



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

(Programa del año 2016)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
METODOS FORMALES	ING. INFORM.	026/1	2- 2016	1° cuatrimestre
		08/15		

II - Equipo Docente

Docente	Función	Cargo	Dedicación
FUNES, ANA MARIA	Prof. Responsable	P.Adj Exc	40 Hs
SANCHEZ, ALEJANDRO	Responsable de Práctico	JTP Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
5 Hs	Hs	Hs	3 Hs	8 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
14/03/2016	24/06/2016	15	120

IV - Fundamentación

Las computadoras son 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.

En las últimas décadas, las computadoras han pasado a jugar 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. Por esta razón es que resulta de vital importancia que los sistemas de computación que están por detrás sean altamente confiables.

Para construir software de alta calidad, libre de errores y confiable, una de las vías más importantes, sin descartar otras alternativas, es definiendo el dominio y los requerimientos a través del uso de especificaciones formales. Una de las principales ventajas de las especificaciones formales es que no son ambiguas y permiten "razonar" sobre las propiedades deseables del sistema especificado, conduciendo al desarrollo de software de calidad.

V - Objetivos / Resultados de Aprendizaje

El objetivo de la materia es que los alumnos tengan la posibilidad de incursionar en la aplicación de los métodos formales como herramienta de producción de software de alta calidad.

VI - Contenidos

Unidad I: Introducción

Introducción al curso y a los Métodos Formales. El papel de los Métodos Formales en Ingeniería de Software. Pros y contras de los Métodos Formales. Un panorama de los Métodos Formales. Diversas clasificaciones de los Métodos Formales. Estilos de especificaciones. Estilos de desarrollo. Grados de rigor. Algunas aplicaciones industriales.

Unidad II: Especificaciones formales

Enfoques de especificación de un sistema. Especificaciones basadas en Lógica de Primer Orden y Teoría de Conjuntos. Especificaciones algebraicas. Especificaciones orientadas a la propiedad. Especificaciones orientadas al modelo. Especificaciones aplicativas, imperativas, concurrentes.

Unidad III: Verificación formal de sistemas de software y de sus propiedades.

Verificación formal del software. Sistema formal. Sintaxis, semántica y sistema de prueba. Relación de refinamiento. Stepwise refinement. Refinamiento de funciones. Refinamiento de datos.

Unidad IV: El Método RAISE

Introducción a RAISE. Características generales del lenguaje 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. Módulos.

Unidad V: Lightweight Formal Methods.

Métodos Formales Livianos. Alloy. Especificaciones formales y verificación en Alloy. La lógica de Alloy: átomos, relaciones, constantes, funciones predefinidas, operadores, restricciones, expresiones, declaraciones, multiplicidades. El lenguaje Alloy: módulos, firmas, hechos, predicados, funciones, aserciones y comandos run y check. Uso del analizador.

Unidad VI: Especificación del comportamiento.

Métodos basados en Álgebra de Procesos. Interacción entre procesos concurrentes. Formalización de eventos concurrentes. Métodos basados en lógica temporal.

VII - Plan de Trabajos Prácticos

1. En aula: Introducción a los Métodos Formales

2. En aula: Especificaciones algebraicas en RSL. Definiciones de axiomas. Chaos. Lógica condicional de RAISE. Expresiones if. 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.

En laboratorio: de los temas del práctico de aula.

3. En aula: Especificaciones orientadas al modelo en RSL: Producto Cartesiano. Funciones. Funciones totales y parciales. Conjuntos finitos e infinitos.

En laboratorio: de los temas del práctico de aula.

4. En aula: Especificaciones orientadas al modelo en RSL: Listas finitas e infinitas. Mapas finitos determinísticos y mapas parciales. Expresiones de subtipos. Subtipos vs. axiomas. Tipos maximales. Subtipos vacíos.

En laboratorio: de los temas del práctico de aula.

5. En aula: Especificaciones orientadas al modelo en RSL: Variants. Constructores de constantes. Constructores de records. Destruccion y reconstruccion. Uniones disjuntas de tipos. Definiciones de short records.

En laboratorio: de los temas del práctico de aula.

6. En aula: Especificación y Verificación en Alloy.

En laboratorio: de los temas del práctico de aula.

7. En aula: Algebra de Procesos

En laboratorio: de los temas del práctico de aula.

8. En laboratorio: Especificación de un caso real.

VIII - Regimen de Aprobación

La materia se desarrolla con la modalidad de promoción sin examen final. Existen dos niveles:

a) Regularización solamente: Para regularizar la materia se deberá:

1. Tener como mínimo un 80% de asistencia a clases prácticas.
2. Tener los prácticos, solicitados por la cátedra, aprobados.
3. Aprobar dos parciales o sus respectivas recuperaciones con una nota mínima de seis (6) sobre diez (10).

b) Promoción sin examen final: Para regularizar y aprobar la materia se deberá:

1. Cumplir con los requisitos a.1 y a.2.
2. Aprobar dos parciales o sus respectivas recuperaciones con una nota mínima de siete (7) sobre diez (10).
3. Aprobar una Evaluación Global Integradora con una nota mínima de siete (7) sobre diez (10).

Aquellos alumnos que sólo regularicen la materia deberán rendir un examen final, en los turnos establecidos.

Se admitirán exámenes de alumnos libres, en cuyo caso el alumno deberá aprobar en primera instancia un examen sobre los temas prácticos de la materia para luego pasar a un examen teórico. La nota final del examen será un promedio de ambas evaluaciones.

IX - Bibliografía Básica

- [1] Bowen, J., & Hinchey, M., "Seven more myths of formal methods". IEEE Software, 12(3). Julio, 1995.
- [2] Bowen, J., & Hinchey, M., "Ten commandments of formal methods". IEEE Computer. Abril, 1995.
- [3] Funes, A., Dasso, A. "Formal Methods Overview" en Encyclopedia of Information Science and Technology, 3ra Edición, IGI Global, 2014.
- [4] George, C.; "Introduction to RAISE". Report 249 UNU/IIST.
- [5] J. F. Groote y col. "The Formal Specification Language mCRL2". En: Methods for Modelling Software Systems: Dagstuhl Seminar 06351. 2007.
- [6] Hung Dang Van, Chris George, Tomasz Janowski, and Richard Moore (Eds). "Specification Case Studies in RAISE". FACIT series. Springer-Verlag. 2002.
- [7] Jackson, Daniel; "Software Abstractions", The MIT Press, 2012.
- [8] R. Milner. Communication and Concurrency. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989. isbn: 0-13-115007-3.
- [9] The RAISE Language Group. "The RAISE Specification Language". Prentice-Hall International, 1992.
- [10] The RAISE Method Group. "The RAISE Development Method". Prentice-Hall International, 1995.

X - Bibliografía Complementaria

- [1] Alagar, V. & Periyasamy, K. "Specification of Software Systems". Springer-Verlag New York, Inc. 1998.
- [2] C. A. R. Hoare. "Communicating Sequential Processes", Prentice Hall, ISBN 0-13-153289-8. Este libro ha sido actualizado por Jim Davies (Oxford University Computing Laboratory) y la nueva edición puede obtenerse en formato PDF en <http://www.usingcsp.com/cspbook.pdf>.
- [3] Funes, A., George, C., "Formalizing UML class diagrams", capítulo 8 de UML and the Unified Process, Idea Group, 2003.
- [4] J.C.M. Baeten. "A Brief History of Process Algebra". Report CS-R 04-02, Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, <http://www.win.tue.nl/fm/pubbaeten.html>.
- [5] Jean-Raymond Abrial. "The B-Book: Assigning Programs to Meanings". Cambridge University Press, 1996. ISBN

0-521-49619-5.

[6] Jonathan Bowen. "Formal Specification and Documentation using Z: A Case Study Approach". International Thomson Computer Press (ITCP) Thomson Publishing. ISBN 1850322309.

[7] Luqi & Goguen, J., "Formal methods: Promises and problems", IEEE Software, 14(1), págs. 73-85., Ene-Feb 1997.

[8] Michael Huth and Mark Ryan. "Logic in Computer Science: Modelling and Reasoning about Systems". Cambridge University Press, 2000.

[9] Steve Schneider. "The B-Method: An Introduction". Palgrave, Cornerstones of Computing series, 2001. ISBN 0-333-79284-X.

XI - Resumen de Objetivos

IncurSIONAR en los métodos formales como herramienta de producción de software de alta calidad.

XII - Resumen del Programa

Introducción a los Métodos Formales. Especificaciones formales. Verificación formal de sistemas de software y de sus propiedades. El Método RAISE. Otros estilos de especificaciones formales. Métodos Formales Livianos. Alloy.

XIII - Imprevistos

-

XIV - Otros