

Programa de Asignatura

1. Identificación Asignatura

Nombre:	Taller de Desarrollo Avanzado de Software		Código:	IF1021
Carrera:	Ingeniería Civil Informática	Unidad Académica:	Ciencias Naturales y Tecnología	
Ciclo Formativo:	Ciclo Licenciatura	Línea formativa:	Especializada	
Semestre	VII	Tipo de actividad:	Obligatoria	
N° SCT:	6	Horas Cronológicas Semanales		
		Presenciales:	4,5	Trabajo Autónomo:
Pre-requisitos	Ingeniería de Software			

2. Propósito formativo

El curso Taller de Desarrollo Avanzado de Software es un curso eminentemente práctico, y tiene como propósito que los y las estudiantes enfrenten una experiencia lo más cercana posible al mundo real del desarrollo de software, abordando los desafíos típicos como por ejemplo: llevar una buena relación con el cliente, planificar y adaptar el trabajo de acuerdo a las circunstancias, evaluar los riesgos asociados al proyecto y desarrollar una solución de software real.

En un comienzo, las y los estudiantes deberán seleccionar un proyecto entre las alternativas presentadas, e identificar una metodología de desarrollo. El desarrollo del proyecto involucra iteración y cada iteración representa actividades que deben ser planificadas y presentadas para su evaluación. El o la académica responsable será una guía profesional y apoyo metodológico.

Esta asignatura se ubica en el VII semestre y no es pre-requisito de otros cursos del plan de formación.

3. Contribución al perfil de egreso

Esta asignatura contribuye a los siguientes desempeños o resultados de aprendizaje globales declarados en el Perfil de Egreso de la carrera:

1. Evalúa la implementación de soluciones computacionales, utilizando métodos analíticos y experimentales, para estudiar su eficiencia en virtud de distintas plataformas y lenguajes utilizados.
2. Desarrolla productos y servicios de software, a través de metodologías, de procesos analíticos y de diseño, que consideren las características de las distintas plataformas y lenguajes disponibles, para abordar necesidades de diversos usuarios.
3. Gestiona componentes organizacionales que estén asociados a la captura, almacenamiento, transformación y provisión de información, de forma que se puedan satisfacer necesidades de información con estándares de calidad adecuados

4. Resultados de aprendizaje específicos

Resultado de Aprendizaje Específico	Criterios de evaluación	Evidencia
RA1. Utilizar lenguajes de programación y paradigmas adhoc a las problemáticas a resolver	1.1. Evalúa diferentes alternativas tecnológicas para la resolución de una problemática en particular. 1.2. Selecciona lenguaje y paradigma adecuado para la problemática.	Laboratorios, guías de ejercicio.
RA2. Planificar, ejecutar y evaluar las actividades relacionadas con el desarrollo de un producto software que resuelva un problema particular.	1.1. Diseña un proyecto que incorpora las etapas y planificación de una solución de software (Aplicado, problema real interna o externa) 2.2. Implementa y evalúa plan de desarrollo de software.	Laboratorios, guías de ejercicio. Informes y presentaciones de avance y producto final.

5. Unidades de Aprendizaje

<p>1. Coordinación Inicial Curso</p> <p>1.1. Introducción, reglas del curso y recomendaciones. 1.2. Presentación de proyectos a abordar. 1.3. Planificación y control de actividades y riesgos.</p>
<p>2. Desarrollo de Proyecto</p> <p>2.1. Revisión semanal de:</p> <ul style="list-style-type: none"> - Planificación y Avance de Proyecto. - Gestión de Riesgos. - Especificaciones de Requisitos/Pruebas - Revisión de Software. - Registro y Análisis de Métricas - Proceso de Transferencia

6. Recursos de Aprendizaje

Bibliografía:

B1: Fundamentals of Software Engineering (2nd Edition), Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Prentice Hall; September 29, 2002.

<https://www.abebooks.com/9787508338767/Fundamentals-Software-Engineering-2nd-Edition-7508338766/plp>

B2: Software Engineering: (Update) (8th Edition), Ian Sommerville; Addison Wesley, June 4, 2006

https://doc.lagout.org/science/0_Computer%20Science/Software%20Engineering%2C%208th%20Edition.pdf

Recursos materiales e infraestructura:

- Laboratorio de computación.
- Acceso a Ucampus.

- Acceso a Googlesites con credenciales institucionales.

Computadores debidamente equipados para utilizar lenguajes de alto nivel (por ej.: Python).

7. Comportamiento y ética académica:

Se espera que los estudiantes actúen en sus diversas actividades académicas y estudiantiles en concordancia con los principios de comportamiento ético y honestidad académica propios de todo espacio universitario y que están estipulados en el *Reglamento de Estudiantes de la Universidad de Aysén*, especialmente aquéllos dispuestos en los artículos 23°, 24° y 26°.

Todo acto contrario a la honestidad académica realizado durante el desarrollo, presentación o entrega de una actividad académica del curso sujeta a evaluación, será sancionado con la suspensión inmediata de la actividad y con la aplicación de la nota mínima (1.0).

Planificación del curso

8. Responsables

Académico (s) Responsable (s) y equipo docente	Gabriel Núñez Vivanco		
Contacto	gabriel.nunez@uaysen.cl		
Año	2025	Periodo Académico	1
Horario clases	Lunes 16:15-16:45 hrs Jueves 10:15 – 13:30 hrs	Horario de atención estudiantes	Viernes 8:30 hrs
Sala / Campus			

9. Metodología de Trabajo:

La asignatura contiene:			
Actividades de vinculación con el medio	Si	Actividades relacionadas con proyectos de investigación	No
En el curso se contemplan cuatro tipos de actividades docentes, las cuales se asocian a requerimientos de sala y al nivel de intervención del profesor:			
Actividad docente	Descripción	Intervención del profesor/ayudante	Requerimiento de sala
Exposición conceptual	El profesor introduce conceptos de programación preliminares y necesarios a otras actividades de índole práctica, de forma expositiva. Se dispone de un site de google donde el estudiante podrá encontrar videos y material complementario.	Alta	Sala de clases UCampus Online UCampus Offline
Programación expositiva	El profesor profundiza en la comprensión de elementos conceptuales a través de la exposición directa de la resolución de problemas de programación como ejemplos.	Alta	Sala de clases UCampus Online UCampus Offline
Programación tutorial	Funciona como la programación expositiva, pero el profesor realiza pausas para que los alumnos completen “pasos requeridos” antes de continuar. El objetivo es que todos los alumnos completen un paso definido por el profesor antes de continuar al siguiente.	Media	Laboratorio de computación Computador persona

Actividad práctica / Programación autónoma	Los estudiantes abordan y resuelven problemas de programación de forma autónoma, idealmente en grupos y sólo guiados por el profesor.	Baja	Laboratorio de computación Computador persona
--	---	------	---

10. Evaluaciones:

Evaluación	Ponderaciones específicas	Ponderación nota presentación
Informes de avance	<ul style="list-style-type: none"> Informe N° 1 + Presentación N°1 Informe N° 2 + Presentación N°2 Informe N° 3 + Presentación N°3 	60%
Informe Final y Exposición	10 % Informe Final 30 % Exposición (Evalúa además Socio/a Comunitario/a)	40%

Calificación final:

- Nota de presentación: 70%
- Examen Final: 30 %

Condiciones de eximición:

- Nota de presentación igual o superior a nota 5,0
- Promedio de pruebas de cátedra y proyecto $\geq 4,0$

Derecho a rendir examen:
Nota de presentación $\geq 3,5$

Requisito de Aprobación

- Asistencia: 70%
- \geq Nota Final: 4,0

11. Otros aspectos asociados al funcionamiento del curso:

<ul style="list-style-type: none"> Toda la coordinación del curso (comunicaciones, actualización de notas, material, etc.) será realizada a través de UCampus. El estudiante deberá informar con tiempo suficiente si presenta dificultades de conexión para trasladar el requerimiento a la coordinación de programa. <p>Es deber del estudiante mantenerse informado de las noticias, avisos y material entregado por el profesor a través de estos medios, se sugiere instalar en su dispositivo móvil la aplicación de UCampus.</p>

12. Planificación de las actividades de enseñanza- aprendizaje y de evaluación

Semana	Tema (Unidades de aprendizaje) actividades	Resultado(s) de Aprendizaje	Recursos utilizados o lecturas	Actividad(es) de Trabajo Autónomo
--------	--	-----------------------------	--------------------------------	-----------------------------------

Semana 1	Presentación Programa y detalles de implementación	RA1, RA2 (Unidad I)	B1, B2	Presentación programa y discusión inicial.
Semana 2	Presentación Socio/a Comunitario/a	RA1, RA2	B1, B2	Búsqueda y selección del socio comunitario. Identificación de problemáticas.
Semana 3	Confirmación Socio/a Comunitario/a	RA1, RA2	B1, B2	Validación de la problemática con el socio comunitario. Definición inicial del alcance del proyecto.
Semana 4	Metodologías y materiales de desarrollo de software.	RA1, RA2	B1, B2	Selección y presentación de la metodología y materiales más adecuados para el proyecto
Semana 5	Planificación detallada del desarrollo	RA1, RA2	B1, B2	Captura y especificación de requisitos. Validación de requisitos con el socio comunitario
Semana 6	Desarrollo Iterativo y Gestión del Proyecto	RA1, RA2	B1, B2	Primera iteración: Diseño de arquitectura y prototipado inicial.
Semana 7	Desarrollo Iterativo y Gestión del Proyecto	RA1, RA2	B1, B2	Segunda iteración: Desarrollo y primeras pruebas de funcionalidad básica.
Semana 8	Desarrollo Iterativo y Gestión del Proyecto	RA1, RA2	B1, B2	Revisión de avances con el socio comunitario. Ajustes a la planificación.
Semana 9	Desarrollo Iterativo y Gestión del Proyecto	RA1, RA2	B1, B2	Tercera iteración: Implementación de funciones clave y pruebas
Semana 10	Desarrollo Iterativo y Gestión del Proyecto	RA1, RA2	B1, B2	Cuarta iteración: Integración y mejoras en usabilidad.
Semana 11	Desarrollo Iterativo y Gestión del Proyecto	RA1, RA2	B1, B2	Quinta iteración: Integración y mejoras en usabilidad.
Semana 12	Validación y Transferencia	RA1, RA2	B1, B2	Pruebas finales y validación con el socio comunitario.
Semana 13	Validación y Transferencia	RA1, RA2	B1, B2	Documentación del proceso y elaboración del manual de usuario.
Semana 14	Validación y Transferencia	RA1, RA2	B1, B2	Preparación para la transferencia del software al socio comunitario.
Semana 15	Presentación y Cierre	RA1, RA2	B1, B2	Pruebas de aceptación con el socio comunitario y ajustes finales
Semana 16	Presentación y Cierre	RA1, RA2	B1, B2	Presentación final del proyecto
Semana 17	Presentación y Cierre	RA1, RA2	B1, B2	Reflexión y retroalimentación del curso. Evaluación de aprendizajes
Semana 18	Examen	Examen	Examen	Examen