

# Asignatura: Algoritmos 1

---

## 1. Algoritmos

- **Qué es un algoritmo**

- Definición y características (precisión, finitud, entradas, salidas).
- Diferencia entre algoritmo y programa.
- Ejemplos cotidianos de algoritmos (receta de cocina, instrucciones de un juego, llamada telefónica...).

- **Diagramas y Representaciones**

- Diagramas de flujo (símbolos y reglas).
- Diagramas Nassi-Shneiderman (NSD) o Chapin
- Pseudocódigo.
- Representación narrativa (lenguaje natural estructurado).

- **Tablas de prueba de escritorio**

- Concepto y utilidad.
- Construcción paso a paso.
- Ejemplo con estructuras secuenciales.
- Ejemplo con estructuras de decisión y repetitivas.

## 2. Elementos básicos

- **Datos**

- Tipos de datos primitivos: enteros, reales, caracteres, booleanos.
- Dominio y rango de cada tipo.

- **Variables y constantes**

- Declaración de variables.
- Diferencia entre variable y constante.
- Buenas prácticas de nombramiento.

- **Operadores**

- Aritméticos: suma, resta, multiplicación, división, módulo, potencia
- Relacionales: mayor, menor, igual, diferente.
- Lógicos: and, or, not.
- Precedencia de operadores.

- **Expresiones y funciones predefinidas**

- Expresiones aritméticas simples y compuestas.
- Funciones matemáticas: valor absoluto, potencia, raíz cuadrada, aleatorio, seno, coseno.
- Funciones de cadenas de caracteres: longitud.
- Ejercicios de conversión de expresiones matemáticas a expresiones algorítmicas.

### 3. Estructuras para la construcción de algoritmos

- **Instrucciones de asignación**

- Sintaxis básica.
- Reglas de conversión de expresiones matemáticas.
- Ejemplos.

- **Expresiones relacionales y lógicas**

- Evaluación de expresiones lógicas.
- Tablas de verdad.
- Operadores combinados.
- Ejercicios con expresiones relacionales simples y compuestas (and, or, not).

### 4. Estructura secuencial

- **Definición y utilización**

- Algoritmos que siguen un flujo único.
- Ejemplos prácticos: suma de dos números, conversión de temperaturas, cálculo de la definitiva de un curso

- **Definición de variables y asignación**

- Reglas de declaración en pseudocódigo.
- Ejemplos de declaración y asignación.

- **Ingreso de información (lectura de datos)**

- Concepto de entrada estándar, el verbo Leer
- Validación de entradas simples.

- **Salida de información (escritura de datos)**

- Concepto de salida estándar, el verbo Imprimir [sin saltar]
- Formatos de salida (texto, numérico).

- **Finalización de un algoritmo**

- Instrucciones de fin.

- Ejemplo: mostrar mensaje de conclusión.

## 5. Estructuras de decisión lógica

- **Definición y utilización**

- Importancia en el control del flujo.
- Notación en pseudocódigo y diagramas de flujo.

- **Tipos de estructuras de decisión lógica**

- If simple.
- If... else.
- If... else if.
- If anidado.

- **Estructuras de decisión anidadas**

- Algoritmos con condiciones múltiples.
- Ejemplos: clasificación de números, notas de estudiantes.

- **Selector múltiple**

- Definición y sintaxis.
- Ejemplo: menú de opciones.

## 6. Estructuras iterativas o repetitivas

- **Definición**

- Diferencia entre ciclo controlado por condición y ciclo controlado por contador.

- **Tipos de estructuras repetitivas**

- Mientras (while).
- Para (for).
- Haga mientras (do... while).

- **Contadores, acumuladores, suiches**

- Conceptos y ejemplos.
- Aplicación: sumatoria de números, promedio de notas.
- Ejercicios con promedios, porcentajes, máximos, mínimos, etc

- **Ciclos anidados**

- Definición y uso.

- Ejemplo: generación de tablas de multiplicar, patrones de asteriscos, recolección y resumen de datos

## 7. Subprogramas, funciones, métodos

- **Definición y sintaxis**

- Qué es un subprograma.
- Diferencia entre procedimientos y funciones.

- **Parámetros**

- Parámetros por valor.
- Parámetros por referencia.

- **Valores de retorno**

- Funciones que devuelven un resultado.
- Ejemplo: función que calcula el factorial.

- **Aplicaciones**

- Reutilización de código.
- Ejemplo: librería de funciones matemáticas simples, el concepto de importación

## 8. Arreglos

- **Definición**

- Concepto de colección de datos homogéneos.
- Los arreglos como estructuras de datos estáticas
- Tipos de arreglos: unidimensionales o vectores, bidimensionales o tablas (matrices), multidimensionales

- **Declaración y asignación**

- Arreglos unidimensionales.
- Inicialización de arreglos.

- **Operaciones con arreglos**

- Recorrido.
- Búsqueda secuencial.
- Ordenamiento (método de la burbuja)
- Búsqueda binaria
- Inserción y eliminación.
- Ejercicios con máximos, mínimos, promedios...

- Ejemplo aplicado: manejo de notas de estudiantes.

## Sugerencia de bibliografía, agregue la que Usted crea conveniente

- Algoritmia básica. Roberto Flórez R.
- Lógica de programación. Efraín M. Oviedo Regino
- Algoritmos, estructuras de datos y programación orientada a objetos. Roberto Flórez R.
- Estructura de datos. Osvaldo Cairó & Silvia Guardati

## Software de apoyo, por favor utilice los que se sugieren

- PSEInt: Herramienta de asistencia para estudiantes en sus primeros pasos en programación. <https://pseint.sourceforge.net/>
- Python: Lenguaje de programación con sintaxis simple y cercana al lenguaje natural, curva de aprendizaje suave. <https://www.python.org/>
- PythonTutor: Herramienta web que visualizar cómo se "ejecuta" un algoritmo/programa en la memoria paso a paso. <https://pythontutor.com/visualize.html#mode=edit>
- Se sugiere el editor de código Visual Studio Code. <https://code.visualstudio.com/>