

Modeling of Reactive Systems using Non-Autonomous Petri Nets and Microservices

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

¹ Universidad Nacional de Córdoba, Córdoba, Argentina

² Laboratorio de Arquitectura de Computadoras, FCEfYn

³mauri.ludemann@unc.edu.ar

Abstract: This paper presents an innovative framework for modeling and executing reactive systems by integrating Non-Autonomous Petri Nets (NAPN) with microservices architecture. Reactive systems in critical domains such as industrial control and IoT demand both formal verifiability and operational flexibility, yet existing approaches often struggle to balance these requirements. Our work addresses this gap by combining the mathematical rigor of NAPNs—which enable precise modeling of event-driven concurrency and synchronization—with the scalability and resilience of microservices.

A key contribution is our hierarchical event taxonomy, which systematically classifies stimuli (temporal, asynchronous, or fault-based) to optimize processing in distributed environments. This taxonomy enables adaptive handling of both recognized and unknown events, enhancing system robustness in dynamic scenarios. The proposed architecture features an intelligent orchestrator that coordinates microservices based on event classification while maintaining consistency with the formal NAPN model.

Expected outcomes include portable, scalable, and formally verifiable reactive systems that preserve design-time properties during distributed execution. The framework particularly benefits industrial applications where reliability and real-time responsiveness are paramount. Future work will validate the approach through payment system case studies and explore machine learning extensions for dynamic performance optimization. This research bridges formal methods with modern distributed architectures, offering a principled yet practical solution for mission-critical reactive systems.

Keywords: Reactive systems, Non-Autonomous Petri Nets, microservices, event taxonomy, formal verification, distributed architectures.

Modelado de Sistemas Reactivos mediante Redes de Petri No Autónomas y Microservicios

Resumen. Este artículo presenta un marco innovador para el modelado y ejecución de sistemas reactivos mediante la integración de Redes de Petri No Autónomas (RPNA) con arquitecturas de microservicios. Los sistemas reactivos en dominios críticos como control industrial e IoT requieren tanto verificabilidad formal como flexibilidad operativa, sin embargo, los enfoques existentes suelen tener dificultades para equilibrar estos requisitos. Nuestro trabajo aborda esta brecha combinando el rigor matemático de las RPNA -que permiten modelado preciso de concurrencia y sincronización orientada a eventos- con la escalabilidad y resiliencia de los microservicios.

Una contribución clave es nuestra taxonomía jerárquica de eventos, que clasifica sistemáticamente los estímulos (temporales, asincrónicos o por fallas) para optimizar su procesamiento en entornos distribuidos. Esta taxonomía permite manejo adaptativo tanto de eventos reconocidos como desconocidos, mejorando la robustez del sistema en escenarios dinámicos. La arquitectura propuesta incluye un orquestador inteligente que coordina microservicios según la clasificación de eventos, manteniendo coherencia con el modelo formal RPNA.

Los resultados esperados incluyen sistemas reactivos portables, escalables y formalmente verificables que preservan las propiedades de diseño durante su ejecución distribuida. El marco beneficia particularmente aplicaciones industriales donde la confiabilidad y capacidad de respuesta en tiempo real son críticas. Trabajos futuros validarán el enfoque mediante casos de estudio en sistemas de pagos y explorarán extensiones con aprendizaje automático para optimización dinámica del desempeño. Esta investigación tiende un puente entre métodos formales y arquitecturas distribuidas modernas, ofreciendo una solución tanto teóricamente rigurosa como prácticamente viable para sistemas reactivos de misión crítica.

Palabras clave: Sistemas reactivos, Redes de Petri No Autónomas, microservicios, taxonomía de eventos, verificación formal, arquitecturas distribuidas.

1 Introducción

Los sistemas reactivos son esenciales en dominios donde la capacidad de procesar eventos en tiempo real determina su eficacia, como en control industrial, IoT y sistemas distribuidos. Estos entornos demandan no solo alta disponibilidad y escalabilidad, sino también mecanismos formales para garantizar consistencia en entornos dinámicos y concurrentes. Sin embargo, su diseño enfrenta desafíos persistentes, como la falta de modelos unificados que equilibren expresividad matemática y flexibilidad arquitectónica (Cerny, Donahoo et al. 2018).

Las arquitecturas de microservicios han surgido como un paradigma viable para sistemas distribuidos, permitiendo modularidad y escalamiento independiente (Shadija,

Rezai et al. 2017). No obstante, su adopción en sistemas reactivos requiere enfoques que integren modelos formales para garantizar coherencia en el manejo de eventos. Este trabajo propone un marco que combina Redes de Petri No Autónomas (RPNA) con microservicios, aprovechando las primeras para modelar comportamientos reactivos de manera formal y los segundos para implementación distribuida y resiliente. Las RPNA ofrecen fundamentos matemáticos para representar concurrencia y sincronización, mientras que los microservicios facilitan el despliegue elástico y la tolerancia a fallos (Richards 2015).

Adicionalmente, se introduce una taxonomía de eventos que categoriza estímulos—temporales, asíncronos o de falla—para optimizar su procesamiento en escenarios críticos. Esta clasificación mejora la capacidad de diagnóstico en sistemas industriales, donde la trazabilidad es crucial (Bhamare, Zolanvari et al. 2020). Al unir modelos formales con patrones modernos de arquitectura, este trabajo busca superar las limitaciones de enfoques ad hoc, ofreciendo un método riguroso para sistemas donde la reactividad y la confiabilidad son prioritarias.

2 Fundamentación Teórica

2.1 Redes de Petri No Autónomas

Las Redes de Petri No Autónomas (RPNA) representan una evolución significativa respecto a las redes de Petri clásicas, al incorporar la capacidad de modular el comportamiento del sistema en función de eventos externos. Esta característica las hace particularmente adecuadas para modelar sistemas donde la dinámica no solo depende del estado interno, sino también de interacciones con el entorno (David and Alla 2010). A diferencia de los modelos tradicionales, las RPNA permiten una representación formal de sistemas concurrentes y distribuidos, facilitando el análisis de propiedades como vivacidad, seguridad y fairness. Además, su capacidad para simular comportamientos complejos las convierte en una herramienta valiosa para la verificación de sistemas reactivos antes de su implementación (Murata 1989).

En el contexto de sistemas industriales y de IoT, donde la concurrencia y la respuesta a eventos son críticas, las RPNA ofrecen un marco teórico robusto para garantizar consistencia en el procesamiento de estímulos. Su integración con arquitecturas modernas, como los microservicios, puede cerrar la brecha entre los modelos formales y las implementaciones distribuidas, asegurando que las propiedades verificadas en el diseño se mantengan durante la ejecución (Macaulay and Singer 2011).

2.2 Arquitectura de Microservicios

La arquitectura de microservicios ha surgido como un paradigma dominante en el desarrollo de sistemas distribuidos, superando muchas de las limitaciones de las arquitecturas monolíticas y orientadas a servicios (SOA). Al descomponer las aplicaciones en servicios independientes, cada uno con responsabilidades bien delimitadas, este enfoque promueve la escalabilidad horizontal, la resiliencia y la capacidad de evolución

incremental (Cerny, Donahoo et al. 2018). Sin embargo, la adopción de microservicios en sistemas reactivos introduce desafíos adicionales, particularmente en la gestión de la consistencia y la coordinación entre servicios distribuidos.

Según (Shadija, Rezaei et al. 2017), la granularidad de los microservicios es un factor determinante en su desempeño: mientras que una descomposición excesiva puede aumentar la sobrecarga por comunicación, una modularización insuficiente limita la escalabilidad. Para sistemas reactivos, donde la latencia y el throughput son críticos, es esencial equilibrar estos aspectos mediante patrones como Event Sourcing y CQRS (Command Query Responsibility Segregation), que permiten manejar flujos de eventos de manera eficiente (Richards 2015).

2.3 Sistemas Reactivos Orientados a Eventos

Los sistemas reactivos se caracterizan por su capacidad de responder de manera asincrónica a estímulos externos, priorizando la escalabilidad, la resiliencia y la capacidad de procesamiento en tiempo real. Estos sistemas son fundamentales en dominios como el control industrial, donde la detección y clasificación de eventos deben realizarse sin comprometer la estabilidad del sistema (Macaulay and Singer 2011). Un diseño efectivo requiere no solo una infraestructura que permita el desacoplamiento entre productores y consumidores de eventos, sino también mecanismos para garantizar el orden, la durabilidad y el procesamiento oportuno de los mismos.

La combinación de modelos formales, como las RPNA, con arquitecturas basadas en eventos y microservicios, ofrece un enfoque integral para abordar estos desafíos. Por un lado, las RPNA proporcionan la formalidad necesaria para garantizar propiedades críticas en el diseño; por otro, los microservicios permiten una implementación flexible y escalable. Esta sinergia es particularmente relevante en entornos donde la reactividad debe coexistir con altos requisitos de confiabilidad, como en sistemas SCADA y aplicaciones de IoT industrial (Cerny, Donahoo et al. 2018).

3 Propuesta

3.1 Taxonomía de Eventos para Sistemas Reactivos

El núcleo de nuestra propuesta radica en una taxonomía de eventos (Fig. 1) diseñada para optimizar el procesamiento en sistemas reactivos complejos. Esta clasificación jerárquica aborda una limitación fundamental identificada en la literatura actual: la falta de esquemas formales para categorizar estímulos en entornos distribuidos donde confluyen microservicios y modelos basados en eventos (Cerny, Donahoo et al. 2018). La taxonomía distingue primero entre eventos reconocidos y desconocidos, estableciendo así un primer filtro para el manejo de estímulos inesperados - aspecto crítico en sistemas industriales donde la resiliencia es prioritaria (Macaulay and Singer 2011).

Para los eventos reconocidos, la clasificación se profundiza según su capacidad de ejecución. Los eventos ejecutables se subdividen en simples (disparadores unitarios de

transiciones) y compuestos (agregaciones de eventos simples que requieren coordinación), diferenciación que responde al desafío de granularidad identificado por (Shadija, Rezai et al. 2017) en arquitecturas distribuidas. Los no ejecutables incorporan el concepto de memoria, permitiendo almacenamiento temporal cuando existen dependencias contextuales, mecanismo que mitiga problemas de sincronización en entornos asíncronos (Richards 2015).

Los eventos desconocidos implementan políticas configurables (almacenamiento o descarte), proporcionando flexibilidad para escenarios donde el análisis posterior pueda revelar patrones significativos. Esta dualidad es particularmente relevante en sistemas de control industrial, donde eventos aparentemente anómalos pueden indicar fallas incipientes (Macaulay and Singer 2011). La taxonomía se implementa mediante un motor de reglas acoplado a la capa de orquestación de microservicios, permitiendo que la clasificación influya tanto en el enrutamiento como en las estrategias de procesamiento.

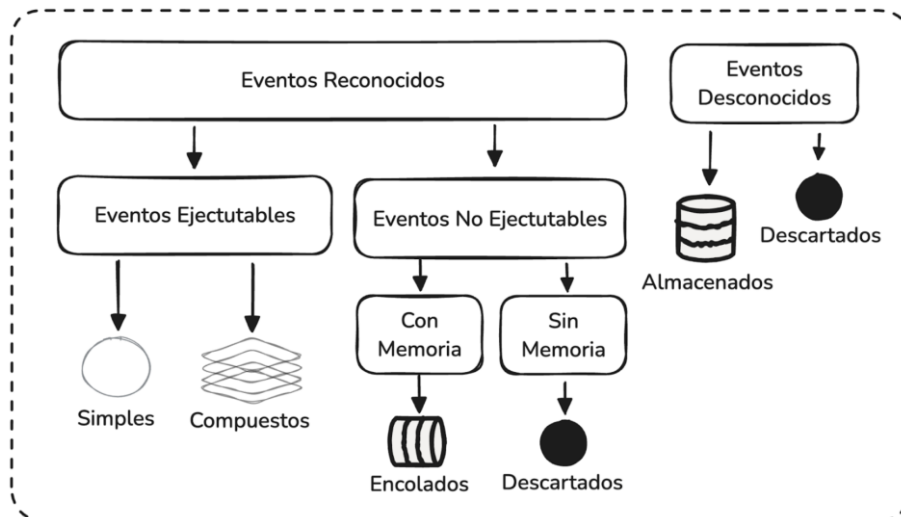


Fig. 1. Taxonomía de Eventos

3.2 Integración con Redes de Petri No Autónomas y Microservicios

La taxonomía propuesta se articula con un modelo formal basado en Redes de Petri No Autónomas (RPNA), donde los tipos de eventos determinan las condiciones de habilitación de transiciones. Esta integración proporciona beneficios significativos: las RPNA ofrecen verificación formal de propiedades del sistema, mientras que la taxonomía garantiza que los eventos se procesen según su naturaleza y prioridad. En el nivel arquitectónico, cada categoría de evento se asocia a microservicios especializados, optimizando así la asignación de recursos computacionales según la complejidad del procesamiento requerido.

Este enfoque híbrido resuelve problemas prácticos identificados en la literatura reciente (Ing. Luis Orlando Ventre 2024). Por un lado, mitiga la sobrecarga de comunicación en microservicios mediante un enrutamiento inteligente basado en la taxonomía (Shadija, Rezai et al. 2017). Por otro, mantiene garantías formales mediante las RPNA, cruciales en dominios donde la corrección del sistema es no negociable (Cerny, Donahoo et al. 2018). La implementación utiliza colas de mensajes con prioridades diferenciadas y mecanismos de retropropagación para eventos compuestos, asegurando consistencia eventual sin comprometer la capacidad de respuesta.

3.3 Solución Propuesta

Nuestra solución integra los principios de las arquitecturas orientadas a microservicios con el rigor formal de las Redes de Petri No Autónomas (RPNA), creando un marco robusto para el desarrollo de sistemas reactivos. El núcleo del sistema consiste en un orquestador inteligente que actúa como intermediario entre los eventos entrantes y los microservicios especializados. Este componente central, inspirado en los patrones de mediación descritos por (Richards 2015), no solo dirige el flujo de eventos según su clasificación taxonómica, sino que también mantiene la coherencia con el modelo formal representado en las RPNA.

La arquitectura propuesta aborda tres desafíos críticos identificados en la literatura reciente. Primero, resuelve el problema de acoplamiento en sistemas distribuidos mediante un patrón de publicación-suscripción que permite a los microservicios operar de manera independiente mientras responden a eventos de su dominio específico (Cerny, Donahoo et al. 2018). Segundo, incorpora mecanismos de retroalimentación que ajustan dinámicamente las prioridades de procesamiento basándose en la taxonomía de eventos, optimizando así el uso de recursos en escenarios de alta carga (Shadija, Rezai et al. 2017). Tercero, implementa un módulo de reconciliación que detecta y resuelve inconsistencias entre el estado real del sistema y su representación en las RPNA, característica esencial para entornos industriales donde la tolerancia a fallos es primordial (Macaulay and Singer 2011).

Cada microservicio en nuestra arquitectura encapsula no solo la lógica de negocio, sino también un fragmento de la RPNA global que modela su comportamiento reactivo. Esta innovadora descomposición espacial del modelo formal permite escalar horizontalmente el sistema sin perder las garantías de corrección proporcionadas por las RPNA. La comunicación entre servicios se realiza mediante un bus de eventos que soporta diferentes protocolos (HTTP/REST, gRPC, mensajería asíncrona), seleccionados automáticamente según el tipo de evento y los requisitos de calidad de servicio.

4 Resultados Esperados

La implementación de esta arquitectura híbrida promete avances significativos en tres dimensiones clave. En cuanto a portabilidad, el uso de contenedores para empaquetar microservicios junto con sus fragmentos de RPNA asociados permite desplegar el sistema en diversas plataformas, desde entornos cloud hasta dispositivos edge en

instalaciones industriales (Macaulay and Singer 2011). La escalabilidad se beneficia de la capacidad de distribuir selectivamente la carga de procesamiento según la categorización taxonómica de eventos, evitando así los cuellos de botella comunes en arquitecturas monolíticas (Shadija, Rezai et al. 2017).

La verificabilidad del sistema constituye quizás su ventaja más distintiva. Al mantener una correspondencia biyectiva entre los microservicios operacionales y el modelo formal en RPNA, es posible realizar análisis estáticos de propiedades como ausencia de deadlocks o cumplimiento de invariantes, incluso en sistemas distribuidos complejos (Cerny, Donahoo et al. 2018). Esto representa un avance significativo respecto a las arquitecturas basadas puramente en eventos, donde tradicionalmente ha sido difícil garantizar propiedades globales.

La taxonomía de eventos incorporada no solo mejora la eficiencia del procesamiento, sino que también dota al sistema de una capacidad adaptativa única. Eventos inicialmente clasificados como desconocidos pueden ser reevaluados y reclasificados dinámicamente a medida que el sistema "aprende" nuevos patrones, una característica particularmente valiosa en entornos industriales donde las condiciones operativas evolucionan constantemente (Macaulay and Singer 2011). Esta flexibilidad, combinada con el formalismo de las RPNA, crea un marco único que equilibra rigor matemático con adaptabilidad práctica.

5 Conclusión

Este trabajo presenta un marco innovador que integra Redes de Petri No Autónomas (RPNA) con arquitecturas de microservicios para abordar los desafíos fundamentales en el diseño de sistemas reactivos. La combinación propuesta logra un equilibrio crucial entre rigor formal y flexibilidad operativa, permitiendo el desarrollo de sistemas que son simultáneamente verificables matemáticamente y altamente adaptables a entornos dinámicos. Como destacan (Cerny, Donahoo et al. 2018), esta sinergia entre modelos formales y arquitecturas distribuidas modernas representa un avance significativo en el campo de los sistemas reactivos empresariales e industriales.

La taxonomía de eventos desarrollada constituye una contribución clave al proporcionar un esquema estructurado para clasificar y gestionar estímulos en sistemas complejos. Este enfoque sistemático, como señala (Richards 2015), es particularmente valioso en escenarios donde la naturaleza heterogénea de los eventos puede comprometer la estabilidad del sistema. Nuestra categorización de eventos reconocidos y desconocidos, con sus respectivas subclases, ofrece un mecanismo robusto para manejar tanto flujos operativos normales como situaciones excepcionales, aspecto crítico en entornos como los sistemas de control industrial (Macaulay and Singer 2011).

Los resultados esperados de nuestra propuesta - portabilidad, escalabilidad y verificabilidad - responden directamente a las limitaciones identificadas en la literatura actual sobre sistemas distribuidos. La capacidad de descomponer un modelo formal de RPNA en microservicios especializados, como analiza (Shadija, Rezai et al. 2017), permite superar el tradicional compromiso entre consistencia global y desempeño distribuido.

Esto es especialmente relevante en dominios como los sistemas de pagos, donde nuestro trabajo futuro planea validar empíricamente el enfoque.

La principal contribución de esta investigación radica en demostrar cómo los paradigmas formales y las arquitecturas modernas pueden complementarse mutuamente. Las RPNA proporcionan la base teórica para garantizar propiedades críticas del sistema, mientras que los microservicios permiten implementaciones flexibles y escalables. Esta dualidad abre nuevas posibilidades para el desarrollo de sistemas reactivos en dominios donde tanto la corrección formal como la adaptabilidad operacional son requisitos fundamentales.

5.1 Trabajos Futuros

La implementación de un prototipo funcional en el dominio de sistemas de pagos representa solo el primer paso en una agenda de investigación más amplia. Como señalan (Cerny, Donahoo et al. 2018), la validación empírica de arquitecturas híbridas que combinan microservicios con modelos formales sigue siendo un área poco explorada, particularmente en escenarios del mundo real con requisitos estrictos de rendimiento y consistencia. Nuestro experimento con pagos instantáneos y programados servirá como banco de pruebas para evaluar no solo los aspectos técnicos de la solución, sino también su capacidad para manejar flujos de trabajo complejos con dependencias temporales y lógicas.

Una línea prometedora de investigación futura explora la integración de técnicas de aprendizaje automático con nuestro marco basado en Redes de Petri No Autónomas (RPNA). Como identifican (Shadija, Rezai et al. 2017), los sistemas de microservicios modernos enfrentan el desafío de optimizar dinámicamente su desempeño ante cargas de trabajo variables. La incorporación de modelos predictivos que anticipen patrones de eventos podría mejorar significativamente la eficiencia del orquestador central, permitiendo un aprovisionamiento preventivo de recursos y una asignación más inteligente de tareas.

En el contexto de sistemas industriales, donde la ciberseguridad es primordial (Macaulay and Singer 2011), planeamos extender nuestra taxonomía de eventos para incorporar categorías específicas de amenazas y anomalías. Esto permitiría desarrollar mecanismos de detección temprana integrados directamente en el modelo formal, combinando así las ventajas de las RPNA para modelar comportamientos del sistema con técnicas avanzadas de monitoreo de seguridad.

Otra dirección importante implica investigar patrones de descomposición óptima para mapear modelos de RPNA complejos a conjuntos de microservicios. Como discute (Richards 2015), el diseño efectivo de arquitecturas de microservicios requiere equilibrar múltiples factores, incluyendo cohesión funcional, granularidad adecuada y minimización de acoplamiento. Nuestro trabajo futuro explorará métricas cuantitativas para evaluar diferentes estrategias de partición del modelo formal, con el objetivo de desarrollar directrices prácticas para ingenieros de software.

References

- Bhamare, D., M. Zolanvari, A. Erbad, R. Jain, K. Khan and N. Meskin (2020). "Cybersecurity for industrial control systems: A survey." computers & security **89**: 101677.
- Cerny, T., M. J. Donahoo and M. Trnka (2018). "Contextual understanding of microservice architecture: current and future directions." ACM SIGAPP Applied Computing Review **17**(4): 29-45.
- David, R. and H. Alla (2010). Discrete, continuous, and hybrid Petri nets. Springer Science & Business Media.
- Ing. Luis Orlando Ventre, D. I. O. M., Ing. Mauricio Ludemann, Agustín Carranza, David D'Andrea1, Enzo Candotti1 (2024). "Caso de estudio: metodología para el diseño y desarrollo de sistemas embebidos distribuidos."
- Macaulay, T. and B. L. Singer (2011). Cybersecurity for industrial control systems: SCADA, DCS, PLC, HMI, and SIS, CRC Press.
- Murata, T. (1989). "Petri Nets: Properties, Analysis and Applications." Proceedings of the IEEE: Vol. 77, No. 74, pp. 541-580.
- Richards, M. (2015). Microservices vs. service-oriented architecture, O'Reilly Media Sebastopol.
- Shadija, D., M. Rezaei and R. Hill (2017). Microservices: granularity vs. performance. Companion Proceedings of the 10th international conference on utility and cloud computing.