

Programa de Asignatura

1. Identificación Asignatura

Nombre:	Lenguajes de Programación		Código:	
Carrera:	Ingeniería Civil Informática	Unidad Académica:	Ciencias Naturales y Tecnología	
Ciclo Formativo:	Ciclo Licenciatura	Línea formativa:	Especializada	
Semestre	VI	Tipo de actividad:	Obligatoria	
Nº SCT:	6	Horas Cronológicas Semanales		
		Presenciales:	4.5	Trabajo Autónomo:
Pre-requisitos	Teoría de la Computación			

2. Propósito formativo

El curso Lenguajes de Programación tiene como propósito que los y las estudiantes adquieran los conceptos fundamentales sobre los diversos lenguajes y estrategias de programación disponibles, sus aplicaciones, semántica, e implementación.

En su primera parte, la asignatura incorpora elementos para identificar mecanismos para entender lenguajes y paradigmas existentes, la programación lógica y funcional en general. Luego, el/la estudiante aprenderá algunos elementos precisos de comparación y descripción de varios lenguajes, algunos ampliamente usados y otros emergentes, como C, Java, Lisp, bash, Haskell, ML, JavaScript, Scala, Self, y Smalltalk.

Esta asignatura es un recurso de apoyo fundamental para el resto de la línea disciplinar, y aunque no es prerrequisito de algún curso en particular, entregará herramientas diversas que serán útiles en todo resto de las asignaturas.

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.
3. Evalúa la implementación de soluciones computacionales, utilizando métodos analíticos y experimentales, para estudiar su eficiencia en virtud de distintas plataformas y lenguajes utilizados.
4. 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.

4. Resultados de aprendizaje específicos

Resultado de Aprendizaje Específico	Criterios de evaluación	Evidencia
<p>RA1. Comprende los conceptos relacionados con la interpretación y compilación de lenguajes de programación. Además, conoce los diferentes paradigmas de programación.</p>	<p>1.1. Conoce las diferentes formas de representación de datos de los lenguajes de programación.</p> <p>1.2. Conoce diferentes paradigmas de programación, y los tipos de lenguajes que pueden ser utilizados en cada caso.</p>	<p>Laboratorios, guías de ejercicio.</p>
<p>RA2. Conoce las ventajas y desventajas del uso de diferentes tipos y lenguajes de programación.</p>	<p>2.1. Resuelve problemas variados utilizando enfoques y lenguajes diversos.</p> <p>2.2. Reconoce ventajas y desventajas en el uso de diferentes lenguajes y estrategias de programación.</p>	<p>Laboratorios, guías de ejercicio.</p>
<p>RA3. Utiliza diferentes estrategias, paradigmas y lenguajes de programación de acuerdo con los requerimientos particulares de diferentes desafíos.</p>	<p>3.1. Diseña soluciones adecuadas según los requerimientos de desafíos de programación.</p> <p>3.2. Compara soluciones e identifica características positivas y negativas de los diferentes enfoques y lenguajes.</p>	<p>Laboratorios, guías de ejercicio.</p> <p>Propuesta: Desarrollo de mini-proyectos de programación, en donde se compartan experiencias entre los compañeros y compañeras, permita abordar de mejor forma un abanico de diferentes paradigmas y lenguajes.</p>

5. Unidades de Aprendizaje

1. Fundamentos de los Lenguajes de Programación

- 1.1. Procesamiento de datos de entrada.
- 1.2. Lenguajes compilados e interpretados
- 1.3. Sintaxis y semántica
- 1.4. Uso de memoria

2. Fundamentos de los tipos de programación

- 2.1. Fundamentos del paradigma de programación lógica.
 - Lógica de primer orden
 - Cláusulas de Horn
 - Resolución y Unificación
 - Algoritmo de resolución SLD
- 2.2. Fundamentos de programación funcional avanzada
 - Estructuras de listas
 - Funciones
 - Lambda cálculo
 - Recursividad
- 2.3. Fundamentos de programación basada en eventos
- 2.4. Fundamentos de la programación concurrente
 - Procesos, hilos y memoria compartida
 - Race condition
 - Paso de mensajes
- 2.5 Fundamentos de la programación paralela y distribuida

6. Recursos de Aprendizaje

Bibliografía:

B1: Shriram Krishnamurthi. Programming Languages: application and interpretation 2012 [<https://cs.brown.edu/~sk/Publications/Books/ProgLangs/>]

B2: Ulf Nilsson, Jan Maluszynski. Logic Programming and prolog. 1995 [<https://www.ida.liu.se/~ulfni53/lpp/bok/bok.pdf>]

Recursos materiales e infraestructura:

- Laboratorio de computación.
- Acceso a Ucampus.
- 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 aquellos 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	Andrés Ruiz-Tagle		
Contacto	Correo: Andrés.Ruiz-Tagle @uaysen.cl		
Año	2025	Periodo Académico	Segundo Semestre
Horario clases	Martes 16:15 – 17:45 Jueves 10:15 – 13.30	Horario de atención estudiantes	
Sala / Campus			

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:

- **Laboratorio:** Evaluación individual o grupal, que se realiza en el computador.
- **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:

Evaluaciones	Ponderación	fecha
Evaluación1	30%	Semana 06 -
Evaluación 2	30%	Semana 12 -
Evaluación 3	40%	Semana 16 -
Examen	30%	Semana 17 -

Calificación final

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

Condiciones de eximición:

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

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.

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	Resultados de Aprendizaje	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas	Actividad(es) de Trabajo Autónomo
Semana 1	RA1.	<p>Revisión del programa -Introducción a los Lenguajes de Programación</p> <p>Unidad 1: Fundamentos de los Lenguajes de Programación 1.1. Procesamiento de datos de entrada. 1.2. Lenguajes compilados e interpretados</p>	Apunte de clases	Lecturas Instalación y configuración de aplicaciones
Semana 2	RA1.	<p>1.3. Sintaxis y semántica</p> <p>1.4. Uso de memoria</p>	Apunte de clases Libros Acceso a un servidor	Lecturas
Semana 3	RA2.	<p>Unidad 2 2.1. Fundamentos del paradigma de programación lógica. -Lógica de primer orden -Cláusulas de Horn</p>	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 4	RA2.	<p>-Resolución y Unificación -Algoritmo de resolución SLD</p>	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 5	RA2.	<p>2.2. Fundamentos de programación funcional avanzada</p> <p>Estructuras de listas Funciones</p>		Guía de Laboratorio
Semana 6	RA2.	Prueba 1	Apunte de clases Libros	Guía de Laboratorio

		Lambda cálculo Recursividad	Acceso a un servidor	
Semana 7	Receso			
Semana 8	RA2.	2.3. Fundamentos de programación basada en eventos	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 9	RA2.	2.4. Fundamentos de la programación concurrente	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 10	RA3.	Procesos, hilos con Pthreads		
Semana 11	RA3.	Programación en memoria compartida OpenMP	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 12	RA3.	Prueba 2		
Semana 13	RA3.	Programación en memoria compartida OpenMp	Apunte de clases Libros	Guía de Laboratorio
Semana 14	RA3.	2.5. Fundamentos de la programación paralela y distribuida	Apunte de clases Libros Acceso a un servidor	Guía de Laboratorio
Semana 15	RA3.	2.5. Fundamentos de la programación paralela y distribuida con MPI	Apunte de clases Libros	Guía de Laboratorio
Semana 16		Trabajo Final con Exposición	Apunte de clases Libros	Guía de Laboratorio
Semana 17		Examen	Apunte de clases Libros	Guía de Laboratorio
Semana 18		Cierre de Actas		