

Programa de Asignatura

1. Identificación Asignatura

Nombre:	Teoría de la Computación		Código:	
Carrera:	Ingeniería Civil Informática	Unidad Académica:	Ciencias Naturales y Tecnología	
Ciclo Formativo:	Ciclo Licenciatura	Línea formativa:	Especializada	
Semestre	IV	Tipo de actividad:	Obligatoria	
N° SCT:	6	Horas Cronológicas Semanales		
		Presenciales:	3	Trabajo Autónomo:
Pre-requisitos	Álgebra Lineal			

2. Propósito formativo

El curso Teoría de la Computación tiene como propósito que los y las estudiantes adquieran los conceptos fundamentales de lenguajes formales, computabilidad y complejidad computacional, estableciendo así algunos límites de la computación. Junto con esto, al finalizar el curso los y las estudiantes serán capaces de utilizar dichas herramientas para modelar y resolver, de forma precisa y rigurosa, una amplia variedad de problemas computacionales.

En su primera parte, la asignatura incorpora el uso de herramientas matemáticas formales para analizar y resolver problemas que involucren elementos discretos, a través de su aplicación con el lenguaje de la lógica formal, con demostraciones, combinatoria y estructuras discretas como relaciones y grafos. Luego, el/la estudiante aprenderá a clasificar lenguajes formales en la jerarquía de regulares, libres del contexto, aceptables y no aceptables, así como la jerarquía de lenguajes P, NP, y NP-completos. Adquirirá herramientas formales para clasificar lenguajes, así como herramientas para resolver problemas prácticos que involucren describir y procesar lenguajes regulares y libres del contexto, particularmente, usando gramáticas y autómatas.

Esta asignatura es un recurso de apoyo fundamental para los temas que impliquen el desarrollo de software utilizando algoritmos eficientes. Además, es prerrequisito del curso Arquitectura de Computadores y de Lenguajes de Programación, correspondientes al semestre VI.

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. Utiliza los conocimientos de las Ciencias Básicas, en el contexto de la Ingeniería, para aplicarlos en el proceso de resolución de problemas complejos.

4. Resultados de aprendizaje específicos

Resultado de Aprendizaje Específico	Criterios de evaluación	Evidencia
RA1. Modela problemas computacionales en término de estructuras discretas y lenguaje de matemática discreta, tales como relaciones, lógica, combinatoria y grafos, con criterios de correctitud y eficiencia computacional.	<p>1.1 Modela problemas reales, traduciéndolos forma clara y coherente, a fórmulas de lógica proposicional y de primer orden.</p> <p>1.2 Traduce o modela problemas, expresándolos en términos de funciones o relaciones.</p> <p>1.3 Modela problemas de conteo y los resuelve considerando técnicas básicas como regla de la suma, del producto, permutaciones y combinaciones.</p>	Laboratorios, guías de ejercicio.
RA2. Distingue y expresa lenguajes regulares en base al uso de autómatas finitos y expresiones regulares, así como los lenguajes libres de contexto, usando autómatas de pila y gramáticas libres del contexto.	<p>2.1. Distingue y expresa lenguajes regulares, mediante el uso de autómatas finitos y expresiones regulares.</p> <p>2.2. Diseña y programa soluciones computacionales simples para problemas prácticos de procesamiento de textos mediante el uso de autómatas finitos.</p>	Laboratorios, guías de ejercicio.
RA3. Diseña y programa soluciones computacionales a problemas de procesamiento de textos mediante el uso de autómatas, utilizando conceptos de lenguajes regulares y libres del contexto.	<p>3.1. Distingue y expresa lenguajes libre del contexto, usando autómatas de pila y gramáticas libres del contexto.</p> <p>3.2. Demuestra que un lenguaje es libre o no, en base al teorema de bombeo y/o usando propiedades de clausura.</p>	Laboratorios, guías de ejercicio.
RA4. Analiza y explica el concepto de complejidad computacional, el significado de las clases de problemas P y NP, demostrando la NP-completitud de problemas mediante el uso de reducciones.	<p>5.1. Analiza y explica el concepto de complejidad computacional y las clases de problemas P, NP, y NP-completos.</p> <p>5.2. Identifica y analiza, a partir de un corpus, ejemplos emblemáticos de lenguajes NP-completos como el problema de satisfactibilidad.</p> <p>5.3 Demuestra mediante reducciones que ciertos lenguajes son NP-completos.</p>	Laboratorios, guías de ejercicio.

5. Unidades de Aprendizaje

<p>1. Introducción a las matemáticas discretas</p> <p>1.1. Lógica</p> <p>1.2. Técnicas de demostración</p> <p>1.3. Relaciones</p> <p>1.4. Combinatoria</p>
<p>2. Lenguajes Regulares y Libres de Contexto</p> <p>2.1. Alfabetos, cadenas y lenguajes.</p> <p>2.2. Representación finita del lenguaje.</p> <p>2.3 Gramáticas Regulares</p> <p>2.4. Autómatas finitos determinísticos y no determinísticos.</p>

- 2.5. Expresiones regulares.
- 2.6. Lema del bombeo.
- 2.7. Gramáticas libres del contexto.
- 2.8. Propiedades de clausura, algorítmicas y de periodicidad.

3. Complejidad Computacional

- 4.1. Longitud de una computación.
- 4.2. Lenguajes y problemas
- 4.3. Complejidad de un problema.
- 4.4. Abstracciones.
- 4.5. Notación O. Las clases P y NP.
- 4.6. Reducción polinomial. - NP-completitud.
- 4.7. El problema NP-completo de satisfactibilidad de fórmulas booleanas.

6. Recursos de Aprendizaje

Bibliografía:

B1: Michael Sipser. Introduction to the theory of computation. 2012 (3rd Edition). Cengage Learning. <http://fuuu.be/polytech/INFOF408/Introduction-To-The-Theory-Of-Computation-Michael-Sipser.pdf>

B2: Ullman, J. D. E. Hopcroft, J. y Motwani, R. (2007). Introducción a la teoría de autómatas, lenguajes y computación (3a. ed.). Pearson Educación. <https://elibro.net/es/lc/uaysen/titulos/52537>

B3: John Hopcroft, Rajeev Motwani, Jeffrey Ullman. Introduction to the automata theory, languages and computation. 3Rd Edition. Pearson, 2006. [http://ce.sharif.edu/courses/94-95/1/ce414-2/resources/root/Text%20Books/Automata/John%20E.%20Hopcroft.%20Rajeev%20Motwani.%20Jeffrey%20D.%20Ullman-Introduction%20to%20Automata%20Theory.%20Languages.%20and%20Computations-Prentice%20Hall%20\(2006\).pdf](http://ce.sharif.edu/courses/94-95/1/ce414-2/resources/root/Text%20Books/Automata/John%20E.%20Hopcroft.%20Rajeev%20Motwani.%20Jeffrey%20D.%20Ullman-Introduction%20to%20Automata%20Theory.%20Languages.%20and%20Computations-Prentice%20Hall%20(2006).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 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	Gabriel Núñez Vivanco		
Contacto	gabriel.nunez@uaysen.cl		
Año	2024	Periodo Académico	1
Horario clases	Jueves 10:15-13:30 hrs	Horario de atención estudiantes	Viernes 8:30 grs
Sala / Campus			

9. Metodología de Trabajo:

La asignatura contiene:			
Actividades de vinculación con el medio	N o	Actividades relacionadas con proyectos de investigación	N o
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
<p>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:</p> <ul style="list-style-type: none"> • Laboratorio: evaluación individual o grupal, que se realiza de manera presencial. • Tarea: Evaluación grupal que se realiza fuera del horario de clases. <p>Al final del semestre, se eliminarán las dos peores evaluaciones menores.</p> <p>Como parte de la nota de presentación del curso, los estudiantes en grupo deberán hacer una exposición de un proyecto, considerando su análisis, diseño e implementación de solución en diagrama de flujo. Se entregará un instructivo específico para esta evaluación.</p>			

Finalmente, se contempla la realización de **pruebas de cátedras**, que son individuales, escritas y que ocupan toda una clase. En cada una se evaluará distintos contenidos asociados a las unidades de aprendizaje de la asignatura.

10. Evaluaciones:

Evaluación	Ponderaciones específicas	Ponderación nota presentación
Pruebas de cátedra	<ul style="list-style-type: none"> Prueba 1 (P1, Unidad I): 50% Prueba 2 (P2, Unidad II): 50% 	50%
Evaluaciones menores (tareas y laboratorios)	El promedio simple entre las notas consideradas corresponde al 100% (Unidad I y II. Tareas y laboratorios semanales)	20%
Proyecto	Nota única (acorde a pauta específica)	30%

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:

- 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	Resultado(s) de Aprendizaje	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas	Actividad(es) de Trabajo Autónomo
1	Presentación Programa y detalles de implementación			Presentación y discusión inicial.
2	RA1	Unidad I: Introducción a las matemáticas	BO.1, BS.1	Clases Expositivas, Tareas, Ejercicios

		discretas 1.1. Lógica		
3	RA1	Unidad I: Introducción a las matemáticas discretas 1.2. Técnicas de demostración	BO.1, BS.1	Clases Expositivas, Tareas, Ejercicios
4	RA1	Unidad I: Introducción a las matemáticas discretas 1.3. Relaciones	BO.1, BS.1	Clases Expositivas, Tareas, Ejercicios
5	RA1	Unidad I: Introducción a las matemáticas discretas 1.4. Combinatoria	BO.1, BS.1	Clases Expositivas, Tareas, Ejercicios
6	RA1	Unidad I: Introducción a las matemáticas discretas Repaso Unidad I	BO.1, BS.1	Clases Expositivas, Tareas, Ejercicios
7	Prueba 1	Prueba 1	Prueba 1	Prueba 1
8	RA2	Unidad II: Lenguajes Regulares y Libres de Contexto 2.1. Alfabetos, cadenas y lenguajes. 2.2. Representación finita del lenguaje. 2.3 Gramáticas Regulares	BO.2, BS.1	Clases Expositivas, Tareas, Ejercicios
9	RA2, RA3	Unidad II: Lenguajes Regulares y Libres de Contexto 2.4. Autómatas finitos determinísticos y no determinísticos.	BO.2, BS.1	Taller grupal
10	RA2,RA3	Unidad II: Lenguajes Regulares y Libres de Contexto	BO.2, BS.1	Taller Grupal

		2.5. Expresiones regulares. 2.6. Lema del bombeo.		
11	RA2,RA3	Unidad II: Lenguajes Regulares y Libres de Contexto 2.7. Gramáticas libres del contexto. 2.8. Propiedades de clausura, algorítmicas y de periodicidad.	BO.2, BS.1	Exposición Taller
12	Semana de Estudio Autónomo	Semana de Estudio Autónomo	Semana de Estudio Autónomo	Semana de Estudio Autónomo
13	Prueba 2	Prueba 2	Prueba 2	Prueba 2
14	RA4	Unidad III: Complejidad Computacional 4.1. Longitud de una computación. 4.2. Lenguajes y problemas	BO.2, BS.1	Clases Expositivas, Tareas, Ejercicios
15	RA4	Unidad III: Complejidad Computacional 4.3. Complejidad de un problema. 4.4. Abstracciones.		Clases Expositivas, Tareas, Ejercicios
16	RA4	Unidad III: Complejidad Computacional 4.5. Notación O. Las clases P y NP. 4.6. Reducción polinomial. - NP-completitud.	BO.2, BS.1	Clases Expositivas, Tareas, Ejercicios
17	Prueba 3	Prueba 3	Prueba 3	Prueba 3
18		Semana de exámenes		Examen final