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 2024)
(Programa en trámite de aprobación)
(Presentado el 25/03/2024 11:51:33)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
METODOS FORMALES	ING. INFORM.	026/12-08/15	2024	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
FUNES, ANA MARIA	Prof. Responsable	P.Asoc Exc	40 Hs
SANCHEZ, ALEJANDRO	Prof. Colaborador	P.Adj Exc	40 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	Hs	5 Hs

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

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
11/03/2024	21/03/2024	15	75

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 la verificación formal de 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 métodos formales como herramienta de producción de software de alta calidad, con especial énfasis en el uso de lenguajes de especificación formal en las etapas iniciales del desarrollo de software.

VI - Contenidos

Unidad I: Aspectos generales de los Métodos Formales

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. Diversas clasificaciones de los Métodos Formales. Estilos de especificaciones de software. Características. Estilos de desarrollo formal. Grados de rigor.

Unidad II: Especificaciones formales

Enfoques de especificación de un sistema. Especificaciones basadas en lógica de primer orden y teoría de conjuntos. Especificaciones aplicativas, imperativas, concurrentes. Especificaciones orientadas al modelo. Especificaciones algebraicas. Especificaciones orientadas a la propiedad.

Unidad III: Integración de técnicas formales y semi-formales con OCL.

Limitaciones de las especificaciones semiformales. Limitaciones de los diagramas de clase UML. El lenguaje OCL. Características. Especificación de valores iniciales de atributos. Especificación de reglas de derivación (valores de atributos derivados). Especificación del cuerpo de operaciones de consulta. Especificación de invariantes. Diseño por contrato. Especificación de pre y post condiciones de operaciones.

Unidad IV: Especificación formal del comportamiento con Redes de Petri.

Definición formal de Red de Petri. Modelado del comportamiento con Redes de Petri. Análisis de una Red de Petri. Árbol de alcanzabilidad. Ecuaciones Matriciales. Algunas extensiones.

Unidad V: Métodos Formales livianos.

Características de los Métodos Formales livianos. Alloy. Especificaciones y verificación en Alloy. La lógica de Alloy: átomos, relaciones, constantes, funciones predefinidas, operadores, restricciones, expresiones, declaraciones, multiplicidades. El lenguaje Alloy: módulos, signaturas, hechos, predicados, funciones, aserciones y comandos run y check. Especificaciones estáticas y dinámicas. Uso del analizador.

Unidad VI: Verificación formal de sistemas de software

Aproximaciones a la verificación formal. Model Checking. Verificación deductiva por demostración de teoremas. Sistema formal. Sintaxis, semántica y sistema de prueba. Relación de refinamiento. Refinamiento por pasos sucesivos. Refinamiento de funciones. Refinamiento de datos.

Unidad VII: Especificación Formal en JML

Lenguajes de especificación de la interfaz de comportamiento. El lenguaje JML. Aplicaciones de JML. Aserciones en JML y su uso en tiempo de ejecución. Runtime-assertion-checking (RAC). Diseño por Contrato. Verificación deductiva en JML.

VII - Plan de Trabajos Prácticos

1. Introducción a los Métodos Formales.
2. OCL
3. Modelado del comportamiento con Redes de Petri
4. Análisis de Redes de Petri
5. Especificación y Verificación en Alloy.
6. Java Modeling Language (JML)

VIII - Regimen de Aprobación

Condiciones para REGULARIZAR la asignatura:

- Haber asistido al menos al 80% de las clases teóricas y prácticas de la asignatura.
- Haber aprobado dos parciales (o sus respectivas recuperaciones) con nota mayor o igual a 6 sobre 10.

Condiciones para PROMOCIONAR la asignatura:

- Haber asistido al menos al 80% de las clases teóricas y prácticas de la asignatura.
- Haber aprobado dos parciales (o sus respectivas recuperaciones) con nota mayor o igual a 7 sobre 10.
- Haber aprobado, al final de la cursada, un examen integrador con nota mayor o igual a 7 sobre 10.

- En caso de haber obtenido nota de promoción en todas las instancias de evaluación, el alumno aprobará la materia con una nota que surgirá del promedio de las mismas.

Recuperaciones:
-Se otorgarán dos recuperaciones para cada examen parcial. Se considera como válida la última nota obtenida en cada recuperación.
-El examen integrador no se recupera.

Examen Final:
En caso de regularizar la materia, el alumno deberá rendir un examen final el cual podrá ser oral o escrito, en cualquiera de los turnos de examen establecidos.

Exámenes Libres:
Para aprobar un examen libre, el alumno deberá aprobar, en una primera instancia, una evaluación sobre temas prácticos de la materia con una nota de al menos 6 sobre 10. En caso aprobar la evaluación práctica, el alumno además deberá aprobar una evaluación sobre temas teóricos con una nota de 4 sobre 10. La nota final surgirá del promedio de ambas notas.

IX - Bibliografía Básica

[1] Specification of Software Systems, 2nd edition, Springer. V.S. Alagar K. Periyasamy, (2011).
[2] An Introduction to Formal Methods for the Development of Safety-critical Applications. Technical report. A. Haxthausen (2010).
[3] Formal Methods Overiew, Encyclopedia of Information Science and Technology, 3ra Edición, IGI Global. Ana Funes y Aristides Dasso, (2014).
[4] Software Abstractions, The MIT Press, Jackson, Daniel (2012).
[5] Formal Software Design with Alloy 6 (https://haslab.github.io/formal-software-design/). Julien Brunel, David Chemouil, Alcino Cunha, and Nuno Macedo (2021).
[6] Object Constrain Language version 2.4 (https://www.omg.org/spec/OCL/About-OCL/), OMG (2014)
[7] The Object Constraint Language: Getting Your Models Ready for MDA, Second Edition, Addison-Wesley Professional. Warmer, Jos & Kleppe, Anneke (2003).
[8] Petri Net Theory and the Modeling of Systems, Prentice Hall. Peterson, James Lyle (1981).
[9] Tutorial de JML, https://www.openjml.org/tutorial

X - Bibliografia Complementaria

[1] A Language and Tool for Exploring Software Designs. Commun. ACM 62(9): 66-76. Daniel Jackson (2019).
[2] Specification Case Studies in RAISE. FACIT series. Springer-Verlag. Hung Dang Van, Chris George, Tomasz Janowski, and Richard Moore (Eds) (2002).
[3] The RAISE Specification Language. Prentice-Hall International, The RAISE Language Group (1992).
[4] The RAISE Development Method. Prentice-Hall International, The RAISE Method Group (1995).
[5] System and Software Verification, Model-Checking Techniques and Tools. Springer. B. Bérard, M. Bidot, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci et al. (1999).
[6] Formalizing UML class diagrams, capítulo 8 de UML and the Unified Process, Idea Group, Funes, A., George, C. (2003).
[7] Logic in Computer Science: Modelling and Reasoning about Systems, 2nd Edition. Cambridge University Press, Huth, Michael & Ryan, Mark (2004).
[8] Introduction to CafeOBJ, https://cafeobj.org/intro/en/, Masaki Nakamura (2009).

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. Aproximaciones a la verificación formal de sistemas de software y de sus propiedades. Métodos Formales livianos. Alloy. Model Checking. Verificación deductiva. JML. Otros estilos de

especificaciones formales. Redes de Petri. OCL.

XIII - Imprevistos

XIV - Otros

Contacto: afunes@email.unsl.edu.ar

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	