



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

## IIC2113 — Diseño Detallado de Software

### Programa de curso

Actualización: 11 de marzo de 2022

<b>Profesor</b>	Antonio Ossa Guerra ( <a href="mailto:aaossa@ing.puc.cl">aaossa@ing.puc.cl</a> )
<b>Horario de cátedras</b>	Viernes, módulos 4 y 5 (14:00-15:20 y 15:30-16:50)
<b>Horario de ayudantías</b>	Viernes, módulo 6 (17:00-18:20)
<b>Requisitos</b>	IIC2143 — Ingeniería de Software

## 1. Descripción

En la actualidad, nos encontramos en un mundo donde los dispositivos tecnológicos cada vez están más presentes en nuestras vidas. La capacidad de desarrollar software para interactuar con estos dispositivos se hace cada vez más popular, atractiva y hasta necesaria para la sociedad. En específico, la capacidad de desarrollar software de alta calidad es aún más valiosa para lograr desarrollar programas, paquetes, herramientas y sistemas de alta importancia y complejidad.

Este curso está orientado a estudiantes con conocimientos sobre técnicas para llevar a cabo un proyecto de desarrollo de software, y provee con nuevos conocimientos para diseñar e implementar sistemas de software siguiendo buenas prácticas de desarrollo, habilidad crítica para quien desarrolla software de alto nivel, ya sea en un contexto profesional o académico.

## 2. Objetivo General y Competencias

Este curso busca que sus estudiantes conozcan técnicas, prácticas y patrones que les permitan realizar software de alta calidad, que resuelva un problema y que perdure en el tiempo, ya sea en un contexto de desarrollo individual o colectivo.

Al finalizar el curso, cada estudiante será capaz de:

1. **Aplicar** técnicas y herramientas de construcción de software, incluyendo enfoques basados en estados y dirigidos por tablas para diseño de bajo nivel de software.
2. **Usar** patrones de diseño en la modelación de software.
3. **Realizar** diseño y programación orientados a objetos con pericia.
4. **Analizar** software para mejorar su eficiencia, confiabilidad y mantenibilidad.
5. **Evaluar** la calidad de un diseño, utilizando criterios teóricos y prácticos.
6. **Modificar** diseños usando enfoques rigurosos de control de cambios.
7. **Usar** técnicas de ingeniería inversa para recuperar el diseño de un producto de software.

### **3. Contenidos**

#### **1. Motivación**

- ¿Qué es el diseño detallado de software?
- Diseño en el proceso de desarrollo
- Principios fundamentales

#### **2. Diagramas UML**

- Modelo 4+1
- UML 2.0

#### **3. Buenas prácticas de desarrollo**

- SOLID
- Métricas de calidad
- Code Smells
- Refactoring
- Testing

#### **4. Patrones**

- Patrones de diseño
- Patrones de arquitectura

#### **5. Ingeniería inversa**

#### **6. Paradigmas de programación**

### **4. Metodología**

- Clases y ayudantías (expositivas y prácticas)
- Controles y actividades en clases
- Proyecto grupal

#### **4.1. Comunicación**

Para comunicar temas personales, cualquier estudiante puede contactar de forma directa y privada a cualquier persona del equipo docente. Si la persona contactada cree que el equipo docente puede hacer algo para ayudar y apoyar a quien le contacte, la situación y los pasos a seguir serán decididos siempre considerando la confidencialidad necesaria. Esta forma de contacto puede usarse para cualquier efecto que estime posible, ya sea como buscar ayuda o simplemente para expresar su situación.

Para enviar comentarios, sugerencias y opiniones al equipo docente, se habilitará un formulario que permanecerá abierto durante toda la duración del curso, el que tendrá la opción de ser respondido de forma anónima.

Adicionalmente, el profesor del curso contará con un horario de consultas, fuera del horario de clase, para atender de forma directa al estudiantado, ya sea de forma remota o presencial según disponibilidad y preferencia.

## 5. Evaluación

Las evaluaciones del curso se dividen en dos partes: una parte teórica, en forma de **controles**; y una parte práctica, en forma de **proyecto**.

### 5.1. Parte Teórica - Controles

Su objetivo es evaluar el aprendizaje conceptual acumulado de cada estudiante de manera individual, mediante una evaluación escrita. Esta evaluación tendrá un foco en los conceptos vistos durante las cátedras del curso, que forman parte de los contenidos listados en el programa. Cada trabajo será evaluado y se asignará una nota de evaluación y retroalimentación personalizada a cada estudiante. Se realizarán cuatro (4) durante el semestre, generando las siguientes notas:  $C_1$ ,  $C_2$ ,  $C_3$  y  $C_4$ . De estas, solo se considerarán las tres (3) mejores notas para el cálculo de la nota  $C$  de cátedra:

$$C = \frac{C_1 + C_2 + C_3 + C_4 - \min(C_1; C_2; C_3; C_4)}{3}$$

El ausentarse a una evaluación teórica implica directamente nota mínima en la evaluación correspondiente, a menos de que se justifique correctamente ante la Dirección de Pregrado. En este último caso, debe notificarse dicha justificación al equipo docente con la mayor anticipación posible, para resolver el caso particularmente.

### 5.2. Parte Práctica - Proyecto

Su objetivo es evaluar el aprendizaje práctico acumulado de cada estudiante de manera individual y grupal, mediante un proyecto aplicado. Esta evaluación tendrá como foco llevar a la práctica los conceptos vistos durante las cátedras y ayudantías del curso, que forman parte de los contenidos listados en el programa. Cada entrega del proyecto será acorde a los contenidos vistos hasta el momento en de publicación del enunciado. Cada entrega será evaluada y se asignará una nota de evaluación grupal, la que contará con una ponderación a nivel individual y retroalimentación a nivel grupal. Solo las dos primeras entregas serán evaluadas únicamente de forma individual, aunque requerirán coordinación dentro del grupo. A cada grupo se le asignará a una persona del equipo docente del curso, quien les guiará y evaluará durante las entregas del proyecto. Se realizarán cinco (5) entregas durante el semestre, generando las siguientes notas:  $E_1$ ,  $E_2$ ,  $E_3$ ,  $E_4$  y  $E_5$ , y la nota ponderada total de proyecto se calcula como:

$$P = \frac{E_1 + 2 \times E_2 + 3 \times E_3 + 3 \times E_4 + 2 \times E_5}{11}$$

### 5.3. Recorrección

Toda entrega del curso, tanto de la parte teórica como de la parte grupal, contará con un proceso de **recorrección**. Una vez se publiquen las notas de una evaluación, se abrirá un periodo de una semana para solicitar una corrección. Toda solicitud debe estar debidamente justificada y ser enviada por los canales que el curso disponga para este propósito.

En caso de que la respuesta a la solicitud de corrección no sea satisfactoria, se deberá llenar un formulario — dentro de una semana de publicada la corrección— para solicitar que el profesor revise el caso. La decisión que se tome en esta instancia es inapelable.

### 5.4. Calificaciones y aprobación

La nota del curso se calcula como  $NC = 0,4 \times C + 0,6 \times P$ . La nota final del curso **NF** se calculará como:

$$\mathbf{NF} = \begin{cases} \mathbf{NC} & \text{si } \mathbf{P} \geq 3,95 \text{ y } \mathbf{C} \geq 3,95 \\ \min(\mathbf{NC}; 3,9) & \text{en otro caso.} \end{cases}$$

Cada estudiante aprobará el curso si su nota final del curso **NF** es mayor a 3,95.

Todas las notas serán calculadas con **dos decimales**, salvo la nota final del curso que se calculará con **un decimal**.

### 5.5. Evaluación de última instancia

Se dará la oportunidad de rendir una evaluación oral el **penúltimo día hábil del semestre** a quienes se encuentren al borde de reprobación el curso. Cualquier estudiante podrá optar a realizar esta evaluación si su nota del curso **NC** es mayor o igual a 3.95, pero alguna de sus notas parciales (**C** o **P**) se encuentra entre 3.80 y 3.94 (ambos valores inclusive). El aprobar esta instancia permitirá calcular la nota final del curso como **NF** = 4,0.

## 6. Integridad académica

Cualquier situación de **copiar** detectada en alguna evaluación tendrá como **sanción un 1,1 final en el curso**. Esto sin perjuicio de sanciones posteriores que estén de acuerdo a la Política de Integridad Académica de la Escuela de Ingeniería y de la Universidad, que sean aplicables al caso. Rige para este curso tanto la política de integridad académica del Departamento de Ciencia de la Computación (ver anexo) como el [Código de honor de la Escuela de Ingeniería](#).

Debido a la naturaleza de la disciplina en la que se enmarca el curso, está permitido el uso de código escrito por una tercera parte, pero solo bajo ciertas condiciones. Primero que todo, el uso de código ajeno **siempre debe estar correctamente referenciado**, indicando la fuente de donde se obtuvo. Y por otro lado, se permite el uso de código encontrado en internet u otra fuente de información similar, siempre y cuando su autor sea **externo al curso**, o en su defecto, sea parte del **equipo docente** del curso. Es decir, se puede hacer referencia a código ajeno al curso y código perteneciente al curso pero solo aquel escrito por el equipo docente, como material o ayudantías. Luego, compartir o usar código de una evaluación **actual o pasada** se considera una falta a la ética.

## 7. Bibliografía

- R. Pressman. *Software Engineering: A Practitioner's Approach*.
- R. Martin. *Agile Principles, Patterns, and Practices in C#*.
- R. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*.
- R. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*.
- M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*.
- M. Fowler. *Refactoring: Improving the Design of Existing Code*.
- M. Fowler. *Patterns of Enterprise Application Architecture*.
- C. Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*.
- S. McConnell. *Code Complete: A Practical Handbook of Software Construction, Second Edition*.
- E. Gamma. *Design Patterns: Elements of Reusable Object-Oriented Software*.
- E. Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*.

## **Anexo: Política de integridad académica del Departamento de Ciencia de la Computación**

Se espera los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile mantengan altos estándares de honestidad académica, acorde al Código de Honor de la Universidad. 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 Pregrado de la Escuela de Ingeniería (Disponible en SIDING, en la sección Pregrado/Asuntos Estudiantiles/Reglamentos/Reglamentos en Ingeniería/Integridad Académica).

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 Pregrado 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 Pregrado 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 (<http://admisionyregistros.uc.cl/alumnos/informacion-academica/reglamentos-estudiantiles>). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.