

**SÍLABO**  
**INGENIERÍA DE SOFTWARE I**

**ÁREA CURRICULAR: INGENIERÍA DE SOFTWARE**

**CICLO:** VI

**SEMESTRE ACADÉMICO:** 2017-I

- I. CÓDIGO DEL CURSO** : 09011906050
- II. CRÉDITOS** : 05
- III. REQUISITOS** : 09093205051 Gestión de Procesos
- IV. CONDICIÓN DEL CURSO** : Obligatorio

**V. SUMILLA**

El curso es naturaleza formación especializada; dirigido a que el estudiante sea capaz de realizar las actividades de gestión de proyectos de software, análisis de requisitos, realizar de manera básica el diseño de la arquitectura, construcción e implementación de software.

Unidades: Ingeniería de software – Gestión de proyecto de software – Proceso de desarrollo e implementación de software

**VII. FUENTES DE CONSULTA**

**Bibliográficas**

- Bruegge, B; Dutoit, A (2009). *Object-Oriented Software Engineering con UML, Patrones y Java*. 3ra edición. New York. Pearson education SA.
- Gomaa, H (2011), *Modelado y Diseño de Software: UML, casos de uso, patrones y Arquitecturas Software*. New York. Cambridge University Press.
- IEEE (2014). *Guide to the Software Engineering Body of Knowledge Version 3.0 SWEBOK. 3.0 edition*. Washington: Copyright © 2014 IEEE
- Kim, R.; (2006) *Learning UML 2.0*. First Edition, California USA: Ed. O'Reilly Media.
- Pressman, R. (2009). *Ingeniería de Software*. 7ma edición. México: Ed. Mc Graw Hill.
- Sommerville, I (2010). *Ingeniería del software*. 9na edición. Madrid: Pearson educación. S.A
- Quatrani, T.; Palistrant, J. (2006) *Visual Modeling with IBM Rational Software Architect and UML*. Estados Unidos de Norteamérica. IBM Press.

**Electrónicas**

- Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual (INDECOPI)  
<http://bvirtual.indecopi.gob.pe/normas/29110-5-1-2.pdf>
- Software Engineering Body of Knowledge (SWEBOK)  
<http://www.computer.org/portal/web/swebok>
- UML, ejemplo sencillo sobre Modelado de un Proyecto (Microsoft)  
<https://msdn.microsoft.com/es-es/library/bb972214.aspx#XSLTsection130121120120>
- Diagramas de casos de uso de UML: Instrucciones  
<https://msdn.microsoft.com/es-es/library/dd409432.aspx>
- CRUD Pattern in Use Cases (Software Engineering)  
<http://www.se.cs.put.poznan.pl/knowledge-base/software-engineering-blog/crud-pattern-in-use-cases>

**VII. UNIDADES DE APRENDIZAJE**

**UNIDAD I. INGENIERÍA DE SOFTWARE**

**OBJETIVOS DE APRENDIZAJE**

- Reconocer los conceptos de software.
- Describir el marco de proceso de software, los métodos y modelos de procesos de software.

- Diseñar estructuras de proyectos de software.

## **PRIMERA SEMANA**

### **Primera sesión**

Software: definición, características, dominios de aplicación y software heredados.

Ingeniería de software: definición y capas.

### **Segunda sesión**

Proceso de Ingeniería de software: definición, ciclos de vida del software, captura de datos y mejora y herramientas.

### **Laboratorio**

Formación del equipo de trabajo. Configuración del repositorio de documentación y completar la plantilla. Elección de la empresa en donde se desarrollará el proyecto final.

## **SEGUNDA SEMANA**

### **Primera sesión**

Modelos de ingeniería de software: modelos principales y tipos de modelos.

### **Segunda sesión**

Métodos de ingeniería de software: tipos de métodos

Grupo de trabajo: identificar el modelo o método a emplear en su proyecto.

### **Laboratorio**

Plantilla enunciado del trabajo: completar el modelo o método a emplear.

## **UNIDAD II. GESTIÓN DE PROYECTO DE SOFTWARE**

### **OBJETIVOS DE APRENDIZAJE**

- Reconocer el proceso de planificación de un proyecto de software.
- Describir el proceso de planificación de un proyecto de software.
- Aplicar el proceso de planificación de un proyecto de software.

## **TERCERA SEMANA**

### **Primera sesión**

Planificación de proyectos de software: planificación, entregables, costos, cronograma, recursos, riesgos, calidad y gestión.

### **Segunda sesión**

Grupo de trabajo: bosquejar la planificación de su proyecto de software.

### **Laboratorio**

Microsoft Office Project: calendario, tareas, hitos, duración, recursos y predecesoras.

## **CUARTA SEMANA**

### **Primera sesión**

NTP-29110 - Proceso de gestión del proyecto: propósito, objetivos, productos de entrada, salida, internos, roles involucrados y actividades.

### **Segunda sesión**

NTP-29110 - Proceso de implementación de software: propósito, objetivos, productos de entrada, salida, internos, roles involucrados y actividades.

### **Laboratorio**

Microsoft Office Project: costos, control, recursos, vistas, línea base, seguimiento e informes.

## **QUINTA SEMANA**

### **Primera sesión**

Plantillas: del proceso de gestión del proyecto y del proceso de implementación de software.

### **Segunda sesión**

Grupo de trabajo: acoplar las fases/actividades de su modelo o método elegido con la NTP. Avanzar con el cronograma y las plantillas.

### **Laboratorio**

Microsoft Office Project: cada grupo de trabajo realizará su cronograma de proyecto empleando la herramienta.

### **UNIDAD III. PROCESO DE DESARROLLO DE SOFTWARE E IMPLEMENTACIÓN DE SOFTWARE**

#### **OBJETIVOS DE APRENDIZAJE**

- Identificar los conceptos fundamentales de la Ingeniería de Requerimientos

#### **SEXTA SEMANA**

##### **Primera sesión**

Requerimientos: definición, principales dificultades e impacto de requerimientos incorrectos.

Ingeniería de requerimientos: definición

##### **Segunda sesión**

Requerimientos: funcionales, no funcionales y cuantificables.

##### **Laboratorio**

Rational Software Architect: introducción al uso de la herramienta.

UML: metamodelo. Resumen de notación UML.

Modelo basico de un modelo de requerimiento

#### **SÉPTIMA SEMANA**

##### **Primera sesión**

Proceso de requerimientos y agente del proceso

Captura de requerimiento: definición y técnicas (Análisis de documentación existente, Glosario, Observación, Entrevista, Cuestionario)

##### **Segunda sesión**

Captura de requerimiento: técnicas (Historia del usuario, Control de problemas)

Del modelo de proceso al requerimiento

##### **Laboratorio**

Identificación de los requerimientos a partir de l procesos de negocio con BIZAGI

#### **OCTAVA SEMANA**

Examen Parcial.

#### **NOVENA SEMANA**

##### **Primera sesión**

Clasificación de los requisitos: Funcionales y no funcional, si el requisito es derivado de uno o mas requisitos de alto nivel, si el requisito está en el producto o proceso, la prioridad del requisito, el alcance del requisito, volatilidad estabilidad.

##### **Segunda sesión**

Modelo de actores

Paquetes lógicos

Diagrama de casos de uso

Relaciones entre Casos de Uso: include, extends, precedence, generalization

##### **Laboratorio**

Rational Software Architect: Modelo de caso de uso básico

#### **DÉCIMA SEMANA**

##### **Primera sesión**

Patrones de caso de uso

Prototipos de interfaces de usuario

##### **Segunda sesión**

Especificación de requerimientos

Revisión del documento de requerimientos

Nivel de detalle de la especificación de requerimientos

##### **Laboratorio**

Rational Software Architect: Modelo de caso de uso completo.

## **UNDÉCIMA SEMANA**

### **Primera sesión**

Asignación arquitectónica del diseño de los requisitos

### **Segunda sesión**

Diagrama de actividades

### **Laboratorio**

Rational Software Architect: Diagrama de actividades

## **DUODÉCIMA SEMANA**

### **Primera sesión**

Diagrama de clases

### **Segunda sesión**

Diagrama de paquetes

### **Laboratorio**

Rational Software Architect: Diagrama de clases

## **DECIMOTERCERA SEMANA**

### **Primera sesión**

Diagrama de secuencia

### **Segunda sesión**

Validación de requisitos

### **Laboratorio**

Rational Software Architect: Diagrama de secuencia

## **DECIMOCUARTA SEMANA**

Exposiciones de proyectos por equipos.

## **DECIMOQUINTA SEMANA**

Exposiciones de proyectos por equipos.

## **DECIMOSEXTA SEMANA**

Examen Final.

## **DECIMOSÉPTIMA SEMANA**

Entrega de promedios finales y acta del curso.

## **VIII. CONTRIBUCIÓN DEL CURSO AL COMPONENTE PROFESIONAL**

- |                                  |          |
|----------------------------------|----------|
| a. Matemática y Ciencias Básicas | <b>0</b> |
| b. Tópicos de Ingeniería         | <b>5</b> |
| c. Educación General             | <b>0</b> |

## **IX. PROCEDIMIENTOS DIDÁCTICOS**

- **Método Expositivo – Interactivo.** Comprende la exposición del docente y la interacción con el estudiante.
- **Método de Discusión Guiada.** Conducción del grupo para abordar situaciones y llegar a conclusiones y recomendaciones.
- **Método de Demostración – Ejecución.** Se utiliza para ejecutar, demostrar, practicar y retroalimentar lo expuesto.
- **Método de la Casuística.** El docente presenta casos tipo y los resuelve en clase.
- **Método Flipped Classroom.** El aula volteada, para el contenido virtual, invierte los métodos tradicionales de enseñanza, la entrega de la instrucción se realiza en línea, fuera de la clase y la tarea se realiza en el aula.

## **X. MEDIOS Y MATERIALES**

- **Equipos:** Computadora, écran y proyector multimedia.
- **Materiales:** Manual Universitario, material docente, prácticas dirigidas de laboratorio y textos bases (ver fuentes de consultas).

- **Software:** Herramienta de diseño, modelado y desarrollo de software, herramientas para crear prototipos, Herramientas de gestión de documentos en la nube, herramientas de gestión de requisitos.

## XI. EVALUACIÓN

El promedio final se obtiene del modo siguiente:

$$PF = (2*PE + EP+EF) / 4$$

$$PE = ( (P1+P2+P3+P4-MN)/3 + W1 + PL) / 3$$

$$PL = (Lb1+Lb2+Lb3+Lb4) / 4$$

**Donde:**

**PF** = Promedio final del curso.

**PE** = Promedio de evaluación.

**EP** = Examen parcial (escrito).

**EF** = Examen final (escrito).

**P1...P4** = Evaluaciones teoría

**W1** = Trabajo final (impreso y digital)

**PL** = Promedio de Laboratorio.

**Lb1...Lb4** = Evaluaciones de laboratorio

## XII. APOORTE DEL CURSO AL LOGRO DE RESULTADOS

El aporte del curso al logro de los resultados (Outcomes), para la Escuela Profesional de Ingeniería de Computación y Sistemas, se establece en la tabla siguiente:

**K** = clave

**R** = relacionado

**Recuadro vacío** = no aplica

a.	Habilidad para aplicar conocimientos de computación y matemáticas apropiadas para los resultados del estudiante y las disciplinas enseñadas.	
b.	Habilidad para analizar un problema e identificar y definir los requerimientos apropiados para su solución.	K
c.	Habilidad para diseñar, implementar y evaluar un sistema basado en computadoras, procesos, componentes o programa que satisfagan las necesidades requeridas.	
d.	Habilidad para trabajar con efectividad en equipos para lograr una meta común.	
e.	Comprensión de los aspectos y las responsabilidades profesional, ética, legal, de seguridad y social.	
f.	Habilidad para comunicarse con efectividad con un rango de audiencias.	
g.	Habilidad para analizar el impacto local y global de la computación en los individuos, organizaciones y la sociedad.	R
h.	Reconocer la necesidad y tener la habilidad para comprometerse a un continuo desarrollo profesional.	
i.	Habilidad para usar técnicas, destrezas, y herramientas modernas necesarias para la práctica de la computación.	K
j.	Comprensión de los procesos que soportan la entrega y la administración de los sistemas de información dentro de un entorno específico de aplicación.	

## XIII. HORAS, SESIONES, DURACIÓN

a) Horas	de	<table><tr><th>Teoría</th><th>Práctica</th><th>Laboratorio</th></tr><tr><td>3</td><td>2</td><td>2</td></tr></table>	Teoría	Práctica	Laboratorio	3	2	2	clase:
Teoría	Práctica	Laboratorio							
3	2	2							

b) **Sesiones por semana:** Tres sesiones.

c) **Duración:** 7 horas académicas de 45 minutos

## XIV. PROFESOR DEL CURSO

Ing. Palomino Guerrero, Carla Rocío

**XV. FECHA**

La Molina, marzo de 2017.