



UNIVERSIDAD NACIONAL DE SAN LUIS

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS Y NATURALES

**Estrategias para Interconectar el Dominio del  
Problema con el Dominio del Programa en Sistemas  
Multiparadigmas**

**Tesis**

Para optar por el grado de Doctor en Ingeniería Informática

**Autor**

Enrique Alfredo Miranda

**Director:** Dr. Daniel Riesco (Universidad Nacional de San Luis)

**Co-Director:** Dr. Mario Berón (Universidad Nacional de San Luis)

San Luis - Argentina

-Marzo 2018-



*A mis queridos viejos,  
por su apoyo incondicional.*

*Para Santi, Juli y Viki,  
por su amor y paciencia infinita.*

*Para el abu Daniel, ingeniero de las cosas simples,  
con un doctorado en la vida misma.*



Serás lo que hay que ser, si no, eres nada.

*José de San Martín*

La frase citada es una más resonantes del General José de San Martín y se preserva en el inconsciente de muchas personas como máxima a la que nunca se puede dar por terminada. Sin embargo, considero que para que alguien sea lo que debe ser, su curso de vida se ve alterado por distintos factores que hacen que éste llegue a destino de forma tal que nunca vuelve a ser el mismo. Uno de los factores más influyentes es el humano; es decir, todas aquellas personas que hacen posible que llegues a destino como una persona distinta a aquella que empezó el camino. En esta etapa donde realicé el doctorado mi destino se vio alterado de manera positiva e irreversiblemente por muchas personas a las cuales no puedo dejar de mencionar y plasmar en este documento.

En primer lugar, quiero agradecer a mi familia por ser incondicionales y brindar todo su apoyo y comprensión, mucho más cuando las cosas se pusieron difíciles. A mi familia de siempre: Ruth, “Cabezón”, Caro y Richard; también a mi “nueva familia”: Viki, Juli y Santi. Quisiera pedirles perdón por haberlos abandonado en varias ocasiones en donde el trabajo requería de mi mayor dedicación y esfuerzo. Ojala pueda compensar ese tiempo con más momentos compartidos con todos ustedes.

Por otro lado quisiera agradecer a esos amigos con los que tuve la oportunidad de compartir la docencia y la investigación: Corina, Edgardo y Hernán. Me brindaron mucho de su tiempo donde vivimos momentos divertidos y trabajamos arduamente con objetivos en común. Siempre estaré agradecido por la dedicación que pusieron cuando los necesité y considero que son excelentes profesionales y mucho mejores compañeros.

Quiero agradecer profundamente a Mario Berón y a Daniel Riesco, no sólo por ser mis directores, sino también por ser grandes colegas y mucho más importante, nobles orientadores. Su dedicación a la investigación y la docencia inspiró en mi la motivación de investigar y perseguir un posgrado de esta magnitud. Quiero que sepan que tienen el mayor de mis respetos y que fue un honor haber trabajado a su lado; ambos son para mí un modelo a seguir desde muchos aspectos.



# Resumen

Sin lugar a dudas, una de las tareas más complejas y que más tiempo consume en el ciclo de vida de una aplicación es la de Mantenimiento y Evolución de Software (MES). Dentro del entorno de MES, las tareas que más tiempo demandan son aquellas que debe ejecutar el ingeniero de software para lograr un completo entendimiento del sistema. A partir de la necesidad de asistir al arduo proceso de comprensión mencionado anteriormente, surge una disciplina de la Ingeniería de Software denominada Comprensión de Programas (CP). La CP se presenta como un área de investigación interesante para impulsar el trabajo de MES a través de técnicas y herramientas que asistan al ingeniero de software en la difícil tarea de analizar y comprender sistemas. En esta tesis doctoral se presenta una estrategia de CP que busca asistir al arduo proceso cognitivo que implica la comprensión de sistemas desarrollados usando lenguajes multiparadigma, mediante la interconexión de los Dominios del Problema y del Programa. El primero hace referencia a la salida del sistema en estudio, el segundo se relaciona con los artefactos de software utilizados para producir dicha salida. Esta vinculación es muy importante ya que permite establecer puentes cognitivos que asisten de manera sólida al ingeniero de software en las tareas de comprensión. Para alcanzar este objetivo, se utilizan distintas técnicas de Ingeniería Reversa que permitirán obtener un modelo de caso de uso de UML el cual servirá como medio para lograr la vinculación. Este tipo de modelo es aceptado como un componente provechoso para describir los requisitos de comportamiento para un sistema y como un medio efectivo de comunicación entre los involucrados entorno al mismo. Para derivar este modelo, la estrategia realiza, en términos generales, los siguientes pasos: i) extracción de información estática del sistema relacionada con diferentes tipos de artefactos de software, estableciendo cierto énfasis en los widgets de GUI, los cuales están estrechamente relacionados con el Dominio del Problema del sistema; ii) filtrado de la información extraída; iii) implementación de un proceso de agrupamiento específico que tiene en cuenta la información extraída; y iv) mapeo del modelo de cluster a un modelo de casos de uso de UML. Aunque las actividades desarrolladas en estos pasos se usan con frecuencia en el contexto de Ingeniería Reversa, la estrategia introduce enfoques inusuales con respecto a las propuestas que se encuentran en la literatura dis-

ponible. Más específicamente, la estrategia presenta: un conjunto de métricas que infiere la importancia relativa de un método o función dentro del sistema analizado, un proceso de reducción de información irrelevante y una nueva técnica para agrupar artefactos de software para luego mapear los mismos en un modelo de caso de uso. La evaluación del enfoque sugiere que la estrategia asiste al ingeniero de software a comprender un sistema que proporciona un modelo de caso de uso detallado.