
Formulario de Aprobación Curso de Actualización 2011

Asignatura: Arquitectura de Software

Profesor de la asignatura ¹: MSc. Daniel Calegari, Prof. Adjunto, Gr. 3, InCo

Profesor Responsable Local ¹: MSc. Daniel Calegari, Prof. Adjunto, Gr. 3, InCo
(título, nombre, grado, Instituto)

Instituto ó Unidad: Instituto de Computación

¹ Agregar CV si el curso se dicta por primera vez.
(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

Fecha de inicio y finalización: 2do semestre 2011
Horario y Salón: Lunes a Jueves de 19 a 22 hs. Salón de Seminarios (5to.Piso INCO)

Horas Presenciales: 57

Arancel: \$ 17.000

Público objetivo y Cupos: El público objetivo son profesionales que trabajen en desarrollo de software. Tendrán preferencia aquellas personas inscriptas en programas de posgrado del Instituto. Existe un cupo máximo de 35 personas.

Objetivos:

En la actualidad, las funcionalidades esperadas de sistemas de software han crecido considerablemente en cantidad y complejidad. Un problema crucial que enfrenta el desarrollo de este tipo de sistemas de gran porte es definir una estructura manejable y controlable. En este contexto, la Arquitectura de Software ha emergido como disciplina cumpliendo ya un rol significativo en el desarrollo de software.

Los objetivos de este curso son introducir al estudiante en los conceptos básicos de la Arquitectura de Software, transmitir su importancia y su rol dentro del proceso de desarrollo, brindar los conceptos técnicos más relevantes e identificar el rol del Arquitecto en el proceso de desarrollo.

Conocimientos previos exigidos:

Conocimientos básicos de desarrollo de sistemas orientados a objetos y de bases de datos.

Conocimientos previos recomendados:

Experiencia o conocimientos en procesos de desarrollo e ingeniería de software, así como en modelado con UML.

Metodología de enseñanza:

Consta de dos partes. La primera parte corresponde a exposición teórica-práctica, correspondiente al 60% de la dedicación total del curso. La segunda parte representa el 35% del curso y está dedicada a un trabajo de laboratorio

grupal que será realizado a lo largo del curso, en paralelo con la exposición teórico-práctica. El 5% restante se emplea en evaluación.

El curso será dictado en 19 clases teórico-prácticas de 3 horas. PDe esta forma, previendo una dedicación del estudiante de 0,7 horas por cada hora dictada (al no dictarse en días corridos. se espera esta dedicación adicional por parte del alumno), los totales de horas se computan de la siguiente forma:

Horas teórico-prácticas: 98

Horas de laboratorio: 22

Horas de Estudio Asistido para preparación de trabajos y defensa: 25

Horas de evaluación: 5

Total horas: 150

Forma de evaluación:

Presentación y defensa del trabajo de laboratorio (grupal) y prueba final (individual).

Temario:

1. Introducción

Definición del concepto Arquitectura de Software e importancia del mismo dentro de un proceso de desarrollo. Motivación para la adopción de un enfoque centrado en la arquitectura, características y beneficios.

2. Procesos de Desarrollo y Arquitectura de Software

Introducción a los procesos de desarrollo y su relación con la Arquitectura de Software, identificación de involucrados (stakeholders), uso de escenarios y rol del arquitecto de software.

3. Representación de la Arquitectura

Modelos de representación de la arquitectura, vistas para representar diferentes aspectos de la arquitectura, documento de especificación de la arquitectura (SAD).

4. Perspectiva de Calidad

Atributos de calidad (requerimientos no funcionales) como influenciadores de la arquitectura y tácticas para su resolución.

5. Estilos de Arquitectura

Patrones de organización de un sistema a nivel arquitectónico: Capas, Pipes&Filters, Blackboard, Orientación a Servicios, arquitecturas heterogéneas.

6. Tecnología y Arquitecturas

Plataformas de desarrollo más importantes (JEE, .NET) y resolución de problemas arquitectónicos mediante su uso.

7. Evaluación de Arquitecturas

Mecanismos de evaluación de arquitecturas de software para el aseguramiento de los requerimientos funcionales y no funcionales.

8. Temas Actuales

Model-Driven Architecture, Product Lines, Software Factories, entre otros.

Bibliografía:

- [1] Software Architecture in Practice, Second Edition.
Len Bass, Paul Clements, Rick Kazman. Addison Wesley, 2003, ISBN 0321154959
- [2] Software System Architecture: Working With Stakeholders Using Viewpoints and Perspectives.
Nick Rozanski, Eoin Woods. Addison Wesley, 2005, ISBN 0321112296
- [3] The Art of Software Architecture: Design Methods and Techniques.
Stephen T. Albin. John Wiley & Sons, 2002, ISBN 0471228869
- [4] Documenting Software Architectures: Views and Beyond.
Paul Clements, Felix Bachmann, Len Bass, et al. Addison Wesley, 2002, ISBN 020170482X
- [5] Evaluating Software Architectures: Methods and Case Studies.
Len Bass, Paul Clements, Rick Kazman. Addison Wesley, 2001, ISBN 020170482X
- [6] Software Architecture: Perspective on an Emerging Discipline
M. Shaw, D. Garlan. Prentice Hall, 1996, ISBN 0131829572
- [7] Pattern-Oriented Software Architecture: A System of Patterns
F. Buschman et al. John Wiley & Sons, 1996, ISBN 0471958697
- [8] Documentación de UML: <http://www.uml.org>