# Programa de Asignatura

## 1. Identificación Asignatura

<table>
<tr><td>Nombre:</td><td colspan="3">Teoría de la Computación</td><td colspan="2">Código:</td><td></td></tr>
<tr><td>Carrera:</td><td colspan="2">Ingeniería Civil Informática</td><td>Unidad Académica:</td><td colspan="2">Ciencias Naturales y Tecnología</td></tr>
<tr><td>Ciclo Formativo:</td><td colspan="2">Ciclo Licenciatura</td><td>Línea formativa:</td><td colspan="2">Especializada</td></tr>
<tr><td>Semestre</td><td colspan="2">IV</td><td>Tipo de actividad:</td><td colspan="2">Obligatoria</td></tr>
<tr><td rowspan="2">N° SCT:</td><td rowspan="2">6</td><td colspan="4">Horas Cronológicas Semanales</td></tr>
<tr><td>Presenciales:</td><td>3</td><td>Trabajo Autónomo:</td><td>6</td></tr>
<tr><td>Pre-requisitos</td><td colspan="5">Álgebra Lineal</td></tr>
</table>

## 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 involucran 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. | 1.1 Modela problemas reales, traduciéndolos forma clara y coherente, a fórmulas de lógica proposicional y de primer orden.<br>1.2 Traduce o modela problemas, expresándolos en términos de funciones o relaciones.<br>1.3 Modela problemas de conteo y los resuelve considerando técnicas básicas como regla de la suma, del producto, permutaciones y combinaciones. | 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. | 2.1. Distingue y expresa lenguajes regulares, mediante el uso de autómatas finitos y expresiones regulares.<br>2.2. Diseña y programa soluciones computacionales simples para problemas prácticos de procesamiento de textos mediante el uso de autómatas finitos. | 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. | 3.1. Distingue y expresa lenguajes libre del contexto, usando autómatas de pila y gramáticas libres del contexto.<br>3.2. Demuestra que un lenguaje es libre o no, en base al teorema de bombeo y/o usando propiedades de clausura. | 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. | 5.1. Analiza y explica el concepto de complejidad computacional y las clases de problemas P, NP, y NP-completos.<br>5.2. Identifica y analiza, a partir de un corpus, ejemplos emblemáticos de lenguajes NP-completos como el problema de satisfactibilidad.<br>5.3 Demuestra mediante reducciones que ciertos lenguajes son NP-completos. | Laboratorios, guías de ejercicio. |

## 5. Unidades de Aprendizaje

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

| |
|---|
| 2.5. Expresiones regulares.<br>2.6. Lema del bombeo.<br>2.7. Gramáticas libres del contexto.<br>2.8. Propiedades de clausura, algorítmicas y de periodicidad. |
| **3. Complejidad Computacional**<br>4.1. Longitud de una computación.<br>4.2. Lenguajes y problemas<br>4.3. Complejidad de un problema.<br>4.4. Abstracciones.<br>4.5. Notación O. Las clases P y NP.<br>4.6. Reducción polinomial. - NP-completitud.<br>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

**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).