

Planificación de Ingeniería del Software I



Código/s: R-411

Identificación y características del Espacio Curricular

| | | | |
|----------------------|--|---------------------|--|
| Carrera/s: | Licenciatura en Ciencias de la Computación | | |
| Plan de Estudios: | 2010, TO2024 | Carácter: | Obligatoria |
| Bloque/Campo: | | Área: | Ingeniería de Software, Bases de Datos y Sistemas de Información |
| Régimen de cursado: | Cuatrimestral | | |
| Cuatrimestre: | 7º [LCC], 7º [LCC] | | |
| Carga horaria: | 135 hs. / 9 hs. semanales | Formato curricular: | Asignatura |
| Escuela: | Ciencias Exactas y Naturales | Departamento: | Ciencias de la Computación |
| Docente responsable: | CRISTIA, Maximiliano | | |

Programa Sintético

Introducción a la Ingeniería de Software: el proceso de desarrollo de software; ciclos de vida de los sistemas de software. Introducción a la Ingeniería de Requerimientos: el proceso de la ingeniería de requerimientos, productos de la ingeniería de requerimientos. Lenguajes formales para especificación de sistemas de software; introducción a la especificación formal de sistemas concurrentes y de tiempo real. Verificación formal de especificaciones.

Espacios Curriculares Relacionados

| | |
|---------------------------|---|
| Previos Aprobados: | R-223 - Lógica, R-312 - Estructuras de Datos y Algoritmos II, R-314 - Examen de Suficiencia de Inglés |
| Simultaneos Recomendados: | R-412 - Sistemas Operativos II, R-413 - Introducción a la Inteligencia Artificial |
| Posteriores: | R-421 - Ingeniería del Software II, R-511 - Seguridad Informática |

Vigencia desde 2024

Firma Profesor

Fecha

Firma Aprob. Escuela

Fecha

Con el aval del Consejo Asesor:

Fundamentación

Los sistemas de software de dimensión industrial suelen contener un número significativo de errores y deficiencias de distintos tipos. Una de las metodologías aceptadas, por la comunidad académica de la Ingeniería de Software, para mitigar estos problemas es la introducción de métodos formales de desarrollo. También se acepta que no existe un único método formal que pueda cubrir todas las necesidades. Por estos motivos la asignatura se centra en un menú variado de métodos formales con los cuales podrán modelar y especificar sistemas de software. La asignatura incluye la realización de un proyecto de desarrollo usando métodos formales que se completa durante el cursado de Ingeniería de Software II. Con este fin los alumnos seleccionan un requerimiento funcional de baja complejidad, escriben una especificación formal de aquel, traducen la especificación formal a un lenguaje de implementación de alto nivel y realizan un par de actividades de verificación. Por un lado, la asignatura se relaciona con los cursos dedicados a la programación y a la lógica en tanto muestra cómo se puede modelar un programa usando formalismos lógicos, pero sin usar código fuente. Por otro lado, se complementa con Ingeniería de Software II en tanto allí se introducen las nociones de diseño y arquitectura de software que constituyen otra mirada posible sobre un sistema de software. Finalmente, se relaciona con Seguridad Informática dado que allí se usan formalismos lógicos para estudiar propiedades de seguridad de la información.

Resultados del aprendizaje

Al finalizar el cursado los/las estudiantes serán capaces de:

RA1 Escribir especificaciones formales de programas (o software) de complejidad reducida en la notación Z

RA2 Escribir especificaciones formales de programas (o software) de complejidad reducida que incluyan requisitos temporales en Statecharts

RA3 Escribir especificaciones formales de programas (o software) de complejidad reducida que incluyan requisitos temporales en la notación Communicating Sequential Processes (CSP)

RA4 Escribir especificaciones formales de programas (o software) de complejidad reducida que incluyan requisitos temporales en Temporal Logic of Actions (TLA)

RA5 Utilizar un asistente de pruebas interactivo para demostrar propiedades simples de sistemas de software

RA6 Implementar un proyecto de desarrollo de software de baja complejidad basado en el uso de técnicas formales de desarrollo

Competencias / Ejes transversales y Resultados del Aprendizaje

| Competencia/Eje transversal al que tributa | Nivel | Resultados del Aprendizaje |
|---|-------|------------------------------|
| CGT1-Identificación, formulación y resolución de problemas de informática | Bajo | RA1, RA2, RA3, RA4, RA5, RA6 |
| CGT2-Concepción, diseño y desarrollo de proyectos de informática | Medio | RA1, RA2, RA3, RA4, RA5, RA6 |
| CGT3-Gestión, planificación, ejecución y control de proyectos de informática | Medio | RA1, RA2, RA3, RA4, RA5, RA6 |
| CGT4-Utilización de técnicas y herramientas de aplicación en la informática | Bajo | RA1, RA2, RA3, RA4, RA5, RA6 |
| CGS2-Fundamentos para la comunicación efectiva | Bajo | RA6 |
| CGS4-Fundamentos para la evaluación y actuación en relación con el impacto social de su actividad en el contexto global y local | Bajo | RA6 |

Programa Analítico

1. Unidad I: Introducción

1.1. Categorización epistemológica y problemática de la Ingeniería de Software

1.1.1. Definición y alcances.

1.1.2. Diferencias entre las Ciencias de la Computación y la Ingeniería de Software.

1.1.3. Problemas en la construcción de sistemas de software de dimensión industrial.

1.2. Modelos de Ciclo de Vida del Software

1.2.1. El modelo en cascada puro y con prototipos.

1.2.2. El modelo evolutivo.

1.2.3. El modelo de transformaciones formales.

1.3. Principios de la Ingeniería de Software y su aplicación al Análisis de Sistemas

1.4. Cualidades del software y su proceso de producción

1.4.1. Calidad del software; atributos de calidad; atributos de calidad del proceso de producción y del producto.

1.5. Nociones de administración y control de proyectos

2. Unidad II: Sistemas de Información

2.1. Teoría general de sistemas

2.1.1. Orígenes y alcance tradicional

2.1.2. Visión holística

2.1.3. La teoría general de sistemas como interdisciplina

2.1.4. Ingeniería de sistemas

2.2. Clasificación de los sistemas organizacionales y de información administrativa

2.2.1. Definición y el impacto de los modelos económicos

2.2.2. Data warehouses. ERP, sistemas empresariales, sistemas expertos, GIS, ofimática.

2.2.3. Ingeniería de sistemas de información

2.3. Seguridad de la información

2.3.1. Confidencialidad, integridad y disponibilidad

2.3.2. Autenticación, autorización y auditoría

2.3.3. Control y peritaje de sistemas

3. Unidad III: Especificaciones formales

3.1. Métodos Formales y Especificaciones

3.1.1. Métodos formales.

3.1.2. Especificaciones: vocabulario y conceptos.

3.1.2.1. Máquinas, Qué y Cómo, modelo WRSPM, fenómenos.

3.1.2.2. Abstracción, modelo e implementación.

3.1.2.3. Lenguajes formales de especificación.

3.1.2.4. Condiciones de corrección, invariantes, comportamiento observable y propiedades de las componentes de estado.

3.1.2.5. Precondiciones, postcondiciones, especificaciones totales, operaciones atómicas.

3.2. Introducción a la ingeniería de requerimientos

3.3. Especificación de sistemas de información en lenguaje Z

3.3.1. Modelando máquinas de estados mediante matemática y lógica simplemente tipadas.

3.3.2. Esquemas de estado y operación.

3.3.3. Promoción de operaciones.

3.3.4. Composición de operaciones.

3.3.5. Aplicación del modelo WRSPM a especificaciones Z.

3.3.6. Z/EVES: un editor de especificaciones Z.

3.4. Statecharts: modelado de sistemas reactivos simples y con requisitos temporales simples

3.5. Communicating Sequential Processes (CSP) para la especificación de sistemas concurrentes

3.5.1. Eventos y procesos; external choice y alternativas guardadas; definición de procesos recursivos y

mutuamente recursivos; operador de interrupción; renombramiento funcional e indexado; internal choice; leyes CPS para los operadores estudiados.

3.5.2. Leyes para la concurrencia: sincronización, abrazo mortal, eventos independientes primero, intercalaciones.

3.5.3. Especificación de requisitos temporales simples mediante timers.

3.5.4. Procesos y eventos parametrizados; canales de comunicación; operadores imperativos; composición secuencial; procesos STOP y SKIP.

3.5.5. El modelo semántico de fallas y divergencias de CSP; la necesidad de distinguir formalmente external e internal choice; rechazos, fallas, divergencias.

3.6. Seguridad (safety), vitalidad y equidad

3.6.1. Sistemas concurrentes y propiedades; el marco teórico de Alpern-Schneider.

3.6.2. Pasos de ejecución repetitivos y composición de modelos.

3.6.3. El teorema de Alper-Schneider; propiedades de seguridad y vitalidad; formalización de seguridad y vitalidad; cómo reconocer seguridad y vitalidad.

3.6.4. Problemas al intersecar propiedades de seguridad con propiedades de vitalidad arbitrarias; el concepto de máquina cerrada de Abadi-Lamport.

3.6.5. Propiedades de equidad para obtener máquinas cerradas; introducción básica a los operadores de la lógica temporal; equidad débil y equidad fuerte.

3.7. Especificación de sistemas concurrentes complejos utilizando Temporal Logic of Actions Plus (TLA+)

3.7.1. El concepto de estado.

3.7.2. Módulos TLA+; cláusulas EXTENDS, INSTANCE, CONSTANT, ASSUMES y VARIABLE; instanciación múltiple; el concepto de acción; definición de acciones y predicados de estado.

3.7.3. La forma de una especificación TLA; la semántica de una especificación TLA.

3.7.4. Regla de conjunción para weak y strong fairness.

3.7.5. Cuantificación existencial universal.

3.7.6. Especificación de requisitos temporales mediante timers.

4. Unidad IV: Análisis de especificaciones Z

4.1. El satisfiability solver {log} (setlog)

4.1. El asistente de pruebas de Z/EVES

Modalidades de enseñanza

El docente a cargo de la asignatura dicta las clases teóricas mientras que el resto de los docentes se dedican a las clases prácticas. En las clases teóricas se exponen los temas usando un problema disparador para cada tema. Por ejemplo para explicar la notación formal Z se plantea el problema de especificar el funcionamiento de un sistema de cajas de ahorro; para introducir Statecharts se hace lo propio con el sistema de control de un cruce ferroviario. Las clases de teoría son expositivas; el recurso áulico usado es el pizarrón. En las clases prácticas se resuelven problemas de la misma índole que los problemas disparadores vistos en clase. Las clases prácticas consisten en la exposición parcial de la especificación de un sistema de software de manera tal que los alumnos luego la puedan completar; más adelante los alumnos deben intentar resolver por sí solos otros problemas similares para lo cual cuentan con la asistencia de los docentes. Los exámenes, sean parciales como finales, consisten esencialmente en resolver problemas similares a los vistos en teoría y práctica. Fuera de los horarios de clase los alumnos comienzan a desarrollar (en forma parcial) un trabajo práctico/proyecto individual que se describe más abajo.

Recursos

Aula de clase; ocasionalmente proyector multimedia, software específico relacionado con los contenidos. El curso cuenta con un sitio web (www.fceia.unr.edu.ar/is1), además se usa el sitio de Comunidades de la UNR y existe un canal YouTube gestionado por el docente responsable (www.youtube.com/c/maximilianocristiais).

Actividades de Formación Práctica

| Nº | Título | Descripción |
|----|---|---|
| 1 | Práctica 1 | Resolución de problemas sobre especificación formal de software en Z, Statecharts, CSP y TLA+. Cada problema consiste en darle al alumno una descripción informal de cierto sistema de software y solicitarle que escriba una especificación formal en cierto lenguaje de especificación. |
| 2 | Primera parte del Proyecto de Especificación y Verificación (PEV) | El PEV consiste de lo siguiente: los alumnos deben seleccionar un requerimiento simple de un sistema informático; escribir la especificación Z correspondiente la cual debe incluir al menos un invariante de estado; traducir la especificación a {log}; utilizar {log} para correr simulaciones; y utilizar Z/EVES y {log} para demostrar que las operaciones definidas en la especificación preservan el invariante. La descripción del proyecto se encuentra aquí: https://www.fceia.unr.edu.ar/ingsoft/tp.html . Durante la realización del proyecto los alumnos cuentan con la asistencia del responsable del curso. |

Evaluación

Regularización-Promoción. Al finalizar el curso el alumno alcanzará una de las tres condiciones: Libre, Regular o Promovido. Cada una de estas condiciones está en relación con las notas que haya obtenido en las evaluaciones parciales.

Parciales. Se toman dos parciales y un recuperatorio durante el cuatrimestre. El primer parcial cubre hasta el ítem 3.3 del Programa Analítico mientras que el segundo cubre los ítems 3.4 y 3.5. Para obtener la condición de Regular es necesario haber resuelto de manera satisfactoria al menos el 65% de cada parcial. Para obtener la condición de Promovido es necesario haber resuelto de manera satisfactoria al menos el 80% de cada parcial. La condición de Promovido dura hasta que comienza a dictarse nuevamente el curso.

El recuperatorio, que se toma al finalizar el cuatrimestre, puede utilizarse para mejorar la nota de uno de los parciales, tanto sea para obtener la condición de Regular como la de Promovido. En caso de concurrir al recuperatorio el alumno perderá la nota obtenida en el parcial que está recuperando. La condición de Libre se obtiene si no se cumplen los requisitos para obtener las otras dos condiciones.

Finales. El alumno que haya obtenido la condición de Promovido solo deberá rendir, en el examen final, TLA+ más los temas correspondientes al parcial que no haya logrado promover, si lo hubiere. El alumno que haya obtenido la condición de Regular deberá rendir un examen final en el que se incluyen problemas prácticos que cubren la totalidad del curso. El alumno que haya quedado en condición de Libre deberá rendir un examen final en el que se incluyen contenidos de todo el curso.

Todos los exámenes (parciales y finales) consisten en resolver problemas prácticos. La única excepción es el examen final de un alumno libre el cual incluye, además, la evaluación de la teoría.

Trabajo Práctico. El Trabajo Práctico consiste en desarrollar el PEV mencionado más arriba. Como se indicó, el PEV se comienza a realizar durante el cursado de esta asignatura pero se completa durante el cursado de Ingeniería de Software 2.

| Resultado de Aprendizaje | Actividades/Modalidad de Enseñanza | Modalidad de Evaluación |
|--------------------------|---|--|
| RA1 | Clases teóricas y clases prácticas | Resolución de problemas |
| RA2 | Clases teóricas y clases prácticas | Resolución de problemas |
| RA3 | Clases teóricas y clases prácticas | Resolución de problemas |
| RA4 | Clases teóricas y clases prácticas | Resolución de problemas |
| RA5 | Clases teóricas y clases prácticas | Resolución de problemas |
| RA6 | Guía por parte de los docentes para la selección de bibliografía y otros materiales | Presentación del programa del proyecto e informe de avance |

Bibliografía básica

| Autores (Apellido, Inicial nombre) | Año de edición | Título de la obra | Editorial o Revista | Ejemplares disponibles o sitio web |
|------------------------------------|----------------|---|---------------------------------|---|
| Cristiá, M. | 2015 | Introducción a la notación Z | Apunte de clase | https://www.fceia.unr.edu.ar/asist/z-a.pdf |
| Herel, D. | 1987 | Statecharts: a visual formalism for complex systems | Science of Computer Programming | https://doi.org/10.1016/0167-6423(87)90035-9 |
| Cristiá, M. | 2015 | Statecharts Parametrizados | Apunte de clase | https://www.fceia.unr.edu.ar/asist/statecharts-param-a.pdf |
| Cristiá, M. | 2015 | Introducción a CSP | Apunte de clase | https://www.fceia.unr.edu.ar/asist/csp-a.pdf |
| Cristiá, M. | 2015 | Seguridad, Vitalidad y Equidad | Apunte de clase | https://www.fceia.unr.edu.ar/asist/safety.pdf |
| Lamport, L. | 2022 | Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers | Addison-Wesley | 1 |
| Cristiá, M. | 2015 | Un modelo simple para crond en TLA+ | Apunte de clase | https://www.fceia.unr.edu.ar/asist/tla-e.pdf |
| Saaltink, M. | 1999 | The Z/EVES 2.0 user's guide | ORA Canada | https://citeseerx.ist.psu.edu/search_result?query=The%20Z%2FEVES%202.0%20user%E2%80%99s%20guide&pdf=true |
| Rossi, G., Cristiá, M. | 2023 | {log}'s User Manual | Manual de usuario | https://www.clpset.unipr.it/SETLOG/setlog-man.pdf |
| Cristiá, M. | | {log}: Aplicaciones a la Especificación, Prototipado y Verificación de Software | Apunte de clase | https://www.fceia.unr.edu.ar/ingsoft/tp/setlog.pdf |

Bibliografía complementaria

| Autores (Apellido, Inicial nombre) | Año de edición | Título de la obra | Editorial o Revista | Ejemplares disponibles o sitio web |
|---|-----------------------|--|----------------------------|---|
| Roscoe, W. | 1998 | The theory and practice of concurrency | Addison-Wesley | https://www.cs.ox.ac.uk/people/publications/personal/Bill.Roscoe.html |
| Ghezzi, C., Jazayeri, M., Mandrioli, D. | 2003 | Fundamentals of Software Engineering | Prentice Hall | 1 |
| Jacky, J. | 1997 | The way of Z | Cambridge University Press | 1 |
| Diller, A. | 1990 | Z: An Introduction fo Formal Methods | John Wiley Press | 1 |

Distribución de la carga horaria**Presenciales**

| | | |
|--------------|---|----------------|
| Teóricas | | 52 Hs. |
| Prácticas | Formación Experimental | |
| | Resolución de Problemas vinculados a la Profesión | 10 Hs. |
| | Resolución de Problemas y Ejercicios | 40 Hs. |
| | Actividades de Proyecto y Diseño | 25 Hs. |
| | Formación en la Práctica Profesional | |
| Evaluaciones | | 8 Hs. |
| | Total | 135 Hs. |

Dedicadas por el alumno fuera de clase

| | | |
|--|---|---------------|
| | Preparación Teórica | 20 Hs. |
| | Preparación Práctica | 30 Hs. |
| | Elaboración y redacción de informes, trabajos, presentaciones, etc. | 30 Hs. |
| | Total | 80 Hs. |

Cronograma de actividades

| Semana | Unidad | Tema | Actividad |
|---------------|---------------|--|---------------------------------------|
| 1 | 1-3 | Introducción a la Ingeniería de Software – Sistemas de Información – Introducción a Z: introducción al lenguaje, teoría de conjuntos para especificar software | Clases de teoría |
| 2 | 3 | Z: esquemas y lenguaje de esquemas | Clases de teoría y clases de práctica |
| 3 | 3 | Z: promoción y composición de operaciones | Clase de teoría, clase de práctica |
| 4 | 3 | Statecharts: introducción al lenguaje, estados OR y AND | Clase de teoría, clase de práctica |
| 5 | 3 | Statecharts parametrizados | Clase de teoría, clase de práctica |

| | | | |
|----|---|--|--|
| 6 | 3 | CSP: introducción al lenguaje, eventos, procesos, álgebra de procesos, primeros operadores algebraicos | Clase de teoría, clase de práctica |
| 7 | 3 | CSP: leyes de composición paralela | Clase de teoría, clase de práctica |
| 8 | 3 | CSP: requisitos temporales, semántica del lenguaje | Clase de teoría, clase de práctica |
| 9 | 3 | Seguridad (safety), vitalidad y equidad: pasos de ejecución repetitivos, teorema de Alpern-Schneider, formalización de seguridad y vitalidad | Clases de teoría y clases de práctica, primer parcial |
| 10 | 3 | Seguridad (safety), vitalidad y equidad: máquina cerrada, introducción a la lógica temporal, definición de equidad | Clase de teoría, clase de práctica |
| 11 | 3 | TLA+: introducción al lenguaje, módulos, variables de estado, operadores matemáticos y lógicos | Clase de teoría, clase de práctica |
| 12 | 3 | TLA+: requisitos temporales, semántica del lenguaje | Clase de teoría, clase de práctica |
| 13 | 4 | Z/EVES: introducción a la prueba asistida de teoremas, el asistente Z/EVES | Clases de teoría y clases de práctica, segundo parcial |
| 14 | 4 | Z/EVES - Comandos de prueba | Clase de teoría, clase de práctica |
| 15 | 4 | Z/EVES - Comandos de prueba | Clases de teoría y clases de práctica, recuperatorio |