

PetriNet Studio: Plataforma SaaS para Simulación, Análisis y Generación Automática de Código Concurrente a partir de Redes de Petri

Gabriel Emanuel Valenzuela¹ Mauricio Ludemann^{1,2} Luis Omar Ventre² Orlando Micolini²

¹Maestría en Ingeniería de Software – Facultad de Informática, UNLP

²Laboratorio de Arquitectura de Computadoras (LAC) – FCEfyN, UNC

gabriel.valenzuela@unc.edu.ar | mauri.ludemann@unc.edu.ar

Contexto

Este trabajo se enmarca en el proyecto “**Diseño e implementación de software y hardware optimizados para sistemas de computación paralelos en ingeniería**” (Proyectos Estimular 2026), ejecutado en el **Laboratorio de Arquitectura de Computadoras (LAC)** de la FCEfyN, UNC, en colaboración con la Facultad de Informática de la UNLP. El LAC lleva más de una década de investigación sobre procesadores de Petri implementados en hardware, metodologías de diseño de sistemas embebidos y reactivos basados en Redes de Petri (RdP), y algoritmos para la determinación automática de hilos en redes S3PR.

Se presenta **PetriNet Studio**, una plataforma SaaS basada en microservicios para modelado, simulación y análisis de RdP y Redes de Petri No Autónomas (NAPN), con **generación automática de código concurrente** (monitores e hilos) para modelos S3PR. La plataforma elimina las barreras de instalación propias de las herramientas de escritorio existentes, integrando nativamente editor gráfico, simulación interactiva, análisis estructural y generación de código en un único entorno web accesible. El trabajo se articula directamente con la tesis de maestría del primer autor (UNLP), la tesis doctoral de Ventre (UNC) y el trabajo de Ludemann et al. sobre arquitecturas reactivas basadas en NAPN y microservicios.

Líneas de Investigación y Desarrollo

- **Arquitecturas SaaS para herramientas de modelado formal.** Diseño e implementación de plataformas basadas en microservicios que trasladen las capacidades de simulación y análisis de RdP a entornos web accesibles sin instalación local, con soporte multiusuario y despliegue en nube.
- **Simulación distribuida con preservación semántica.** Estudio de mecanismos de sincronización para garantizar la atomicidad del disparo de transiciones en entornos de microservicios, centralizando la gestión del marcado en el *simulation-engine*: cada disparo se ejecuta como transacción atómica protegida por exclusión mutua.
- **Generación automática de código concurrente a partir de modelos S3PR.** Implementación del algoritmo de Ventre y Micolini para determinación de cantidad y responsabilidad de hilos en redes S3PR, con generación de código en C++, Java y Go, y verificación de ausencia de condiciones de carrera mediante ThreadSanitizer.
- **Redes de Petri No Autónomas (NAPN) en arquitecturas reactivas.** Empleo de NAPN como núcleo de decisión para microservicios reactivos; extensión del motor de simulación y del módulo de análisis para soportar la ecuación de estado generalizada y la taxonomía de eventos (simples, compuestos, temporales y de estado).
- **Validación empírica comparativa.** Comparación de grafos de alcanzabilidad con PIPE y LoLA sobre modelos del repositorio MCC; benchmarks de rendimiento frente a PIPE 4 e IOPT-Tools; verificación formal de corrección del código generado.

Formación de Recursos Humanos

Tesis de posgrado vinculadas:

- **Tesis de Maestría en Ingeniería de Software (UNLP)** – Gabriel E. Valenzuela: “*PetriNet Studio: Diseño e implementación de una plataforma SaaS para la simulación y análisis de Redes de Petri.*” En curso. Director: Dr. Orlando Micolini. Codirector: Mg. Pablo Thomas.
- **Tesis Doctoral (UNC)** – Luis Omar Ventre: “*Análisis de paralelismo y determinación de hilos en sistemas embebidos modelados con Redes de Petri S3PR.*” En curso. Director: Dr. Orlando Micolini.
- **Tesis de Maestría (UNLP)** – Mauricio Ludemann: “*Arquitectura de microservicios reactivos basada en Redes de Petri No Autónomas.*” En curso. Director: Dr. Orlando Micolini.

Docencia vinculada:

- Gabriel Valenzuela dicta **Sistemas Operativos I y Algoritmos y Estructuras de Datos** en la carrera de Ingeniería en Computación (UNC), incorporando el uso de Redes de Petri para la enseñanza de concurrencia y sistemas reactivos.
- Formación continua en modelos formales y arquitecturas de software distribuido en el ámbito del LAC-UNC, articulando investigación y actividad docente.

Resultados Obtenidos / Esperados

- **Prototipo funcional parcial.** Módulos operativos verificados: *editor gráfico SPA* con soporte para plazas, transiciones, arcos con peso, marcado inicial y asociación de tipos de evento para NAPN; *motor de simulación* en C++ con disparo atómico, tres modos de selección de transición (manual, aleatorio, por prioridades) y evaluación de condiciones de evento externo; *módulo de análisis* en Python con cálculo de matriz de incidencia extendida, P-invariantes, T-invariantes, detección de sifones/trampas y clasificación estructural; *servicio de generación de código* con descomposición S3PR y generación de monitor en C++.
- **Arquitectura de cinco microservicios independientes.** (1) *editor-service*: interfaz gráfica SPA; (2) *simulation-engine*: motor C++ con gestión atómica del marcado; (3) *analysis-service*: módulo Python para invariantes y clasificación estructural (incluye S3PR y NAPN); (4) *codegen-service*: implementación del algoritmo Ventre-Micolini para generación de monitores e hilos en C++ y Java; (5) *api-gateway*: punto de entrada único con autenticación y enrutamiento.
- **Verificación funcional inicial.** Todos los módulos han sido verificados mediante casos de prueba manuales comparando resultados con PIPE 4 para los mismos modelos de referencia.
- **Resultados esperados al completar el trabajo:** (a) Plataforma SaaS completa desplegada en nube pública con autenticación multiusuario; (b) validación empírica formal con resultados cuantitativos de corrección y rendimiento; (c) extensión del generador de código a redes NAPN con semántica de paso máximo; (d) publicación de al menos un artículo en congreso o revista indexada con resultados de la evaluación.

Palabras clave: Redes de Petri · NAPN · SaaS · microservicios · generación automática de código · S3PR · sistemas concurrentes · sistemas embebidos