

Programa de Asignatura

1. Identificación Asignatura

Nombre:	Teoría de la Computación				Código:	IF1012	
Carrera:	Ingeniería Civil Informática		Unidad Académica:		Ciencias Naturales y Tecnología		
Ciclo Formativo:	Ciclo Licenciatura		Línea formativa:		Especializada		
Semestre	V (Quinto)		Tipo de actividad:		Obligatoria		
N° SCT:	6	Horas Cronológicas Semanales					
		Presenciales:	3	Trabajo Autónomo:		3	
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:

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

- para estudiar su eficiencia en virtud de distintas plataformas y lenguajes utilizados.
- 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. Modelar problemas reales, traduciéndose de forma clara y coherente, a fórmulas de lógica proposicional y de primer orden. 1.2. Traduce o modela problemas, expresándose en términos de funciones o relaciones. 1.3. Modela problemas de conteo y los resuelve considerando técnicas básicas como regla de la suma, del producto, permutaciones y combinaciones. 1.4. Modela y resuelve problemas usando los métodos de la teoría de grafos.	Laboratorios, guías de ejercicio y actividades.
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. 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 y actividades.
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 libres del contexto, usando autómatas de pila y gramáticas libres del contexto. 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 y actividades.
RA4. Demuestra formalmente, mediante reducciones, que ciertos problemas no se pueden resolver computacionalmente, clasificándolos en aceptables y no aceptables, extendiendo la validez de estas conclusiones a otros modelos de computación a partir de la tesis de Church.	4.1. Utiliza Máquinas de Turing básicas y extendidas, determinísticas y no determinísticas, para calcular funciones y reconocer lenguajes. 4.2. Analiza y argumenta cómo la Tesis de Church da validez universal a resultados demostrados con Máquinas de Turing y que es extensible su validez a otros modelos de computación.	Laboratorios, guías de ejercicio y actividades.

RA5. 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. 5.2. Identifica y analiza, a partir de un corpus, ejemplos emblemáticos de lenguajes NP-completos y sus problemas característicos. 5.3. Demuestra mediante reducciones que ciertos lenguajes son NP-completos.	Laboratorios, guías de ejercicio y actividades.
---	--	--

5. Unidades de Aprendizaje

1. Introducción a las matemáticas discretas 1.1. Lógica 1.2. Técnicas de demostración 1.3. Relaciones 1.4. Combinatoria
2. Lenguajes Regulares y Libres de Contexto 2.1. Alfabetos, cadenas y lenguajes. 2.2. Representación finita del lenguaje. 2.3 Gramáticas Regulares 2.4. Autómatas finitos determinísticos y no determinísticos. 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. Maquinas de Turing y Computabilidad 3.1. Máquinas de Turing (MTs). 3.2. MTs no determinísticas y su simulación 3.3. La Tesis de Church y otros formalismos. 3.4. Equivalencia con el modelo RAM. 3.5. Gramáticas dependientes del contexto. 3.6. Equivalencia con MTs. Numerabilidad.
4. 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	Roberto Rivera Mardones		
Contacto	roberto.rivera@uaysen.cl		
Año	2023	Periodo Académico	Primer Semestre
Horario clases	Jueves 10:15 a 13:30 Hrs.	Horario de atención estudiantes	Miércoles 10:00 a 11:00 Hrs.
Sala / Campus	Sala de Computación - Campus Río Simpson (Obispo Vielmo N°62)		

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: <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr style="background-color: #d3d3d3;"> <th style="width: 20%;">Actividad Docente</th> <th style="width: 45%;">Descripción</th> <th style="width: 15%;">Intervención del profesor</th> <th style="width: 20%;">Requerimiento Sala</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Exposición conceptual</td> <td>El profesor introduce de forma expositiva conceptos preliminares y necesarios a las actividades de índole práctica, buscando la reflexión y opinión en torno a estos conceptos. Se dispone de materiales complementarios en la plataforma Ucampus.</td> <td style="text-align: center;">Alta</td> <td style="text-align: center;">Sala de Computación y Acceso a UCampus</td> </tr> <tr> <td style="text-align: center;">Programación expositiva</td> <td>El profesor profundiza en la comprensión de elementos conceptuales a través de la exposición directa de la resolución de problemas de programación como ejemplos.</td> <td style="text-align: center;">Alta</td> <td style="text-align: center;">Sala de Computación y Acceso a UCampus</td> </tr> <tr> <td style="text-align: center;">Programación tutorial</td> <td>Funciona como la programación expositiva, pero el profesor realiza pausas para que los alumnos completen “pasos requeridos” antes de continuar. El objetivo es completar un paso definido por el profesor antes de continuar al siguiente.</td> <td style="text-align: center;">Media</td> <td style="text-align: center;">Sala de Computación con equipos habilitados</td> </tr> <tr> <td style="text-align: center;">Actividad práctica / Programación autónoma</td> <td>Los estudiantes abordan y resuelven problemas de programación de forma autónoma, algunas con guía y apoyo docente, otras personalmente o en grupo.</td> <td style="text-align: center;">Baja/Media</td> <td style="text-align: center;">Sala de Computación con equipos habilitados</td> </tr> </tbody> </table>				Actividad Docente	Descripción	Intervención del profesor	Requerimiento Sala	Exposición conceptual	El profesor introduce de forma expositiva conceptos preliminares y necesarios a las actividades de índole práctica, buscando la reflexión y opinión en torno a estos conceptos. Se dispone de materiales complementarios en la plataforma Ucampus.	Alta	Sala de Computación y Acceso a UCampus	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 Computación y Acceso a UCampus	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 completar un paso definido por el profesor antes de continuar al siguiente.	Media	Sala de Computación con equipos habilitados	Actividad práctica / Programación autónoma	Los estudiantes abordan y resuelven problemas de programación de forma autónoma, algunas con guía y apoyo docente, otras personalmente o en grupo.	Baja/Media	Sala de Computación con equipos habilitados
Actividad Docente	Descripción	Intervención del profesor	Requerimiento Sala																				
Exposición conceptual	El profesor introduce de forma expositiva conceptos preliminares y necesarios a las actividades de índole práctica, buscando la reflexión y opinión en torno a estos conceptos. Se dispone de materiales complementarios en la plataforma Ucampus.	Alta	Sala de Computación y Acceso a UCampus																				
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 Computación y Acceso a UCampus																				
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 completar un paso definido por el profesor antes de continuar al siguiente.	Media	Sala de Computación con equipos habilitados																				
Actividad práctica / Programación autónoma	Los estudiantes abordan y resuelven problemas de programación de forma autónoma, algunas con guía y apoyo docente, otras personalmente o en grupo.	Baja/Media	Sala de Computación con equipos habilitados																				
En cualquier semana del semestre en curso se podría realizar una evaluación sobre las temáticas estudiadas a la fecha. Esta evaluación 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 Actividades: Evaluación individual que se realiza en computador durante el tiempo de trabajo autónomo. Prueba de Cátedra: 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	- Prueba Unidad 1 y 2 (P1): 50% - Prueba Unidad 3 y 4 (P2): 50%	50%
Laboratorio y Guía de Actividades	- Guía de Actividades 1 (A1): 25% - Guía de Actividades 2 (A2): 25% - Laboratorio 1 (L1): 25% - Laboratorio 2 (L1): 25%	30%
Proyecto Semestral	Proyecto Semestral (P1)	20%

Calificación Final:

- Nota de presentación: 70%
- Exámen 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 algún tipo para trasladar el requerimiento a la coordinación del programa. Adicionalmente los y las estudiantes deberán acceder al material disponible en UCampus que el profesor preparó para este propósito. Es deber de las y los estudiantes mantenerse informados 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
1	RA1	Unidad 1: Introducción a las matemáticas discretas 1.1. Lógica 1.2. Técnicas de demostración	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
2	RA1	Unidad 1: Introducción a las matemáticas discretas 1.3. Relaciones 1.4. Combinatoria Guía de Actividades N°1: Se analizan problemas que estudiantes deben resolver según lo aprendido en clases de cátedra.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
3	RA2 y RA3	Unidad 2: Lenguajes Regulares y Libres de Contexto 2.1. Alfabetos, cadenas y lenguajes. 2.2. Representación finita del lenguaje. 2.3 Gramáticas Regulares	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
4	RA2 y RA3	Unidad 2: Lenguajes Regulares y Libres de Contexto 2.4. Autómatas finitos determinísticos y no determinísticos.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
5	RA2 y RA3	Unidad 2: Lenguajes Regulares y Libres de Contexto 2.5. Expresiones regulares. 2.6. Lema del bombeo. Guía de Actividades N°2: Se analizan problemas que estudiantes deben resolver según lo aprendido en clases de cátedra.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
6	RA2 y RA3	Unidad 2: Lenguajes Regulares y Libres de Contexto 2.7. Gramáticas libres del contexto. 2.8. Propiedades de clausura, algorítmicas y de periodicidad.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.

7	RA1, RA2 y RA3	Prueba de Cátedra Unidad 1 y 2	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
8	RA4	Unidad 3: Máquinas de Turing y Computabilidad 3.1. Máquinas de Turing (MTs). 3.2. MTs no determinísticas y su simulación	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Sugerencia: película The Imitation Game , año 2014 (Contexto histórico y base de antecedentes científicos). Revisión también de bibliografía del curso.
9	RA4	Unidad 3: Máquinas de Turing y Computabilidad 3.3. La Tesis de Church y otros formalismos.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
10	RA4	Unidad 3: Máquinas de Turing y Computabilidad 3.4. Equivalencia con el modelo RAM. 3.5. Gramáticas dependientes del contexto.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
11	RA4	Unidad 3: Máquinas de Turing y Computabilidad 3.6. Equivalencia con MTs. Numerabilidad. Laboratorio N°1: Se analizan problemas que estudiantes deben resolver según lo aprendido en clases de cátedra.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
12	RA5	Unidad 4: Complejidad Computacional 4.1. Longitud de una computación. 4.2. Lenguajes y problemas	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
13	RA5	Unidad 4: Complejidad Computacional 4.3. Complejidad de un problema. 4.4. Abstracciones.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
14	RA5	Unidad 5: Complejidad Computacional 4.5. Notación O. Las clases P y NP. 4.6. Reducción polinomial. - NP-completitud.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.

		Laboratorio N°2: Se analizan problemas que estudiantes deben resolver según lo aprendido en clases de cátedra.		
15	RA5	Unidad 5: Complejidad Computacional 4.7. El problema NP-completo de satisfactibilidad de fórmulas booleanas.	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	Lectura de material utilizado en clases y disponible en UCampus. Revisión de bibliografía del curso.
16	RA4 y RA5	Prueba de Cátedra Unidad 3 y 4	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	
17	RA1, RA2, RA3, RA4 y RA5	Entrega Proyecto Semestral	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	
18	RA1, RA2, RA3, RA4 y RA5	Exámen	Se utilizarán equipos de la sala de computación y ejercicios de libros presentes en la bibliografía.	