



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

## IIC2113 Diseño Detallado de Software (II/2016)

### Programa del Curso

<b>Profesor</b>	: Rodrigo Arturo Saffie Kattan
<b>Correo</b>	: <a href="mailto:rasaffie@ing.puc.cl">rasaffie@ing.puc.cl</a>
<b>Horario de clases</b>	: Martes y Jueves M3 (11:30 - 12:50)
<b>Horario de ayudantía</b>	: Viernes M6 (17:00 - 18:20)
<b>Sala de clases</b>	: Javier Pinto (DCC)
<b>Sala de ayudantías</b>	: E16
<b>Requisitos</b>	: IIC2143
<b>Ayudantes</b>	: <ul style="list-style-type: none"><li>- Nebil Kawas (<a href="mailto:nakawas@uc.cl">nakawas@uc.cl</a>)</li><li>- Patricio López (<a href="mailto:pelopez2@uc.cl">pelopez2@uc.cl</a>)</li><li>- Sebastián Salata (<a href="mailto:sasalata@uc.cl">sasalata@uc.cl</a>)</li></ul>

## 1. Objetivos

Al finalizar el curso, el estudiante estará capacitado para:

- Realizar diseño y programación orientados a objetos con pericia
- Evaluar la calidad de un diseño, utilizando criterios teóricos y prácticos
- Analizar software para mejorar su eficiencia, confiabilidad, y mantenibilidad
- Usar patrones de diseño en el desarrollo de software

## 2. Contenidos

La siguiente lista contiene algunos de los tópicos que se abordarán durante el curso:

### 1. Introducción

- a) ¿Qué es el diseño detallado de software?
- b) Modelo 4+1

### 2. Principios del diseño detallado de software

- a) Cohesión
- b) Acomplamiento

- c) Abstracción
- d) Ocultamiento
- e) SOLID

### 3. Diseño detallado aplicado a componentes

- a) Patrones de diseño
- b) *Frameworks*

### 4. Testing y refactoring

- a) Técnicas de testing
- b) *Code Smells*
- c) Técnicas de refactoring
- d) Métricas de calidad de código

### 5. Ingeniería reversa

## 3. Canales de comunicación

El curso contará con dos canales oficiales para comunicar información. Las notas y anuncios serán publicados a través de la página del curso en la plataforma SIDING. Todo lo demás relacionado con el curso se distribuirá a través de la comunidad de GitHub del curso.

## 4. Controles

El curso contará con controles ( $C$ ) periódicos sobre los contenidos del curso y/o lecturas.

## 5. Presentaciones

Los alumnos del curso deberá realizar 2 presentaciones ( $P$ ) sobre distintos temas relacionados con los contenidos del curso. Los temas a investigar serán anunciados con anticipación. Las presentaciones serán en horario de clases, en las siguientes fechas:

<b>Presentación 1</b>	jueves 13 de octubre
<b>Presentación 2</b>	martes 8 de noviembre

## 6. Tareas

El curso contará con 3 tareas ( $T$ ) a lo largo del semestre. Las fechas de publicación de los enunciados y entrega de las tareas son:

	<b>Publicación</b>	<b>Entrega</b>
<b>Tarea 1</b>	jueves 25 de agosto	viernes 9 de septiembre
<b>Tarea 2</b>	jueves 22 de septiembre	viernes 7 de octubre
<b>Tarea 3</b>	jueves 27 de octubre	viernes 11 de noviembre

## 7. Interrogaciones

El curso contará con 2 interrogaciones ( $I$ ) a lo largo del semestre. Estas serán en horario de clases en las siguientes fechas:

<b>Interrogación 1</b>	martes 13 de septiembre
<b>Interrogación 2</b>	martes 18 de octubre

## 8. Examen

El curso contará con un examen ( $E$ ) obligatorio. La fecha del examen es el día viernes 25 de noviembre, a las 8:30.

## 9. Evaluación

La nota final del curso ( $N'_f$ ) se calcula de la siguiente manera:

$$N'_f = 0,15 * \bar{C} + 0,15 * \bar{P} + 0,30 * \bar{T} + 0,20 * \bar{I} + 0,20 * E$$

Las condiciones para aprobar el curso son las siguientes:

$$\bar{T} \geq 3,95$$

$$E \geq 2,95$$

$$N'_f \geq 3,95$$

Considerando lo anterior, la nota final será:

$$N_f = \begin{cases} N'_f & \text{si se cumplen todos los requisitos} \\ \min\{3,9, N'_f\} & \text{en caso contrario} \end{cases}$$

## 10. Bibliografía

- FOWLER, M., ET AL., *Patterns of Enterprise Application Architecture*, ADDISON-WESLEY, 2003.
- FOWLER, M., ET AL., *Refactoring: Improving the Design of Existing Code*, ADDISON-WESLEY, 2000.
- FOWLER, M., ET AL., *UML Distilled, (3<sup>d</sup> ed.)*, ADDISON-WESLEY, 2004.

- GAMMA, E., ET AL., *Design Patterns: Elements of Reusable Object-Oriented Software*, (1<sup>st</sup> ed.), ADDISON-WESLEY, 1994.
- MARTIN, R. C. *Clean Code: A Handbook of Agile Software Craftsmanship*, (1<sup>st</sup> ed.), PRENTICE HALL, 2008.
- PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*, (7<sup>th</sup> ed.), MCGRAW-HILL EDUCATION, 2009.

## 11. Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que **mantengan altos estándares de honestidad académica**. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería (disponible en SIDING).

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno, sin apoyo en material de terceros. Por *trabajo* se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

**En particular, si un alumno copia un trabajo, o si a un alumno se le prueba que compró o intentó comprar un trabajo, obtendrá nota final 1.1 en el curso y se solicitará a la Dirección de Docencia de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.**

Por *copia* se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. **En caso que corresponda a copia a otros alumnos, la sanción anterior se aplicará a todos los involucrados.** En todos los casos, se informará a la Dirección de Docencia de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile. Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.