

## FORMATO 1

### PLANIFICACIÓN DE CURSO

Primer Semestre académico 2022 - Docencia Presencial

#### I. ACTIVIDAD CURRICULAR Y CARGA HORARIA

Asignatura:	Análisis y Diseño de Software	Código: COM3301-1
Semestre de la Carrera:	5° semestre	
Carrera:	Ingeniería Civil en Computación	
Escuela:	Ingeniería	
Docente(s):	Elizabeth Soto Gajardo	
Ayudante(s):		
Horario:	Lunes y miércoles 18:00 a 19:30	

Créditos SCT:	6
Carga horaria semestral <sup>1</sup> :	180 horas
Carga horaria semanal:	13 horas

Tiempo de trabajo sincrónico semanal:	4 horas
Tiempo de trabajo asincrónico semanal:	9 horas

#### II. RESULTADOS U OBJETIVOS DE APRENDIZAJE ESPERADOS ESTE SEMESTRE

1)	Conocer y aplicar técnicas para captar, formalizar y validar los requisitos de un proyecto de software.
2)	Conocer y describir atributos de calidad de software, y explicar su relación con los principios de diseño de software.
3)	Identificar una gama de soluciones para un proyecto de software, formalizando estos diseños usando los lenguajes y modelos adecuados.
4)	Evaluar críticamente y justificar diseños de software propuestos.

<sup>1</sup> Considere que 1 crédito SCT equivale a 30 horas de trabajo total (presencial/sincrónico y autónomo/asincrónico) en el semestre.

### III. UNIDADES, CONTENIDOS Y ACTIVIDADES

UNIDAD 1: Repaso de orientación a objetos				
Semana	Contenidos	Actividades de enseñanza y aprendizaje		Actividades de evaluación diagnóstica, formativa y/o sumativa
		Tiempo sincrónico	Tiempo asincrónico (trabajo autónomo del o la estudiante)	
1	<i>Presentación curso Metodología orientada a objetos. Repaso de conceptos claves.</i>	4	9	Lectura complementaria
2	<i>Metodologías de desarrollo del software. Ciclo de vida del software</i>	4	9	Lectura complementaria

UNIDAD 2: Análisis de software				
Semana	Contenidos	Actividades de enseñanza y aprendizaje		Actividades de evaluación diagnóstica, formativa y/o sumativa
		Tiempo sincrónico	Tiempo asincrónico (trabajo autónomo del o la estudiante)	
3	<i>Levantamiento de requerimientos Actores, glosario y otros. Artefactos. Eje práctico</i>	4	9	
4	<i>Casos de uso Historias de Usuario</i>	4	9	
5	<i>Modelo conceptual Análisis del dominio</i>	4	9	Tarea 1: Planteamiento del proyecto (04-05-2022)

6	<i>Diagramas de secuencia Comportamiento de los sistemas: contratos</i>	4	9	
7	<i>Diagramas de estado y Actividad.</i>	4	9	Control n°1 (18-05-2022)

**UNIDAD 3: Diseño de software**

Semana	Contenidos	Actividades de enseñanza y aprendizaje		Actividades de evaluación diagnóstica, formativa y/o sumativa
		Tiempo sincrónico	Tiempo asincrónico (trabajo autónomo del o la estudiante)	
8	<i>Del análisis al diseño</i>	4	9	Tarea 2: Análisis (01-06-2022)
9	<i>Diagrama de clase</i>	4	9	
10	<i>Diagramas de colaboración Diagrama de secuencia</i>	4	9	
11	<i>Principios de Diseño Patrones de Diseño</i>	4	9	
12	<i>Componentes y Framework</i>	4	9	Tarea 3: diseño detallado (29-06-2022)
13	<i>Generación de Prototipos</i>	4	9	Control n°2 (06-07-2022)

14	<i>Generación de Prototipos</i>	4	9	
15				Tarea 4: prototipo y presentación final (20-07-2022 sujeto a cambios)

#### IV. CONDICIONES Y POLÍTICAS DE EVALUACIÓN

Nota de Cátedra: Controles y examen

- Ser realizarán 2 controles.
- El examen pondera por 50% de la Nota de Cátedra
- Cualquier estudiante con nota mayor o igual a 5.0 está eximido de rendir el examen
- $NC = C1 * 25\% + C2 * 25\% + E * 50\%$
- El examen, aparte de ser una nota por sí sola, reemplaza la peor nota parcial.
- Si después del examen un alumno tiene nota final entre 3.7 y 3.9, tiene la opción de dar un examen recuperativo.

Nota de Actividades Complementarias: Entregas asociadas a un proyecto

- Trabajo grupal de 3 personas.
- Los estudiantes deberán desarrollar un proyecto de Software, realizando una entrega (informe y exposición) de las etapas del mismo: Presentación Proyecto (L1), Análisis (L2), Diseño (L3), Prototipo y Presentación Final(L4)
- $NAC = L1 * 20\% + L2 * 25\% + L3 * 25\% + L4 * 30\%$

Algoritmo para calcular Nota Final (NF)

- Si  $\text{minimo}(NC, NAC) \geq 4.0$
- ENTONCES  $NF = NC * 60\% + NAC * 40\%$
- SINO  $NF = \text{minimo}(NC, NAC)$

#### V. BIBLIOGRAFÍA Y RECURSOS OBLIGATORIOS

- Vélez Serrano, J. (2011). Diseñar y programar, todo es empezar: una introducción a la Programación Orientada a Objetos usando UML y Java.

## VI. BIBLIOGRAFÍA Y RECURSOS COMPLEMENTARIOS

- Craig Larman, *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and the Unified Process*, Prentice Hall, 2002.
- Erich Gamma, John Vlissides, Ralph Johnson, y Richard Helm, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.