

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información

Ingeniería de Software II (CI-4712)
Nota Informativa Septiembre-Diciembre 2007

Profesores

Teoría:	Marlene Goncalves	mgoncalves@ldc.usb.ve	MYS 202A
Práctica:	Pedro García	pgarcia@ldc.usb.ve	MYS

Objetivo General

Introducir al estudiante en los procesos característicos de la ingeniería de software orientado a objetos. El curso enfatizará:

1. Desarrollo incremental e iterativo;
2. Manejo de requerimientos, el modelado, diseño y programación de software orientado a objetos de mediana envergadura;
3. El trabajo disciplinado en equipos de 4-12 personas (*teamwork*);

Créditos: 4 créditos (6 horas semanales)

Programa Sinóptico

Proceso de desarrollo y mantenimiento de software en el marco de desarrollo iterativo e incremental. Arquitecturas de software orientadas a objetos. Patrones. Criterios y evaluación del diseño orientado a objetos. Elementos de Gerencia de Proyectos: planificación, control, riesgos, calidad. Administración de Requerimientos. Manejo de configuraciones. Trabajo en equipo.

Programa Detallado

Revisión de elementos de la Ingeniería del software orientada a objetos. El proceso de mantenimiento de software y el desarrollo iterativo incremental. Evaluación de un proyecto de software con miras a su mantenimiento.

Comparación de metodologías de desarrollo de Software.

Importancia de la administración de requerimientos. El ciclo de vida de una solicitud. Uso de atributos asociados a requerimientos. Herramientas de soporte.

Arquitectura y diseño orientado a objetos: Arquitectura de capas. Arquitecturas de 2,3 y 4 niveles (*tiers*) para Sistemas de Información. Patrones: MVC (*model-view-controller*), publicar-suscribir (*publish-subscribe*), estrategia, estado, comando, peso mosca, fachada, decorador, mediador, constructor, fábrica abstracta, unitario (*singleton*). Evaluación de diseños según criterios como facilidad de uso, eficiencia, seguridad, complejidad, flexibilidad, auditabilidad y esfuerzo.

Control de configuraciones: Arquitectura de una configuración, versión, variante, promoción, liberación (*release*) y rama. Manejo de cambios. Versión base (*baseline*), deltas positivos y negativos, operaciones de solicitud, reserva, bloqueo, préstamo y actualización de elementos de una configuración. Plan de gestión de configuración.

Ingeniería de software orientada a componentes. Programación Java y uso de componentes.

El rol de métodos formales en la Ingeniería de Software.

Trabajo en equipos y control de proyectos: Introducción a los procesos grupales efectivos. Fases de la etapa de formación de grupos. Funciones de cumplimiento de tareas y funciones de mantenimiento del grupo. Liderazgo. Roles. Manejo efectivo y eficiente de una sesión de trabajo (agenda, minutas, conducción de sesiones). Planificación y distribución de tareas. Introducción a la estimación de complejidad de un proyecto y puntos de función como modelos de estimación. Seguimiento y visibilidad del progreso. Identificación y manejo de riesgos, confidencialidad, relaciones con el cliente y análisis post-mortem. Toma de decisiones en un grupo. Comunicación. Administración de diferencias y conflictos. Negociación. Manejo de "stress".

Libros recomendados

No hay un texto para la asignatura ya que el material a cubrir se encuentra disperso en la literatura. Recomiendo que cada *empresa* forme una biblioteca con los siguientes textos:

- El proceso unificado de desarrollo de software. Jacobson, I. – Booch, G. – Rumbaugh, J. Addison Wesley 2000 (ISBN 84-7829-036-2)
- The Rational Unified Process. Ph. Kruchten. Addison Wesley 2000
- Ingeniería de software. Un enfoque práctico. Pressman, R. Quinta edición. Mc. Graw Hill 2002 (ISBN 84-481-3214-9)
- Ingeniería de software. Sommerville, I. Sexta edición. Addison Wesley 2002 (ISBN 970-26-0206-8)
- Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Addison-Wesley, 1995.
- Clay Carr: *Team Leader's Problem Solver*. Prentice-Hall, 1996
- Craig Larman: *UML y Patrones: Introducción al análisis y diseño orientado a objetos y proceso unificado*. Prentice-Hall Hispanoamericana, 2002. disponible a mediados de octubre 2002

- Cay Horstmann, Gary Cornell. *Core Java 2. Volume 1: Fundamentals*. Prentice-Hall, 1999.

Nota: Estos libros de carácter general podrán ser complementados con lecturas específicas para cada tópico tratado.

Algunas Referencias Complementarias

Desarrollo de software orientado por objetos

- Michael Blaha y William Premerlani: *Object-Oriented Modeling and Design for Database Applications*. Prentice-Hall, 1998.
- Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal. *A System of Patterns: Pattern-Oriented Software Architecture*. Wiley, 1996.
- Luiz Fernando Capretz y Miriam A. M. Capretz: *Object-Oriented Software: Design and Maintenance*. Series on Software Engineering and Knowledge Engineering, vol. 6. World Scientific, 1996.
- Martin Fowler: *Analysis Patterns: Reusable Object Models*. Addison-Wesley, 1997.
- Martin Fowler: *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 1999.
- Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- Ivar Jacobson, Grady Booch y James Rumbaugh: *The Unified Software Development Process*. Addison-Wesley, 1999.
- John Vlissides: *Pattern Hatching: Design Patterns Applied*. Addison-Wesley, 1998.
- Jos Warmer y Anneke Kleppe: *The Object Constraint Language: Precise Modeling with UML*. Addison-Wesley, 1999.
- [Láminas sobre Internet, HTML y Java.](#)
- [Object Management Group \(OMG\).](#)
- Sitio web base para los patrones: <http://hillside.net/patterns/patterns.html>
- [Repositorio Cetus sobre Tecnología Orientada a Objetos.](#)
- Tutorial SUN, documentos y enlaces sobre Java (accesible desde dominio ldc.usb.ve):

file:/usr/local/docs/java/index.html

- [Tutorial SUN \(remoto\) sobre Java.](#)

Desarrollo de software orientado por componentes

- Sametinger: *Component Oriented Software Engineering*. Springer ...
- Clemens Szyperski: *Component Based Software Engineering: Beyond Object-Oriented Programming*. Addison-Wesley, 1997.

Dinámica de grupos

- Clay Carr: *Team Leader's Problem Solver*. Prentice-Hall, 1996.
- S. Robbins: *Comportamiento Organizacional: Teoría y Práctica*. Octava edición. Prentice-Hall Hispanoamericana, 1999.

Otros tópicos en Ingeniería de Software

- Tom Gilb. *Requirements-Driven Management Using Planguage*. [Software Technology Support Center Web site](#), 1996-7. El documento también está disponible localmente en formato [doc](#) y comprimido en formato [zip](#).
- Watts Humphrey: *Introduction to the Personal Software Process (TM)*. Addison-Wesley, 1997.
- Capers Jones: *Assessment and Control of Software Risks*. Yourdon Press, 1994.
- Roger S. Pressman. *Ingeniería del Software: Un Enfoque práctico*. Cuarta edición. McGraw-Hill, 1998.
- [Sitio www de apoyo al texto de R. Pressman](#).
- [Webmaestro: un curso sobre HTML](#).

Evaluación

La asignatura tiene un componente de teoría y un componente de práctica.

1. La nota definitiva del curso se obtiene ponderando el 40 % de la nota de la teoría con el 60 % de la nota de práctica.

Evaluación de la teoría

El componente de teoría de la asignatura será evaluado por dos exámenes, presentaciones, trabajos dirigidos, trabajos de investigación etc.