



Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

## PROGRAMA ANALÍTICO DE ASIGNATURA

**DEPARTAMENTO:** Ingeniería en Sistemas de Información

**CARRERA:** Ingeniería en Sistemas de Información

**NOMBRE DE LA ACTIVIDAD CURRICULAR:** Patrones Algorítmicos

**Año Académico:** 2023

**Área:** Desarrollo de Software

**Bloque:** Electivas

**Tipo:** Electiva

**Modalidad:** Cuatrimestral

**Cargas horarias totales:**

<i>Horas reloj</i>	<i>Horas cátedra</i>	<i>Horas cátedra semanales</i>
72	96	6

## FUNDAMENTACIÓN

El estudiante de Ingeniería en Sistemas de Información se forma con un perfil profesional cuya base analítica le permite interpretar y resolver problemas computacionales. Esta formación surge como resultado de una rigurosa incorporación de conocimientos que adquiere, progresiva y sistemáticamente, durante la cursada de las diferentes materias de la carrera. Atento a esto, la presente asignatura, se propone integrar y profundizar saberes en relación a los patrones algorítmicos, especialmente, para estructuras avanzadas, a través de un espacio de análisis, discusión, y experimentación de algoritmos complejos, con *nombre propio*, con el fin de extraer, analizar y extrapolar estrategias para aplicarlas a la solución de problemas.

## OBJETIVOS

- Identificar algoritmos correspondientes que permitan diseñarlos de forma genérica.
- Realizar una herramienta genérica para incrementar la productividad en el desarrollo de cualquier aplicación vertical.
- Reconocer estructuras de datos para lograr un diseño de algoritmos genérico.
- Reconocer una perspectiva abstracta e introspectiva de la programación para pensar y diseñar algoritmos genéricos.
- Experimentar algoritmos complejos para extraer y analizar estrategias que puedan ser aplicables a la resolución de problemas.



*Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires*

## **CONTENIDOS**

### **Contenidos analíticos**

#### **Unidad Temática 1: Algoritmia y programación**

Revisión de los conceptos previos: lenguajes, algoritmia, paradigmas. Especificaciones de Pre y Pos condición. Invariantes de ciclos. Pruebas de corrección, pruebas de terminación y técnicas de transformación de programas. Estructuras de datos fundamentales. Abstracción mediante Tipos Abstractos de Datos (TAD) y Clases. Interfaces e Implementaciones. Análisis de algoritmos. Concepto de tiempo de ejecución. Reglas de análisis de programas imperativos y recursivos.

#### **Unidad Temática 2: Recursividad**

Conceptos, principios de los algoritmos recursivos, programación con recursividad, finalización de la recursión, tipos de recursividad, transformación de algoritmos recursivos a iterativos, el algoritmo recursivo de Fibonacci.

#### **Unidad Temática 3: Árboles**

Estructura arbórea, tipos de árbol, tipos de recorrido: en amplitud, en profundidad. Pre orden, pos orden y orden simétrico. Árbol binario de búsqueda. árboles balanceados y auto balanceados.

#### **Unidad Temática 4: Complejidad algorítmica**

La *complejidad algorítmica* como una herramienta de comparación de algoritmos equivalentes. Funciones de complejidad: cota asintótica superior, media y ajustada. Peor caso: notación  $O$ , caso promedio: notación  $\Theta$ , mejor caso: Notación  $\Omega$ . Relación entre complejidad espacial y temporal. Clasificación de algoritmos en función de su complejidad.

#### **Unidad Temática 5: Métodos de ordenamiento**

Principales algoritmos de ordenamiento: Bubble sort, Quicksort, Heapsort. Ordenamiento por inserción, ordenamiento de raíz y otras técnicas conocidas. Clasificación de los algoritmos estudiados según sus funciones de complejidad.

#### **Unidad Temática 6: Patrones algorítmicos**

Técnicas de diseño de algoritmos: Divide y conquista, Algoritmos voraces (*greedy*), Programación dinámica, Backtracking, búsqueda exhaustiva o por fuerza bruta.

#### **Unidad Temática 7: Algoritmos sobre grafos**

Estructura de grafo. Grafos dirigidos, no dirigidos, ponderados, conexos o inconexos. Representación de la estructura, matriz de adyacencia. Nodos adyacentes y nodos vecinos. Recorridos en profundidad y anchura, topológicos. Camino más corto, árbol de cubrimiento mínimo, algoritmos de Dijkstra, Prim, Kruskal y Warshall-Floyd.

## **BIBLIOGRAFÍA OBLIGATORIA**



*Universidad Tecnológica Nacional*  
*Facultad Regional Buenos Aires*

- Arias, A. (2014). Fundamentos de Programación y Bases de Datos. Ed. CreateSpace.
- Revilla, M. (2002). Desafíos de Programación. Ed. Lulu.com.
- Siviani, N. (2007). Diseño de Algoritmos. Ed. Tomson.
- Sznajdleder, P. (2020). Java a fondo 4ta. edición. Ed. Alfaomega.
- Sznajdleder, P. (2013). Algoritmos a fondo. Ed. Alfaomega.

## **CORRELATIVAS**

Para cursar y rendir

- Cursadas:
  - Análisis de Sistemas de Información
  - Sintaxis y Semántica de los Lenguajes
  - Paradigmas de Programación