

**1. INFORMACIÓN DEL CURSO**

<b>Nombre:</b> INGENIERÍA DE SOFTWARE AVANZADO	<b>Número de créditos:</b> 8	<b>Prerrequisitos:</b>
<b>Departamento:</b> Ciencias Computacionales e Ingenierías	<b>Tipo:</b> Presencial	<b>Nivel:</b> Optativa abierta
<b>Horas teoría:</b> 40	<b>Horas práctica:</b> 40	<b>Total de horas por cada semestre:</b> 80

**2. DESCRIPCIÓN****Objetivo general**

El estudiante aplicará distintas técnicas de desarrollo de software cuidando la calidad de los procesos utilizando el diseño orientado a objetos para desarrollar y documentar un sistema real.

**Contenido temático sintético**

1. Introducción
2. Construcción
3. Pruebas de Software
4. Mantenimiento de Software
5. Calidad
6. Gestión de la configuración del software

**CONTENIDO DESARROLLADO****1. Introducción**

Objetivo específico: El estudiante conocerá el estado actual de la disciplina, su importancia y su aplicación, del mismo modo definirá, organizará y estructurará un proyecto de informática para controlar su desarrollo a través de técnicas y metodologías.

Carga Horaria teórica: 4 Carga Horaria práctica: 6

Contenido programático.

**1. Introducción.**

- 1.1. La ingeniería del software como disciplina profesional.
- 1.2. Procesos de software.
  - 1.2.1. Ciclos de vida de software.
- 1.3. Requisitos.
  - 1.3.1. Definiciones preliminares y características.
  - 1.3.2. Tipos de requisitos.
  - 1.3.3. Las actividades de los requisitos.
- 1.4. Notaciones para el modelo conceptual.
  - 1.4.1. Casos de uso.
  - 1.4.2. Modelos entidad relación.
  - 1.4.3. Diagramas UML.
  - 1.4.4. Notaciones formales.

**2. Construcción**

Objetivo: El estudiante conocerá los principios básicos para la construcción de software de calidad,

de acuerdo con técnicas de Ingeniería de Software.

Carga Horaria teórica: 6

Carga Horaria práctica: 9

Contenido programático.

## 2. Construcción

2.1. Lenguajes de construcción.

2.2. Realización de código.

2.3. Reutilización de código.

2.4. Principios fundamentales de la construcción de software.

2.4.1. Minimizar la complejidad.

2.4.2. Anticipar los cambios.

2.4.3. Construir para verificar.

2.4.4. Utilización de los estándares.

2.5. La calidad de la construcción de software.

2.5.1. Aserciones y diseño por contrato.

2.5.2. Análisis de rendimiento.

2.6. Gestión de la construcción.

2.6.1. Planificación de la construcción.

2.6.2. Métricas de la construcción.

## **3. Pruebas de software**

Objetivo: El estudiante conocerá los aspectos esenciales de las pruebas de software y aplicara distintas técnicas de pruebas de software existentes.

Carga Horaria teórica: 5

Carga Horaria práctica: 5

Contenido programático.

### 3. Pruebas de software

3.1. Conceptos fundamentales

3.1.1. Limitación en la realización de pruebas

3.1.2. Pruebas y riesgos.

3.2. Técnicas de pruebas.

3.2.1. Pruebas de caja blanca

3.2.2. Pruebas de caja negra

3.3. Niveles de pruebas

3.3.1. Pruebas según su objeto.

3.3.2. Pruebas según el objetivo.

3.3.3. Pruebas unitarias.

3.4. Métricas relacionadas con las pruebas.

3.4.1. Medidas durante las pruebas

3.4.2. Evaluación de las pruebas.

## **4. Mantenimiento de software**

Objetivo: El estudiante entenderá la necesidad de cambios en el sistema y creará un modelo de mejora.

Carga Horaria teórica: 4

Carga Horaria práctica: 6

Contenido programático.

4. Mantenimiento de software.

4.1. Conceptos fundamentales

4.1.1. Que es el mantenimiento de software

- 4.1.2. Características de mantenimiento.
- 4.1.3. La facilidad de mantenimiento.
- 4.1.4. Mantenimiento de calidad.
- 4.2. Las practicas del mantenimiento de software
- 4.2.1. La evolución del software y sus leyes.
- 4.3. El proceso de mantenimiento.
- 4.3.1. Las actividades de mantenimiento.
- 4.3.2. El mantenimiento como preparación.
- 4.4. Técnicas para el mantenimiento de software
- 4.4.1. Ingeniería inversa.
- 4.4.2. Reingeniería.
- 4.4.3. Reestructuración.
- 4.4.4. Métricas de mantenimiento.

## **5. Calidad**

Objetivo: El estudiante comprenderá la conceptualización de calidad y su en foque en el desarrollo de software.

Carga Horaria teórica: 6

Carga Horaria práctica: 4

Contenido programático.

### 5. Calidad

#### 5.1.Introducción

- 5.1.1. Cultura y ética de la calidad
- 5.1.2. Valor y costes de la calidad
- 5.1.3. Los múltiples aspectos de la calidad
- 5.2.Calidad del producto
- 5.2.1. El modelo de calidad de McCall
- 5.2.2. El modelo de Boehm
- 5.2.3. El modelo de calidad ISO/IEC 9126
- 5.3.Calidad del proceso
- 5.3.1. Aseguramiento de la calidad
- 5.3.2. El modelo CMMI
- 5.3.3. Modelo SPICE: Estándar ISO/IEC 15504
- 5.3.4. Estándares ISO 9000

## **6. Gestión de la configuración del software**

Objetivo: El estudiante conocerá e implementara las distintas técnicas de gestión de la configuración del software.

Carga Horaria teórica: 5

Carga Horaria práctica: 4

Contenido programático.

### 6. Gestión de la configuración del software

- 6.1.La configuración del software
- 6.2. Actividades de gestión de configuración de software.
- 6.2.1. Identificación de la configuración de software.
- 6.2.2. Control de los cambios en el software.
- 6.2.3. Gestión de entregas.
- 6.3.Planificación y gestión
- 6.3.1. Contabilidad y medición en gestión de la configuración.
- 6.3.2. Auditoria de la configuración del software

- 6.4. Técnicas y herramientas para la configuración del software.
- 6.4.1. Versiones divisiones y deltas.
- 6.4.2. Políticas de control versiones.

**Modalidades de enseñanza aprendizaje**

Curso- Taller

**Modalidad de evaluación**

Conocimientos: (Ensayos, casos, resolución de problemas, exámenes, etc.)

Habilidades y Destrezas: (actividades prácticas para el desarrollo de habilidades del pensamiento, de las capacidades motrices, etc.)

Exámenes 30 %

Actividades extra- áulicas: Tareas 20 %

Actividades áulicas: prácticas y ejercicios en el aula 20 %

Proyecto final 20 %

Actitud y valores: (interés, participación, asistencia a asesorías, trabajo en equipo, puntualidad, responsabilidad, trato, tolerancia etc.)

Autoevaluación 5%

**Competencia a desarrollar**

Al término de la asignatura de Ingeniería de Software avanzado se conocerán y aplicarán los conceptos fundamentales de Diseño e Implementación de Software, por medio de la realización de un proyecto software real.

**Campo de aplicación profesional**

Desarrollador en la industria del software, aplicando procesos de aseguramiento de la calidad.

**3. BIBLIOGRAFÍA**

**BIBLIOGRAFÍA BÁSICA**

Pressman, Roger, Ingeniería del Software Un enfoque práctico, McGraw-Hill, 2005

Kendall y Kendall, Análisis y Diseño de Sistemas, Prentice Hall, 2005

Sommerville, Ian, Ingeniería del Software, Addison Wesley, 2005

**BIBLIOGRAFÍA COMPLEMENTARIA**

IEEE Std 830, IEEE Guide for Software Requirements Specifications (ANSI).

IEEE Std 1074, IEEE Standard for Developing Software Life Cycle Processes.

IEEE/EIA Std 12207.0 , IEEE Standard for Software Life Cycle Processes.

IEEE Std 1016, IEEE Recommended Practice for Software Design Descriptions.

IEEE Standard 1012 for Software Verification and Validation