

# ESCUELA DE INGENIERÍA

## Informática Y Sistemas

<b>ASIGNATURA</b>	DESARROLLO DE SOFTWARE BASADO EN COMPONENTES
<b>CODIGO</b>	ST0791
<b>SEMESTRE</b>	2019-2
<b>INTENSIDAD HORARIA</b>	32 horas semestral
<b>CARACTERÍSTICAS</b>	Suficientable
<b>CRÉDITOS</b>	

---

### 1. JUSTIFICACIÓN CURSO

El curso es pertinente y se justifica dentro del plan de estudios puesto que le permite al estudiante reconocer los enfoques utilizados para abordar el desarrollo de soluciones software aplicando la reutilización como estrategia básica.

### 2. OBJETIVOS GENERALES DEL CURSO

#### 2.1. Objetivo General:

- El curso desarrollará y ampliará las capacidades del estudiante para especificar una solución software siguiendo el enfoque de componentes que pueden ser modelados utilizando UML. El curso enfatiza la importancia de definir esquemas de trabajo y mecanismos que favorezcan una reutilización sistémica de los módulos o componentes (desarrollados o adquiridos) que conforman una solución software.

#### 2.2. Objetivos Específicos:

- 2.2.1. Entender las diferentes perspectivas del desarrollo basado en componentes
- 2.2.2. Analizar los modelos de ciclo de vida asociados a un enfoque basado en componentes
- 2.2.3. Entender y aplicar el proceso general que se sigue para el desarrollo de componentes
- 2.2.4. Aplicar UML como lenguaje para especificar tanto el diseño de alto nivel, como el diseño detallado
- 2.2.5. Reconocer las características requeridas de un modelo de componentes
- 2.2.6. Indagar acerca de servicios y facilidades disponibles para búsqueda y adquisición de componentes

### 3. DESCRIPCIÓN ANALÍTICA DE CONTENIDOS

- 3.1. Visión General del Desarrollo Basado en Componentes:
  - 3.1.1. Dimensiones del desarrollo basado en componentes
  - 3.1.2. Modelos de ciclo de vida del enfoque de componentes
  - 3.1.3. Evolución de las aproximaciones de desarrollo de software
  - 3.1.4. Clasificación de los componentes
  - 3.1.5. Lecturas Requeridas:
    - 3.1.5.1. Lectura 1. Component-based Development Process and Component Lifecycle. Ivica Crnkovic 1, Stig Iarsson2, Michel Chaudron3
    - 3.1.5.2. Lectura 2. Towards a Suitable and Systematic Approach for Component Based Software Development. Kuljit Kaur, Parminder Kaur, Jaspreet Bedi, and Hardeep Singh.
- 3.2. Conceptos básicos del desarrollo basado en componentes :
  - 3.2.1. Definición de componente e interface
  - 3.2.2. De los objetos a los componentes
  - 3.2.3. Arquitectura y componentes
  - 3.2.4. Modelo de componentes
  - 3.2.5. Plataforma de componentes
  - 3.2.6. Marcos de trabajo y patrones de diseño
  - 3.2.7. Lecturas Requeridas:
    - 3.2.7.1. Lectura 3. Desarrollo de Software Basado en Componentes. Lidia Fuentes, José M. Troya y Antonio Vallecillo
    - 3.2.7.2. Lectura 4. Classification Framework for Business Components. Klement J. Fellner and Klaus Turowski
    - 3.2.7.3. Lectura 5. Szyperski, Capítulo
- 3.3. Análisis de Requisitos:
  - 3.3.1. Visión del proceso de negocio
  - 3.3.2. Modelo de conceptos del negocio
  - 3.3.3. Definición del alcance: Casos de uso, actores
  - 3.3.4. Realización de casos de uso
  - 3.3.5. Atributos de calidad del servicio
  - 3.3.6. Lecturas Requeridas:
    - 3.3.6.1. Lectura 6. Chessman & Daniels, Capítulo
- 3.4. Identificación de Componentes:
  - 3.4.1. Principios de la arquitectura de layers
  - 3.4.2. Refinando modelo de tipos de negocio.
    - 3.4.2.1. Tipo de datos asociados a conceptos del dominio (DataType, Value Objects)
    - 3.4.2.2. Agregados/cluster del dominio

- 3.4.3. Identificando interfaces del negocio: Interface de servicios, interfaces de repositorio
- 3.4.4. Identificando interfaces de aplicación
  - 3.4.4.1. Definición de los servicios requeridos ( a partir de los casos de uso)
  - 3.4.4.2. Definición de contratos
- 3.4.5. Especificación de interfaces (IDL)
- 3.4.6. Lecturas Requeridas:
  - 3.4.6.1. Lectura 7. Chessman & Daniels; Capítulo
  - 3.4.6.2. Lectura 8. Domain Driven Design. Páginas 36/54  
Lectura 9. A Domain enguage Approach to Designing Dynamic Enterprise Component â based Architectures to Support Business Services. Ali Arsanjani
  - 3.4.6.3.
- 3.5. Arquitectura de especificación de componentes :
  - 3.5.1. Caracterizando un componente de aplicación
  - 3.5.2. Caracterizando un componente de negocio
  - 3.5.3. Interacción entre componentes
  - 3.5.4. Lecturas Requeridas:
    - 3.5.4.1. Lectura 10. Chessman & Daniels, Capítulo
    - 3.5.4.2. Lectura 11. Domain Driven Design. Páginas 54 / 64
- 3.6. Modelos de componentes:
  - 3.6.1. Estándares de comunicación entre componentes
  - 3.6.2. Características deseables de un modelo de componentes
  - 3.6.3. Clasificación de los modelos de componentes
  - 3.6.4. Reconociendo algunos modelos de componentes
  - 3.6.5. Lecturas Requeridas:
    - 3.6.5.1. Lectura 12. Szyperski, Capítulo
    - 3.6.5.2. Lectura 13. Software Component Models. Kung / Kiu Lau and Zheng Wang
- 3.7. Consideraciones de Implementación de componentes:
  - 3.7.1. Diseño interno de componentes: el papel de los patrones de diseño; consideraciones de persistencia
  - 3.7.2. Consideraciones de persistencia
  - 3.7.3. Consideraciones para el ensamble y despliegue de componentes
  - 3.7.4. Lectura 14. Chessman & Daniels, Capítulo
- 3.8. Hacia un mercado de componentes . Estos temas serán trabajados por medio de exposiciones de los estudiantes
  - 3.8.1. Decisiones acerca de construir o comprar

- 3.8.2. Gestión de librería de componentes
- 3.8.3. Servicios para búsqueda y recuperación de componentes
- 3.8.4. Artículos Sugeridos:
  - 3.8.4.1. A DYNAMIC BUSINESS MODEL FOR COMPONENT-BASED SIMULATION SOFTWARE Stephan Onggo, Didier Soopramanien, and Mike Pidd
  - 3.8.4.2. A Case Study on the Multicriteria Selection of Software Components. Thomas Neubauer, Jan Pichler, Christian Stummer
  - 3.8.4.3. Modeling COTS/Based Development and Related Selection Methods Processes with MAP. Sihem Ben Sassi 1 Lamia Labeled Jilani2 Henda Hajjami Ben Ghezala3
  - 3.8.4.4. A Business Perspective on Component Trading: Criteria, Immaturities, and Critical Success Factors. Sven Overhage, Peter Thomas
- 3.9. Metodología (estrategias de enseñanza / aprendizaje):
  - 3.9.1. Enfoque teórico-práctico, mediante el cual se realizan talleres para aplicar y asimilar los conceptos teóricos vistos.

#### 4. EVALUACIÓN

- 4.1. 2 parciales del 15% cada uno
- 4.2. Proyecto: 40%
  - 4.2.1. Entrega 1 Especificación del sistema 10%
  - 4.2.2. Entrega 2 Especificación de la arquitectura y los componentes 15%
  - 4.2.3. Entrega 3 Implementación de un componente 15%
- 4.3. Informes de lectura y talleres en clase: 15%
- 4.4. Profundización en un tema (incluye un informe tipo artículo y exposición): 15%

#### 5. BIBLIOGRAFIA GENERAL

- 5.1. Cheesman John, Daniels John. UML Components. A simple process for specifying component based software. Addison Wesley, 2001.
- 5.2. Szyperski C., Component Software âBeyond Object Oriented Programming. Addison Wesley, 1998
- 5.3. Avram, Abel, Marinescu Floyd. Domain Driven Design Quickly
- 5.4. Brown, Alan. Large Scale Component Base Development. Prentice Hall,2000
- 5.5. Guide to the Software Engineering Body of Knowledge. SWEBOK Versión 2004.Capítulo 3.