



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

IIC2143 Ingeniería de Software (2018/I)

Programa del Curso - Sección 2

Profesor	: Jaime Navón Cohen
Correo	: jnavon@uc.cl
Horario de clases	: Lunes y Miércoles M3 (11:30 - 12:50)
Horario de ayudantía	: Viernes M3 (11:30 - 12:50)
Sala de clases	: B16
Sala de ayudantías	: B21
Requisitos	: IIC2233

1. Objetivos

Al finalizar el curso, el estudiante estará capacitado para:

- Llevar a cabo el desarrollo de un sistema de una manera metódica, comenzando por la formulación de los requisitos del sistema, desarrollando un diseño modular, refinando este diseño en una implementación que identifique y minimice los riesgos, codificando de manera que se pueda integrar con el trabajo de un equipo, y usando métodos para identificar y prevenir fallas.
- Desarrollar requisitos claros, concisos y precisos para el desarrollo de un nuevo producto de software (sistema), basados en las necesidades de los usuarios y otros interesados.
- Aplicar principios y patrones al diseñar un sistema y al evaluar el diseño de un sistema: abstracción, descomposición, ocultación de información, acoplamiento, cohesión, etc.
- Crear diagramas de clases en UML que modelen el dominio de un problema y la arquitectura de software de un sistema.
- Crear diagramas de secuencia, de estados, y de actividades en UML que modelen los casos de uso y, más en general, el comportamiento de un sistema.
- Aplicar técnicas de testing simples a distintos niveles de un producto de software; por ejemplo, escribir pruebas de caja negra básicas para clases y métodos.

2. Contenidos

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

1. Motivación

- la necesidad de ingeniería de software
- software como servicio
- desafíos y oportunidades

2. Proceso

- la necesidad de proceso
- modelo de cascada
- procesos iterativos: prototipos y RUP
- procesos incrementales
- métodos ágiles
- Scrum y Kanban

3. Requisitos

- funcionales y no funcionales
- casos de uso
- relatos de usuarios

4. Diseño

- modelo de dominio
- atributos de un buen diseño
- acoplamiento y cohesión
- diagramas UML de clases, secuencia y estados
- patrones de diseño

5. Arquitectura

- conceptos fundamentales
- atributos que impactan la arquitectura
- patrones arquitectónicos
- arquitecturas cliente servidor y multicapas
- arquitectura orientada a servicios
- microservicios

6. Gestión del Proyecto

- actividades de gestión
- estimaciones
- planeación de producto, release y sprint
- gestión de personas

7. Aseguramiento de Calidad (QA)

- definiciones de calidad
- prevención de defectos
- detección y eliminación de defectos
- testing

8. Métodos Estadísticos en Ingeniería de Software

- principales desafíos
- métricas para ingeniería de software
- aplicaciones de distribución binomial y geométrica
- aplicación de teorema de Bayes

3. Canales de comunicación

El curso contará con dos canales oficiales para comunicar información. Las notas y anuncios serán publicados a través de la plataforma SIDING. Todo lo demás relacionado con el curso se distribuirá a través de su [organización de GitHub](#).

4. Proyecto

A lo largo del semestre los estudiantes trabajarán en grupos de 3 personas para desarrollar una aplicación web. El proyecto (P) será evaluado periódicamente, con al menos 5 entregas parciales (EP), una entrega final (EF) y una presentación final (PF).

5. Interrogaciones

El curso contará con 3 interrogaciones (I) a lo largo del semestre.

6. Inasistencia Interrogación

La asistencia a todas las interrogaciones es obligatoria. Cualquier inasistencia a una evaluación debe ser justificada como lo explica la DIPRE. De no cumplir con este procedimiento se evaluará con nota mínima la interrogación faltante, la cual no podrá ser eliminada.

7. Examen

El curso contará con un examen (E) obligatorio. La nota del examen podrá reemplazar la peor nota obtenida en las interrogaciones si esto mejora el promedio final del alumno y si este no se ausentó de forma injustificada a alguna interrogación.

8. Evaluación

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

$$N'_f = 0,5 * P + 0,3 * \bar{I} + 0,2 * E$$

$$P = 0,5 * \bar{E}P + 0,2 * EF + 0,3 * PF$$

Las condiciones para aprobar el curso son las siguientes:

$$P \geq 3,95$$

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

$$E \geq 2,95$$

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

Considerando lo anterior, la nota final será:

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

9. Bibliografía

- D. BELL, *Software Engineering for Students*, (4th ed.), ADDISON-WESLEY, 2005.
- P. JALOTE, *Concise Introduction to Software Engineering*, SPRINGER-VERLAG, 2008.
- J. LOELIGER AND M. MCCULLOUGH, *Version Control with GIT* (2nd ed.), O'REILLY 2012.
- K. RUBIN, *Essential Scrum* ADDISON-WESLEY 2013.
- A. PHAM AND P. PHAM, *Scrum in Action*, COURSE TECHNOLOGY PTR 2012.
- R. MILES AND K. HAMILTON, *Learning UML 2.0*, O'REILLY 2006.
- C. LASATER, *Design Patterns*, WORDWARE PUBLISHING 2007.
- M. HARTL, *Ruby on Rails Tutorial* (3rd ed.), ADDISON WESLEY 2015.
- R. OLSEN, *Design Patterns in Ruby*, ADDISON WESLEY 2007.
- D. CHELIMSKY, *The RSpec Book*, PRAGMATIC BOOKSHELF 2012.
- M. WYNNE AND A. HELLESoy, *The Cucumber Book*, PRAGMATIC BOOKSHELF 2012.

10. 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.