

Programa de Asignatura

1. Identificación Asignatura

Nombre:	Programación Avanzada Programación II	Código:	IF 1002 IF 1015
Carrera:	Ingeniería Civil Informática Ingeniería Civil Industrial	Unidad Académica:	Ciencias Naturales y Tecnología
Ciclo Formativo:	Ciclo Inicial	Línea formativa:	Básica
Semestre	III	Tipo de actividad:	Obligatoria
N° SCT:	6	Horas Cronológicas Semanales	
		Presenciales:	4.5
		Trabajo Autónomo:	4.5
Pre-requisitos	Fundamentos de Programación		

2. Propósito formativo

La asignatura de Programación Avanzada tiene como propósito que el/la estudiante aprenda a desarrollar soluciones de programación siguiendo el paradigma orientado a objetos (POO).

En su primera parte, la asignatura aborda conceptos de orientación a objetos al mismo tiempo que los aplica usando el lenguaje de programación Python. Luego, en una transición al lenguaje JAVA el/la estudiante ira conociendo y aplicando elementos como objetos, clases, herencia, abstracción y polimorfismo, entre otros. El contexto del desarrollo de habilidades de aplicación metodológica será a través de problemas específicos, definidos en diversos dominios de aplicación cuyas soluciones se encontrarán delimitadas en cuanto al alcance y complejidad.

Esta asignatura plantea nuevos paradigmas de programación a los vistos hasta este momento, y promueve su aplicación en una asignatura práctica del mismo nivel que es Taller de Ingeniería III. De forma incipiente, los y las estudiantes comenzarán a trabajar con diferentes estructuras de datos que serán profundizadas en la siguiente asignatura de especialidad que es Algoritmos y Estructuras de Datos.

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. Entiende problemas a través de la construcción de abstracciones conceptuales, cualitativas y cuantitativas, utilizando formalismos establecidos, que permitan formular soluciones
2. Diseña y programa soluciones, utilizando estrategias algorítmicas, que permitan resolver problemas de forma eficaz y acorde a múltiples objetivos de diseño

Resultados de aprendizaje específicos

Resultado de Aprendizaje Específico	Criterios de evaluación	Evidencia
RA1. Reconoce los componentes del paradigma de programación orientado a objetos, definiendo cada uno de ellos y reconociéndolos en contextos de uso.	<p>1.1. Comprende los conceptos del paradigma de programación orientada a objetos tales como: polimorfismo, encapsulamiento, herencia, sobrecarga, funciones virtuales, etc.</p> <p>1.2. Define y diagrama clases y sus componentes: campos, constructor, métodos accesorios y mutadores.</p> <p>1.3. Modelar problemas utilizando el paradigma de programación orientada a objetos.</p>	Laboratorio ejercicio. guías de
RA2. Diseña e implementa soluciones en Java aplicando los principios de la programación orientada a objetos para el desarrollo de software estructurado, modular y reutilizable.	<p>1.4. Identifica y compara las diferencias entre Python y Java en la implementación de la programación orientada a objetos, incluyendo clases, objetos, herencia, polimorfismo y encapsulamiento.</p> <p>1.5. El/la estudiante logra elaborar de manera autónoma programas en JAVA utilizando POO para resolver problemas simples, aplicando principios de modularidad y reutilización</p>	Laboratorio ejercicio. guías de
RA3. Desarrolla aplicaciones en Java utilizando clases, objetos y librerías para la creación de soluciones de software, ya sea adaptando código existente o implementando nuevos requerimientos según especificaciones dadas	1.6. El/la estudiante logran desarrollar soluciones de forma autónoma a problemas de diversa complejidad utilizando el lenguaje JAVA y con un POO.	Laboratorio ejercicio. guías de

4. Unidades de Aprendizaje

1. Introducción a la modelación orientada a objetos – relaciones de asociación
<p>1.1. Paradigmas de programación</p> <p>1.2. Lenguajes orientados a objetos</p> <p>1.3. Identificación y evolución de los esquemas de abstracción.</p> <p>1.4. Conceptos de clase, atributo, operación y objeto.</p> <p>1.5. Aliasing, manejo de referencias</p>

2. Programación Orientada a Objetos en JAVA
2.1. Introducción al lenguaje JAVA
2.2. Aplicación de Abstracción: Clases, Atributos, Operaciones (métodos, clases, herencia)
2.3. Modularidad (criterios, principios y reglas)
2.4. El concepto de interfaz
2.5. El concepto de objeto
2.6. Asociaciones, agregaciones y composiciones
3. Herencia, Polimorfismo y Gestión de Errores
3.1. Abstracción
3.2. Tipos de jerarquías y costes de la herencia
3.3. Tipos de polimorfismos
3.4. Relaciones entre polimorfismos
3.5. Gestión de errores. Persistencia y concurrencia.
3.6. Relaciones entre polimorfismos
3.7. Gestión de errores. Persistencia y concurrencia.

5. Recursos de Aprendizaje

Bibliografía:

-[Programación orientada a objetos en Java](#) Autores: [Blasco, Francisco](#) ISBN: 9789587921014, 9789587921007
Editorial: [Ediciones de la U 2019](#) Disponible en: Pdf

-Programación orientada a objetos Autores: Moreno Pérez, Juan Carlos ISBN: 9788499644875 Editorial: RA-MA 2015

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).

6. 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	Profesor: Claudio Levicán Jaque			
Contacto	Correo: claudio.levican@uaysen.cl			
Año	2025	Periodo Académico	Primer semestre	
Horario clases	Cátedra: Lunes: 14:30 a 17:45 hs. Viernes: 12:00 a 13.30 hs.	Horario de atención estudiantes	Miércoles 14.30-15.30 hs.	
Sala / Campus	Sala			

9. Metodología de Trabajo:

La asignatura contiene:			
Actividades de vinculación con el medio	No	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 materiales complementarios en la plataforma Ucampus.	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, algunas con guía y apoyo docente y otras no.	Baja/Media	Laboratorio de computación Computador persona
En cualquier semana del semestre en curso se podría realizar una evaluación menor sobre las temáticas estudiadas a la fecha. Esta evaluación menor puede ser de los siguientes tipos:			
<ul style="list-style-type: none"> ● Laboratorio: Evaluación individual o grupal, que se realiza en el computador. Ocupará los bloques del día jueves. ● Guía de ejercicios: Evaluación individual que se realiza en computador durante el tiempo de trabajo autónomo. ● Prueba Parcial: Evaluación individual que se realiza en computador en el horario de clases. ● Proyecto: Evaluación individual o grupal, que se realiza en el computador. Ocupará los bloques horario autónomo. 			

10. Evaluaciones:

Evaluación	Ponderación	Fecha
Prueba Unidad 1 (P1):	30%	07/04/2025
Prueba Unidad 2 (P2):	35%	14/05/2025
Trabajo- presentación Unidad 3 (P3):	35%	27/06/2025
Prueba recuperativa		30/06/2025
Examen		11/07/2025

Calificación final:

-Nota de presentación: 70%

-Examen Final: 30 %

Condiciones de eximición:

-Nota de presentación igual o superior a nota 5,0 -

Ponderación específica pruebas de cátedra \geq 4,0

-Derecho a rendir examen:

-Nota de presentación \geq 3,5

Requisito de Aprobación

Asistencia: 70%

Nota Final: 4,0

11. Otros aspectos asociados al funcionamiento del curso:

- 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. Adicionalmente los estudiantes deberán acceder al material disponible en el google sites que el profesor preparó para este propósito.

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.

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

Semana / Sesión 2025	Resultado(s) de Aprendizaje	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas	Actividad(es) de Trabajo Autónomo
Semana 1 (10 - 14 de Marzo)	RA1. Reconoce los componentes del paradigma de programación orientado a objetos, definiendo cada uno de ellos y reconociéndolas en contextos de uso.	Revisión del programa Acceso al servidor Guía de Linux Unidad 1: 1.1 Paradigmas de programación 1.2 Lenguajes orientados a objetos	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio de Linux
Semana 2 (17 - 21 de marzo)	RA1.	1.3 Identificación y evolución de los esquemas de abstracción.	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 3 (24 - 28 de marzo)	RA1.	1.4 Conceptos de clase, atributo, operación y objeto. 1.5 Manejo de referencias Unidad 2. Programación Orientada a Objetos en JAVA	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 4 (31 al 4 de abril)	RA1.	2.1. Introducción al lenguaje JAVA	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 5 (7 al 11 de abril)	RA1.	Prueba 1: 07/04/2025		
Semana 6 (14 al 18 de abril)	RA2.	2.2. Aplicación de Abstracción Clases, Atributos.	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 7 (21 al 25 de abril)	RA2.	2.2. Aplicación de Abstracción: Operaciones métodos, clases, herencia.	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 8 (28-04 al 2/05)	RA2.	2.3 Modularidad (criterios, principios y reglas) 2.4 El concepto de interfaz	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio

Semana / Sesión 2025	Resultado(s) de Aprendizaje	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas	Actividad(es) de Trabajo Autónomo
Semana 9 (5 - 9 de mayo)	RA2.	2.5 Concepto de objetos 2.6 Asociaciones, agregaciones y composiciones	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 10 (12 al 16/05)		Prueba 2: 14/05/2025		
Semana 10 (12 al 16/05)		Unidad 3: Herencia, Polimorfismo y Gestión de Errores 3.1 Abstracción		
Semana 11 (19 -23 de mayo)	RA1.	3.2 Tipos de jerarquías y costes de la herencia Actividad Práctica	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 11 (26 -30 de mayo)		3.3 Tipos de polimorfismos 3.4. Relaciones entre polimorfismos	Apunte de clases Libros Acceso a un servidor	
Semana 12 (6 al 6 de junio)		3.5. Gestión de errores. Persistencia y concurrencia 3.6 Aplicaciones básicas de Programación orientada a objetos en JAVA	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 13 (9 al 13 de junio)		3.7 Aplicaciones avanzadas de Programación orientada a objetos en JAVA	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio

Semana / Sesión 2025	Resultado(s) de Aprendizaje	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas	Actividad(es) de Trabajo Autónomo
Semana 14 (16 al 20 de junio)		3.8 Aplicaciones en uso de servidores	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 15 (23 al 27 de junio) Sesion1		-Repaso de contenidos -actividad practica	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 15 (23 al 27 de junio) Sesion2		Prueba 3: 27 de julio 2025	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 16 (30/06 al 4 de julio) Sesion1		Prueba recuperativa 30/06/2025		
Semana 16 (30/06 al 4 de julio) Sesion2		Repaso de contenidos		
Semana 17 (7 al 11 de julio)		Examen: 11 de julio 2025 Unidades 1, 2 y 3		
Semana 18 (14 al 18 de julio)		Revisión de examen y cierre de actas		