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

IIC2113 - Diseño Detallado de Software (II/2020)

Programa

Profesora	M. Fernanda Sepúlveda
Correo	mfsepulveda@uc.cl
Horario del curso	Viernes módulos 4 y 5 (14:00-15:20 y 15:30-16:50)
Horario Ayudantía	Viernes módulo 6 (17:00-18:20)
Requisitos	IIC2143 - Ingeniería de Software

Objetivos

- Aplicar técnicas y herramientas de construcción de software, incluyendo enfoques basados en estados y dirigidos por tablas para diseño de bajo nivel de software.
- Usar patrones de diseño en la modelación de software.
- Realizar diseño y programación orientados a objetos con pericia.
- Analizar software para mejorar su eficiencia, confiabilidad, y mantenibilidad.
- Modificar diseños usando enfoques rigurosos de control de cambios.
- Usar técnicas de ingeniería inversa para recuperar el diseño de un producto de software.

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

Metodología

- Clases expositivas y prácticas
- Ejercicios en clases
- Prueba teórica y evaluación posterior práctica
- Proyecto grupal

Canales de comunicación

- Clases vía Zoom. Canal privado.
- Entrega de evaluaciones teóricas vía Canvas UC.
- Entrega de evaluaciones prácticas vía Github (en repositorio asignado al grupo).
- Foro de conversaciones, preguntas y discusiones a través del Repositorio de Recursos en Github.

Evaluación

El curso se divide de una parte teórica y una parte práctica que se deben aprobar por separado.

Evaluación teórica

Habrán dos interrogaciones y un exámen (eximible). Cada interrogación constará de dos partes:

- Una parte escrita que se resolverá durante medio modulo.
- Un ejercicio práctico que se podrá resolver de forma asíncrona y podrá entregarse durante las 24 horas siguientes. Podrá ser resuelto en Ruby o en C#. Al menos una de las evaluaciones que haga el alumno debe ser en C#.

Evaluación	**Fecha**
I1	Viernes 11 de Septiembre
I2	Viernes 13 de Noviembre
EX	Martes 15 de Diciembre (15:30)

Evaluación práctica

Existirá un proyecto con 3 entregables. El objetivo del proyecto es aplicar el contenido estudiado en el curso y fomentar el trabajo colaborativo entre los alumnos.

Criterio de aprobación

La cátedra (NC) y el proyecto (NE) se aprueban por separado, con nota igual o superior a $3,95$.

NP : Nota de Presentación. Se calcula como el promedio simple entre las interrogaciones:

$$NP = \frac{(I1 + I2)}{2}$$

NEX : Nota considerada del exámen. El exámen es eximible con nota NP superior o igual a $5,4$.

$$NEX = \begin{cases} NP & \text{ssi } NP >= 5,4 \\ max(NP, EX) & \text{ssi } NP < 5,4 \end{cases}$$

NC : Nota de cátedra. Se calcula como sigue:

$$NC = \frac{(I1 + I2 + 2 * EX)}{4}$$

NE : Nota del proyecto. Se calcula como una suma de las ponderaciones de cada entrega.

$$NE = 0,2 * E1 + 0,4 * E2 + 0,4 * E3$$

NF : La nota final se calcula de la siguiente manera

$$NF = \begin{cases} \frac{(NC+NE)}{2} & \text{ssi } NC >= 3,95 \text{ y } NE >= 3,95 \\ min(NC, NE) & \text{ssi } NC < 3,94 \text{ y } NE < 3,94 \end{cases}$$

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 3Complexity in the Heart of Software.

Política de Integridad Académica

Este curso se adscribe a la política de integridad académica de la Escuela de Ingeniería y el Departamento de Computación.

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 y sea autorizado por los ayudantes.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile1. Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

Reglamento del Alumno de la Pontificia Universidad Católica de Chile disponible en: Reglamentos Estudiantiles.