

PROGRAMA DE ASIGNATURA

Nombre	Computación I	
Carrera	Ingeniería Matemática	
Código	22146	
Créditos SCT-Chile	5 Sct	<i>Tbjo. Directo: 6 hrs. pedag. – Tbjo. Autónomo: 6 hrs. cronolog.</i>
Nivel	2do semestre	
Requisitos	Ingreso	
Categoría	Obligatorio	
Área de conocimiento según OCDE	Ingeniería y Tecnología	
Descripción	Contribución al Perfil de Egreso La asignatura contribuye a los siguientes desempeños integrales: 1. Desarrollar constructos matemáticos teóricos y prácticos para estudiar problemas que surgen del ámbito académico o profesional, utilizando herramientas matemáticas avanzadas y el pensamiento abstracto y/o estructurado. 2. Formular modelos matemáticos complejos para estudiar problemas pertenecientes a otras disciplinas, interactuando en contextos interdisciplinarios o multidisciplinarios, con una actitud crítica frente a distintas situaciones de aplicación. 3. Resolver problemas de la ingeniería y las ciencias mediante la aplicación de conceptos avanzados de matemática y herramientas computacionales y/o tecnológicas para generar soluciones que ayuden a la toma de decisiones en distintos contextos laborales, actuando con rigurosidad en el proceso. 4. Interpretar los resultados obtenidos de la resolución de problemas, analizando la información desde una perspectiva cualitativa y cuantitativa para la toma de decisiones en función de los distintos contextos de aplicación, con responsabilidad y ética en el quehacer profesional.	
	Resultado de aprendizaje general Al finalizar el curso, el estudiante será capaz de conocer los problemas y los algoritmos clásicos de la computación. Conocer los conceptos y las técnicas para analizar la correctitud y el desempeño de algoritmos. Diseñar algoritmos para resolver problemas computacionales básicos.	
	Resultados de aprendizaje específicos	Unidades temáticas
	• Conocer las nociones básicas de la teoría de algoritmos y complejidad. • Conocer algunos de los algoritmos básicos y sus desempeños. • Reconocer algunos elementos básicos de algoritmos como expresiones condicionales, repeticiones, etc.	Unidad N°1: Computación y algoritmos 1. Definiciones generales 2. Relación con el álgebra y el cálculo 3. Conceptos básicos 4. Cultura general en relación con computación y algoritmos

	- Entender las expresiones y conectores lógicos (“y”, “o”, “no”, “si, entonces”, “si y sólo si”). - Conocer las expresiones de algoritmos: condicionales (“if”, “if else”) y bucles (“for”, “while”) y sus operaciones de control (“break”, iteradores). - Reconocer el concepto de funciones/procesos en programación. - Identificar la diferencia entre variables locales y globales y entre variables de entrada y salida. - Conocer la diferencia entre algoritmos directos, iterativos y algoritmos recursivos y como se refleja en la estructura de un algoritmo.	Unidad N°2: Diseño y control de programas - Lógica computacional - Sentencias de control: if/else, for, while, break, iteradores - Uso de funciones - Variables: globales, locales, entradas y salidas, utilización de archivos - Estructuras de algoritmos: algoritmos directos, algoritmos iterativos y recursividad general
	- Conocer las estructuras básicas encontradas en programación y cómo se pueden utilizar. - Distinguir los ventajas y desventajas de las estructuras de datos básicas en algunas situaciones comunes.	Unidad N°3: Estructuras de datos - Arreglos - Matrices - Listas
	- Entender la necesidad de demostrar la correctitud de un algoritmo. - Saber demostrar la correctitud de algoritmos básicos. - Tener conocimientos sobre la medición de la eficacia y eficiencia de los algoritmos para ofrecer soluciones de buen rendimiento.	Unidad N°4: Eficiencia y correctitud de algoritmos. - Correctitud. - Eficiencia temporal. - Eficiencia espacial. - Medición de error y eficiencia computacional.
	- Aplica la noción fundamental de algoritmos de descenso, basado en la búsqueda sobre una dirección dada. - Identifica los distintos tipos de convergencia: lineal, superlineal, cuadrática, a través de la revisión de métodos específicos.	Unidad N°5: Métodos numéricos para optimización - Método de direcciones admisibles (caso restricciones lineales). 6.2. Método de penalización/barrera.

	Laboratorios (aprendizaje) • Saber implementar algunos algoritmos básicos en el lenguaje de programación Python. • Conocer las etapas claves en la programación de algoritmos (diseño, verificación de correctitud, medición de eficacia y eficiencia, testeo) y aplicarlas en casos prácticos. • Saber utilizar funciones para la programación de un algoritmo. • Entender y aplicar herramientas básicas de recursividad en programación.	Laboratorios (aprendizaje) • Saber implementar algunos algoritmos básicos en el lenguaje de programación Python. • Conocer las etapas claves en la programación de algoritmos (diseño, verificación de correctitud, medición de eficacia y eficiencia, testeo) y aplicarlas en casos prácticos. • Saber utilizar funciones para la programación de un algoritmo. • Entender y aplicar herramientas básicas de recursividad en programación.
	Metodologías de enseñanza y de aprendizaje La metodología contempla clases expositivas, resolución plenaria de problemas, y aplicación guías de aprendizaje, lo cual se trabajará de forma individual o colaborativa. El trabajo autónomo se desarrollará en base a tareas e implementaciones de algoritmos, las cuales pueden ser resueltas de forma individual y grupal.	
	Procedimientos de evaluación Los procedimientos de evaluación del curso contemplan evaluaciones diagnósticas, formativas y sumativas, alineadas con el contenido de las unidades y los resultados de aprendizaje definidos. Las actividades evaluativas concretas para realizar durante el curso serán claramente definidas al inicio de este.	

	Bibliografía básica ● Parker, J.R. (2016) <i>Python. An Introduction to Programming</i>. Mercury Learning & Information, Berlin ● Üçoluk, G; Kalkan, S. (2012) <i>Introduction to Programming Concepts with Case Studies in Python</i>. Springer, Vienna. ● Cormen, T.H.; Leiserson, C.E.; Rivest, R.L. (1994) <i>Introduction to Algorithms</i>. MIT Press / McGraw-Hill, Cambridge. ● Yang, X.-S. (2014) <i>Introduction to Algorithms</i>. Elsevier, EE.-UU. Bibliografía avanzada:
--	---