

Documentación Técnica – ICPC 2025

Equipo XYZ – Universidad ABC

7 de octubre de 2025

Índice

1. Plantillas de Código	2
1.1. I/O rápido en C++	2
1.2. Macros utiles para el lenguaje	2
2. Tablas de Complejidad	2
3. Fórmulas Matemáticas	3
4. Estrategia de Equipo	3

1. Plantillas de Código

1.1. I/O rápido en C++

Se encargan de desincronizar el cin y el cout con el fin de eliminar el overhead y alcanzar una velocidad similar al printf y scanf de C.

```
1 ios::sync_with_stdio(false);
2 cin.tie(nullptr);
```

1.2. Macros utiles para el lenguaje

a. Atajos de sintaxis

```
1 #define pb push_back //metodo de insercion comun en vectores
2 #define mp make_pair //construccion de pares en ejecucion para maps
3 #define all(x) (x).begin(), (x).end() //util para sort y otros metodos
    que requieren iteradores inicio y fin
4 #define rall(x) (x).rbegin(), (x).rend() //igual que all pero en orden
    inverso
5 #define sz(x) (int)((x).size()) //longitud de las principales DS std
```

b. Tipos de datos frecuentes

```
1 #define ll long long //entero 64 bits (maximo espacio c++)
2 #define ull unsigned long long //igual a ll, para casos sin negativos
3 //vectores de los tipos principales
4 #define vi vector<int>
5 #define vll vector<long long>
6 #define vd vector<double>
7 #define vs vector<string>
```

c. Bucles rapidos

```
1 #define FOR(i,a,b) for(long long i=(a); i<(b); i++) //Bucle for 1 a 1
    desde a hasta b
2 #define REP(i,n) for(long long i=0; i<(n); i++) //Bucle for 1 a 1 de
    0 a n (Ideal para recorrer todo un vector)
3 #define ROF(i,a,b) for(long long i=(a); i>=(b); i--) //Bucle
    descendente 1 a 1 de a hasta b
```

d. Constantes

```
1 #define INF 1000000000 //Posible infinito para int
2 #define LINF 1000000000000000000LL //Posible infinito para long long
3 #define MOD 1000000007 //Constante 10^9 + 7 para problemas gigantes
4 #define MOD2 998244353 //Otra constante modular menos usada
5 #define EPS 1e-9 //Margen de error para operaciones punto flotante
```

e. Debugging rapido (No sabemos debuggear)

```
1 #define debug(x) cerr << #x << " = " << (x) << endl //Se usa cerr ya
    que no es evaluado por el juez
2 #define debugv(v) { cerr << #v << " = "; for(auto _ : v) cerr << _ <<
    " "; cerr << endl; }
```

2. Tablas de Complejidad

- **vector**: acceso $O(1)$, inserción al final $O(1)$ amortizado
- **set/map**: inserción/búsqueda $O(\log n)$
- **unordered_map**: inserción/búsqueda $O(1)$ promedio

3. Fórmulas Matemáticas

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

$$\text{Área de triángulo (Herón)} = \sqrt{s(s-a)(s-b)(s-c)}$$

4. Estrategia de Equipo

Roles, protocolos de clarifications, rotación de teclado.