



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:** Algoritmos y Estructuras de Datos

**Año Académico:** Plan 2023

**Área:** Desarrollo de software

**Bloque:** Tecnologías Básicas

**Nivel:** 1º

**Tipo:** Obligatoria

**Modalidad:** Anual

**Cargas horarias totales:**

<i>Horas reloj total</i>	<i>Horas cátedra total</i>	<i>Horas cátedra semanales</i>
120	160	10

### OBJETIVOS

- Identificar problemas algorítmicos.
- Comprender el proceso de desarrollo de software.
- Resolver problemas aplicando soluciones algorítmicas y estructuras de datos.
- Desarrollar un pensamiento creativo para resolver situaciones problemáticas.
- Utilizar enfoques lógicos y estructurados para reconocer la información contenida en una situación a resolver.
- Identificar la eficiencia computacional, en cuanto a recursos y tiempos de ejecución (en el paradigma imperativo), para distinguir problemas en términos.
- Desarrollar habilidades prácticas y experiencia en la escritura de programas en lenguajes formales de programación estructurada.
- Aplicar el proceso de pensamiento sistemático para resolver problemas con la computadora y tomar decisiones.
- Distinguir técnicas de resolución de problemas y programación para promover la producción y la construcción de conocimiento.
- Utilizar el razonamiento lógico en entornos gráficos y de implementación para predecir el comportamiento de programas.
- Reconocer la noción de algoritmo como un método de resolución de problemas en contextos diferentes.
- Relacionar una representación gráfica con una secuencia ordenada de acciones para resolver problemas de información.



- Utilizar programación estructurada para diferenciar soluciones concretas a una situación problemática.
- Trabajar con variables y diversas formas de entrada y salida, para incorporar el concepto de estructuras de datos persistentes o volátiles.
- Comparar diferentes algoritmos para la solución de un mismo problema en los términos de refinamiento y optimización.

## **CONTENIDOS**

### **Contenidos mínimos**

- Programación Imperativa y Concepto de algoritmo
- Concepto de Dato
- Tipos de Datos Simples
- Tipo Abstracto de datos
- Estructuras de Control Básicas
- Estrategias de Resolución de problemas
- Estructuras de Datos
- Abstracciones con procedimientos y funciones
- Estructuras de Datos lineales y no lineales
- Algoritmos de Búsqueda, Recorrido y Ordenamiento
- Archivos de Acceso Secuencial y Aleatorio
- Recursividad

### **Contenidos analíticos**

#### **Parte I: Del Universo real al universo computacional**

##### **Unidad 1: Definición de términos y frases**

Definición de paradigmas de programación. Profundización del paradigma imperativo, modular y estructurado. Concepto de Computación, Informática, Sistemas. Tipos de problemas: problemas computacionales. Definición de asignación, sentencia, declaración, dato, información, conocimiento, lenguajes de programación. Partes de un programa, Abstracción, Modularización, Especificación

##### **Unidad 2: Conceptos Fundamentales de Algoritmia**

Del problema real a su implementación en computadoras. Análisis y comprensión de problemas de información planteados en lenguaje natural. Representaciones gráficas para la formalización y resolución estratégica de problemas computables de información. Concepto de algoritmo. Refinamientos sucesivos. Léxico y algoritmo. Características, propiedades y eficiencia de los algoritmos. Estructura de los algoritmos. Proceso computacional.

##### **Unidad 3: Tipos de datos primitivos y operadores**

Definición de tipo de dato. Tipos de datos primitivos (definición de los conjuntos de valores y



operaciones aplicadas a los mismos. Definición y alcance de tipos de datos simples, ordinales y no ordinales, introducción al concepto de cadenas de caracteres. Operadores lógicos, aritméticos de relación de asignación, utilización y combinaciones de los mismos. Patrones algorítmicos con datos simples: secuencias, seguidillas, selecciones.

**Logros Pedagógicos:** Comprender los problemas de información con la utilización de datos de tipo simple y utilizar el concepto de refinamientos sucesivos para la eficaz resolución de un problema computacional Mediante el desarrollo conceptual y/o gráfico del algoritmo más adecuado para la resolución.

### **Parte II: Constructos del Lenguaje: Expresiones, Declaraciones, Sentencias y Estructura de Programa**

**Unidad 4: Tipos, Operadores, y Expresiones**  
Tipo, precedencia, asociatividad, orden de evaluación. Declaraciones. Definiciones, Alcance. Identificadores, Constantes, Variables, valorL.

**Unidad 5: El lenguaje de programación C++**  
Estructura general de un programa. Concepto de bibliotecas, introducción al concepto de la descomposición: prototipos, firmas y desarrollo de funciones. Concepto de espacios de nombres. Expresiones y sentencias: sentencias simples estructuradas, compuestas. Análisis de casos simples, múltiples, completos, incompletos. Composiciones iterativas: la necesidad de iterar. Composiciones exactas, no exactas, precondicionales, poscondicionales. Iteración versus recursividad. Composiciones recursivas.

**Unidad 6: Descomposiciones y Funciones**  
La necesidad de la descomposición. Declaración y definición de funciones: argumentos y parámetros; invocación y declaración. La propiedad de la Reusabilidad. Ventajas de la generalidad de tipo y de criterio de selección

**Logros Pedagógicos:** Aplicar las abstracciones de los lenguajes de programación para la implementación de los algoritmos mediante la descomposición a través de funciones.

### **Parte III: Estructura de datos estáticos y almacenamiento lógico**

**Unidad 7: Concepto de tipos de datos derivados y abstractos**  
Definición de tipo de dato derivado y estructura de dato. Operadores de acceso a cada miembro de la estructura. Estructuras de datos con almacenamiento lógico: Estructuras de posiciones contiguas de memoria de datos no homogéneos y estructuras de datos homogéneos. Combinaciones. Concepto de estructuras estáticas y estructuras con asignación dinámica en memoria. Concepto de estructuras con almacenamiento físico. Profundización de estudio de estructuras de posiciones contiguas no homogéneas en memoria. Introducción al concepto de abstracción de datos

**Unidad 8: Estructuras indexadas**  
Definición y declaración de array. Array de tipo de dato simple. Array de tipos de datos



derivados. Tamaño lógico y tamaño físico. Estructura unidimensional o multidimensional. Array de caracteres. Concepto de puntero constante, desplazamiento.

**Unidad 9: Patrones algorítmicos**

Patrones de carga: en la definición. Secuencial, directa, ordenada. Patrones de recorrido: total, parcial, en ambas direcciones. Con corte de control, apareando dos o más estructuras. Patrones de búsqueda: directa, binaria, lineal. Análisis de eficiencia. Patrones de ordenamiento: Con posición Única predecible, mediante métodos de ordenamiento. Análisis de eficiencia

**Logros Pedagógicos:** Implementar patrones algorítmicos de carga, recorrido y búsqueda en estructuras con posiciones contiguas de memoria, homogéneas, no homogéneas y combinadas

**Parte IV: Almacenamiento Físico - Estructuras con asignación dinámica**

**Unidad 10: Estructuras de Almacenamiento físico**

Concepto de almacenamiento físico. Archivos de texto y archivos binarios. Archivos con acceso directo. Análisis de bibliotecas para el manejo de archivos. Nombre lógico y físico. Patrones algorítmicos: comparación con patrones de otras estructuras de manejos de conjunto de datos del mismo tipo, conservando la lógica y modificando el modo de acceso a cada uno de los datos particulares

**Unidad 11: Punteros, asignación dinámica en memoria de estructuras enlazadas lineales**

Definición de punteros. La necesidad de la asignación dinámica en memoria. Constructores y destructores. Concepto de instancia y operadores de dirección y des referenciación. Punteros a datos y punteros a código. Estructuras enlazadas con asignación dinámica en memoria: Estructuras lineales, ventajas y debilidades. Definiciones y declaraciones. Estructura de tipo pila: Patrones de carga y eliminación de nodos. Estructura de tipo cola: Patrones de carga y eliminación de nodos. Estructura de tipo lista: listas simplemente enlazadas, listas circulares, listas doblemente enlazadas y listas circulares doblemente enlazadas. Patrones de carga sin restricciones, de carga sin repetir del dato, de búsqueda, de recorrido (completo, parcial, con corte de control o apareo) de eliminación de nodos

**Unidad 12: Estructuras arbóreas**

Definición de árbol, árbol binario de búsqueda, árbol balanceado inserción y recorrido recursivo e iterativo: análisis de eficiencia.

**Unidad 13: Combinaciones complejas de estructuras**

Resolución de problemas complejos combinando las estructuras analizadas y justificando la elección de las mismas.

**Logros Pedagógicos:** Aplicar la especificación precisa del algoritmo de resolución de un problema complejo de información mediante la utilización de las estructuras más idóneas para la solución.

**BIBLIOGRAFÍA OBLIGATORIA**



*Universidad Tecnológica Nacional*  
*Facultad Regional Buenos Aires*

- Aguilar, L. J (2004) .Algoritmos y estructura de datos, una perspectiva en C. Ed. Mc Graw Hill .
- Forouzan, Behrouz (2005). Introducción a la ciencia de la computación. Ed. Ciencias e Ingeniería.
- Garcia Molina (2005). Una introducción a la programación. Ed. Thomson.
- Iranzo, J. (2006). Lógica simbólica para informáticos. Ed. Alfaomega.
- Kernighan, R. (1988). El lenguaje de programación C. Ed. Pearson.
- Kernighan, P. (2000). La práctica de la programación. Ed. Pearson.
- Perkins, D. (2004). La bañera de Arquímedes y otras historias del descubrimiento científico. Ed. Paidós.
- Stroustrup, B. (2013). A Tour of C++. Second Edition. Ed. Pearson Education.
- Stroustrup, B. (2008). Programming: Principles and Practice Using C++. Ed. Addison Wesley.
- Stroustrup, B. (2002) El lenguaje de Programación C++. Ed. Addison Wesley