



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

IIC2143 Ingeniería de Software (2021/1) **Programa de Curso**

Horario de clases : Lunes y Miércoles Módulo 3
Horario de ayudantía : Viernes Módulo 3
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, mas 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

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. Gestión del Proyecto

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

5. Diseño

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

6. Arquitectura

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

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 CANVAS. Todo lo demás relacionado con el curso se distribuirá a través de su [organizacion de GitHub](#).

4. Proyecto

A lo largo del semestre los estudiantes trabajarán en grupo 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 2 interrogaciones (I) y un examen final (Ex) a lo largo del semestre. No hay posibilidad de eximirse del examen final.

6. Inasistencia a una 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. En el caso en que el alumno haya rendido la interrogación, tendrá derecho a sustituirla por la nota del examen solo si su nota ha sido superior a un 3.

7. Examen

El curso contará con un examen (E) obligatorio. La nota del examen podrá reemplazar la peor nota obtenida en una de las interrogaciones si dicha mala nota es a lo menos un 3. Si el alumno falta a una interrogación por enfermedad y lo justifica adecuadamente puede reemplazar la nota de esa interrogación por la nota del examen.

8. 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. Chelmsky, The RSpec Book, Pragmatic Bookshelf 2012.
- M. Wynne and A. Hellesoy, The Cucumber Book, Pragmatic Bookshelf 2012.

10. Política de Integridad Académica

COMPROMISO DEL CODIGO DE HONOR

Este curso adscribe el Código de Honor establecido por la Universidad, el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso de que exista colaboración permitida con otros alumnos, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como alumno es su deber conocer el Código de Honor (www.uc.cl/codigodehonor).

POLÍTICA DE INTEGRIDAD ACADÉMICA DEL DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

Se espera que 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.