

GUÍA DE APRENDIZAJE PARA EL ESTUDIANTE

I. Datos de Identificación General

Datos del curso

Nombre del Curso	Diseño de software				Código	Créditos	P	M	A	PMA Semestral	Semestre - Año	Año Versión
					INFO1156	6	3	2	5	170	1-2025	1
Curso con Ayudantía		Horas ayudantía		Nombre Ayudante	Prof Guido Mellado: Nombre - Email Prof: Ignacio Lincolao: Jorge Soto - jorge.soto2022@alu.uct.cl							

Datos del o la Docente (Repita este cuadro en función del número de docentes)

Nombre y Apellidos	Ignacio Lincolao Venegas Guido Mellado	Grado Académico	Ingeniero Civil en Informática Mención Tecnologías de Información, Licenciado en ciencias de la Ingeniería Doctor en ciencias mención biofísica y biología computacional
Fono Oficina		Email Institucional	gmellado@uct.cl ilincolao@uct.cl
Horario de Atención	Por definir	Unidad Académica a la que pertenece el curso	DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

II. Descripción del curso

Este curso se ubica en el primer semestre del segundo año y tributa a la línea de formación disciplinaria. Desarrolla la competencias genéricas de creatividad e innovación y trabajo colaborativo; así como la competencia específica gestión de las tecnologías de información. El tema central de este curso corresponde a una fase fundamental del ciclo de vida de un software, el diseño. Mediante el diseño se establece con precisión cómo traducir los modelos creados durante la fase de análisis (requerimientos, arquitectura del software, modelo de datos) en un producto realizable, confiable y re-utilizable.

III. Competencias a las que tributa el curso

Competencias Genéricas (Máximo dos competencias)

Nombre		Creatividad e Innovación	
Definición		Genera ideas, estrategias, métodos, procesos sustentables y socialmente responsables, para responder con productos o servicios originales a necesidades individuales y colectivas originadas en diversos entornos socioculturales, profesionales, laborales, científicos o tecnológicos.	
Nivel	1	Definición del nivel	Identifica soluciones novedosas para problemas o necesidades de variada naturaleza y contexto, y las adapta a otras necesidades, o bien plantear

Nombre		Trabajo colaborativo	
Definición		Demuestra participación activa en contextos de trabajo colaborativo desarrollando, movilizand o integrando recursos y habilidades, tanto individuales como grupales (intercambiar información, asumir responsabilidades, resolver dificultades) para lograr eficientemente los objetivos comunes para los cuales se ha constituido un equipo de trabajo.	
Nivel	1	Definición del nivel	Reconoce habilidades y necesidades personales disponibles para el desarrollo de trabajo colaborativo en aula, comprometiéndose individualmente en acciones vinculadas al logro de los objetivos planteados.

Competencias Específicas (Copiar y pegar tabla dependiendo del número de competencias específicas a las que tributa el curso)

Nombre		Desarrollo de Software	
Definición		Desarrolla soluciones de software apropiadas para uno o más dominios de aplicación, usando un enfoque de ingeniería de software que integra aspectos técnicos, éticos, sociales, legales, económicos e incluidos el aprendizaje del idioma inglés.	
Nivel	2	Definición del nivel	Analiza, diseña y prueba soluciones de software a problemas de mayor complejidad no alcanzables adecuadamente desde la perspectiva individual.

IV. Resultados de Aprendizaje (RA) (Señalar entre paréntesis qué CG y CE integra cada Resultado de Aprendizaje)

RA1: Diseña software de manera colaborativa, utilizando herramientas de modelado para resolver problemas de mediana complejidad, mediante técnicas y buenas prácticas aplicables al diseño.

RA2: Soluciona de manera creativa problemas aplicando patrones y estilos arquitectónicos para la construcción de software en entornos colaborativos.

V. Coherencia entre Resultados de Aprendizaje, contenidos asociados, evaluación y métodos de enseñanza y aprendizaje.





Resultados de Aprendizaje		Procedimientos evaluativos (Explicitar ponderación por cada procedimiento evaluativo)		Métodos de Enseñanza y aprendizaje / Técnicas didácticas
		E. Focalizadas de contenidos	E. Integradas de desempeño	
RA1: Diseña software de manera colaborativa, utilizando herramientas de modelado para resolver problemas de mediana complejidad, mediante técnicas y	Conceptuales: <ul style="list-style-type: none"> • Conceptos fundamentales del diseño de software. • Diseño arquitectónico y detallado. • Técnicas, métodos y herramientas de diseño. • Patrones de diseño y estilos arquitectónicos. • Principios de buenas prácticas y refactorización (SOLID). Procedimentales: <ul style="list-style-type: none"> • Aplicación de procedimientos de ingeniería 	Evaluaciones Focalizadas de Contenidos: - Ev1 (20% – Perfil de Proyecto): Entrega del plan de pruebas y documento de requerimientos. (Semana 6: 21/04-25/04).	Evaluaciones Integradas de Desempeño: - Ev2 (20% – Sprint 1): Presentación del avance del proyecto integrador, donde se evidencia la aplicación práctica de técnicas de	<ul style="list-style-type: none"> • Aprendizaje Basado en Proyectos y Casos: Trabajo en equipos para desarrollar y mejorar progresivamente el proyecto integrador. • Talleres Prácticos y Demostraciones: Sesiones de modelado UML, elaboración de diagramas y



FACULTAD DE INGENIERÍA - DEPARTAMENTO DE INGENIERÍA INFORMÁTICA




<p>buenas prácticas aplicables al diseño. RA2: Soluciona de manera creativa problemas aplicando patrones y estilos arquitectónicos para la construcción de software en entornos colaborativos.</p>	<p>inversa y directa en el análisis y modelado.</p> <ul style="list-style-type: none"> • Desarrollo de diseños arquitectónicos a través de diagramas UML (clases, secuencia, actividad). • Ejecución de técnicas y procedimientos de diseño en actividades prácticas y ejercicios grupales. • Aplicación práctica de patrones y estilos a través de casos reales y ejercicios de resolución de problemas. <p>Actitudinales:</p> <ul style="list-style-type: none"> • Capacidad de indagar y utilizar diversas fuentes de información. • Uso de estrategias propias y creatividad en la resolución de problemas. • Confianza, perseverancia y compromiso en el trabajo colaborativo. 	<p>- Ev Continúa (30% – Test de Certeza): Se compone de 4 test de 30 preguntas cada uno, aplicados en:</p> <ul style="list-style-type: none"> • Semana 3 • Semana 7 • Semana 13 • Semana 15 	<p>modelado y patrones de diseño. (Semana 11). - Ev3 (30% – Sprint 2 / Proyecto Final): Presentación final del proyecto integrador orientado a objeto, integrando todos los elementos aprendidos durante el curso. (Semana 16).</p>	<p>aplicación práctica de patrones de diseño.</p> <ul style="list-style-type: none"> • Metodología Scrum: Uso de backlog y revisiones quincenales para integrar avances y ajustar el proyecto. • Clases Teóricas y Lecturas Guiadas: Uso de textos como "Código Limpio" y "Arquitectura Limpia" para fundamentar la teoría y guiar la práctica. • Dinámicas Colaborativas: Ejercicios grupales y resolución de casos que fomenten el pensamiento crítico y la creatividad.
--	---	--	--	--



VI. Programación de actividades formativas (Repetir tabla para cada resultado de aprendizaje)



Semana Fecha	Actividades formativas (de enseñanza y aprendizaje y de evaluación)		
	Presenciales (3 horas)	Mixtas (2 horas)	Autónomas (5 horas)
RA1 RA2 Sem 1 17/03-21/03	- Presentación del curso y de la Guía de Aprendizaje. - Introducción a los principios de diseño y toma de requerimientos. - Formación de grupos y asignación de roles. - Revisión del proyecto base e identificación de problemas.	- Formación de grupos y discusión inicial sobre el proyecto. -Revisión del proyecto base, identificación de problemas.	📖 Obligatoria: - "Código Limpio": Prefacio e Introducción. -Preparar y llevar “resumen de proyecto” necesario para la semana 2 (¼ a ½ página Impresa) (obligatorio)
Sem 2 24/03-28/03	Clase expositiva sobre: Casos de uso, historias de usuario, diagramas de contexto (Condiciones, precondiciones flujo alterno. caso de uso con tablas). Ejemplos en clase. -Actividad “Modelado de requisitos para el proyecto” Parte 1: utilizando el “resumen de proyecto”.	- Continúan con la actividad “Modelado de requisitos para el proyecto” realizando su diagrama de caso de uso -Creación de diagramas de casos de uso para el proyecto.	📖 Obligatoria: - "Código Limpio": Capítulo 2

<p>Sem 3 31/03-04/04</p>	<p>- Demostración de herramientas UML y elaboración de diagramas (clases, secuencia y actividad).</p>	<p>-Evaluación CONTINUA 1: Test de certeza (40 min):</p> <ul style="list-style-type: none"> • Todo lo pasado en clases (Hasta la semana 2) • Lectura: <ol style="list-style-type: none"> 1. "Código Limpio": Capítulo 2 <p>- Trabajando en el proyecto creando sus diagramas de clases, secuencia y actividad</p> <p>- Ejercicios guiados de modelado en grupos. programado.</p> <p>- Modelado de la estructura y flujo del proyecto</p>	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Código Limpio": Capítulo 3 (Funciones).
<p>Sem 4 07/04-11/04</p>	<p>- Sesión de mentorías y asesorías para la revisión de avances:</p> <ul style="list-style-type: none"> - Backlog - Perfil de proyectos - Herramientas utilizadas - Diagramas - Horas trabajadas - etc.. 	<p>- Sesión de mentorías y asesorías para la revisión de avances:</p> <ul style="list-style-type: none"> - Backlog - Perfil de proyectos - Herramientas utilizadas - Diagramas - Horas trabajadas - etc.. 	<p> Opcional: Repaso de los Capítulos 1–3 de "Código Limpio".</p> <p> Obligatoria:</p> <ul style="list-style-type: none"> - "Código Limpio": Capítulo 4 (*Comentarios*).
<p>Sem 5 14/04-18/04</p>	<p>- Clase expositiva sobre Principios SOLID y buenas prácticas en el diseño.</p> <p>- Análisis de ejemplos prácticos.</p> <p>- Discusión sobre buenas prácticas en el diseño de software.</p> <p>-Actividad grupal - SOLID</p>	<p>- Aplicación de SOLID en el proyecto</p>	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Arquitectura Limpia": Capítulo 7 (SRP: El principio de responsabilidad única). - "Arquitectura Limpia"Capítulo 8. (OCP: EL PRINCIPIO ABIERTO-CERRADO)

<p>Sem 6</p> <p>21/04-25/04</p>	<p>Presentaciones de Evaluación 1:</p> <ul style="list-style-type: none"> Plan de Pruebas Detalle de Requerimientos del Proyecto Antiguo PLANTILLA A PARTIR DEL 8030 PPT PRESENTACIÓN PERFIL DE PROYECTO 	<p>Presentaciones de Evaluación 1:</p> <ul style="list-style-type: none"> Plan de Pruebas Detalle de Requerimientos del Proyecto Antiguo PLANTILLA A PARTIR DEL 8030 PPT PRESENTACIÓN PERFIL DE PROYECTO 	<ul style="list-style-type: none"> Los estudiantes se preparan para realizar la presentación de la evaluación 1.
<p>Sem 7</p> <p>28/04-02/05</p>	<ul style="list-style-type: none"> Clase sobre Patrones de Diseño – Parte 1: Introducción a patrones creacionales y estructurales. Demostración de ejemplos prácticos. 	<p>- Evaluación CONTINUA 2 - Test de certeza contenido (40 min):</p> <ul style="list-style-type: none"> Test de certeza anterior Contenidos en clases (hasta la semana 5) Lecturas indicadas: <ol style="list-style-type: none"> "Código Limpio": Capítulo 3 (Funciones). "Código Limpio": Capítulo 4 (*Comentarios*). "Arquitectura Limpia": Capítulo 7 (SRP: El principio de responsabilidad única). "Arquitectura Limpia": Capítulo 8. (OCP: EL PRINCIPIO ABIERTO- CERRADO) <p>- Aplicación de patrones en el proyecto</p>	<p> Obligatoria:</p> <ul style="list-style-type: none"> "Arquitectura Limpia": Capítulo 10 (ISP: El principio de segregación de interfaces). "Arquitectura Limpia": Capítulo 11 (DIP: LA INVERSIÓN DE LA DEPENDENCIA PRINCIPIO).
<p>Sem 8</p> <p>05/05-09/05</p>	<ul style="list-style-type: none"> Sesión expositiva sobre Patrones de Diseño – Parte 2: Patrones de comportamiento. Ejercicios de aplicación en el proyecto. 	<p>- Actividad grupal - Diagramas con patrones: elaborar diagramas y casos de uso que integren patrones de comportamiento en el proyecto</p>	<p> Obligatoria:</p> <ul style="list-style-type: none"> "Código Limpio": Capítulo 5 (*Formato y estructura de código*). "Arquitectura Limpia": Capítulo 12 (Componentes y Cohesión).

<p>Sem 9</p> <p>12/05-16/05</p>	<ul style="list-style-type: none"> - Clase sobre Diseño Basado en Componentes y Modularidad. - Demostración práctica de modularización del código. Alta cohesión y bajo acoplamiento 	<ul style="list-style-type: none"> - Modularización del código en el proyecto 	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Código Limpio" Capítulos 6-7 (*Objetos y clases, manejo de errores*). - Lectura: "Arquitectura Limpia": Capítulo 14 (Acoplamiento de componentes).
<p>Sem 10</p> <p>19/05-23/05</p>	Semana de Receso	Semana de Receso	Semana de Receso
<p>Sem 11</p> <p>26/05-30/05</p>	<p>Evaluación 2 - Presentación proyectos con los elementos:</p> <ul style="list-style-type: none"> - SOLID - Patrones de diseño creacionales - Patrones de diseño estructurales - Patrones de comportamiento - Diseño basado en componentes y modularidad - Backlog + PPT 	<p>Evaluación 2 - Presentación proyectos con los elementos:</p> <ul style="list-style-type: none"> - SOLID - Patrones de diseño creacionales - Patrones de diseño estructurales - Patrones de comportamiento - Diseño basado en componentes y modularidad - Backlog + PPT 	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Código Limpio": Capítulo 8 (*Límites en el código*).
<p>Sem 12</p> <p>02/06-06/06</p>	<ul style="list-style-type: none"> -Enfoques en capas, arquitectura hexagonal y aspectos sobre endpoints/APIs. - Ejemplos de rediseño y definición de interfaces. <p>Evaluación de métricas de código, code smells y refactorización</p>	<p>- Trabajo en grupal - API: proponer un rediseño arquitectónico que incluya el diseño de endpoints/APIs, al proyecto.</p>	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Arquitectura Limpia": Capítulo 15 (¿Qué es la arquitectura?), complementado con ejemplos de diseño de APIs.

<p>Sem 13</p> <p>09/06-13/06</p>	<p>Evaluación de métricas de código, code smells y refactorización</p>	<p>- Evaluación CONTINUA 3 - Test de certeza</p> <p>Contenido (40 min):</p> <ul style="list-style-type: none"> • Test de certeza anterior • Contenidos en clases (Hasta la semana 12) • Lecturas indicadas: <ol style="list-style-type: none"> 1. "Código Limpio": Capítulo 5 (*Formato y estructura de código*). 2. "Código Limpio": Capítulos 6-7 (*Objetos y clases, manejo de errores*). 3. "Código Limpio": Capítulo 8 (*Límites en el código*). 4. "Arquitectura Limpia": Capítulo 10 (ISP: El principio de segregación de interfaces). 5. "Arquitectura Limpia": Capítulo 11:(DIP: LA INVERSIÓN DE LA DEPENDENCIA PRINCIPIO) 6. "Arquitectura Limpia": Capítulo 12 (Componentes y Cohesión). 7. "Arquitectura Limpia": Capítulo 14 (Acoplamiento de componentes). 8. "Arquitectura Limpia": Capítulo 15 (¿Qué es la arquitectura?), complementado con ejemplos de diseño de APIs. <p>-Herramientas de evaluación de diseño.</p>	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Código Limpio": Capítulos 9-10 (*Pruebas unitarias, sistemas y clases limpias*).
<p>Sem 14</p> <p>16/06-20/06</p>	<p>Introducción a TDD y BDD</p> <p>-Pruebas automatizadas</p> <p>-Actividad grupal - Pruebas Automatizadas</p>	<p>Implementación de pruebas en el proyecto</p> <p>-Pruebas automatizadas</p>	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Código Limpio": Capítulo 11

<p>Sem 15</p> <p>23/06-27/06</p>	<p>- Sesión para el inicio del Proyecto Final – Parte 1: Lineamientos para la documentación y planificación final.</p>	<p>- Evaluación CONTINUA 4 - Test de certeza</p> <p>Contenido (40 min):</p> <ul style="list-style-type: none"> • Test de certeza anterior • Contenidos en clases (Hasta la semana 14) • Nuevas Lecturas indicadas: <ol style="list-style-type: none"> 1. "Código Limpio": Capítulos 9-10 (*Pruebas unitarias, sistemas y clases limpias*). 2. "Código Limpio": Capítulo 11. <p>- Sesión para el inicio del Proyecto Final – Parte 2: Lineamientos para la documentación y planificación final.</p>	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Código Limpio": Capítulo 12
<p>Sem 16</p> <p>30/06-04/07</p>	<p>Evaluación 3 - Presentación proyectos con los elementos:</p> <ul style="list-style-type: none"> - Rediseño arquitectónico que incluya el diseño de endpoints/APIs, al proyecto. - Métricas de código - Code smells - Refactorización - Pruebas automatizadas - Backlog + PPT 	<p>Evaluación 3 - Presentación proyectos con los elementos:</p> <ul style="list-style-type: none"> - Rediseño arquitectónico que incluya el diseño de endpoints/APIs, al proyecto. - Métricas de código - Code smells - Refactorización - Pruebas automatizadas - Backlog + PPT 	<p> Obligatoria:</p> <ul style="list-style-type: none"> - "Arquitectura Limpia": Capítulo 16 (Enfoque en independencia y modularidad en el diseño).
<p>Sem 17</p>	<p>- Actividades presenciales para ajustes finales y exámenes.</p> <p>- Sesión de cierre y evaluación global del curso.</p>	<p>- Actividades presenciales para ajustes finales y exámenes.</p> <p>- Sesión de cierre y evaluación global del curso.</p>	<p>- Actividades presenciales para ajustes finales y exámenes.</p> <p>- Sesión de cierre y evaluación global del curso.</p>

VII. Bibliografía (Obligatoria (5 títulos como máximo) y complementaria (5 títulos como máximo))

Bibliografía Obligatoria	Bibliografía Complementaria
<ol style="list-style-type: none">1. Código limpio / Clean code: Manual de estilo para el desarrollo ágil de software. Robert C. Martin2. Arquitectura Limpia / clean architecture: A craftsman's Guide to Software Structure and Design. Robert C. Martin	<ol style="list-style-type: none">1. Software para desarrollo programación orientada a objetos2. Guías y apuntes elaborados por el profesor3. Material disponible en Internet sugerido por el profesor

VIII. Disposiciones o Lineamientos Reglamentarios

- El estudiante aprobará el curso cuando evidencie todos los resultados de aprendizaje y el Promedio ponderado de las instancias evaluativas de curso sea igual o superior a 4,0.
- **El curso se evaluará a través de 4 instancias:**
 - Ev1 (20%): Perfil de proyecto.
 - Ev2 (20%): Sprint 1.
 - Ev3 (30%): Sprint 2.
 - Ev Continúa (30%): Test de certeza.
- **Durante todo el curso, los estudiantes trabajarán en un proyecto que se irá mejorando e incorporando progresivamente, de acuerdo con el contenido enseñado a lo largo del semestre. La Ev1 y la Ev2 evaluarán avances parciales, y en la Ev3 se presentará el proyecto final, que integrará todos los elementos aprendidos, con especial énfasis en la orientación a objetos.**
- Además, deberán trabajar con Git y aplicar la metodología Scrum para integrar de manera sistemática los conocimientos adquiridos en el proyecto. Se requiere llevar un backlog de las tareas semanales, el cual se basará en la guía de aprendizaje y se actualizará de forma continua según los contenidos y las necesidades de cada etapa del proyecto.

IX. Anexos
