

EX-2026-00088647- -UNC-ME#FAMAF

PROGRAMA DE ASIGNATURA	
ASIGNATURA: Ingeniería del Software II	AÑO: 2026
CARÁCTER: Obligatoria	UBICACIÓN EN LA CARRERA: 5° año 1° cuatrimestre
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: Cuatrimestral	CARGA HORARIA: 120 horas

FUNDAMENTOS Y OBJETIVOS

El curso introduce metodologías y técnicas avanzadas para la construcción de software confiable y seguro. Los temas tratados a lo largo del curso brindan el conocimiento fundamental y las herramientas para asegurar que el software que será parte de sistemas de alta complejidad, del cual pueden depender vidas humanas o respondan a misiones críticas, brinde un servicio correcto y efectivo.

Objetivos:

Al finalizar la materia los/as estudiantes estarán en condiciones de:

- comprender la problemática de los sistemas críticos (incluyendo sistemas concurrentes y de tiempo real) y los requerimientos fundamentales que estos deben satisfacer;
- elaborar modelos operacionales de estos tipos de modelos en lenguajes formales;
- expresar formalmente los requerimientos de estos sistemas complejos;
- seleccionar y manipular las herramientas y técnicas adecuadas para hacer los distintos tipos de análisis y verificación de modelos y especificaciones;
- comprender los fundamentos matemáticos y algorítmicos detrás de las distintas herramientas de análisis y verificación.

CONTENIDO

1. El problema de la corrección del software

(1) Definición de sistemas críticos, (2) Limitaciones del testing y la simulación, (3) Discusiones sobre verificación.

2. Programación concurrente

(1) Definición de sistemas reactivos, (2) Interacción entre procesos, (3) Los problemas de la concurrencia, (4) Semántica de los programas concurrentes, (5) Interleaving y no determinismo, (6) Razonamiento sobre programas concurrentes, (7) La necesidad de abstraer para modelar, (8) El lenguaje de modelado FSP: sintaxis y semántica, (9) La herramienta LTSA.

3. Sincronización de procesos concurrentes

(1) Recursos compartidos: interferencia y exclusión mutua, (2) Detección de errores, (3) Monitores, sincronización condicional e invariantes del monitor, (4) Semáforos y su invariante, (5) Buffers acotados, (6) Bisimulación como equivalencia de procesos, (7) Comunicación mediante pasaje de mensajes, (8) Pasaje sincrónico de mensajes, (9) Recepción selectiva, (10) Pasaje asincrónico de mensajes, (11) Rendezvous. (12)

Transacciones distribuidas.

4. Propiedades de los sistemas concurrentes

(1) Categorías de propiedades: alcanzabilidad, safety, liveness, y fairness, (2) Necesidad de la categorización de propiedades, (3) Propiedades como conjuntos de trazas, (4) Lenguajes ω -regulares, (5) Formalización de las propiedades de safety y liveness, (6) Otras propiedades, (7) Análisis automatizado de propiedades usando FSP: deadlock, safety y liveness.

5. Lógicas temporales

(1) Limitaciones de los métodos previos y de las lógicas usuales, (2) Lógicas modales, (3) Introducción a las lógicas temporales, (4) La lógica temporal lineal LTL, (5) Sintaxis y semántica, (6) Operadores derivados y leyes, (7) Especificación de propiedades con LTL: Safety y Liveness, (8) Fairness: incondicional, débil y fuerte, (9) Otros tipos propiedades en LTL.

6. Model checking

(1) El modelo de un sistema, (2) Autómatas de Büchi: definición y uso para presentar programas y propiedades, (3) Model Checking de propiedades LTL con enfoque en la teoría de autómatas, (4) Herramientas de model checking, (5) El model checking de propiedades descritas en LTL Spin, (6) Promela: modelado y análisis, (7) El model checker de propiedades descritas en CTL (computational tree logic) SMV, (8) El model checker de propiedades de tiempo Uppaal, (9) Otros model checkers.

7. Especificaciones de sistemas

(1) Características de los lenguajes de especificación, (2) Las lógicas como lenguajes de especificación, (3) Lógica proposicional: Sintaxis, semántica y poder expresivo, (4) SAT solving en la lógica proposicional: ventajas y desventajas, (5) Lógica de primer orden: Sintaxis, semántica y poder expresivo, (6) SAT solving en la lógica de primer orden, (7) El álgebra relacional. Sintaxis, Semántica y Axiomas.

8. El lenguaje de especificación Alloy

(1) Sintaxis del lenguaje Alloy, (2) Características de Alloy, (3) Uso de Alloy para la resolución de problemas con restricciones (constraint solving), (4) Modelos de ejecuciones, (5) Uso de Alloy para verificar refinamientos, (6) Análisis de especificaciones en Alloy: Cotas, cuantificadores no acotados, axiomas de generación.

9. Algoritmos para verificar satisfactibilidad en lógica proposicional

(1) Algoritmos simples: Tablas de verdad y argumentos semánticos, (2) Algoritmos avanzados, (3) Tablas de verdad revisadas, (4) Conversión a forma normal conjuntiva, (5) Regla de resolución clausal, (6) Propagación de restricciones booleanas, (7) El algoritmo de Davis, Putnam, Logemann & Loveland, (8) Cláusulas de Horn, (9) Linealidad de la resolución en la lógica de Horn, (10) La lógica de Horn como base de la programación lógica y los demostradores automáticos de teoremas.

10. Testing

(1) Definición del testing basado en modelos, (2) Testing con modelos formales, (3) El proceso de testing formal, (4) Conformidad corrección y exhaustividad, (5) La teoría de conformidad de testing basada en entradas y salidas (ioco: Input/Output Conformance Testing), (6) Extensión con tiempo y canales de la teoría ioco, (7)

Definición de cubrimiento semántico.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- [1] J. Magee y J. Kramer. Concurrency: State Models & Java Programs, 2nd edition. Wiley 2006.
- [2] C. Baier y J.-P. Katoen. Principles of Model Checking. MIT Press, 2008.
- [3] D. Jackson. Software Abstractions: Logic, Language, and Analysis (Revised Edition). MIT Press, 2011.
- [4] A.R. Bradley y Z. Manna. The Calculus of Computation: Decision Procedures with Applications to Verification. Springer, 2007.

BIBLIOGRAFÍA COMPLEMENTARIA

- [5] A. Biere, M. Heule, H. van Maaren, y T. Walsh (eds.) Handbook of Satisfiability. 2nd ed. IOS Press, 2021.
- [6] G. J. Holzmann. The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley, 2003.
- [7] B. Alpern y F. Schneider. Defining Liveness. Information Processing Letter 21:181-185. 1985
- [8] B. Alpern y F. Schneider. Recognizing Safety and Liveness. Distributed Computing 2 (3): 117-126. 1987.
- [9] B. Berard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schnoebelen. Systems and Software Verification Model-Checking Techniques and Tools. Springer, 2001.
- [10] E.M. Clarke, O. Grumberg, D. Peled. Model Checking. MIT Press, 1999.
- [11] P. Jalote. An Integrated Approach to Software Engineering, Third Edition. Springer. 2005.
- [12] M. Müller-Olm, D. Schmidt, B. Steffen. Model Checking: A Tutorial Introduction. En A. Cortesi, G. Filé (Eds.), Procs. Of SAS'99, LNCS 1694, pp. 330-354. Springer 1999.
- [13] J. Tretmans. A formal Approach to Conformance Testing. PhD Thesis. Univeristeit Twente, The Netherlands, 1992.

METODOLOGÍA DE ENSEÑANZA

La dependencia de las distintas unidades se organiza según la Fig. 1. Las líneas sólidas indican una dependencia necesaria entre las unidades dado que la mayoría de los temas que comprenden la unidad requieren conocimiento de los temas enseñados en las unidades de las cuales dependen. Las líneas de punto, en cambio señalan una dependencia recomendada, en el sentido de que la enseñanza de los temas de dicha unidad se verían facilitados si ciertos conceptos previos ya fueron expuestos.

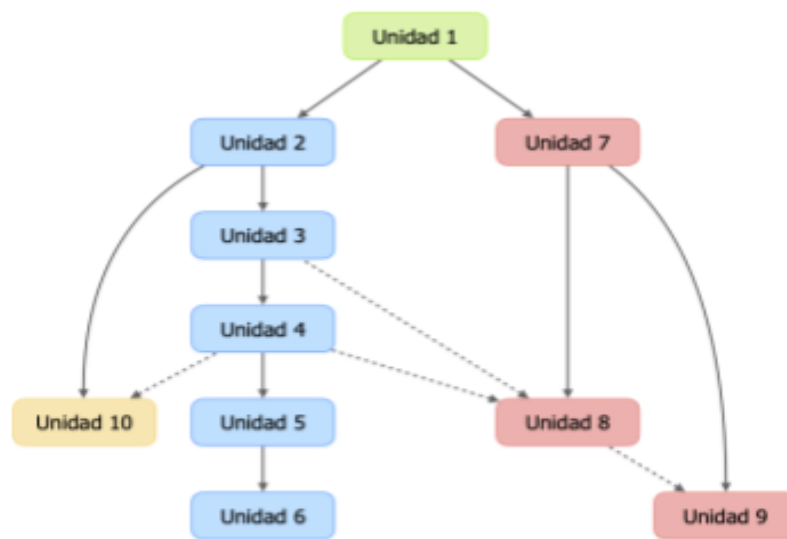


Fig. 1: Dependencia entre las distintas unidades de la materia

Los colores indican grupos temáticos. La primera unidad corresponde a los temas introductorios y es más bien descriptiva de lo que transcurrirá a lo largo del curso. Las Unidades 2 a 6 (en celeste) corresponden a la primera parte de la materia y agrupa lo que corresponde al modelado, análisis y fundamentos comprendiendo a los sistemas de forma operacional. Esta parte agrupa poco más de la mitad del contenido de la materia y es la que se evalúa en el primer parcial. Las Unidades 7 a 9 (en rosado) corresponden al modelado, análisis y fundamentos comprendiendo a los sistemas y sus requerimientos en forma declarativa, es la que se dicta en la segunda parte de la materia y normalmente se evalúa en el segundo parcial. La Unidad 10 (en amarillo) se enfoca en testing basado en modelos operacionales y es la unidad que usualmente queda excluida del dictado normal de la materia por cuestiones de tiempo. Esto no afecta a los contenidos de la carrera ya que esta unidad presenta una visión más específica de conceptos básicos que ya fueron expuestos en la materia de Ingeniería del Software I pero también se trabajaron de manera menos sistemática en casi todas las materias que incluyen un laboratorio.

El dictado de la materia se organiza en teóricos y prácticos, destinando aproximadamente 60 horas al primero y 60 horas al segundo. En cada clase, usualmente, se utilizan las dos primeras horas para el teórico y las dos horas restantes para el práctico. El teórico se dicta mayormente con un programa de presentación multimedia complementado con el uso del pizarrón. Se fomenta la interacción durante las clases buscando que los/as estudiantes puedan ir descubriendo, o al menos intuyendo por sí mismos/as las soluciones a las problemáticas que presentan los distintos temas. Por otro lado, los prácticos tienen como objetivo la consolidación de los temas aprendidos en el teórico a través de la resolución de problemas presentados en guías de ejercicios prácticos. Se espera que los/as estudiantes resuelvan los problemas mayormente de manera autónoma y utilicen el tiempo del práctico para resolver consultas. Estas consultas pueden ser respondidas personalmente o también a través de la elaboración de soluciones de ejercicios paradigmáticos en el pizarrón y con la colaboración de toda la clase.

El trabajo práctico de la materia se realiza en grupo y es un trabajo de investigación integral que comprende el desarrollo de un breve manuscrito asociado y una

presentación oral, que se complementa con las actividades de la materia y orientan a la consolidación de la elaboración y diseminación de trabajos de investigación y desarrollo en el área. Las consultas necesarias para la elaboración del trabajo práctico también se efectúan en el horario del práctico.

Destacamos que la materia hace uso del Aula Virtual, donde se encuentra disponible todo el material necesario para el desarrollo de la materia. Esto incluye: las presentaciones multimedia de cada unidad y las de presentación y explicación del trabajo práctico, las guías de ejercicios prácticos, la bibliografía utilizada incluyendo links en los casos que el material es de público acceso, las herramientas de software utilizadas (tanto LTSA y ALLOY, utilizadas respectivamente en la primera y segunda parte de la materia, como todas las herramientas de software que serán objeto de estudio en el trabajo práctico), las grabaciones de cada clase teórica para poder ser accedida por estudiantes en caso de ausencia a la clase, además de información administrativa, como el programa de la materia, cronograma con fechas importantes del curso (exámenes, entregas, etc) y las condiciones de promoción y regularidad.

Además, el aula virtual provee un foro para anuncios generales de la materia y otro foro para consultas e interacción entre estudiantes (para esto último también hay habilitado un canal de Zulip). Finalmente, el Aula Virtual también provee el marco para seguimientos y entregas del trabajo práctico. Por todo esto, es importante que los/as alumnos/as mantengan activos sus correos electrónicos dado que las comunicaciones de los foros se reenvían a través de email.

En la materia se alienta a que las consultas que corresponden a la resolución de problemas y a la comprensión de los distintos temas se haga de forma pública, sea esto en clases o a través de los foros de consulta. Esta metodología permite enriquecer el aprendizaje del grupo de estudiantes ya sea porque ellos/as mismos/as son quienes realizan las consultas o porque las consultas pueden despertar nuevas dudas que de otra manera no hubieran surgido a término.

EVALUACIÓN

FORMAS DE EVALUACIÓN

La materia consta de dos evaluaciones parciales y la elaboración de un trabajo práctico con múltiples instancias de evaluación.

El trabajo práctico, determinante para la obtención de la regularidad, es un trabajo de investigación integral que comprende el desarrollo de un breve manuscrito asociado y una presentación oral, que se complementa con las actividades de la materia y orientan a la consolidación de la elaboración y diseminación de trabajos de investigación y desarrollo en el área.

Las dos evaluaciones parciales complementan al trabajo práctico en lo que respecta a la promoción de la materia

REGULARIDAD

- Aprobar al menos el 60% de los trabajos prácticos o de laboratorio.

PROMOCIÓN

- Aprobar todas las evaluaciones parciales con una nota no menor a 6 (seis), y obteniendo un promedio no menor a 7 (siete).

- Aprobar todos los trabajos prácticos.