



# Plan temático

(Fecha última actualización de este documento: 5 ene 2025 )

<b>Matemáticas</b> .....	<b>1</b>
<b>Ciencia de la Computación</b> .....	<b>2</b>
<b>Bibliografía</b> .....	<b>4</b>

Nota: Los seis(6) ejercicios de la competencia oficial estarán basados en uno o varios de los temas (y subtemas) que se relacionan a continuación:

## Matemáticas

### 1. Aritmética y Geometría

- Enteros, operaciones (incluyendo exponenciación), comparación.
- Propiedades básicas de los números enteros (signo, paridad, divisibilidad).
- Aritmética modular básica: suma, resta, multiplicación.
- Números primos.
- Fracciones y porcentajes.
- Punto, vector, coordenadas en el plano.
- Distancias euclidianas.
- Teorema de Pitágoras.

### 2. Estructuras Discretas

- Funciones, relaciones y conjuntos.
- Lógica básica.
- Tablas de verdad.
- Definiciones matemáticas recursivas.

### 3. Fundamentos básicos de conteo

- Argumentos de conteo (regla de la suma y producto, progresiones aritméticas y geométricas, números de Fibonacci).
- Permutaciones y combinaciones (definiciones básicas).
- Función factorial, coeficientes binomiales.
- Principio de inclusión-exclusión.

### 4. Grafos y árboles

- Grafos no dirigidos (vértice/nodo, arista, grado, adyacencia).
- Grafos dirigidos.
- Multigrafos, grafos con bucles.
- Caminos en grafos (camino no dirigido y dirigido, ciclo, recorrido, paseo; tour de Euler; camino/ciclo hamiltoniano).
- Alcanzabilidad (distancia más corta).
- Árboles (hoja, diámetro, centro, centroide, bosque).



## Ciencia de la Computación

1. Fundamentos de la Programación
  - Construcciones fundamentales de programación.
  - Sintaxis y semántica básica de un lenguaje de alto nivel (C++).
  - Variables, tipos, expresiones y asignación.
  - Entrada/salida estándar.
  - Estructuras de control (condicional e iterativa).
  - Funciones y paso de parámetros.
2. Algoritmos y resolución de problemas
  - Estrategias para resolver problemas.
  - Estrategias de implementación para algoritmos.
  - Estrategias de depuración.
  - Concepto y propiedades de los algoritmos (corrección, eficiencia).
3. Estructuras de Datos fundamentales
  - Tipos de datos (booleano, entero con/sin signo, carácter).
  - Arreglos (incluyendo arreglos multidimensionales).
  - Cadenas y procesamiento de cadenas.
  - Estructuras enlazadas.
  - Estrategias de implementación para grafos y árboles.
  - Punteros y referencias.
4. Recursión
  - Funciones matemáticas recursivas.
  - Recursividad simple.
  - Recursividad ramificada.
  - Estrategias de divide y vencerás.
  - Recursividad con backtracking.
5. Análisis básico de algoritmos
  - Análisis asintótico de límites superiores de complejidad.
  - Notación Big O.
  - Análisis de complejidad de tiempo: constante, logarítmica, lineal,  $O(n \log n)$ , cuadrática, cúbica, exponencial, etc.
6. Estrategias algorítmicas
  - Búsqueda exhaustiva.
  - Algoritmos voraces (greedy).
  - Divide y vencerás.
  - Programación dinámica.
7. Algoritmos
  - Algoritmos simples que involucran enteros: conversión entre bases, algoritmo de Euclides, test de primalidad, criba de Eratóstenes, factorización, exponenciación eficiente.



## Olimpiada Cubana de Informática 2024-2025

pág. 3

- Operaciones básicas con números enteros de precisión arbitraria (suma, resta, multiplicación simple).
  - Manipulación básica de arreglos (llenado, desplazamiento, rotación, reversión, redimensionamiento, mínimo/máximo, suma de prefijos, histograma).
  - Algoritmos simples en cadenas.
  - Búsqueda secuencial y búsqueda binaria.
  - Algoritmos de ordenación con peor caso  $O(n \log n)$ .
  - Recorridos de árboles.
  - Recorridos en profundidad y en anchura (dfs y bfs).
  - Búsqueda de componentes conexas y cláusulas transitivas.
  - Algoritmos de caminos más cortos (Dijkstra, Bellman-Ford, Floyd-Warshall)
  - Árbol de expansión mínimo (algoritmos de Prim y Kruskal)
  - Bi-conectividad en grafos no dirigidos (puntos de articulación, puentes).
  - Conectividad en grafos dirigidos (componentes fuertemente conexas).
  - Fundamentos básicos de la teoría de juegos combinatorios, posiciones ganadoras y perdedoras, algoritmos minimax para juego óptimo.
8. Estructuras de Datos
- Pilas y colas.
  - Representación de grafos (lista de adyacencia, matriz de adyacencia).
  - Representación de Conjuntos Disjuntos.
  - Árboles de búsqueda binarios estáticamente equilibrados. Incluyendo Árbol Binario Indexado (también conocido como Fenwick Tree) y Segment Tree.
  - Algoritmos en  $O(\log n)$  para responder el Lowest Common Ancestor (LCA) en un árbol estático.
  - Anidamiento de Estructuras de Datos.
  - Tries.
9. Algoritmos geométricos
- Representación de puntos, vectores, líneas, segmentos de línea.
  - Intersección de dos líneas.
  - Computar el área de polígonos dadas las coordenadas de sus vértices.
  - Chequear si un polígono contiene un punto.
  - Compresión de coordenadas.
  - Algoritmos en  $O(n \log n)$  para encontrar la cápsula convexa de un polígono.
  - Método de Sweep Line.



## Bibliografía

1. Tom Verhoef, Gyula Horváth, Krzysztof Diks, and Gordon Cormack, Michal Forisek y otros. [The International Olympiad in Informatics Syllabus](#). 2025.
2. Harvey Deitel y Paul Deitel. C++ How to Program. Décima Edición.
3. Antti Laaksonen. [Competitive Programmer's Handbook](#). 2018.
4. Felix Halim, Steven Halim, y Suhendry Effendy. Competitive Programming 4: The Lower Bound of Programming Contests in the 2020s.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, y Clifford Stein. Introduction to Algorithms, 3rd Edition.