



PROGRAMA DE ASIGNATURA

I. IDENTIFICACIÓN

Carrera o programa: Ingeniería en Tecnologías de Información

Unidad responsable: Escuela de Ingeniería

Nombre de la asignatura: Fundamentos de Ciencias de la Computación

Código: ECIN-00563

Semestre en la malla¹: 5

Créditos SCT – Chile: 5

Ciclo de Formación	Básico		Profesional	X
Tipo de Asignatura	Obligatoria	X	Electiva	

Clasificación de área de conocimiento²

Área: Ingeniería y Tecnología **Subárea:** Ingeniería Informática

Requisitos:

Prerrequisitos:

- Programación Orientada a Objeto

Requisitos para:

- Proyecto Desarrollo Ágil

II. ORGANIZACIÓN SEMESTRAL

Horas Dedicación Semanal (Cronológicas)	Docencia Directa	4.5	Trabajo Autónomo	3.5	Total	8	
Detalle Horas Directas	Cátedra	Ayudantía	Laboratorio	Taller	Terreno	Exp. Clínica	Supervisión
	3		1.5				

III. APORTE AL PERFIL DE EGRESO

La asignatura contribuye al dominio 1 del perfil de egreso, 'Conocimiento científico y disciplinario'. Además, contribuye al dominio 4, 'Habilidades para la Práctica de la Ingeniería'. Al finalizar la asignatura, los estudiantes serán capaces de aplicar los fundamentos matemáticos y de la teoría de autómatas y lenguajes para modelar y resolver problemas en el contexto de las ciencias de la computación.

IV. COMPETENCIAS

La carrera declara las siguientes habilidades:

- 1.1. Aplicar conocimientos de matemáticas y ciencias naturales a la solución de problemas complejos de ingeniería.
- 4.4. Diseñar soluciones que involucren, por ejemplo, aplicaciones TI, infraestructura TI, toma de decisiones, gestión de datos y gestión de proyectos.

V. RESULTADOS DE APRENDIZAJE

¹ Este campo sólo se completa en caso de carreras con programas semestrales.

² Clasificación del curso de acuerdo a la OCDE



1. Calcular las operaciones relacionadas con conjuntos, relaciones, funciones y lenguajes.
2. Diseñar modelos de situaciones de la vida real o computacionales mediante lógica simbólica.
3. Aplicar las técnicas de demostración argumentando una sentencia.
4. Determinar las variables y recurrencia que representan una característica de un problema computacional.
5. Diseñar un reconocedor y generador de lenguajes regulares aplicando la teoría de autómatas finitos.
6. Diseñar un reconocedor y generador de lenguajes libre de contexto aplicando la teoría de lenguajes formales.
7. Identificar los límites de la computabilidad a través del modelo Máquina de Turing.

VI. ÁREAS TEMÁTICAS

1. Conjuntos, Relaciones, Lenguajes
 - 1.1 Conjuntos
 - 1.2 Relaciones
 - 1.3 Funciones
 - 1.4 Clausuras
 - 1.5 Técnicas fundamentales de demostración
 - 1.6 Alfabetos y Lenguajes
 - 1.7 Ejemplo de aplicaciones en el diseño de lenguajes de programación (tipos de datos, lenguajes funcionales)
2. Fundamentos de Lógica
 - 2.1 Lógica proposicional
 - 2.2 Reglas de inferencia proposicional
 - 2.3 Lógica de predicados
 - 2.4 Limitaciones de lógica proposicional y de predicados
 - 2.5 Ejemplo de Aplicación en Inteligencia Artificial (agentes que razonan lógicamente)
3. Teoría del Conteo
 - 3.1 Argumentos del conteo
 - 3.2 Permutaciones y combinaciones
 - 3.3 Teorema Binomial
 - 3.4 Números de Fibonacci
 - 3.5 Solución de ecuaciones de recurrencia
 - 3.6 Aritmética modular básica
 - 3.7 Aplicaciones al análisis de algoritmos
4. Autómatas Finitos
 - 4.1 Autómatas Finitos Determinísticos (AFD)
 - 4.2 Autómatas Finitos No Determinísticos (AFND)
 - 4.3 Expresiones Regulares (ER)
 - 4.4 Equivalencia AFD – AFND – ER
 - 4.5 Demostración que un lenguaje es o no es regular.
 - 4.6 Ejemplo de aplicación a la construcción de un analizador léxico
5. Lenguajes Libre de Contexto
 - 5.1 Gramáticas Libre de Contexto (GLC)



- 5.2 Lenguajes Regulares y Lenguajes Libre de Contexto (LLC)
- 5.3 Autómatas Pushdown
- 5.4 Autómatas Pushdown y GLCs
- 5.5 Propiedades de Clausura de LLC
- 5.6 Parsing (top-down y bottom-up)
- 5.7 Ejemplo de aplicación a la construcción de un analizador de sintaxis

- 6. Máquinas de Turing
 - 6.1 Definición de una Máquina de Turing (MT)
 - 6.2 Cálculos con MTs
 - 6.3 Combinación de MTs
 - 6.4 Extensiones de las MTs
 - 6.5 MT No Determinísticas
 - 6.6 La Jerarquía de Chomsky

VII. ORIENTACIONES METODOLÓGICAS

1. La metodología a desarrollar en esta asignatura debe favorecer la interacción entre las y los estudiantes a través de trabajos prácticos colaborativos que permitan la solución a problemas específicos contextualizados a la asignatura.
 - Se sugiere el uso de clases expositivas y participativas con un método combinado, es decir, clases expositivas alternadas con trabajos en grupo de corta duración para responder preguntas.
 - Se sugiere la utilización de la metodología activa de análisis de casos para desarrollar experiencias que permitan incorporar los elementos teórico-prácticos asociados a los resultados de aprendizaje de la asignatura.
2. Las experiencias de cátedra/laboratorio/taller deben ser realizadas por medio de la utilización de software moderno aplicable a la asignatura.
3. Se recomienda que las y los estudiantes realicen presentaciones periódicas sobre el trabajo realizado que incluya: contextualización, desarrollo y conclusiones.
4. Actividades prácticas recomendadas: cápsulas teóricas, reuniones de trabajo, taller de trabajo en equipo y liderazgo, presentaciones e informes escritos de avance en español, revisión del estado del arte asociado al problema, lluvia de ideas, análisis de alternativas y descripción detallada de la solución.

VIII. ORIENTACIONES Y CRITERIOS PARA EVALUACIÓN

1. Se recomienda la aplicación de una evaluación diagnóstica al inicio de la asignatura.
2. La asignatura podría contemplar dos instancias de evaluación de los resultados de aprendizaje: cátedra y taller/laboratorio.
 - En el caso de existir, ambas debieran ser aprobadas por separado: el porcentaje de cada una de ellas deberá ser de 60% para cátedra y 40% para taller/laboratorio.
 - En el caso que la asignatura tenga actividades de taller/laboratorio, éstas deben ser realizadas en grupos de estudiantes y se recomienda la elaboración por parte de los estudiantes de un informe sobre la actividad desarrollada.
3. Se evaluará el conocimiento conceptual y procedimental mediante la realización de al menos dos pruebas sumativas de carácter presencial.
 - Se recomienda además la aplicación de una evaluación mediante la entrega de un trabajo desarrollado en las horas indirectas asociadas a la asignatura.



- Se recomienda que las y los estudiantes realicen una o más presentaciones de los trabajos realizados, la evaluación de la misma debe ser por medio de la aplicación de una rúbrica.
- 4. Se recomienda realizar evaluaciones de carácter formativo. Esto permite al docente introducir correcciones, añadir alternativas y reforzar los aspectos para ayudar al estudiantado en el logro de sus habilidades.
- 5. La asistencia y condiciones de aprobación de la asignatura debe ser acorde a la aplicación del Reglamento de Docencia de Pregrado.

IX. RECURSOS BIBLIOGRÁFICOS

Bibliografía Mínima

- Rosen, K. H., Krithivasan, K. (2013). Discrete Mathematics and Its Applications. Singapur: McGraw-Hill.
- Sipser, M. (2013). Introduction to the Theory of Computation. Italia: Cengage Learning.

Bibliografía Complementaria

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). Introduction to Algorithms. India: MIT Press.
- Bender, E. A., Williamson, S. G. (2005). Mathematics for Algorithm and Systems Analysis. Estados Unidos: Dover Publications.
- Johnsonbaugh, R. (2005). MATEMÁTICAS DISCRETAS. México: Pearson Educación.
- Kolman, B., Busby, R. C., Ross, S. (1999). Estructuras de matemáticas discretas para la computación. México: Pearson Educación.
- Graham, R. L., Knuth, D. E., Patashnik, O. (1994). Concrete Mathematics: A Foundation for Computer Science. Alemania: Addison-Wesley.
- Knuth, D. E. (1998). The art of computer programming. Alemania: Addison-Wesley.