

# Programa de Asignatura

## 1. Identificación Asignatura

<b>Nombre:</b>	Algoritmos y Estructuras de Datos		<b>Código:</b>	
<b>Carrera:</b>	Ingeniería Civil Informática	<b>Unidad Académica:</b>	Ciencias Naturales y Tecnología	
<b>Ciclo Formativo:</b>	Ciclo Inicial	<b>Línea formativa:</b>	Básica	
<b>Semestre</b>	IV	<b>Tipo de actividad:</b>	Obligatoria	
<b>N° SCT:</b>	6	<b>Horas Cronológicas Semanales</b>		
		<b>Presenciales:</b>	4.5	<b>Trabajo Autónomo:</b>
<b>Pre-requisitos</b>	Programación Avanzada			

## 2. Propósito formativo

La asignatura de Algoritmos y Estructuras de Datos tiene como propósito que el/la estudiante logre un dominio de los conceptos básicos de diseño de algoritmos eficientes (iterativos y recursivos), de análisis de algoritmos, y de estructuras de datos (arreglos, estructuras enlazadas, grafos). Los y las estudiantes deben ser capaces de aplicar estos conceptos al diseño, implementación, y evaluación de los tipos de datos abstractos más usados (listas, pilas, colas, colas de prioridad, diccionario, árboles, tablas de hash), así como de los algoritmos fundamentales (búsqueda, ordenación, algoritmos para grafos y para búsqueda en texto) utilizados en Ciencias de la Computación.

Esta asignatura plantea nuevas estructuras de datos y promueve su aplicación en una asignatura práctica del mismo nivel que es el Taller de Ingeniería IV.

## 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. Gestiona datos de distintas características, asegurando su calidad, acceso eficiente y seguridad, a través de mecanismos que habiliten su almacenamiento y que sean apropiados a las responsabilidades establecidas para su uso.

## 4. Resultados de aprendizaje específicos

Resultado de Aprendizaje Específico	Criterios de evaluación	Evidencia
-------------------------------------	-------------------------	-----------

<p><b>RA1.</b> Aprende a diseñar y analizar algoritmos según su complejidad.</p>	<p>1.1. Realiza procedimientos básicos de análisis de algoritmos.</p> <p>1.2. Utiliza notación formal para describir el tiempo de ejecución de un algoritmo.</p> <p>1.3. Identifica ejemplos prácticos en los que se puede aplicar cada una de las estrategias (fuerza bruta, codicioso, dividir para conquistar, backtracking y programación dinámica) de diseño de algoritmos</p>	<p>Laboratorios, guías de ejercicio.</p>
<p><b>RA2.</b> Aprende a reconocer y utilizar estructuras de datos elementales y tipos de datos abstractos.</p>	<p>2.1 Realiza operaciones con listas, insertar, eliminar o localizar un elemento</p> <p>2.2 Realiza operaciones con pilas:insertar un elemento en la pila, recuperar y eliminar un elemento de la pila.</p> <p>2.3 Realiza operaciones con colas: insertar un elemento en la cola, recuperar y eliminar un elemento de la cola.</p> <p>2.4 Realiza operaciones con colas de prioridad: insertar un elemento en la cola de prioridad, recuperar y eliminar el elemento de mayor prioridad en la cola.</p> <p>2.5 Realiza operaciones de manejo de datos en arreglos, árboles y grafos.</p>	<p>Laboratorios, guías de ejercicio.</p>
<p><b>RA3.</b> Programa algoritmos que resuelven problemas de ordenamiento o búsqueda sobre estructuras de datos avanzadas como por ejemplo, grafos, árboles, tablas hash, etc.</p>	<p>3.1 Implementa los algoritmos de los distintos tipos de árboles, representación de un árbol, representación por arreglos, tipos de recorridos: Pre orden, In orden, Post orden, utilizar los tipos de grafos, dirigidos, no dirigidos, representación de los grafos.</p> <p>3.2 Utiliza tablas de hash como estructura de datos principal para búsqueda de información.</p> <p>3.3 Implementa algoritmos que utilizan diferentes estructuras de datos para búsqueda y ordenación de datos.</p>	<p>Laboratorios, guías de ejercicio.</p>

## 5. Unidades de Aprendizaje

### 1. Introducción – Diseño y Análisis de Algoritmos.

- 1.1. Conceptos matemáticos: Notación “O”, ecuaciones de recurrencia, Teorema Maestro.
- 1.2. Análisis de algoritmos sencillos.
- 1.3. Inducción simple y reforzada.
- 1.4. Recursividad, dividir para reinar.
- 1.5. Programación dinámica.
- 1.6. Algoritmos avaros (greedy).
  
- 1.7. Backtracking

### 2. Estructuras de Datos Elementales y Tipos de Datos abstractos

- 2.1. Arreglos
- 2.2. Matrices
- 2.3. Punteros y listas enlazadas
- 2.4. Árboles
- 2.5. Pilas
- 2.6. Colas
  
- 2.7. Colas de prioridad.

### 3. Algoritmos de búsqueda y diccionarios

- 3.1. Búsquedas
- 3.2. Árboles de búsqueda
- 3.3. Tablas de hash

### 4. Algoritmos de ordenamiento

- 4.1. Cota inferior.
- 4.2. Quicksort.
- 4.3. Heapsort.
- 4.4. Radix Sort.
- 4.5. Mergesort.

### 5. Algoritmos para grafos

- 5.1. Representaciones para grafos.
- 5.2. Recorridos en profundidad y amplitud.
- 5.3. Árbol cobertor mínimo (Kruskal, Prim).
- 5.4. Caminos más cortos (Dijkstra,, Floyd, Warshall)

### 6. Algoritmos para búsqueda en textos y compresión de datos

## 6. Recursos de Aprendizaje

### Bibliografía:

- B1. Tymoschuk, J. Guzman, A. y Fritelli, V. (2020). Algoritmos y estructuras de datos (2a. ed.). Jorge Sarmiento Editor - Universitas. <https://elibro.net/es/lc/uaysen/titulos/175249>
- B2. Joyanes Aguilar, L. (2005). Estructuras de datos en C. McGraw-Hill España. <https://elibro.net/es/ereader/uaysen/50302?page=34> , <https://elibro.net/es/lc/uaysen/titulos/50068>

**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**

<b>Académico Responsable (s) y equipo docente</b>	<b>Profesor:</b> Dra. Mariela Isabel González Flores		
<b>Contacto</b>	<b>Correo:</b> mariela.gonzalez@uaysen.cl		
<b>Año</b>	2024	<b>Periodo Académico</b>	Primer semestre
<b>Horario clases</b>	<b>Cátedra:</b> Martes: 14.30 a 17.45 h. Viernes: 10.15-11.45 h.	<b>Horario de atención estudiantes</b>	Horario por confirmar con los/las estudiantes
<b>Sala / Campus</b>	Sala Virtual Ucampus		

**9. Metodología de Trabajo:**

<b>La asignatura contiene:</b>			
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	Alta	Sala de clases UCampus Online

	directa de la resolución de problemas de programación como ejemplos.		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. 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 del día jueves y también horario autónomo.

## 10. Evaluaciones:

Evaluación	Ponderaciones específicas	Ponderación nota presentación
Pruebas de cátedra	<ul style="list-style-type: none"> <li>● Trabajo con exposición : semana 5, 30%</li> <li>● Trabajo con exposición : semana 12, 35%</li> <li>● Trabajo con exposición: semana 16, 35%</li> </ul>	70%
Laboratorios y Guías	El promedio simple entre las notas consideradas corresponde al 100%.	30%

### 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	Resultado(s) de Aprendizaje	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas	Actividad(es) de Trabajo Autónomo
Semana 1		Revisión del programa Acceso al servidor Guía de Linux  <b>Unidad 1: Introducción – Diseño y Análisis de Algoritmos.</b>  1.1 Conceptos matemáticos: Notación “O”, ecuaciones de recurrencia, Teorema Maestro. 1.2 Análisis de algoritmos sencillos.	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 2		1.3 Inducción simple y reforzada. 1.4 Recursividad, dividir para reinar. 1.5 Programación dinámica.	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 3		1.6 Algoritmos avaros (greedy). 1.7 Backtracking	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 4		<b>Unidad 2: Estructuras de Datos Elementales y</b>	Apunte de clases Libros B1, B2	Guía de Laboratorio Guía de Laboratorio

		<b>Tipos de Datos abstractos</b> 2.1 Arreglos  2.2 Matrices	Acceso a un servidor	Laboratorio
Semana 5		2.3 Punteros y listas enlazadas 2.4 Árboles	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 6		<b>Trabajo 1 con exposición</b>		
Semana 7 (Receso)				
Semana 8		2.5 Pilas  2.6 Colas  2.7 Colas de prioridad.	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 9		<b>Unidad 3: Algoritmos de búsqueda y diccionarios</b> 3.1 Árboles de búsqueda	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 10		3.2 Tablas de hash	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 11		<b>Unidad 4: Algoritmos de ordenamiento</b>  4.4 Cota inferior.  4.5 Quicksort.	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 12		<b>Trabajo 2 con. exposición</b>  4.6 Heapsort.	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 13		4.7 Radix Sort. 4.8 Mergesort.	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio
Semana 14		<b>Unidad 5: Algoritmos para grafos</b>	Apunte de clases Libros B1, B2 Acceso a un servidor	Guía de Laboratorio

		<p>5.1 Representaciones para grafos.</p> <p>5.2 Recorridos en profundidad y amplitud.</p>		
Semana 15		<p>5.3 Árbol cobertor mínimo (Kruskal, Prim).</p> <p>5.4 Caminos más cortos (Dijkstra,, Floyd, Warshall)</p>	<p>Apunte de clases Libros B1, B2 Acceso a un servidor</p>	<p>Guía de Laboratorio</p>
Semana 16		<b>Trabajo con exposición</b>		
Semana 17		<b>Examen</b>		
Semana 18		<b>Cierre Actas</b>		