alto de $AvPI$, reflejando un paralelismo potencial elevado sustentado por la ausencia de restricciones adicionales. Sin embargo, este comportamiento coexiste con estados de bloqueo, lo que evidencia que la maximización del paralelismo estructural no es suficiente para asegurar confiabilidad operativa en sistemas embebidos críticos (Ezpeleta et al., 1995). Esta situación confirma que la concurrencia debe evaluarse conjuntamente con mecanismos que garanticen vivacidad y ausencia de deadlocks, ya que un diseño que prioriza únicamente la eficiencia puede resultar inviable en entornos de misión crítica.

En el extremo opuesto, el enfoque restrictivo basado en control de sifones elimina efectivamente los deadlocks, pero lo hace mediante la imposición de restricciones globales que reducen drásticamente el espacio de estados alcanzables. La caída del $AvPI$ y la contracción del número de configuraciones posibles son consistentes con comportamientos altamente deterministas observados en la literatura cuando se aplican políticas de control excesivamente estrictas (H. Liu et al., 2014). En sistemas embebidos modernos, este tipo de sobre-restricción puede derivar en incremento de latencias, subutilización sistemática de recursos y pérdida de flexibilidad frente a variaciones en la demanda, afectando negativamente la capacidad de respuesta en tiempo real, especialmente en arquitecturas multiprocesador o de IoT industrial (Smiri et al., 2024).

El tercer enfoque, basado en control selectivo, ofrece un punto de equilibrio particularmente relevante: su AvPI se mantiene cercano al de la red original, al tiempo que elimina por completo los deadlocks y conserva propiedades esenciales como vivacidad y boundedness. Estos resultados confirman que restricciones focalizadas pueden modular eficazmente el espacio de estados sin sacrificar la capacidad concurrente inherente del sistema, en línea con las propuestas de control dinámico y supervisión minimalista presentes en la literatura reciente. Para aplicaciones embebidas con requerimientos simultáneos de seguridad, eficiencia y adaptabilidad, este equilibrio es especialmente valioso.

Desde una perspectiva metodológica, el AvPI se consolida como un indicador complementario a las propiedades cualitativas tradicionales de las RdP (Murata, 1989). Mientras que la verificación estructural mediante invariantes garantiza corrección básica, no proporciona información cuantitativa sobre el rendimiento concurrente. En cambio, el AvPI expone diferencias sustanciales entre alternativas que, aun siendo equivalentes en términos de corrección funcional, presentan comportamientos operativos profundamente distintos. Esta capacidad de discriminación resulta crítica para evitar configuraciones sobre-restringidas y para seleccionar políticas de control que maximicen la utilización de recursos disponibles.

Asimismo, la integración del AvPI en etapas tempranas del flujo de diseño, alineada con estrategias shift-left, permite detectar tempranamente patrones de bloqueo potencial, reducciones innecesarias en paralelismo o comportamientos degenerados inducidos por restricciones. La evaluación anticipada facilita decisiones de diseño más informadas, reduce costos de iteración y mejora la trazabilidad de configuraciones a lo largo del ciclo de desarrollo del sistema.

En conjunto, los resultados muestran que la evaluación de políticas de control no debe limitarse a verificar ausencia de deadlocks, sino que debe considerar su impacto en la estructura dinámica del sistema y en su potencial para ejecutar actividades concurrentes. El AvPI constituye una herramienta cuantitativa rigurosa para este análisis, permitiendo comparar configuraciones alternativas y seleccionar aquellas que maximizan el equilibrio entre seguridad, eficiencia y capacidad de respuesta.

5 Conclusiones

Este estudio desarrolla un marco metodológico innovador para cuantificar el paralelismo en sistemas embebidos mediante redes S^3PR , introduciendo el AvPI como métrica para evaluar compensaciones entre restricciones de control y eficiencia concurrente. Sus contribuciones clave son:

1. Herramienta analítica: Cálculo del AvPI basado en análisis del grafo de alcanzabilidad e invariantes de transición, excluyendo plazas asociadas a recursos o buffers.
2. Optimización del diseño: Integración con algoritmos existentes para evaluación de hilos, consolidando un conjunto de herramientas para diseño multi-hilo.
3. Validación empírica: Demostración de que soluciones balanceadas (ej., control selectivo) preservan paralelismo sin comprometer confiabilidad.

Pese a limitaciones inherentes (naturaleza estática, explosión combinatoria, ausencia de temporalidad), el trabajo establece bases para extensiones futuras:

- Incorporación de redes temporizadas para modelar dinámicas operativas.
- Técnicas de abstracción de espacios de estado grandes (Y. Liu et al., 2023).
- Estrategias adaptativas de control en sistemas complejos.

Esta investigación puentea teoría de RdP y necesidades prácticas en sistemas embebidos contemporáneos, ofreciendo un instrumento cuantitativo para toma de decisiones en dominios críticos donde confiabilidad y eficiencia coexisten en tensión permanente (Smiri et al., 2024).

Referencias

- Abdul-Hussin, M. (2015). *Design of a Petri net based deadlock prevention policy supervisor for S3PR* [ponencia]. 2015 6th International Conference on Intelligent Systems, Modelling and Simulation. Kuala Lumpur, Malaysia. <https://doi.org/10.1109/ISMS.2015.5>
- Computing, I. C. L. D. o. (2002). Platform Independent Petri Net Editor 2. *PIPE*. Recuperado el 25 de abril de 2025 de <https://pipe2.sourceforge.net/>
- Cassandras, C. G. y Lafourtune, S. L. (Eds.). (2010). *Introduction to discrete event systems*. (2ª ed.). Springer. <https://doi.org/10.1007/978-0-387-68612-7>
- Culler, D., Singh, J. P. y Gupta, A. (1998). *Parallel computer architecture: A hardware/software approach*. Morgan Kaufmann Publishers. <https://dl.acm.org/doi/book/10.5555/2821564>
- Ezpeleta, J., Colom, J. M. y Martínez, J. (1995). A Petri Net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on Robotics and Automation*, 11(2), 173-184. <http://dx.doi.org/10.1109/70.370500>
- Ezpeleta, J., Tricas, F., García-Valles, F. y Colom, J. M. (2002). A banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states. *IEEE Transactions on Robotics and Automation*, 18(4), 621-625. <https://doi.org/10.1109/TRA.2002.801048>
- Hu, H., Zhou, M., Li, Z. y Tang, Y. (2012). Deadlock-free control of automated manufacturing systems with flexible routes and assembly operations using Petri nets. *IEEE Transactions on Industrial Informatics*, 9(1), 109-121. <https://doi.org/10.1109/TII.2012.2198661>
- Hu, L., Liu, Z., Hu, W., Wang, Y., Tan, J. y Wu, F. (2020). Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *Journal of Manufacturing Systems*, 55, 1-14. <https://doi.org/10.1016/j.jmsy.2020.02.004>
- Li, Z., Wu, N. y Zhou, M. (2011). Deadlock control of automated manufacturing systems based on Petri nets—A literature review. Paper presented at the IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), <https://doi.org/10.1109/TSMCC.2011.2160626>
- Liu, H., Xing, K., Wu, W., Zhou, M. y Zou, H. (2014). Deadlock prevention for flexible manufacturing systems via controllable siphon basis of Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3), 519-529. <https://doi.org/10.1109/TSMC.2014.2347267>
- Liu, Y., Li, X. y Li, Z. (2023). Control strategy of discrete event systems modeled by labeled Petri nets based on transition priority. *IEEE Access*, 11, 45442-45455. <https://doi.org/10.1109/ACCESS.2023.3272404>
- Marwedel, P. (2021). *Embedded system design: Embedded systems foundations of cyber-physical systems, and the internet of things*. (4ª ed.). Springer. <https://doi.org/10.1007/978-3-030-60910-8>
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *IEEE*, 77(4), 541-580. <http://dx.doi.org/10.1109/5.24143>
- Navet, N. y Simonot-Lion, F. (2017). *Automotive embedded systems handbook*. CRC Press. <https://doi.org/10.1201/9780849380273>
- Popova-Zeugmann, L. (2013). *Time and Petri Nets*. Springer. <https://doi.org/10.1007/978-3-642-41115-1>
- Rawson, M. y Rawson, M. G. (2022). Petri nets for concurrent programming. *arXiv*, 2208.02900. <https://doi.org/10.48550/arXiv.2208.02900>
- Real Academia Española. (s.f.). Par. En *Diccionario de la lengua española*. Recuperado el 25 de abril de 2025 de <https://dle.rae.es/par>

- Roszkowska, E. (2004). Supervisory control for deadlock avoidance in compound processes. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(1), 52-64. <https://doi.org/10.1109/TSMCA.2003.820572>
- Smiri, K., Dimerad, R. B. y El Ayeub, F. (2024). Hardware/Software Co-design in Multiprocessors Embedded Systems and IoT. *Journal of Theoretical and Applied Information Technology*, 102(2), 719-726.
- Tanenbaum, A. S. (2016). Structured computer organization. *Computer Languages*, 10(3-4), 237. [https://doi.org/10.1016/0096-0551\(85\)90019-0](https://doi.org/10.1016/0096-0551(85)90019-0)
- Tricas, F., Garcia-Valles, F., Colom, J. M. y Ezpeleta, J. (2005). A Petri net structure-based deadlock prevention solution for sequential resource allocation systems. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, 2005*, 271-277. <https://doi.org/10.1109/ROBOT.2005.1570131>
- Ventre, L. O. y Micolini, O. (2021). Extended Petri net processor and threads quantity determination algorithm for embedded systems. En P. Pesado y J. Eterovic (Eds.), *Computer Science – CACIC 2020. CACIC 2020. Communications in Computer and Information Science* (vol. 1409). Springer. https://doi.org/10.1007/978-3-030-75836-3_14
- Ventre, L. O. y Micolini, O. (2022). *Algoritmos para determinar cantidad y responsabilidad de hilos en sistemas embebidos modelados con Redes de Petri S3PR* [ponencia]. XXVII Congreso Argentino de Ciencias de la Computación. Red de Universidades con Carreras en Informática. Ciudad Autónoma de Buenos Aires, Argentina. <https://sedici.unlp.edu.ar/handle/10915/130440>
- Wolf, W. (2010). *High-performance embedded computing: architectures, applications, and methodologies*. Morgan Kaufmann. <https://doi.org/10.1016/B978-0-12-369485-0.X5000-3>
- Zhao, M. y Hou, Y. (2013). An iterative method for synthesizing non-blocking supervisors for a class of generalized Petri nets using mathematical programming. *Discrete Event Dynamic Systems*, (23), 3-17. <https://doi.org/10.1007/s10626-011-0124-9>
- Zhou, F., Cao, Z., Ge, Q., Zhang, Q., Zhao, D., Liu, S. y Chen, Y. (2023). A real-time phase processing system for phase sensitive optical time domain reflectometer. *Review of Scientific Instruments*, 94(1), 014710. <https://doi.org/10.1063/5.0132420>