



Programa de Asignatura

I. IDENTIFICACIÓN				
Carrera o programa: Ingeniería Civil en Computación e Informática				
Unidad responsable: Escuela de Ingeniería				
Nombre de la asignatura: Estructuras de Datos				
Código: ECIN-00407				
Semestre en la malla ¹ : 4				
Créditos SCT - Chile: 5				
Ciclo de Formación	Básico	X	Profesional	
Tipo de Asignatura	Obligatoria	X	Electiva	
Clasificación de área de conocimiento ²				
Área: Ingeniería y Tecnología		Sub área: Ingeniería Informática		
Requisitos:				
Pre-requisitos:		Requisito para:		
• ECIN-00307 Programación Orientada a Objetos • ECIN-00308 Técnicas y metodologías de Programación Avanzada		• ECIN-00512 Introducción a Data Science • ECIN-00507 Bases de Datos • ECIN-00610 Ingeniería de Software • ECIN-00609 Lenguajes de Programación		

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			

¹Este campo

²Clasificación del curso de acuerdo a la OCDE



III. APOORTE AL PERFIL DE EGRESO

La asignatura contribuye al dominio 1 del perfil de egreso, “Conocimiento científico y disciplinario”. Además, contribuye al dominio 2 “Habilidades y Actitudes Personales y Profesionales”. También contribuye al dominio 4 “Habilidades para la Práctica de la Ingeniería”. También contribuye al dominio 3 “Habilidades Interpersonales”. Al finalizar la asignatura las y los estudiantes serán capaces de seleccionar, diseñar e implementar estructuras de datos y sus algoritmos asociados, para la resolución de problemas específicos, teniendo en cuenta criterios como el tamaño del problema, eficiencia en el uso de recursos y complejidad algorítmica.

IV. HABILIDADES PERFIL DE EGRESO (RELACIÓN)

1.3 Aplicar conocimientos, métodos y herramientas de la especialidad para resolver problemas complejos de Ingeniería de Software, Plataformas y Gestión de Tecnologías.

2.1 Identificación, formulación, modelación y resolución de problemas complejos de ingeniería considerando las interacciones y la dinámica de las variables.

3.1 Liderar y trabajar en equipos de trabajo multidisciplinarios

4.3 Concebir las soluciones TIC requeridas en las organizaciones haciendo uso eficiente de los recursos como personas, información, y procesos. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.

4.4 Diseñar las soluciones TIC para la industria intensiva en procesamiento de la información. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.

4.5 Implementar las soluciones TIC. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.

CG1 Capacidad de Autoaprendizaje: habilidades de tipo metacognitivo que permiten descubrir y aprovechar constantemente oportunidades de aprendizaje, teniendo una permanente actitud de superación y espíritu inquisitivo, logrando con ellos la motivación necesaria para



construir nuevos conocimientos y transferirlos a nuevas situaciones. El egresado de la UCN se hace cargo de construir su destino académico con Libertad, valiéndose de las oportunidades de espacio, contexto y relaciones que se generan en el entorno educativo.

CG6 Trabajo en equipo: Capacidad de desarrollar labores tipo grupal, facilitando el despliegue de las habilidades de sus integrantes, promoviendo el compromiso y un clima respetuoso, que permita facilitar la sinergia con personas de diferentes disciplinas y/o culturas, con el propósito de alcanzar metas colectivas. El egresado UCN, en la interacción del trabajo colaborativo, respeta la diversidad de pensamiento de opinión, de expresión y de conciencia que se genera en la relación con otros, en consecuencia, con el valor de la libertad.

CG7 Uso eficiente de las tecnologías: Competencias informacionales (CI2) que permiten a la persona reconocer cuando se requiere información, como acceder a ella, como evaluarla, seleccionarla y utilizarla de manera responsable, haciendo uso de las TIC. El egresado UCN, genera, transmite y difunde conocimiento con transparencia y honestidad en coherencia con el valor de la verdad. El sentido ético que lo mueve es no poner en riesgo a la “persona en su dignidad”.

V. RESULTADOS DE APRENDIZAJE

1. Programar con un enfoque orientado a objetos verificando que las prueba de los programas satisfacen los requerimientos del usuario
2. Calcular la complejidad algorítmica para algoritmos no recursivos mediante la técnica de conteo en el análisis del algoritmo y la función $O(g(n))$, en el peor y en el mejor caso
3. Programar estructuras de datos básicas utilizando listas enlazadas
4. Programar estructuras de datos utilizando árboles
5. Analizar las relaciones causa efecto de los procesos en estudio.
6. Realizar el modelamiento del diseño de la solución.
7. Desarrollar la solución tecnológica más adecuada en base a las características del problema y los recursos disponibles.

VI. ÁREAS TEMÁTICAS

1. Programación C++
 - 1.1 Aspectos generales de C++



- 1.2 Modelo de programación en C
- 1.3 Punteros/referencias y manejo de memoria dinámica
- 1.4 Abstracción (tipos concretos, tipos abstractos, POO, programación genérica, contenedores)
- 1.5 Programación a gran escala (excepciones, espacio de nombres)
- 1.6 Librería estándar C++ (STL)
- 2. Introducción a Complejidad Algorítmica
 - 2.1 Tamaño de un problema
 - 2.2 Técnica de conteo de instrucciones
 - 2.3 Complejidad asintótica (Notación O)
- 3. Listas enlazadas
 - 3.1 Listas enlazadas para implementar colecciones: simple, doble, circular
 - 3.2 Matrices poco pobladas
 - 3.3 Uso conceptual de listas como implementación de Pilas y Colas
- 4. Grafos
 - 4.1 Definiciones y Clasificación de Grafos
 - 4.2 Representaciones (e.g., matriz de adyacencia, lista de adyacencia) y su impacto en la eficiencia.
 - 4.3 Operaciones sobre Grafos
 - 4.4 Cálculo de la ruta más corta usando grafos
- 5. Árboles
 - 5.1 Conceptos
 - 5.2 Árboles binarios
 - 5.3 Recorrido de árboles binarios
 - 5.4 Heaps
 - 5.5 Árboles de búsqueda binarios
 - 5.6 Árboles de búsqueda binarios balanceados
 - 5.7 Árboles 2-3
 - 5.8 Familia de árboles B (B, B*, B+)
 - 5.9 Desarrollo de un caso de estudio de aplicación de árboles



VII. ORIENTACIONES METODOLÓGICAS

1. La metodología a desarrollar en esta asignatura es principalmente práctica, por lo que se debe favorecer la interacción entre los y las estudiantes, a través de trabajos prácticos colaborativos.
2. Se sugiere trabajar la teoría a través de metodologías activas como clase invertida, ABP, entre otras, generando instancias de presentación oral individual y/o grupal, favoreciendo el aprendizaje contextualizado.
3. Las experiencias de cátedra y taller deben ser realizadas por medio de la utilización de software moderno aplicable a la asignatura.
4. Se recomienda que las y los estudiantes realicen presentaciones periódicas sobre el trabajo realizado en taller.

VII. ORIENTACIONES Y CRITERIOS PARA LA EVALUACIÓN

1. Se recomienda la aplicación de una evaluación diagnóstica al inicio de la asignatura.
2. En cátedra, se sugiere implementar estrategias de evaluación sumativas, con una ponderación del 70% de la nota final de la asignatura.
3. En taller se sugiere implementar estrategias de evaluación sumativas, con una ponderación del 30% de la nota final de la asignatura.
4. Las actividades podrán ser individuales o grupales.
5. La asistencia mínima exigida para las actividades de cátedra es del 70%.
6. Se exigirá un 60% de logro de objetivos para aprobar las actividades de evaluación.
7. Se recomienda realizar evaluaciones de carácter formativo con retroalimentación de carácter personal.



IX. RECURSOS BIBLIOGRÁFICOS

Bibliografía Mínima

- Michael T. Goodrich, Roberto Tamassia, David M. Mount. (2011). Data Structures and Algorithms in C++.
- Horowitz, E., Sahni, S., Mehta, D. (2007). Fundamentals of data structures in C++ (Segunda Edición). Computer Science Press.

Bibliografía Complementaria

- Apuntes de Clases, disponibles en la Plataforma Campus Virtual UCN.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, (2009). Introduction to Algorithms (3rd Edition).
- Drozdek, A. (2005). Data Structures and Algorithms in C++ (Tercera Edición). Brooks/Cole Thomson Learning.
- Hubbard, J. (2000). Data Structures with C++ (Schaum's Outline Series). McGraw-Hill.
- Lafore, R. (2002). Object-Oriented Programming in C++, Sams Publisher.
- Bjarne Stroustrup (1999), "An overview of the C++ programming language"