



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

## Proyecto de Especialidad – IIC2154 (2021-1)

### Programa de Curso

**Profesor:** Andrés Neyem, Luis González y Felipe Rojos  
**E-mail:** aneyem@ing.puc.cl  
**Sitio Web:** <https://intrawww.ing.puc.cl/siding/>  
**Horario de clases:** Sección 1  
Cátedra: Jueves módulo 2 (10:00 – 11:20) y Jueves módulo 3 (11:30 - 12:50)  
Cátedra/Ayudantía/Tutoría: Jueves módulo 4 (14:00 a 15:20)  
  
Sección 2 y 3  
Cátedra: Jueves módulo 5 (15:30 – 16:50) y Jueves módulo 6 (17:00 - 18:20)  
Cátedra/Ayudantía/Tutoría/: Jueves módulo 4 (14:00 a 15:20)  
  
**Sala de clases:** Zoom / Google Meet

## 1. Descripción

El objetivo del curso es proporcionar una experiencia de culminación (Capstone) que integre el conocimiento adquirido en los cursos anteriores del plan de estudios a través del trabajo en un proyecto, con retos y limitaciones realistas, para un cliente (stakeholder) real. Los proyectos son elegidos de propuestas realizadas por diferentes tipos de organizaciones, a fin de proporcionar una fuente valiosa de aprendizaje para los estudiantes y también importantes beneficios para las organizaciones clientes.

El curso hace énfasis en el trabajo en equipo en grupos pequeños de estudiantes, siguiendo las mejores prácticas y pautas sobre la forma de gestionar proyectos de desarrollo de software, tales como las que están siendo llevadas a cabo por los equipos. Los miembros del equipo tienen uno o más roles, y cada miembro del equipo contribuirá al análisis y definición de requerimientos, diseño funcional, implementación de funciones, integración, y otros aspectos de acuerdo a las necesidades y las fases del proyecto.

## 2. Contenido

1. Planificación y control de proyectos
  - Planificación de un proyecto
  - Dimensionamiento de recursos

- Control y monitoreo de un proyecto de software
  - Ambientes de desarrollo, pruebas y producción
  - Herramientas de apoyo, frameworks de diseño
- 2. Coordinación basada en estructura de roles
- 3. Análisis de requisitos
- 4. Testeo de software
- 5. Calidad en el desarrollo de software
- 6. Administración de la configuración
  - Control de versiones de software
  - Herramientas de apoyo

### 3. Logros del Aprendizaje

1. Los estudiantes serán capaces de trabajar en grupos para especificar los requisitos de un producto de software deseado por un stakeholder real.
2. Los estudiantes serán capaces de trabajar en grupos para diseñar un producto de software deseado por un stakeholder real.
3. Los estudiantes serán capaces de trabajar en grupos para implementar un producto de software deseado por un stakeholder real.
4. Los estudiantes serán capaces de trabajar en grupos para testear un producto de software deseado por un stakeholder real.
5. Los estudiantes serán capaces de trabajar en grupos para desarrollar las habilidades de gestión de proyectos, tales como la planificación, programación, seguimiento y trabajo en equipo, presente en un proyecto de software real.
6. Los estudiantes serán capaces de trabajar en grupos para mejorar la capacidad de comunicarse de manera efectiva a nivel oral (por ejemplo, reuniones semanales con los stakeholders y el equipo académico, y presentación final oral para un proyecto).

### 4. Metodología

Por ser un proyecto de especialidad, el curso se basa fuertemente en una experiencia Capstone, lo que se materializa en un proyecto grupal que deben hacer los alumnos. Al inicio del curso se imparten clases expositivas, dos veces por semanas, con el fin de facilitar la planificación, desarrollo y control del proyecto. Posteriormente se divide el trabajo en tres iteraciones (equivalente cada una aproximadamente a un mes de trabajo) en el cual se definen las fechas de presentación de avance del proyecto.

Los alumnos envían su Curriculum y el profesor del curso arma los equipos y se les asigna un proyecto perteneciente a una organización real. Cada equipo es apoyado internamente por una persona con varios años de experiencia en proyectos de desarrollo de software quién asume el rol de tutor. Cada semana se evaluará el avance de cada proyecto en las reuniones de Seguimiento y Control, evaluando el desempeño de los alumnos en los roles asignados, evaluando la calidad de la planificación, control de avance y gestión de riesgos, lo que conformará una fracción de la nota de la iteración. Al finalizar la iteración se realizará una presentación ante el curso y los clientes, donde se evaluará la claridad de los conceptos, calidad de los presentadores y en especial la demostración práctica de los avances en el producto desarrollado. Se realizará además una co-evaluación en donde cada miembro del equipo

evaluará al resto, y se considerará la opinión del cliente acerca del proceso realizado y del producto obtenido.

Debido a que es un curso destinado a alumnos que se encuentran cerca de terminar su carrera, se espera que los alumnos tengan una actitud profesional al desempeñar el proyecto. Se recomienda fuertemente que cuente con las siguientes habilidades y conocimientos: Programación (todos los miembros del equipo DEBEN programar), ingeniería de software, base de datos, tecnologías del WWW, realizar investigación para encontrar una solución a un problema técnico, compromiso y responsabilidad.

## 5. Evaluación

En este curso se busca evaluar el aprendizaje de las materias relacionadas a las Ciencias de la Computación y su aplicación práctica en el análisis de requisitos, diseño, implementación y testeo de un proyecto real de desarrollo de software, así como el trabajo e inmersión del alumno en un equipo de desarrollo, sus habilidades de gestión de un proyecto y capacidad de comunicarse de manera efectiva a nivel oral.

El sistema de evaluación consta de la evaluación de proceso (grupal e individual), evaluación de producto (grupal) y presentación final de ciclo (grupal). Estas evaluaciones se llevan a cabo en cada iteración con diferentes ponderaciones y al final de cada iteración al alumno se le asigna una nota. La nota final (NF) se calcula como el promedio de las notas individuales (NI) obtenidas en cada iteración:  $NF = \text{Min}[(NI1 + NI2 + NI3) / 3; 7]$ . Aprobación del Curso:  $NF \geq 4$ .

## 6. Bibliografía

1. Adkins, L., Coaching agile teams: a companion for ScrumMasters, agile coaches, and project managers in transition. Addison-Wesley Professional, 2010.
2. Cohn, M., Agile Estimating and Planning. Prentice Hall, 2005.
3. Crispin, L., & Gregory, J., Agile testing: A practical guide for testers and agile teams. Pearson Education, 2009.
4. ESA Software Engineering Standard, Issue 2, European Space Agency, 1991.
5. Larman, C., Agile and iterative development: a manager's guide. Addison-Wesley Professional, 2004.
6. McConnell, S., Code Complete: A Practical Handbook Of Software Construction. Microsoft Press, 2004.
7. Neyem, A., Benedetto J., & Chacon, A., Improving Software Engineering Education through an Empirical Approach: Lessons Learned from Capstone Teaching Experiences. ACM Technical Symposium on Computer Science Education, 2014.
8. Pressman, Software Engineering: A Practitioner's Approach, 2004.
9. Sommerville, I., Software Engineering, (9th Edition). Addison-Wesley Professional, 2010.