



## Programa de Asignatura

I. IDENTIFICACIÓN				
<b>Carrera o programa:</b> Ingeniería Civil en Computación e Informática				
<b>Unidad responsable:</b> Escuela de Ingeniería				
<b>Nombre de la asignatura:</b> Proyecto Integrador Plataformas				
<b>Código:</b> ECIN-00708				
<b>Semestre en la malla<sup>1</sup>:</b> 7				
<b>Créditos SCT - Chile:</b> 5				
<b>Ciclo de Formación</b>	Básico		Profesional	X
<b>Tipo de Asignatura</b>	Obligatoria	X	Electiva	
<b>Clasificación de área de conocimiento<sup>2</sup></b>				
<b>Área:</b> Ingeniería y Tecnología		<b>Sub área:</b> Ingeniería Informática		
<b>Requisitos:</b>				
<b>Pre-requisitos:</b>		<b>Requisito para:</b>		
• ECIN-00619 Proyecto Integrador Software		• ECIN-00809 Proyecto Integrador Gestión TI		

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

III. APORTE 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 aplicar los procesos de Ingeniería de Software en el desarrollo de una aplicación web y una aplicación móvil, usando arquitecturas apropiadas a la solución planteada.

<sup>1</sup> Este campo

<sup>2</sup> Clasificación del curso de acuerdo a la OCDE



#### 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. Concebir un producto de software que da solución a las necesidades planteada por un cliente.
2. Diseñar un producto de software utilizando UML considerando el diseño de alto nivel, la arquitectura del sistema y la interfaz de usuario.
3. Implementar un producto de software de calidad, en base a las buenas prácticas recomendadas por los modelos de procesos, considerando la creación de una interfaz web y una aplicación móvil.
4. Identificar los objetivos y requerimientos de las soluciones TIC
5. Seleccionar los procesos, técnicas y herramientas adecuados de acuerdo a los requerimientos.
6. Desarrollar la solución tecnológica más adecuada en base a las características del problema y los recursos disponibles.
7. Organizar información mediante herramientas TIC para mejorar el almacenamiento, la comunicación, la transmisión e intercambio de ésta de manera efectiva.

## VI. ÁREAS TEMÁTICAS

1. Desarrollo basado en plataforma versus desarrollo tradicional
  - 1.1 Visión general de plataformas
  - 1.2 Programación vía APIs específicas a las plataformas
  - 1.3 Visión general de lenguajes para plataformas
  - 1.4 Programación bajo restricciones de la plataforma
2. Plataformas Web
  - 2.1 Lenguajes de programación Web
  - 2.2 Restricciones de la plataforma Web
  - 2.3 Software como un servicio (SaaS)
  - 2.4 Estándares Web
3. Plataformas Móviles
  - 3.1 Lenguajes de programación para móviles
  - 3.2 Desafíos con la movilidad y comunicación inalámbrica
  - 3.3 Aplicaciones conscientes de la localización



- 3.4 Balance rendimiento versus energía
- 3.5 Restricciones de las plataformas móviles
- 3.6 Tecnologías emergentes

## 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**

- M. Murphy. The Busy Coder's Guide to Android Development

### **Bibliografía Complementaria**

- Ian Sommerville. Ingeniería de Software
- G. Gordon Schulmeyer. Handbook of Software Quality Assurance, Cuarta Edición
- Craig Larman. UML