



UNIVERSIDAD CATÓLICA DEL NORTE

PROGRAMA DE CURSO

I. IDENTIFICACIÓN DEL CURSO

Carrera: Magíster en Ingeniería Informática

Unidad responsable: Departamento de Ingeniería de Sistemas y Computación

Nombre del curso: Arquitectura de Software

Código: Por definir

Semestre en la malla:

Cursos Requisitos:

Procesos de desarrollo de software

Requisito para:

II. PROPÓSITO DEL CURSO

Al finalizar el curso, el estudiante estará preparado para construir soluciones computacionales a problemas de ingeniería y ciencias basados utilizando las mejores prácticas en definición y diseño arquitectónico de productos de software que aseguren calidad y mantenibilidad de dicho producto.

III. UNIDADES TEMÁTICAS

Capítulo 1. Introducción

- 1.1. ¿Qué es la Arquitectura de Software?
- 1.2. La arquitectura y el ciclo del proceso del software.
- 1.3. Importancia de una adecuada arquitectura.

Capítulo 2. Definición de la Arquitectura

- 2.1. Atributos de Calidad Asociados a la Arquitectura.
- 2.2. Notación UML y su importancia en la claridad de la arquitectura.
- 2.3. Tipos de Arquitecturas:
 - 2.3.1. Cliente – Servidor
 - 2.3.2. Basada en Capas
 - 2.3.3. Distribuida
 - 2.3.4. Basada en Componentes
 - 2.3.5. Orientada al uso en Sistemas Paralelos
 - 2.3.6. Para Sistemas de Misión Crítica.

Capítulo 3. Patrones de Diseño de Software

- 3.1. ¿Qué es un patrón de diseño?
- 3.2. Tipos de Patrones.
- 3.3. Lectura y aplicación de un patrón desde la bibliografía.
- 3.4. Documentación de patrones identificados en el desarrollo de un producto de software.

Capítulo 4. Lenguajes de Programación y sus Arquitecturas Específicas.

- 4.1. JEE
- 4.2. JME
- 4.3. .NET

Capítulo 5. Evaluación de Arquitecturas de Software.

- 5.1. Preguntas que deben ser contestadas en un proceso de evaluación de arquitectura.
- 5.2. Método ATAM para evaluación de una arquitectura de software
- 5.3. Método FMEA para evaluación de una arquitectura de software

IV. RESULTADOS DE APRENDIZAJE

- A. Diseñar arquitecturas de software basado en las mejores practicas existentes, que permitan asegurar un desarrollo de software de calidad y evolutivo.
- B. Aplicar patrones de diseño en el proceso de definición de una arquitectura para un producto de software.
- C. Seleccionar arquitecturas y patrones propietarios de un lenguaje de acuerdo a los requerimientos específicos de cada producto de software diseñado.

V. RECURSOS BIBLIOGRÁFICOS

Textos Guías:

- Bruegge, Bernd, Object-oriented software engineering: using UML, patterns and Java, 3rd Ed. 2009.

Textos Complementarios:

- Larman, Craig, Applying UML and patterns : an introduction to object-oriented analysis and design and the unified process, 3rd ed. 2004
- Martin, Robert C, Design Principles and Design Patterns
- Gamma, Erich, Patrones de diseño : elementos de software orientado a objetivos reutilizable / Eric Gamma, 2003
- Fowler, Martin, UML distilled : a brief guide to the standard object modeling lenguaje, 3rd ed. 2003.
- Alur, Deepak, Core J2EE patterns : best practices and design strategies, 2nd ed. 2003