

## **FACULTAD/ESCUELA DE BARBERI DE INGENIERÍA, DISEÑO Y CIENCIAS APLICADAS**

### **DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS INTELIGENTES**

#### ***Información de la asignatura***

<b>Nombre de la asignatura</b>	Ingeniería de software II
<b>Código de la asignatura</b>	09773 - TIC
<b>Periodo Académico</b>	202510
<b>Nrc</b>	11177
<b>Grupo</b>	003
<b>Programas/Semestres</b>	SIS 03
<b>Intensidad horaria</b>	2
<b>Intensidad Semanal</b>	4
<b>Créditos</b>	3
<b>Docente(s)</b>	Ingri Lorena Jojoa Lopez

#### ***Introducción o presentación general del curso***

En este curso se continúa el proceso de formación en el ciclo de vida del desarrollo del software que inició con el curso Ingeniería de Software I: Se estudian conceptos, métodos, técnicas y buenas prácticas de ingeniería de software, reconociendo la importancia del mejoramiento continuo en el desarrollo de software. Específicamente, el curso se enfoca en la elicitación, análisis, especificación, y validación de requerimientos funcionales; el diseño detallado de la solución aplicando principios de diseño orientado a objetos; y la administración de la configuración.

En el contexto anterior, se invita al estudiante a hacer un permanente autocuestionamiento y autorreflexión sobre las actividades que realiza y cómo las realiza, a nivel personal y grupal, en el desarrollo del software. Más aún, en cómo podría mejorarlas para diseñar productos de software que cumplan con los requerimientos establecidos por los clientes.

#### ***Formación en competencias***

SO1. Solución de problemas: Identificar, formular y resolver problemas complejos de ingeniería aplicando pensamiento crítico y principios de las ciencias, las matemáticas, la ingeniería y, en particular, de las Ciencias de la Computación y de la Ingeniería de Software.

SO2. Diseño de ingeniería: Diseñar soluciones y procesos basados en software que satisfagan necesidades específicas y generen valor a sus usuarios, considerando la salud pública, la seguridad y el bienestar de las personas, así como factores globales, culturales, sociales, ambientales y económicos.

SO3. Comunicación efectiva: Comunicarse efectivamente de forma oral y escrita, tanto en español como en inglés.

#### ***Objetivo general de aprendizaje***

Aplicar conceptos, métodos, técnicas y buenas prácticas de análisis de requerimientos, diseño detallado de software y control de versiones, en la ingeniería de software de proyectos sencillos, que se deberán trabajar en grupos pequeños, y que impliquen

interactuar con usuarios reales, o de problemas de contextos reales.

### Objetivos Terminales:

(OT1) Aplicar, en el contexto de un proyecto sencillo de ingeniería de software, la administración de la configuración como actividad transversal fundamental en cualquier proyecto de ingeniería de software.

(OT2) Elicitar, analizar, especificar, verificar y validar requerimientos funcionales de software usando el lenguaje natural y la notación del lenguaje de modelamiento unificado UML- diagramas de casos de uso.

(OT3) Realizar el diseño detallado de software, a partir del conjunto de requerimientos funcionales consolidados; en cuanto a la estructura, usando diagramas de clases, y en cuanto al comportamiento, con diagramas de secuencia, empleando la notación UML, y aplicando principios reconocidos de diseño.

### Objetivos terminales - Resultados de aprendizajes

Resultado de aprendizaje del curso o asignatura	Competencia en formación	Resultado de aprendizaje de la competencia de egreso al que se contribuye
OT1	SO1-Solución de problemas	SO1-PI2 Solución de problema (ingeniería/ciencias) (T)
OT2	SO1-Solución de problemas; SO3 - Comunicación efectiva	SO1-PI1 Identificación y formulación de problemas(T); SO3-PI1 Comunicación Escrita(T) SO3-PI2 Comunicación Oral(T)
OT3	SO1-Solución de problemas; SO2- Diseño	SO1-PI2 Solución de problemas (ingeniería y ciencias)(T); SO2-PI1 Diseño detallado(T)

### Unidades de aprendizaje

Al finalizar cada unidad el estudiante estará en capacidad de:

#### Unidad 1 - OT1: Administración de la configuración (4 semanas)

- Explicar la importancia de la administración de la configuración en los proyectos de ingeniería de software.
- Identificar las tareas que se realizan en el proceso de administración de la configuración y versiones del software.
- Realizar control de cambios y de versiones en el código fuente de proyectos de software.

#### Unidad 2 - OT2: Requerimientos funcionales de software (6 semanas)

- Explicar técnicas y métodos para elicitar, analizar y especificar requerimientos funcionales.
- Elicitar requerimientos funcionales para un sistema de software interactuando con un "cliente" real, cuidando la correcta interpretación en la interacción oral.

- Analizar requerimientos funcionales aplicando el método propuesto por Merlin Dorfman, cuidando que la redacción de los requerimientos en lenguaje natural sea correcta.
- Especificar requerimientos funcionales usando casos de uso (diagramas UML y formatos bicolunares, cuidando que en éstos, la redacción en lenguaje natural sea correcta).
- Verificar los requerimientos funcionales especificados: cuantificabilidad, verificabilidad, satisfacibilidad.

### **Unidad 3 - OT3: Diseño detallado (estructura y comportamiento) de software empleando UML (6 semanas)**

- Reconocer y explicar principios generales de diseño de software.
- Realizar el diseño detallado de la estructura del software satisfaciendo los requerimientos analizados y validados, usando correctamente los elementos notacionales (sintaxis y semántica) de los diagramas de clase UML.
- Realizar el diseño detallado del comportamiento del software satisfaciendo los requerimientos analizados y validados, usando correctamente los elementos notacionales (sintaxis y semántica) de los diagramas de secuencia UML.

### ***Metodologías de aprendizajes***

Los estudiantes deberán preparar, antes de la clase, los temas que se asigne. Bajo el esquema de trabajo de este curso, preparar un tema significa ESTUDIARLO siguiendo las recomendaciones dadas por el/la profesor(a) a través de las guías de preparación de los temas. Durante la clase, el estudiante deberá participar en las discusiones del tema, en la formulación de preguntas orientadas a resolver las dudas que hayan surgido al realizar las actividades de preparación del tema, y en la solución de ejercicios prácticos enfocados análisis de requerimientos, diseño detallado de software, control de versiones y aseguramiento de la calidad en proyectos de software. Después de cada clase, además de realizar los trabajos asignados por el/la profesor(a) para poner en práctica y afianzar conceptos, el estudiante deberá reflexionar sobre la relación del tema con temas anteriores de este y otros cursos y con su futuro desempeño profesional como Ingeniero de Sistemas.

Se definirán horarios de atención por fuera de clase para la solución de dudas, y canales de comunicación además del sistema Intu, para el acompañamiento oportuno durante el proceso de aprendizaje de los estudiantes.

Todos los diseños de sistemas de software que se realicen en el curso deben escribirse en inglés.

***Los estudiantes deberán instalar VisualParadigm con la licencia institucional a la que tienen derecho por ser estudiantes de la Universidad Icesi. No se aceptan entregas desarrolladas con otras herramientas o con una licencia de VisualParadigm que no sea la institucional asociada al usuario del estudiante.***

### ***Evaluación de aprendizajes***

Código evaluación	Mecanismo o actividad evaluativa	Porcentaje de la nota final	Relación con objetivos terminales - resultado de aprendizaje del curso	Relación con el resultado de aprendizaje de la competencia de egreso
Parcial 1	Parcial 1	20	OT1	SO1-PI2
Parcial 2	Parcial 2	20	OT2	SO1-PI1; SO3-PI1
Parcial 3	Parcial 3	20	OT3	SO1-PI2; SO2-PI1
Proyecto	Proyecto	30	OT2;OT3	SO1-PI1; SO1-PI2; SO2-PI1; SO3-PI1
Talleres	Talleres	10	OT1; OT2; OT3	SO1-PI1;SO1-PI2; SO2-PI1; SO3-PI1

### ***Recursos de apoyo***

No es necesario comprar ningún libro para el curso. Todo el material necesario para el curso será suministrado a través de Intu, como los materiales bibliográficos, guías de clases, videos, blogs, cursos en línea y otros.

Algunos libros, enlaces y herramientas relevantes:

- P. Bourque and R.E. Fairley, eds., Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society, 2014; [www.swebok.org](http://www.swebok.org).
- ISO/IEC 19505-2:2012. Information technology - Object Management Group Unified Modeling Language (OMG UML), Superstructure.
- Tutorial de diagramas de clases en VisualParadigm: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>
- The Scrum Guide <http://www.scrumguides.org/>
- R. Pressman. Ingeniería del Software: Un Enfoque Práctico. 7 Ed. McGraw Hill, 2010.