

C 1998
④

UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES

1. DEPARTAMENTO: Computación
2. CUATRIMESTRE: Primero de 1998.
3. ASIGNATURA: MODELADO DE SISTEMAS DE SOFTWARE
4. CARRERA: Licenciatura en Ciencias de la Computación
5. CARACTER DE LA MATERIA: Optativa
6. NUMERO DE CODIGO DE CARRERA: 18
7. NUMERO DE CODIGO DE MATERIA:
8. PUNTAJE: 3 (87 y 93)
9. PLAN DE ESTUDIOS AÑO: 1987 y 1993.
10. DURACION DE LA MATERIA: Cuatrimestral
11. HORAS DE CLASE SEMANAL:
a) TEORICAS 3 HS. c) PROBLEMAS 3 hs.
b) LABORATORIO HS. d) SEMINARIOS
12. CARGA HORARIA TOTAL: 6 HORAS
13. ASIGNATURAS CORRELATIVAS: Ingeniería de Software Y, lógica y Computabilidad (Plan 93). especificación y Diseño de Software, laboratorio VI y Lógica (Plan 87)
14. FORMA DE EVALUACION: Examen Final
15. PROGRAMA Y BIBLIOGRAFIA: Adjuntas a esta hoja

FECHA: 1/3/98



Firma del Profesor
Dr. daniel YANKELEVICH



Lic. Irene LOISEAU
DEPTO. DE COMPUTACION
F.C.E.VII UBA

Firma del Director
9 Lic. Irene LOISEAU

Propuesta de Materia Optativa, Segundo Cuatrimestre 1997

Título:

Modelado de Sistemas de Software

Profesor:

Dr. Daniel Yankelevich

Asistente:

Lic. Angel Perez Pulletti

Carrera:

Departamento:

Carácter:

Licenciatura en Ciencias de la Computación
Departamento de Computación
Optativa

Puntaje:

[REDACTED]

Asig. Correlativas:
APROBADO.

Ingeniería del Software I, Lógica; CON FINAL

Duración:

Cuatrimestral

Modalidad:

Teórico-Práctica

Carga Horaria Semanal:

preparar (papers y material de estudio).

5 horas de clase (4 teóricas + 1 práctica), más temas a

Total de Horas:

90 horas

Objetivos:

Introducir a los alumnos al uso de modelos de sistemas de software. Presentar técnicas modernas de modelado de sistemas. Que los alumnos apliquen dichas técnicas mediante un trabajo práctico, para incorporarlas en su forma de enfrentar el desarrollo de sistemas.

Lic. IRENE LOISEAU
DIRECTORA
DEPTO. DE COMPUTACION
F. C. E. y N. UBA

Descripción:

La materia presenta el uso de modelos de software en la construcción de sistemas. Parte de la base de que un adecuado nivel de abstracción es necesario para poder describir los sistemas en forma adecuada, y que las técnicas utilizadas para modelar sistemas influyen en forma decisiva en el desarrollo y la calidad de los mismos.

La materia se divide en tres partes, que no necesariamente deben dictarse en forma secuencial. La primera parte introduce modelos formales de software, con fuertes fundamentos matemáticos. La segunda parte introduce el UML, un lenguaje unificado basado en las técnicas orientadas a objetos para el modelado de sistemas. La tercera parte se apoya en la participación de los estudiantes, y propone una tema de trabajo que constituye prácticamente un tema de investigación. Esta parte es un trabajo práctico, que permite que los alumnos utilicen las técnicas estudiadas, a la vez que permite que los alumnos trabajen en un proyecto original en el que se pueden obtener, potencialmente, importantes resultados.

Finalmente, y como anexo, se proponen temas adicionales, que se podrán dictar, como extensión a los temas dictados si el tiempo lo permitiera y fuera considerado conveniente. Estos temas no son estrictamente necesarios, pero pueden complementar el programa. Dado que es la primera vez que esta materia se dicta, creemos conveniente contar con un temario recargado y ajustarlo en función del background de los alumnos.

Creemos que los lenguajes presentados tienen dos ventajas de importancia: constituyen técnicas de interés para el futuro profesional y cuentan con un importante futuro, a la vez que forman parte de los fundamentos necesarios para comprender parte del espectro de técnicas usadas en ciencias de la computación.

Contenidos:

Uso de las técnicas de modelado en el desarrollo de sistemas. Técnicas formales de modelado de sistemas. Técnicas de modelado orientadas a objetos y lenguaje UML.

Metodología

La materia contará con clases teóricas. Desde el punto de vista de forma de la dictada, la materia se divide en dos partes. En la primera parte se dictarán, en forma intensiva, los temas teóricos. La dictada se reforzará con artículos y temas de estudio. La segunda parte consistirá en el desarrollo del trabajo práctico, con clases de consulta y clases teóricas de temas especiales en forma esporádica. Los contenidos teóricos de la materia, por lo tanto, se concentran en la primera mitad del cuatrimestre, mientras que la segunda parte tiene como objetivo fundamental el desarrollo del trabajo práctico.

Durante toda la dictada se entregarán artículos de posición y técnicos para su lectura, que serán de estudio obligatorio, y necesarios para participar en las clases.

Desde la parte de dictada intensiva, se reforzarán los conceptos presentados con ejercicios prácticos.

Se estima que la primera parte (dictada intensiva) requerirá 10 semanas, dictándose una clase semanal para temas formales y una clase para UML.

Lic. IRENE LOISEAU
DIRECTORA
DEPTO. DE COMPUTACION
F. C. E. y N. UBA

Evaluación

La evaluación se basará en el proyecto y un coloquio final. Es posible que, durante la primera mitad del cuatrimestre, se incorpore un parcial, consistente en un par de ejercicios (uno de UML y uno de métodos formales) a ser aprobados para continuar la cursada.

Cupo

Debido al carácter intensivo planteado para el seguimiento de los alumnos no se podrán inscribir más de 18 estudiantes.

Programa:

1. Modelos. Importancia de los modelos, para qué sirven, por qué usarlos. Uso de modelos en desarrollo de sistemas. Abstracción. Propiedades sobre abstracciones.
2. Qué abstraer. Máquinas abstractas; SOS. Modelos operacionales. Comportamiento y equivalencias de comportamiento.
3. Uso de lógica en construcción de modelos.
4. Abstracción en datos. Álgebras universales como modelos. Signatura y álgebra inicial.
5. De álgebras a objetos.
6. Modelado con objetos. Metodología de análisis orientada a objetos.
7. Introducción al UML, historia, herramientas. Uso de UML para modelar.
8. UML como lenguaje gráfico. UML y sus diagramas.
9. Semántica de UML en UML.
10. Anexo: temas adicionales. Modelos y lenguajes de programación. Elección de un lenguaje de programación.
11. Trabajo práctico: formalización de UML (UML en lógica/UML y sus álgebras).

Lic. IRENE LOISEAU
DIRECTORA
DEPTO. DE COMPUTACION
F. C. E. y N. UBA

Bibliografía

Nota: dado que toda la bibliografía no está disponible en biblioteca, se seleccionará un subconjunto o se reemplazarán algunos trabajos.

Bidoit, M.; Kreowski, H.; Lescanne, P.; Orejas, F.; Sannella, D. (editores); Algebraic System Specification and Development. A Survey and Annotated Bibliography. Volumen 501 de Lecture Notes in Computer Science; Springer-Verlag, 1991.

Collins; Nicholls; Sorncsen; Introducing formal methods: the CICS experience with Z; Technical Report TR12.260, IBM, IBM United Kingdom Laboratories; 1987.

Ehrig, H.; Mahr, B; Fundamentals of algebraic specifications I; Volumen 6 de EATCS Monographs on Theoretical Computer Science; Springer-Verlag, 1985.

Gutttag; Horning; The algebraic specification of abstract data types; Acta Informatica; (10):27--52, 1978.

Halmos, P. R.; Naive set theory; Litton Ed. Publishers, 1960.

Liskov; Gutttag; Abstraction and specification in program development; MIT Press; 1986.

Mendelson, E.; Introduction to mathematical logic; Van Nostrand, 1979.

Milner, Robin; Communication and Concurrency; Prentice-Hall International, Englewood Cliffs, 1989.

Plotkin, G. D.; A structural approach to operational semantics; Reporte DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

Goguen, J.A.; Thatcher, J.W.; Wagner, E.G.; An Initial Algebra Approach to the Specification, Correctness, and Implementation of Abstract Data Types.

Hoare, C.A.R.; Jones, C.B.; Essays in Computing Science; Prentice Hall; International Series in Computer Science; Cambridge; 1989.

Potter, Ben; Sinclair, Jane; Till, David; An introduction to formal specification and Z; Prentice Hall, International Series in Computer Science, 1991.

Booch, Grady; Rumbaugh, James; Jacobson, Ivar; Unified Modeling Language Document Set; Rational Software Corporation; 1997.

Jacobson, Ivar et. Al.; Object-Oriented Software Engineering; Addison-Wesley & ACM; 1992.

Rumbaugh, James et. Al.; Object-Oriented Modeling and Design; Prentice Hall; 1991.

Booch, Grady; Benjamin/Cummings; 1993.

Wirfs-Brock, Rebeca, etc. Al.; Designing Object-Oriented Software; Prentice Hall; 1990.