

# Utilización de Redes de Petri para la enseñanza de la concurrencia en la Ingeniería en Computación

Luis O. Ventre<sup>1</sup>, Orlando Micolini<sup>1</sup>, Armando Micolini<sup>2</sup>

<sup>1</sup> UNC-FCEFyN, Córdoba, Argentina, <sup>2</sup> UNC y UNdeC Argentina  
(luis.ventre, orlando.micolini, armando.micolini) @unc.edu.ar

**Abstract.** La creciente importancia de los sistemas reactivos (SR) y ciberfísicos (CF) ha hecho que el desarrollo práctico de software concurrente sea un componente clave en los planes de estudio de informática. En los últimos años, se ha establecido una conexión sólida entre la teoría de la concurrencia expresada con redes de Petri (RdP) y la práctica de la implementación de programas concurrentes. Este artículo presenta un mapeo semántico innovador en la enseñanza de la programación concurrente para obtener código a partir de una RdP. Este enfoque está totalmente integrado, convirtiendo el modelo en código de lógica ejecutable por el lenguaje, lo que permite a los estudiantes interactuar entre el modelo basado en RdP y el código.

**Keywords:** Programación Concurrente, Redes de Petri, Enseñanza, Competencias

## 1 Introducción

La computación concurrente ha ganado importancia en informática debido a avances en modelado teórico de sistemas y desarrollo práctico de programas. La Industria 4.0 requiere técnicas formales en diseño de CF, embebidos, críticos, reactivos y dirigidos por eventos. Los sistemas concurrentes son multi-hilo, interactúan con variables y eventos heterogéneos y no determinísticos. La programación concurrente (PC) es fundamental en educación informática y requiere abordar comunicación, coordinación, modelos de memoria compartida, atomicidad, sincronización y espera condicional para garantizar eficacia y eficiencia.

Como antecedente a este trabajo, y como aporte del presente grupo de investigación, se encuentra los siguientes trabajos [1] [2, 3] que dan respuesta a los detalles y fundamentos para las soluciones a implementar por los estudiantes.

En el siguiente apartado, se presentan los objetivos de este trabajo, en la sección dos se presentan los fundamentos. La sección tres detalla las competencias y se explica el mecanismo propuesto para la implementación. Finalmente, las conclusiones.

### 1.1 Objetivos

Este trabajo presenta la metodología utilizada para la enseñanza de la PC haciendo uso de la competencia [4], la cual se divide en cuatro actividades: tarea, conocimiento, habilidad y disposición.

### 1.2 Características de los SR y CF y su relación con la PC

Los SR y CF comparten la interacción en tiempo real con el entorno y la integración de hardware y software. Los sistemas CF son más complejos al incluir sistemas físicos y digitales, mientras que los SR se enfocan en la respuesta a eventos en tiempo real. Para el diseño y desarrollo efectivo de estos sistemas, es fundamental la PC que permite coordinar la interacción con dispositivos físicos y manejar múltiples tareas simultáneamente. En definitiva, la PC es clave para el desarrollo de SR y CF.

### 1.3 Justificación del uso de Redes de Petri

Las redes de Petri son útiles para modelar y analizar formalmente sistemas complejos con múltiples procesos o eventos simultáneos[5]. Pueden identificar conflictos y problemas de exclusión mutua y sincronización, y permiten la simulación de diferentes escenarios y análisis del comportamiento del sistema en diferentes situaciones. Las características que se pueden modelar incluyen conflictos de recursos, sincronización y competencia, y estos conflictos pueden ser resueltos mediante técnicas como la asignación de prioridades, la sincronización y la exclusión mutua. Las RdP son aplicables a técnicas de diseño orientadas a procesos, datos, control y objetos, y pueden incluir restricciones temporales como tiempos de espera o límites[6].

## 2 Competencia para la enseñanza de la PC

Desde 2005, la asignatura de PC es obligatoria en la FCEFYN de la UNC (Universidad Nacional de Córdoba), y en la última parte del curso se lleva a cabo una tarea integradora. El objetivo de esta tarea es acompañar a los estudiantes a comprender los mecanismos y problemas que implica diseñar e implementar sistemas concurrentes, desarrollando habilidades en programación de SR y CF, fomentando la creatividad y el pensamiento innovador en la solución de problemas complejos en un entorno cambiante y desafiante.

### 2.1 Competencia

Aprender y adquirir habilidades para la PC aplicada a SR y CF.

**Tarea:** Diseñar y realizar un sistema de control autónomos para resolver un trabajo específico en un entorno dinámico y cambiante.

Los estudiantes tienen que diseñar y programar un sistema de control (hardware y software) que permita interactuar con su entorno en tiempo real, recolectando datos de sensores y tomando decisiones en función de esos datos. Además, el sistema debe ser capaz de coordinar la acción de varios hilos simultáneamente para lograr la tarea encomendada de manera eficiente y efectiva.

Para el diseño del sistema se utilizarán RdP y la programación del mismo se realizará con Java.

El alumno evaluará la capacidad del sistema para responder en tiempo real a los cambios en el entorno (eventos), la eficiencia en el uso de recursos, la capacidad para

coordinar y controlar la acción de múltiples recursos simultáneamente, así como la capacidad para manejar situaciones imprevistas y condiciones inesperadas.

**El conocimiento** es decir "saber qué", el estudiante emplea conocimientos previos de modelos, principios fundamentales y técnicas de la POO, testing y expresiones regulares. Asimismo, aplica los conocimientos adquiridos en la asignatura, tales como RdP, vivacidad, interbloqueo, inanición, semáforos, sección crítica, sincronización, recursos compartidos y conflictos [7].

**Las habilidades** y estrategia necesarias para "aplicar el saber qué". El estudiante debe ser capaz de diseñar la arquitectura de la solución desacoplando la política de la lógica y las acciones, y establecer una estrategia adecuada para probar el sistema, utilizando los conocimientos adquiridos. El estudiante debe establecer parámetros de cuantificación y mostrar que el sistema resultante se adapta a entornos cambiantes.

**Las disposiciones**, aquí se aborda el "saber por qué". En PC de envergadura, no es posible determinar la corrección únicamente mediante el testing, por lo que es necesario recurrir a métodos formales, como las RdP. Es crucial encontrar un equilibrio entre el uso de testing y los métodos formales, moderando el conocimiento y la habilidad. Dado que se trata de sistemas críticos en los que la vida o los bienes pueden estar en riesgo.

Se espera que el estudiante realice procesos cognitivos de orden superior de forma progresiva. Se evalúan estas dimensiones a través de la observación del proceso y la calidad del trabajo producido.

### 3 Como generar código a partir del modelo

Una vez que se ha diseñado un modelo de RdP, es necesario generar código ejecutable para implementar el sistema en un entorno de software o hardware. El proceso de generación de código se puede dividir en dos etapas principales:

**Etapas:** el uso de un Monitor de Concurrencia y la expresión del modelo haciendo uso de la ecuación de estado generalizada [2]

**Etapas:** la determinación de la cantidad de Hilos y sus secuencias de ejecución [3], la determinación de la política del sistema y codificación de las acciones

La Fig. 1 muestra la arquitectura de la solución, donde se destacan los siguientes componentes: **Monitor:** es responsable de gestionar el acceso de los hilos en exclusión mutua, para determinar la próxima acción a ejecutar. **RdP:** modela y representa la lógica del sistema, y es el mecanismo utilizado por el monitor para determinar las posibles acciones a ejecutar por los hilos. **Política:** es el mecanismo utilizado por el monitor para resolver los conflictos del sistema. **Colas de hilos bloqueados:** se encargan de bloquear los hilos que solicitan el disparo de una transición requerida. **Acciones:** son las tareas que deben realizar los hilos en función del estado actual del sistema. **Secuencia de disparos:** representa las transiciones correspondientes con las acciones a ejecutar. En este trabajo se utiliza el modelo y la ecuación de estado para generar automáticamente el código a ejecutar. Esta metodología ha sido empleada para implementar sistemas complejos en distintos entornos de hardware y software [referencias].

4

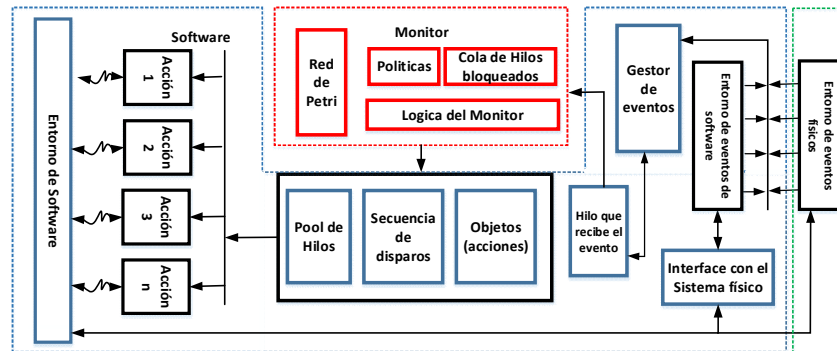


Fig. 1. Arquitectura de la implementación del sistema

#### 4 Conclusiones

Este artículo presenta como llevar adelante la competencia inherente a PC en un entorno de SR y CF haciendo uso de RdP, ejecutando el modelo, de la ecuación de estado sin perder la validación en la codificación. La depuración está basada en que la lógica es ejecutada por el modelo, por lo que hay que validar que la implementación del monitor preserva la ejecución prescripta por este; esto se realiza controlando el cumplimiento de los invariantes de transición y plaza. La extensión de ejecutar RdP es completamente nueva y, como tal, hasta ahora no se han encontrado publicaciones realizadas por otros grupos de investigación.

#### References

1. Micolini, O., L.O. Ventre, and M. Ludemann. Methodology for design and development of Embedded and Reactive Systems Based on Petri Nets. in 2018 IEEE Biennial Congress of Argentina (ARGENCON). 2018. IEEE.
2. Micolini, O., et al. Ecuación de estado generalizada para redes de Petri no autónomas y con distintos tipos de arcos. in XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016). 2016.
3. Ventre, L.O. and O. Micolini. Algoritmos para determinar cantidad y responsabilidad de hilos en sistemas embebidos modelados con Redes de Petri S3PR. in XXVII Congreso Argentino de Ciencias de la Computación (CACIC)(Modalidad virtual, 4 AL 8 DE OCTUBRE DE 2021.). 2022.
4. Force, C.T., Computing Curricula 2020: Paradigms for Global Computing Education November 2020. 2020.
5. Murata, T., Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, 1989: p. Vol. 77, No. 4, pp. 541-580.
6. Diaz, M., Petri nets: fundamental models, verification and applications. 2013: John Wiley & Sons.
7. Ausubel, D.P., J.D. Novak, and H. Hanesian, Psicología educativa: un punto de vista cognoscitivo. Vol. 3. 1976: Trillas México.