

Ciencias de la Computación

Temario de curso

Adscripción	
Programa de posgrado	Ciencias de la Computación
Orientación	
Fecha de registro en el DSE	Haga clic aquí para escribir una fecha.

Información del curso		
Nombre del curso		
Análisis y Diseño Orientado a Objetos		
Periodo lectivo	Tipo	
Cuatrimestre III (agosto-diciembre)	Optativo	
Cursos previos		
Materias que se requiere haber tomado antes (una por renglón) o escriba Ninguna.		
Créditos	Horas de teoría	Horas de laboratorio
5	40	Escriba un número.
Elaborado por		
Ana Isabel Martínez García		
Aprobado en reunión de Consejo de Programa de Posgrado (CPP)		
Haga clic aquí para escribir una fecha.		

[Click here to enter a date.](#)

Objetivos generales

Al finalizar el curso el alumno manejará los elementos relevantes para el análisis y diseño de sistemas con una filosofía orientada a objetos y será capaz de involucrarse en el desarrollo de sistemas complejos de cómputo utilizando este enfoque.

Contenido temático

1. Introducción a la ingeniería de software

Industria de software
 Ingeniería de software es diferente a la ingeniería del hardware
 Proceso de desarrollo
 Ciclos de vida
 Cascada vs. Agil
 Assurance (Garatía y seguridad)
 Productividad

2. Requerimientos

Definición: Requerimientos funcionales y no funcionales
 Problemas asociados a la ausencia de requerimientos
 Proceso para la obtención de requerimientos
 Estandar IEEE
 Ejemplos

Ciencias de la Computación

3. Conceptos básicos de orientación a objetos

Antecedentes de POO
Cohesión y acoplamiento
Clases y Objetos
Encapsulación, Herencia, Polimorfismo
Ejemplos y ejercicios

4. Modelado de sistemas orientado a objetos

Modelado visual
Evolución del lenguaje UML
UML 2.0
Herramientas CASE
Argo UML y estereotipos

5. Diagramas de casos de uso

Definición
Notación
Actores y Relaciones
Flujos alternativos
Descripción general y extendida
Ejemplos
Ejercicios de casos de uso

6. Diagramas de objetos, clases y relaciones

Definición
Notación
Actores y Relaciones
Flujos alternativos
Descripción general y extendida
Ejemplos
Ejercicios de objetos, clases y relaciones

7. Diagramas de secuencia y colaboración

Notación de diagramas de secuencia
Diagramas de secuencia en uso
Ejemplos
Ejercicios de diagramas de secuencia

8. Diagramas de actividad y de estado

Notación de diagramas de estados
Transiciones y estados compuestos
Ejemplos
Notación de diagramas de actividad
Flujos de actividades y particiones
Ejemplos
Ejercicios de diagramas de actividad y de estado
Ejercicios de modelado completos

Ciencias de la Computación

9. Arquitectura de software

Introducción a arquitecturas de software
Descomposición del sistema en subsistemas
Decisiones de diseño arquitectónico
Paquetes
Componentes
Emplazamiento
Ejercicios de arquitecturas

10. Metodologías Ágiles

Introducción a las metodologías ágiles
Manifiesto
Características
Métodos Ágiles (Scrum, AUP, DSDM, XP)
Técnicas para el desarrollo de software ágil (BDD, TDD)
Agilidad

11. Desarrollo basado en comportamiento (Behaviour driven development)

Definición
Casos de uso “inteligentes”
Ejemplos y ejercicios
Scenarios
Cucumber

12. Prototipado de baja fidelidad

Storyboards
Sketches
Ejemplos y ejercicios

13. Test driven development

Repaso de pruebas
El ciclo de TDD: Rojo-Verde-Refactor
Cobertura, Pruebas de unidad e integración
Perspectivas y otros conceptos de pruebas

Crterios y mecanismos de evaluación

Exámenes parciales 40%
Lecturas 10%
Exposiciones 10%
Proyecto 40%

Comentarios

Haga clic aquí para escribir texto.

Ciencias de la Computación

Referencias bibliográficas

1. Alistair Cockburn, 2002, *Use cases, ten years later*, STQE magazine Armstrong Deborah J.; 2006. The quarks of object-oriented development. *Commun. ACM* 49, 2 (February 2006), 123-128
2. Bell Alex E., 2004. Death by UML Fever. *Queue* 2, 1 (March 2004), 72-80.
3. Bennet, S., Skelton, J., Lunn, K.; UML Schaum's Outlines. McGraw Hill, 2001.
4. Bersoff Edward H., Elements of Software Configuration Management. *IEEE Trans. Software Eng.* 10, 1 (January 1984), 79-8
5. Booch G., Jacobson I. and Rumbaugh J., The Unified Modeling Language User Guide. Addison-Wesley, 1998.
6. Cusumano M.A. and Selby R.W., How Microsoft Builds Software, *Communications of the ACM*, pp. 53-61, June 1997.
7. Fowler M. and Scott K., UML Distilled (UML Gota a Gota), Addison Wesley, 1997
8. Garlan David and Shaw Mary. 1994. *An Introduction to Software Architecture*. Technical Report. Carnegie Mellon Univ., Pittsburgh, PA, USA.
9. Greenberg Saul, Carpendale Sheelagh, Marquardt Nicolai, Buxton Bill. 2012. Sketching user experiences, Chapter 3 (3.1-3.4), pages 69-91
10. Jacobson I., Booch G., Rumbaugh J.; Unified Software Development Process, Addison Wesley, 1999.
11. Janzen, D.; Saiedian, H.; , "Test-driven development concepts, taxonomy, and future direction," *Computer* , vol.38, no.9, pp. 43- 50
12. North Dan (2006). Introducing Behavior Driven Development, Better Software, <http://dannorth.net/introducing-bdd/>
13. Lawrence, Brian, Karl Wiegers, and Christof Ebert. "The Top Risks of Requirements Engineering." *IEEE Software* (November/December 2001): 62-63.
14. Parnas D.L., Why Software Jewels are Rare, *Computer*, 29(2), 1996.
15. Patterson David, Engineering Long-Lasting Software: An Agile Approach Using SaaS and Cloud Computing, Beta Edition, Armando Fox, 2011
16. Pooley R. and Stevens P., Using UML, Software Engineering with Objects and Components, Addison Wesley, 1999.
17. Rosenberg Doug and Kendall Scott, Driving Design with Use Cases, December 2000, www.sdmagazine.com.
18. Rosenberg Doug and Kendall Scott, Top Ten Use Case Mistakes, Doug Rosenberg and Kendall Scott, Februa
19. Rosenberg Doug and Kendall Scott, Sequence Diagrams: One Step at a time. Software Development Online 2001, www.sdmagazine.com.
20. Rosenberg Doug and Kendall Scott, Driving Design: The Problem Domain, January 2001. www.sdmagazine.com.
21. Rosenberg Doug and Kendall Scott, Give Them What They Want June 2001. www.sdmagazine.com.
22. Sridhar Nerur, RadhaKanta Mahapatra, and George Mangalaraj. 2005. Challenges of migrating to agile methodologies. *Commun. ACM* 48, 5 (May 2005), 72-78
23. Rumbaugh J., Booch G., Jacobson I., Unified Modeling Language Reference Manual. Addison Wesley, 1998.