

**FACULTAD DE CIENCIAS EXACTAS, INGENIERIA Y AGRIMENSURA  
U.N.R.**

**PROGRAMA ANALITICO DE LA ASIGNATURA:** Ingeniería de Software I

Código R-411

<p><b>PLAN DE ESTUDIOS:</b> 2010  <b>CARRERA:</b> Licenciatura en Ciencias de la Computación  <b>DEPARTAMENTO:</b> Cs. de la Computación (ECEN)  <b>PROFESOR:</b> Maximiliano Cristiá</p> <p align="center">2013      HASTA AÑO</p> <p align="center"><del>TENTATIVO</del>      <del>DEFINITIVO</del>      DE EXAMEN</p> <p><b>PROGRAMA</b></p> <p align="center"><del>ANUAL</del>      <del>SEMESTRAL</del>      CUATRIMESTRAL</p> <p align="center">Táchese lo que no corresponda.</p> <p><b><u>OBSERVACIONES:</u></b></p>	<p align="center"><b>PRESUPUESTO HORARIO SEMANAL PROMEDIO</b></p> <table border="0"> <tr> <td>TEORÍA:</td> <td align="right">4</td> <td align="right">1</td> </tr> <tr> <td>PRÁCTICA:</td> <td align="right">4</td> <td align="right">2</td> </tr> <tr> <td>LABORATORIO:</td> <td align="right">1</td> <td align="right">3</td> </tr> <tr> <td>TOTAL ASIGNADO:</td> <td align="right">9</td> <td align="right">4 1+2+3</td> </tr> <tr> <td>DEDICACIÓN DEL ALUMNO FUERA DE CLASE:</td> <td align="right">8</td> <td align="right">5</td> </tr> <tr> <td>PRESUPUESTO TOTAL:</td> <td align="right">17</td> <td align="right">6 5+4</td> </tr> <tr> <td>PROGRAMA BASADO EN SEMANAS ÚTILES :</td> <td align="right">15</td> <td align="right">7</td> </tr> <tr> <td>HORAS TOTALES ASIGNADAS:</td> <td align="right">135</td> <td align="right">7x4</td> </tr> <tr> <td>HORAS TOTALES PRESUPUESTAS:</td> <td align="right">255</td> <td align="right">7x6</td> </tr> </table>	TEORÍA:	4	1	PRÁCTICA:	4	2	LABORATORIO:	1	3	TOTAL ASIGNADO:	9	4 1+2+3	DEDICACIÓN DEL ALUMNO FUERA DE CLASE:	8	5	PRESUPUESTO TOTAL:	17	6 5+4	PROGRAMA BASADO EN SEMANAS ÚTILES :	15	7	HORAS TOTALES ASIGNADAS:	135	7x4	HORAS TOTALES PRESUPUESTAS:	255	7x6
TEORÍA:	4	1																										
PRÁCTICA:	4	2																										
LABORATORIO:	1	3																										
TOTAL ASIGNADO:	9	4 1+2+3																										
DEDICACIÓN DEL ALUMNO FUERA DE CLASE:	8	5																										
PRESUPUESTO TOTAL:	17	6 5+4																										
PROGRAMA BASADO EN SEMANAS ÚTILES :	15	7																										
HORAS TOTALES ASIGNADAS:	135	7x4																										
HORAS TOTALES PRESUPUESTAS:	255	7x6																										

**OBJETIVOS: (qué debe saber el alumno al concluir el curso)**

- (1) Debe comprender que la construcción de grandes sistemas de software es esencialmente diferente a la programación de aplicaciones pequeñas o caseras.
- (2) Debe comprender los conceptos fundamentales sobre construcción de modelos funcionales de sistemas de software.
- (3) Debe manejar con fluidez varios lenguajes de especificación formal.
- (4) Debe conocer los rudimentos para la prueba formal de propiedades de modelos.

**UBICACIÓN EN LA CARRERA Y CARACTERISTICAS GENERALES:**

Materia obligatoria del 4º año de la carrera.  
 Está orientada a que el alumno se introduzca en la construcción de modelos y especificaciones funcionales formales de sistemas de software de tamaño industrial.

**MATERIAS RELACIONADAS:**

**Previas:** R-223 Lógica, R-312 Estructuras de Datos y Algoritmos II, R-314 Examen de Suficiencia de Inglés.  
**Simultáneas recomendadas:** - - -  
**Posteriores:** R-422 Ingeniería de Software II, R-511 Seguridad Informática.

.....

Firma Profesor	Fecha	Aprob. Escuela	Fecha
----------------	-------	----------------	-------

**Aprobado en reunión de Consejo Académico de fecha:** .....

## CONTENIDO TEMATICO

### Ordenar temas utilizando codificación decimal

#### 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**
- 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: Especificaciones formales

- 2.1. **Métodos Formales y Especificaciones**
  - 2.1.1. Métodos formales.
  - 2.1.2. Especificaciones: vocabulario y conceptos.
    - 2.1.2.1. Máquinas, Qué y Cómo, modelo WRSPM, fenómenos.
    - 2.1.2.2. Abstracción, modelo e implementación.
    - 2.1.2.3. Lenguajes formales de especificación.
    - 2.1.2.4. Condiciones de corrección, invariantes, comportamiento observable y propiedades de las componentes de estado.
    - 2.1.2.5. Precondiciones, postcondiciones, especificaciones totales, operaciones atómicas.
- 2.2. **Introducción a la ingeniería de requerimientos**
- 2.3. **Especificación de sistemas de información en lenguaje Z**
  - 2.3.1. Modelando máquinas de estados mediante matemática y lógica simplemente tipadas.
  - 2.3.2. Esquemas de estado y operación.
  - 2.3.3. Promoción de operaciones.
  - 2.3.4. Composición de operaciones.
  - 2.3.5. Aplicación del modelo WRSPM a especificaciones Z.
  - 2.3.6. Z/EVES: un editor de especificaciones Z.
- 2.4. **Statecharts: modelado de sistemas reactivos simples y con requisitos temporales simples**

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

- 2.5.1. 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.
- 2.5.2. Leyes para la concurrencia: sincronización, abrazo mortal, eventos independientes primero, intercalaciones.
- 2.5.3. Especificación de requisitos temporales simples mediante timers.
- 2.5.4. Procesos y eventos parametrizados; canales de comunicación; operadores imperativos; composición secuencial; procesos *STOP* y *SKIP*.
- 2.5.5. El modelo semántico de fallas y divergencias de CSP; la necesidad de distinguir formalmente *external* e *internal choice*; rechazos, fallas, divergencias.

**2.6. Safety, Liveness y Fairness**

- 2.6.1. Sistemas concurrentes y propiedades; el marco Alpern-Schneider.
- 2.6.2. Pasos intrascendentes y composición de modelos.
- 2.6.3. El teorema de Alper-Schneider; propiedades de *safety* y *liveness*; formalización de *safety* y *liveness*; cómo reconocer *safety* y *liveness*.
- 2.6.4. Problemas al interesechar propiedades de *safety* con propiedades de *liveness* arbitrarias; el concepto de máquina cerrada de Abadi-Lamport.
- 2.6.5. Propiedades de *fairness* para obtener máquinas cerradas; introducción básica a los operadores de la lógica temporal; *weak* y *strong fairness*.

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

- 2.7.1. El concepto de estado.
- 2.7.2. Módulos TLA+; cláusulas EXTENDS, INSTANCE, CONSTANT, ASSUMES y VARIABLE; instanciación múltiple; el concepto de acción; definición de acciones y predicados de estado.
- 2.7.3. La forma de una especificación TLA; la semántica de una especificación TLA.
- 2.7.4. Regla de conjunción para *weak* y *strong fairness*.
- 2.7.5. Cuantificación existencial universal.
- 2.7.6. Especificación de requisitos temporales mediante timers.

**3. Unidad III: Análisis de especificaciones Z**

**3.1. El asistente de pruebas de Z/EVES**

**3.2. Teoremas en Z/EVES**

**3.3. El teorema de inicialización**

**3.4. Cálculo de precondiciones**

**3.5. Errores de dominio**

**3.6. Inconsistencias**

**3.7. Propiedades del modelo**

## BIBLIOGRAFIA

a) Adecuada al programa. Ordenada por temas y con su codificación de biblioteca, incluidas las publicaciones de la Cátedra con su código de publicación.

### 4. Unidad I

- 4.1. Ghezzi, C., Jazayeri, M. y Mandrioli, D., *Fundamentals of Software Engineering*, Prentice Hall, Upper Saddle River, 1991.
- 4.2. Gibbs, W.W., "La crisis crónica de la programación", *Investigación y Ciencia*, 72-81, noviembre 1994.
- 4.3. Brooks, F.P., "No silver bullet: essence and accidents of software engineering", *IEEE Computer*, 20(4): 10-19, abril 1987.
- 4.4. Harel, D., "Biting the silver bullet: toward a brighter future for system development", *IEEE Computer*, 25(1): 8-20, enero 1992.

### 5. Unidad II

#### 5.1. Introducción a la ingeniería de requerimientos

- 5.1.1. Sommerville, I., *Software Engineering*, Addison-Wesley, Harlow, 1995.

#### 5.2. Métodos Formales y Especificaciones

- 5.2.1. Sommerville, I., *Software Engineering*, Addison-Wesley, Harlow, 1995.
- 5.2.2. Bowen, J., Hinchey, M., "Seven more myths of formal methods", *IEEE Software*, 12(4):34-40, julio 1995.
- 5.2.3. Clarke, E., Wing, J., "Formal methods: state of the art and future directions", *ACM Computing Surveys*, 18(4):626-643, diciembre 1996.
- 5.2.4. Jackson, M.; *Software requirements & specifications: a lexicon of practice, principles and prejudices*, Addison-Wesley, 1995
- 5.2.5. Wing, J.; "Hints to specifiers"; CMU-S-95-118r; Carnegie Mellon University, mayo 1995.

#### 5.3. Z

- 5.3.1. Maximiliano Cristiá; *Especificación Z de parte de los sistemas de un banco; apunte de clase; www.fceia.unr.edu.ar/asist.*
- 5.3.2. Antoni Diller; *Z: An Introduction to Formal Methods*; John Wiley Press; 1990.
- 5.3.3. Ben Potter and Jane Sinclair and David Till; *An Introduction to Formal Specification and Z*; Prentice Hall International; 1996.
- 5.3.4. Jonathan Jacky; *The Way of Z*; Cambridge University Press; 1997.
- 5.3.5. Jim Woodcock and Jim Davies; *Using Z: specification, refinement, and proof*; 1996; ISBN 0-13-948472-8; Prentice-Hall, Inc.; Upper Saddle River, NJ, USA.

#### 5.4. Statecharts

- 5.4.1. Maximiliano Cristiá; *Statecharts parametrizados; apunte de clase; www.fceia.unr.edu.ar/asist.*
- 5.4.2. David Harel; *Statecharts: A visual formalism for complex systems*; *Science of Computer Programming*; 1987; 8, pp. 231-274.

#### 5.5. CSP

- 5.5.1. Hinchey, M., Jarvis, S., *Concurrent systems: formal development in CSP*, McGraw Hill, 1995.

#### 5.6. Elementos de concurrencia

- 5.6.1. Alpern, B., Schneider, F., "Defining liveness", *Information Processing Letters*, 21(4): 181-185, octubre 1985.
- 5.6.2. Leslie Lamport; *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*; Addison-Wesley Professional; 2002.

#### 5.7. TLA

- 5.7.1. Leslie Lamport; *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*; Addison-Wesley Professional; 2002.

- 5.7.2. Lamport, L., "Specifying concurrent systems with TLA+", en *Calculational System Design*, editores M. Broy y R. Steinbrüggen, IOS Press, Amsterdam, 1999.
- 5.7.3. Lamport, L., "The temporal logic of actions", *ACM Transactions on Programming Languages and Systems*, 16(3): 872-923, mayo 1994.

## 6. Unidad III

- 6.1. Ghezzi, C., Jazayeri, M. y Mandrioli, D., *Fundamentals of Software Engineering*, Prentice Hall, Upper Saddle River, 1991.
  - 6.1.1. Saaltink, M., *The Z/EVES 2.0 user's guide*, ORA Canada, octubre 1999.
  - 6.1.2. Potter, B., Sinclair, J., Till, D., *An introduction to formal specification and Z*, Prentice Hall International, Hertfordshire, 1996.

## b) Complementaria para profundización o extensión de temas.

### 7. Unidad I

- 7.1.1. Maibaum, T.S.E., "What we teach software engineering in the university: do we take engineering seriously?", *Software Engineering Notes*, 22(6): 40-50, noviembre 1997.
- 7.1.2. Parnas, D.L., "Software engineering: an unconsumated marriage", *Software Engineering Notes*, 22(6): 1-3, noviembre 1997.

### 8. Unidad II

- 8.1.1. Jackson, M.; *Software requirements & specifications: a lexicon of practice, principles and prejudices*, Addison-Wesley, 1995.
- 8.1.2. Wing, J.; "A specifier's introduction to formal methods"; *IEEE Computer*, 23(9):8-24, septiembre 1990.
- 8.1.3. Abadi, M., Alpern, B., Apt, K., Francez, N., Katz, S., Lamport, L., Schneider, F., "Preserving liveness: comments on 'Safety and liveness from a methodological point of view'", *Information Processing Letters* 40(3): 141-142, noviembre 1991.
- 8.1.4. Abadi, M., Lamport, L., "An old-fashioned recipe for real time", *ACM Transactions on Programming Languages and Systems*, 16(4): 1543-1571, septiembre 1994.
- 8.1.5. Abadi, M., Merz, S., "On TLA as a logic", en *Deductive Program Design* (Manfred Broy, ed.), NATO ASI Series, Springer-Verlag (1995), 235-271. *Proceedings of the NATO Advanced Study Institute on Deductive Program Design*, held in Marktoberdorf, Germany (1994).
- 8.1.6. Harel, D., Naamad, A., "The STATEMATE semantics of Statecharts", *ACM Transactions on Software Engineering and Methodology*, 5(4), octubre 1996.
- 8.1.7. Roscoe, A., *The theory and practice of concurrency*, Prentice Hall International, 1998. Páginas 1-145.