



Programa de Asignatura

I. IDENTIFICACIÓN				
Carrera o programa: Ingeniería Civil en Computación e Informática				
Unidad responsable: Escuela de Ingeniería				
Nombre de la asignatura: Proyecto Integrador Software				
Código: ECIN-00619				
Semestre en la malla ¹ : 6				
Créditos SCT - Chile: 5				
Ciclo de Formación	Básico		Profesional	X
Tipo de Asignatura	Obligatoria	X	Electiva	
Clasificación de área de conocimiento ²				
Área: Ingeniería y Tecnología		Sub área: Ingeniería Informática		
Requisitos:				
Pre-requisitos:		Requisito para:		
• ECIN-00513 Proyecto Integrador Programación Avanzada		• ECIN-00708 Proyecto Integrador Plataformas		

II. ORGANIZACIÓN SEMESTRAL							
Horas Dedicación Semanal (Cronológicas)	Docencia Directa	3	Trabajo Autónomo	4.5	Total	7.5	
Detalle Horas Directas	Cátedra	Ayudantía	Laboratorio	Taller	Terreno	Exp. Clínica	Supervisión
			3				

¹Este campo

²Clasificación del curso de acuerdo a la OCDE



III. APOORTE AL PERFIL DE EGRESO

La asignatura contribuye al dominio 1 del perfil de egreso, “Conocimiento científico y disciplinario”. Además, contribuye al dominio 4 “Habilidades para la Práctica de la Ingeniería”. Al finalizar la asignatura las y los estudiantes serán capaces de planificar, administrar y ejecutar un proyecto de software, integrando herramientas y mejores prácticas tanto de diferentes modelos de desarrollo de software como del ciclo de vida de desarrollo para crear productos proactivos a los ataques maliciosos. El proyecto de software considera el desarrollo desde cero o la mantención de un producto ya existente.

IV. HABILIDADES PERFIL DE EGRESO (RELACIÓN)

1.3 Aplicar conocimientos, métodos y herramientas de la especialidad para resolver problemas complejos de Ingeniería de Software, Plataformas y Gestión de Tecnologías.

4.3 Concebir las soluciones TIC requeridas en las organizaciones haciendo uso eficiente de los recursos como personas, información, y procesos. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.

4.4 Diseñar las soluciones TIC para la industria intensiva en procesamiento de la información. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.

4.5 Implementar las soluciones TIC. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.

CG7 Uso eficiente de las tecnologías: Competencias informacionales (CI2) que permiten a la persona reconocer cuando se requiere información, como acceder a ella, como evaluarla, seleccionarla y utilizarla de manera responsable, haciendo uso de las TIC. El egresado UCN, genera, transmite y difunde conocimiento con transparencia y honestidad en coherencia con el valor de la verdad. El sentido ético que lo mueve es no poner en riesgo a la “persona en su dignidad”.



V. RESULTADOS DE APRENDIZAJE

1. Integrar las buenas prácticas para el desarrollo de software seguro en los procesos de producción de software.
2. Aplicar las técnicas y modelos para la mantención de sistemas de software
3. Gestionar un proyecto de desarrollo de software, incorporando actividades de gestión de calidad
4. Ejecutar un proyecto de desarrollo de software utilizando un modelo de proceso iterativo e incremental.
5. Identificar los objetivos y requerimientos de las soluciones TIC
6. Seleccionar los procesos, técnicas y herramientas adecuados de acuerdo a los requerimientos.
7. Desarrollar la solución tecnológica más adecuada en base a las características del problema y los recursos disponibles.
8. Aplicar herramientas TIC para el almacenamiento, la comunicación, la transmisión e intercambio de información de manera efectiva.

VI. ÁREAS TEMÁTICAS

1. Evolución de Software
 - 1.1 Fundamentos
 - 1.2 Problemas claves de la mantención
 - 1.3 Proceso de mantención
 - 1.4 Técnicas para mantención
2. Desarrollo de Software Seguro
 - 2.1 Requisitos de seguridad
 - 2.2 Ciclo de vida del desarrollo de software seguro
 - 2.3 Vulnerabilidades y amenazas
 - 2.4 Patrones de Ataques
3. Gestión de Configuración
 - 3.1 Administración del proceso de manejo de la configuración
 - 3.2 Ciclo de vida de la configuración
 - 3.3 Auditoría



3.4 Reportabilidad

3.5 Administración de la liberación de versiones

4. Gestión de Calidad

4.1 Fundamentos de Calidad del Software

4.2 Procesos de Gestión de Calidad del Software

4.3 Modelos de Calidad

4.4 Consideraciones prácticas

5. Gestión Proyectos Software

5.1 Iniciación y definición de alcance

5.2 Estimación de tiempo y costo para un proyecto de software

5.3 Planificación de proyectos de software

5.4 Promulgación de proyectos de software

5.5 Revisión y evaluación

5.6 Cierre

VII. ORIENTACIONES METODOLÓGICAS

1. Los y las estudiantes se aproximarán gradualmente a los resultados de aprendizaje mediante el desarrollo de un proyecto complejo, trabajando en equipos autogestionados con la guía del equipo de profesores y ayudantes. El énfasis está en el fortalecimiento las habilidades técnicas para el ejercicio de la ingeniería integrando conocimientos previos con nuevos aprendizajes necesarios para concebir y diseñar soluciones al problema central. Asimismo, a través del trabajo equipo, los y las integrantes fortalecerán sus habilidades comunicacionales.
2. Las actividades en aula serán esencialmente prácticas, en las clases los y las estudiantes trabajarán en sus proyectos con la guía y asistencia del profesor tanto en el diagnóstico y la definición del problema, como en la concepción y diseño de la solución. De manera complementaria se impartirán talleres introductorios con herramientas técnicas que apoyarán el desarrollo de los proyectos, las que ampliarán el ámbito soluciones que se pueden adoptar.
3. Cada equipo es responsable de la integración de conocimientos previos, así como los nuevos aprendizajes que sean pertinentes a la solución concebida y su posterior diseño.



VII. ORIENTACIONES Y CRITERIOS PARA LA EVALUACIÓN

1. La evaluación, tanto las formativas como las sumativas, se basarán en un conjunto de entregables que den cuenta del desarrollo del proyecto, en sus fases de diagnóstico, concepción y diseño. Cada entregable debe abordar dos aspectos, por un lado, la definición del problema y/o la solución y, por otro, la justificación de las decisiones de diseño. Asimismo, los y las estudiantes comunicarán formalmente los resultados a través de la presentación y defensa de su propuesta. Finalmente, el profesor evaluará a cada estudiante a través de la observación del trabajo de los equipos de clases.
2. Ponderación de las evaluaciones:
 - (a) 25% Diagnóstico del problema.
 - (b) 15% Concepción de la solución.
 - (c) 15% Diseño primer prototipo.
 - (d) 25% Diseño segundo prototipo.
 - (e) 10% Plan de implementación.
 - (f) 10% Observación en aula
3. Se requiere una asistencia mínima del 90%.
4. Las condiciones de aprobación de la asignatura serán de acuerdo al Reglamento de Docencia de Pregrado.

IX. RECURSOS BIBLIOGRÁFICOS

Bibliografía Mínima

- Pierre Bourque, Richard E. Fairley. SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge

Bibliografía Complementaria

- Ian Sommerville. Ingeniería de Software
- Craig Larman. UML
- G. Gordon Schulmeyer. Handbook of Software Quality Assurance, Cuarta Edición
- VV.AA. Patrones De Diseño: Elementos De Software Orientado a Objetos