

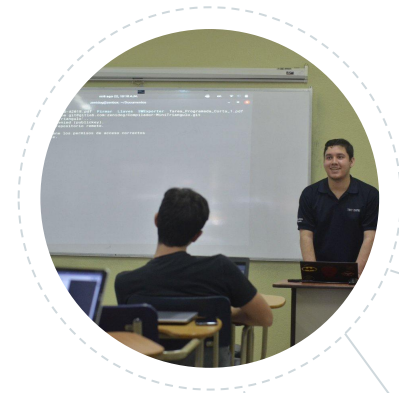
Introducción a la programación competitiva

¡Hola!

Soy José Godínez

Soy egresado del TEC y
soy un SDE en Amazon

Telegram:
[@zenidog8](https://t.me/zenidog8)



¿Qué es?

Es un deporte mental basado en la solución de problemas mediante la programación.

Usualmente se califica basado en la cantidad de soluciones y el tiempo utilizado por cada solución. NO SE EVALÚA SI UNA SOLUCIÓN DURA MENOS EN EJECUCIÓN QUE OTRA.

Habilidades

- Abstracción
- Modelado
- Cálculo de complejidad
- Estructuras de datos
- Creatividad
- Matemática
- Algoritmos
- Trabajo duro

Beneficios

- Amigos
- Excusa para aprender
- Networking profesional
- Habilidades técnicas
- Medición de las habilidades
- Facilidad en entrevistas técnicas

ICPC

Torneo más antiguo de programación competitiva. Se compete a nivel universitario, primero de manera regional y posteriormente a nivel mundial. ¡Están invitados a participar!

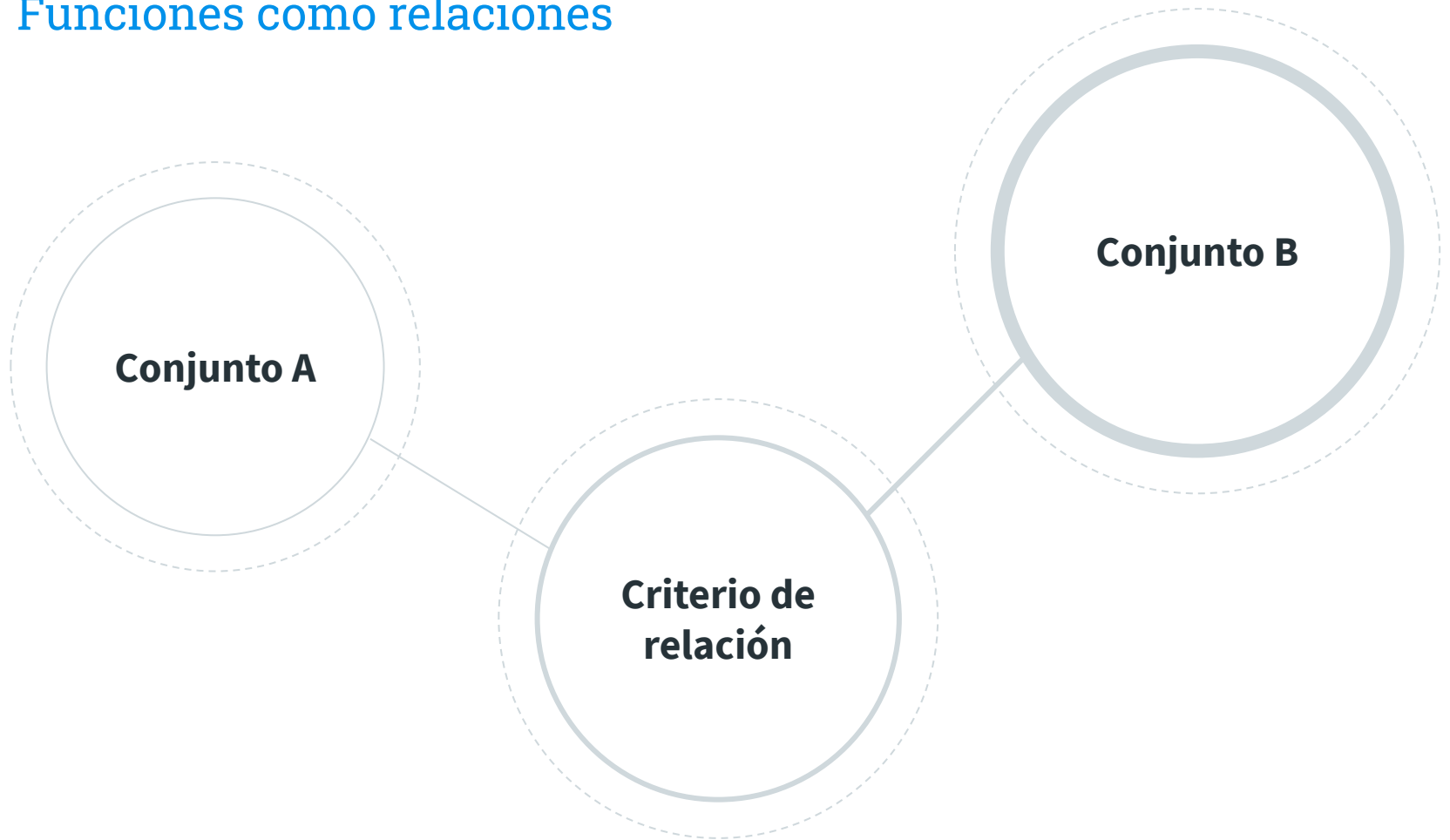
A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a grey outline. The network is dense and irregular.

1. **Funciones**

A modo de repaso

A decorative network diagram in the bottom-right corner, similar to the one in the top-left, featuring a cluster of interconnected nodes and edges.

Funciones como relaciones





“

*En programación competitiva no se espera una aplicación o un sistema, **se espera una función o un conjunto de funciones que resuelvan el problema***

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by circles of varying sizes, some with concentric rings, and the lines are thin and grey. The diagram is partially cut off by the left edge of the frame.

2.

Los problemas

Hablemos de táctica y estrategia

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It shows a complex web of interconnected nodes and lines, with nodes represented by circles of varying sizes and lines as thin grey connections. The diagram is partially cut off by the right edge of the frame.

Partes de un problema

Límites

La mayoría de problemas cuentan con un límite de tiempo y memoria para su ejecución.

Explicación

Se brinda una descripción en prosa, normalmente con el origen del problema a resolver.

Ejemplos de I/O

El juez entrega al programa una serie de datos a través de la entrada estándar y evalúa la respuesta desde la salida estándar. En esta sección se ejemplifica **ligeramente** un posible escenario.

Mi heurística personal para competir en el torneo (estrategia)

- Leer todos los problemas rápidamente
- Buscar el que se ve más sencillo y leerlo con más detalle
- Si el equipo tiene una idea se puede proceder a resolverlo
- En caso de tener un scoreboard disponible, se puede continuar con los problemas más resueltos por otros equipos
- Si el problema es fácil pero requiere mucho tiempo para programar, otro compañero puede escribir el algoritmo en la computadora y/o en papel mientras se buscan problemas más rápidos de resolver

Mi heurística personal para resolver un problema (táctica)

- Leer la descripción y los ejemplos de I/O
- Identificar el tipo de problema (Arrays, DP, grafos, teoría de números...)
- Verificar si hay alguna estructura de datos que me pueda ayudar (cola, pila, heaps, hashmaps)
- Verificar el tamaño de las entradas, casos extremos, tiempo y memoria disponibles
- Hacer un intento con la forma más simple (el tiempo es oro)
- Al comprobar que se ocupa una mejor solución proceder a mejorarla o continuar con otro problema

Un ejemplo práctico de un problema

- Write a program that multiplies pairs of integers
- Input starts with one line containing an integer T , where $1 \leq T \leq 100$, denoting the number of test cases. Then T lines follow, each containing a test case. Each test case consists of two integers A, B , where $-2^{20} \leq A, B \leq 2^{20}$, separated by a single space.
- For each test case, output one line containing the value of $A \times B$.

Un ejemplo práctico

Sample input	Sample output
4	12
3 4	0
13 0	8
1 8	10000
100 100	

Un ejemplo práctico

```
#include <iostream>
using namespace std;

int main() {
    int T;
    cin >> T;

    for (int t = 0; t < T; t++) {

        int A, B;
        cin >> A >> B;

        cout << A * B << endl;
    }

    return 0;
}
```

Un ejemplo práctico

```
#include <iostream>
using namespace std;

int main() {
    int T;
    cin >> T;

    for (int t = 0; t < T; t++) {

        int A, B;
        cin >> A >> B;

        cout << A * B << endl;
    }

    return 0;
}
```

¿Qué pasa si $A = B = 2^{20}$?

Ó

¿Qué pasa si $A = B = -2^{20}$?

Un ejemplo práctico

```
#include <iostream>
using namespace std;

int main() {
    int T;
    cin >> T;

    for (int t = 0; t < T; t++) {

        int A, B;
        cin >> A >> B;

        cout << A * B << endl;
    }

    return 0;
}
```

¿Qué pasa si $A = B = 2^{20}$?

Ó

¿Qué pasa si $A = B = -2^{20}$?

Desborde y WA

Un ejemplo práctico

```
#include <iostream>
using namespace std;

int main() {
    int T;
    cin >> T;

    for (int t = 0; t < T; t++) {

        int A, B;
        cin >> A >> B;

        cout << A * B << endl;
    }

    return 0;
}
```

```
#include <iostream>
using namespace std;

int main() {
    int T;
    cin >> T;

    for (int t = 0; t < T; t++) {

        long long A, B;
        cin >> A >> B;

        cout << A * B << endl;
    }

    return 0;
}
```


Tipos de errores (Su máximo enemigo)

Codigo

Que significa

WA

Wrong answer

TLE

Time limit exceed

MLE

Memory limit exceed

PE

Presentation error

RTE

Runtime error

CE

Compilation error



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles inside, suggesting a hierarchical or multi-layered structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

3. **Lenguajes**

No todo lo que brilla es oro

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It shows a cluster of nodes connected by lines, with some nodes being larger and having concentric circles, indicating a similar hierarchical or multi-layered structure. The lines are thin and gray.

Algunos de los lenguajes más usados



C

Uno de los lenguajes más importantes en la historia, usado en el desarrollo de sistemas operativos, tiene una sintaxis sencilla y muchos lenguajes heredan elementos de la misma.



C++

Todo código de C es válido en C++. Agrega orientación a objetos útiles especialmente para problemas con strings. Permite explotar el poder y los algoritmos de la STL. Es el más usado en el mundo.



Python

Es un lenguaje sencillo de aprender, posee muchas bibliotecas útiles. Al ser interpretado es más lento que C. pero facilita el trabajo con listas y sus **variables numéricas no poseen un límite de bytes.**



Java

Casi en todas las universidades de aprende Java. Tiene bibliotecas importantes para problemas de Big Numbers. Otros lenguajes requieren menos líneas de código en comparación y esto puede representar una desventaja.



Kotlin

Es el lenguaje del patrocinador global. Es altamente utilizado para el desarrollo móvil. Es interesante para tenerlo en cuenta para el futuro, pero si no lo conocen no es buena idea aprenderlo desde 0 para competir.



Otros

Es permitido en algunos jueces usar otros lenguajes populares como JavaScript. La ICPC no permite otros lenguajes a parte de los mencionados anteriormente.

Opinión personal

Es mejor usar C++ como lenguaje primario y python como lenguaje secundario para casos específicos.



Conocer C++ permite estudiar soluciones de otros programadores competitivos y apoya al competidor con la STL. Python permite trabajar con problemas de números grandes, arrays y strings con mucha mayor facilidad.

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are highlighted with a double-circle effect. The lines are thin and gray, creating a mesh-like structure.

4. **Conclusiones**

Hablemos de resultados

A decorative network diagram in the bottom-right corner, similar to the one in the top-left. It shows a cluster of nodes connected by lines, with some nodes having a double-circle highlight. The overall style is minimalist and technical.

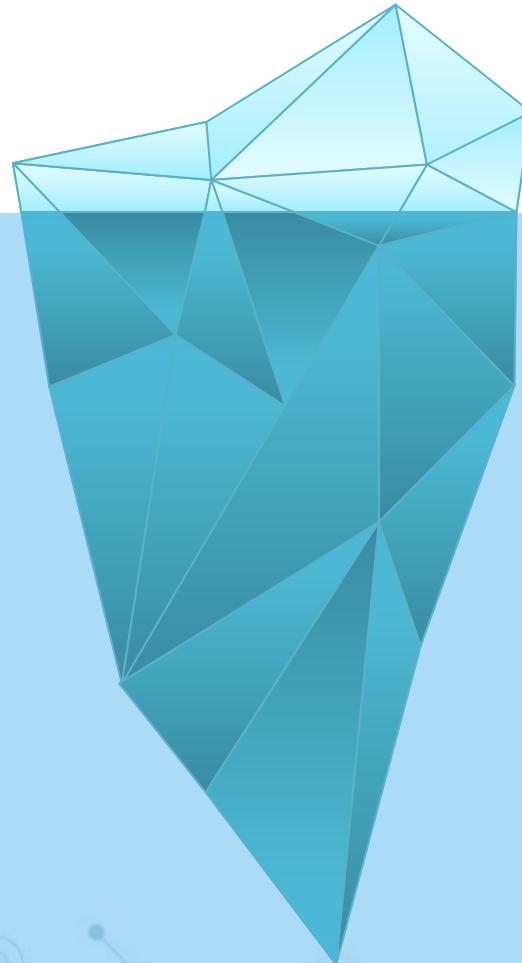
El secreto del éxito



El éxito como a nivel profesional

Saber que hacer

En la superficie del éxito está el saber que hacer para cumplir con una tarea determinada.



Entender el porqué

En lo más profundo del éxito está el entender porqué las cosas se solucionan de determinada forma. Los problemas más difíciles se logran resolver a través del entendimiento de la teoría y no por seguir patrones a ciegas.



Tarea

(No hay nota, su propio progreso es el medidor de su trabajo)

URI 1000 al 1021

Para esto van a requerir una cuenta en:

<https://www.urionlinejudge.com.br/judge/en/login>



¡Gracias!

¿Preguntas?

Pueden escribirme cuando
gusten a mi telegram

@zenidog8



Créditos

¡Gracias en especial a nuestros amigos de la ICPC Colombia Juan Felipe, Lina y Edward. Creadores del contenido original del que este material se deriva!

© Material original:

<https://sites.google.com/view/campamento-de-programacion-cr/clases/d%C3%ADa-1>