



## PROGRAMA DE ASIGNATURA

### I. IDENTIFICACIÓN

**Carrera o programa:** Ingeniería en Tecnologías de Información

**Unidad responsable:** Escuela de Ingeniería

**Nombre de la asignatura:** Estructuras de Datos

**Código:** ECIN-00416

**Semestre en la malla<sup>1</sup>:** 4

**Créditos SCT – Chile:** 5

<b>Ciclo de Formación</b>	Básico		Profesional	X
---------------------------	--------	--	-------------	---

<b>Tipo de Asignatura</b>	Obligatoria	X	Electiva	
---------------------------	-------------	---	----------	--

**Clasificación de área de conocimiento<sup>2</sup>**

<b>Área:</b> Ingeniería y Tecnología	<b>Subárea:</b> Ingeniería Informática
--------------------------------------	--

**Requisitos:**

**Prerrequisitos:**

- Programación Orientada a Objeto

**Requisitos para:**

- Ingeniería de Software
- Bases de Datos
- Proyecto IoT

### II. ORGANIZACIÓN SEMESTRAL

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

### 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 3 “Habilidades Interpersonales”. Por último, contribuye al dominio 4 “Habilidades para la Práctica de la Ingeniería”. 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. COMPETENCIAS

<sup>1</sup> Este campo sólo se completa en caso de carreras con programas semestrales.

<sup>2</sup> Clasificación del curso de acuerdo a la OCDE



La carrera declara las siguientes habilidades:

- 1.3. Aplicar conocimientos, métodos y herramientas con un enfoque sistémico principalmente en la ejecución de proyectos TI en el ámbito de las ciencias de la computación, infraestructura TI e ingeniería de software.
- 2.1. Identificar y resolver problemas con un razonamiento analítico.
- 3.1. Liderar y trabajar en equipos multidisciplinares y multiculturales.
- 4.3. Concebir soluciones que involucren, por ejemplo, aplicaciones TI, infraestructura TI, toma de decisiones, gestión de datos y gestión de proyectos.
- 4.4. Diseñar soluciones que involucren, por ejemplo, aplicaciones TI, infraestructura TI, toma de decisiones, gestión de datos y gestión de proyectos.
- 4.5. Implementar soluciones que involucren, por ejemplo, aplicaciones TI, infraestructura TI, toma de decisiones, gestión de datos y gestión de proyectos.

## V. RESULTADOS DE APRENDIZAJE

1. Programar con un enfoque orientado a objetos y verificar que las pruebas de los programas satisfagan 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. Realizar el modelamiento del diseño de la solución.
6. 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.

## VIII. ORIENTACIONES Y CRITERIOS PARA 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

– Goodrich, M. T., Tamassia, R., Mount, D. M. (2011). Data Structures and Algorithms in C++. Reino Unido: Wiley.

– Horowitz, E., Sahni, S., Anderson-Freed, S. (2007). Fundamentals of data structures in C. Estados Unidos: Silicon Press.

### Bibliografía Complementaria



# Universidad Católica del Norte

- Apuntes de Clases, disponibles en la Plataforma Campus Virtual UCN.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). Introduction to Algorithms. India: MIT Press.
- Drozdek, A. (2003). Data Structures and Algorithms in C++. islas Cocos: Tsinghua University Press.
- Hubbard, J. R. (2000). Schaum's Outline of Theory and Problems of Data Structures with C++. Reino Unido: McGraw-Hill.
- Lafore, R. (2002). Object-Oriented Programming in C++, Sams Publisher.
- Bjarne Stroustrup (1999), "An overview of the C++ programming language".