



Ministerio de Cultura y Educación  
 Universidad Nacional de San Luis  
 Facultad de Ciencias Físico Matemáticas y Naturales  
 Departamento: Informatica  
 Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2009)  
 (Programa en trámite de aprobación)  
 (Presentado el 18/06/2010 19:24:30)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
(OPTATIVAS) TOPICOS AVANZADOS EN INGENIERIA DE SOFTWARE	LIC.EN CS.DE LA COMPUTACION		2009	2° cuatrimestre

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
FUNES, ANA MARIA	Prof. Responsable	P.Adj Exc	40 Hs
BERON, MARIO MARCELO	Responsable de Práctico	JTP Exc	40 Hs

### III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	3 Hs	1 Hs	2 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
31/08/2009	04/12/2009	14	84

### IV - Fundamentación

Las computadoras están siendo empleadas para múltiples tareas donde la presencia de fallas puede traer aparejado severas consecuencias no sólo a nivel de pérdidas económicas sino de vidas humanas.

Esta tendencia se está incrementando cada vez más. Las computadoras juegan un rol esencial en el control aeroespacial, en la aeronavegación, en el control de trenes, autos, reactores nucleares y equipamiento hospitalario, por dar simplemente algunas pocas de las aplicaciones más críticas. Si las personas están preocupadas por un aterrizaje seguro de su avión o que su cuenta bancaria tenga el saldo correcto, entonces concordarán que es de vital importancia que los sistemas de computación que están por detrás de ellos sean altamente confiables.

Para construir software de alta calidad, sin descartar otras alternativas, una de las vías más importantes es definiendo el dominio y los requerimientos a través de especificaciones formales.

El método RAISE es un método formal que está basado sobre principios tales como: desarrollo separado, desarrollo en pasos sucesivos, inventar y verificar, y rigurosidad.

El lenguaje de especificación formal usado en este caso es el RAISE Specification Language (RSL) dado que es reconocido en la industria del software para especificaciones formales de desarrollos reales. RSL es un lenguaje de especificación formal que tiene tanto sintaxis como semántica formal. Por ser un lenguaje formal está basado en el formalismo de la matemática, usando conceptos tales como la teoría de conjuntos, lógica de primer orden, lógica de orden superior, que están netamente orientados a construir modelos, ya sea describiendo un dominio de la realidad o describiendo una herramienta a desarrollar así como sus requerimientos.

## V - Objetivos / Resultados de Aprendizaje

El objetivo de la materia es que los alumnos tengan la posibilidad de incursionar en los métodos formales como herramienta de producción de software de alta calidad, en particular que logren conocer a nivel de detalle el método RAISE, "Rigorous Approach to Industrial Software Engineering". Se llevarán a cabo un número de casos de estudios en especificación y diseño usando RAISE, en diferentes áreas de aplicación, usando diferentes estilos (aplicativo, imperativo, concurrente). Adicionalmente, se presenta al alumno la alternativa del uso combinado de técnicas formales y semi-formales de producción de software. Para cubrir este objetivo, se usan los diagramas de clase de UML y especificaciones formales en RSL, derivadas a partir de este tipo de diagrama.

## VI - Contenidos

### Unidad I: Introducción

- Introducción al curso y a los Métodos Formales.
- Pro y contra de los Métodos Formales.
- Estilos de especificaciones
- Estilos de desarrollo.
- Grados de rigor.
- Introducción al método de RAISE.

### Unidad II: Especificaciones formales en RSL

- Características generales de RSL
- Especificaciones orientadas a la propiedad
- Especificaciones orientadas al modelo.
- Estilos: aplicativo, imperativo, concurrente.

### Unidad III: Aspectos Sintácticos de RSL

- Expresión básica de clase.
- Definiciones de tipos.
- Expresiones de tipo: built-in y definidos por el usuario. Sorts. Constructores de tipo.
- Definiciones de valores: funciones y constantes. Estilos: explícito e implícito.
- Definiciones de axiomas. Chaos. Lógica condicional de RAISE. Expresiones if.

### Unidad IV: Constructores de Tipos

- Producto Cartesiano.
- Funciones. Funciones totales y parciales.
- Conjuntos finitos e infinitos. Definiciones explícitas e implícitas. Operadores.
- Listas finitas e infinitas. Definiciones explícitas e implícitas. Operadores.
- Mapas finitos determinísticos y mapas parciales. Definiciones explícitas e implícitas. Operadores.

### Unidad V: Subtipos

- Expresiones de subtipos.
- Subtipos vs. axiomas
- Tipos maximales
- Subtipos vacíos.

### Unidad VI: Variants

- Constructores de constantes
- Constructores de records
- Destruccion y reconstruccion
- Uniones disjuntas de tipos
- Definiciones de short records.

### Unidad VII: Expresiones Case y Let

- Forma general de una expresión CASE. Uso.
- Patrones: Literales, Wildcard, de Productos, de Nombres, de Records, de Listas.

- Forma general de expresiones Let implícitas y explícitas. Uso.
- Equivalencias entre Let y Case.
- Anidamiento.

#### **Unidad VIII: Módulos**

- Expresión básica de clase.
- Esquemas
- Objetos
- Expresión de clase extendida.
- Renombramiento
- Ocultamiento
- Esquemas parametrizados.

#### **Unidad IX: Combinación de Técnicas Formales y semi-formales**

- Captura de requerimientos.
- Semántica formal en RSL de diagramas de clases en UML.
- Uso de UML2RSL en un caso práctico.

### **VII - Plan de Trabajos Prácticos**

Práctico I: Especificaciones formales en RSL: orientadas a la propiedad, orientadas al modelo. Estilos: aplicativo, imperativo, concurrente.

Práctico II: Aspectos Sintácticos de RSL: clases, tipos, built-in y definidos por el usuario. Constructores de tipo, valores: funciones y constantes. Estilos: explícito e implícito. Axiomas. Expresiones if.

Práctico III: Lógica condicional de RAISE.

Práctico IV: Sets y Producto.

Práctico V: Listas y Mapas.

Práctico VI: Subtipos. Tipos maximales. Subtipos vacíos.

Práctico VII: Variants. Records. Constructores, destructores y reconconstructores. Uniones disjuntas de tipos.

Práctico VIII: Desarrollo de una especificación formal para un caso real.

### **VIII - Regimen de Aprobación**

Para regularizar o aprobar la asignatura se deberá presentar un proyecto final sobre la temática del curso. El mismo consistirá en el desarrollo de una especificación formal haciendo uso de las herramientas de RAISE para un caso práctico particular. Si la nota obtenida está entre 6 y 7 el alumno regulariza la asignatura y si es mayor o igual a 7 promociona.

En caso de regularizar, el alumno deberá rendir un examen final.

No se aceptan alumnos libres.

### **IX - Bibliografía Básica**

- [1] - Dang Van, H.; George, C.; Janowski, T.; Moore, R. (Eds.). Specification Case Studies in RAISE. Springer. 2002.
- [2] - The RAISE Language Group. "The RAISE Specification Language". Prentice-Hall International, 1992.
- [3] - The RAISE Method Group. "The RAISE Development Method". Prentice-Hall International, 1995.
- [4] - Dasso, A., Funes, A., "Formal Methods in Software Engineering" en Encyclopedia of Information Science and Technology, Idea Group, Enero de 2005.
- [5] - Funes, A., George, C., "Formalizing UML class diagrams", "capítulo 8 de UML and the Unified Process, Idea Group, 2003.

## X - Bibliografía Complementaria

- [1] - Sten Agerholm and Peter G. Larsen, "A Lightweight Approach to Formal Methods", In Proceedings of the International Workshop on Current Trends in Applied Formal Methods, Boppard, Germany, Springer-Verlag, October 1998.
- [2] - Daniel Jackson and Jeannette Wing, "Lightweight Formal Methods", IEEE Computer, April 1996.
- [3] - Bowen, J., & Hinchey, M., "Seven more myths of formal methods". IEEE Software, 12(3). Julio, 1995.
- [4] - J.A. Hall, Seven Myths of Formal Methods. IEEE Software, 7(5):11-19, September 1990.
- [5] - J.P. Bowen and M.G. Hinchey, "Ten Commandments of Formal Methods ... Ten Years Later". IEEE Computer, 39(1):40-48, January 2006.
- [6] - Bowen, J., & Hinchey, M., "Ten commandments of formal methods". IEEE Computer. Abril, 1995.
- [7] - Richard Sharpe, "Formal Methods Start to Add up Again", Computing, 8 January 2004.
- [8] - Axel van Lamsweerde, "Formal specification: a roadmap", Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland, Pages: 147 – 159, 2000. ACM New York, NY, USA.
- [9] - Ann Sobel and Michael R. Clarkson, "Formal Methods Application: An Empirical Tale of Software Development", IEEE Transactions on Software Engineering, Vol. 28, N° 3, Marzo 2002.
- [10] - Michael Huth and Mark Ryan. "Logic in Computer Science: Modelling and Reasoning About Systems", Cambridge University Press, 2004.

## XI - Resumen de Objetivos

El objetivo de la materia es que los alumnos incursionen en los métodos formales especificando casos de estudio reales, aplicando diferentes estilos de especificación en RSL.

## XII - Resumen del Programa

Introducción a los Métodos Formales. Pros y contras de los Métodos Formales. Estilos de especificaciones. Estilos de desarrollo. Grados de rigor. Introducción al método de RAISE. Especificaciones formales en RSL. Características de RSL. Aspectos Sintácticos de RSL. Constructores de Tipos. Subtipos. Variants. Expresiones Case y Let. Módulos. Esquemas. Objetos. Combinación de Técnicas Formales y semi-formales. Captura de requerimientos. Semántica formal en RSL de diagramas de clases en UML. UML2RSL.

## XIII - Imprevistos

--

## XIV - Otros

--

### ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA

#### Profesor Responsable

Firma:

Aclaración:

Fecha: