

Introducción a la Programación Competitiva (CP)

Ian González Vidalón - Club de Programación Competitiva UTEC

1 Introducción

La CP consiste en resolver, mediante la programación, problemas algorítmicos con limitaciones de tiempo y memoria. Por lo general, cuando uno compite en CP, se enfrenta con un conjunto de problemas de variable grado de dificultad, con los siguiente objetivos:

1. Resolver la mayor cantidad de problemas.
2. Minimizar la cantidad de envíos erróneos.
3. Minimizar la suma de los tiempos (respecto al inicio de la competencia) en que cada envío correcto fue realizado.

De este modo, la CP requiere e incentiva el desarrollo de los siguientes atributos:

- Resolución de problemas.
- Pensamiento algorítmico.
- Gestión de la presión.
- Velocidad de pensamiento y programación.

Embarcarse en la travesía de programar de forma competitiva es un viaje que puede ser largo y requiere una inversión de tiempo significativa. Los beneficios de desarrollarse en esta área son amplios y variados. En primer lugar, la capacidad de adaptarse a las restricciones de recursos es una habilidad clave para desarrollar proyectos de *software*. Es más, en general, lo que se desarrolla es una habilidad de optimizar los recursos para cumplir con requisitos explícitamente detallados, lo cual puede ser aplicado para optimizar cualquier proceso.

En segundo lugar, la gran mayoría de entrevistas para entrar a compañías *top*, como **Big Tech**, involucra tener que resolver frente al entrevistador problemas de algoritmos y/o estructuras de datos. Por lo tanto, ser exitoso en CP equivale a no tener problemas para triunfar en este tipo de entrevistas.

En definitiva, CP abre muchas puertas, tanto en el ámbito laboral como en el social, con la oportunidad de conocer a otras personas con pasiones e intereses similares. Asimismo, hay múltiples oportunidades para ganar premios en las competencias, a su vez como viajar para competir.



Figure 1: Algunas fotos del equipo de UTEC "WA doesn't go away with tears" en su trayecto ICPC 2024-2025

Más allá de los posibles beneficios en la proyección profesional, es importante considerar que dedicarse a CP no es un requisito para tener éxito en entrevistas de trabajo, y en la mayoría de casos la razón de perseguir un camino en CP es una pasión por la resolución de problemas.

Al respecto, aquí hay algunos *blogs* donde personas exitosas en CP hablan acerca de su trayecto y experiencia:

- 10 años en CP de un estudiante iraní que pasó de viajar a una nueva ciudad para prepararse para la Olimpiada Internacional de Informática (IOI) a obtener el puesto 24 a nivel mundial en la ICPC (como estudiante universitario) [Parte 1, 2, 3, 4]
- Un estudiante universitario en Líbano que, luego de participar por primera vez en la regional ICPC, se enamoró de la programación competitiva y entrenó hasta clasificar al mundial en 1 año [[From 0 to ICPC in one year](#)]
- Un programador competitivo retirado que volvió para cumplir su meta de llegar a **Maestro** en [Codeforces](#) antes de graduarse [Parte 1, 2, 3, 4]
- Un blog que explora la comunidad de CP [[Jollybee CP Team, the Luxor WF, and the Indomie Aftermath](#)]
- Una opinión (ligeramente controversial) de una leyenda de CP sobre lo que implica ser un programador competitivo [[My opinion on how to practice competitive programming](#)]
- Otra opinión (también ligeramente controversial) de otra leyenda de CP sobre cómo practicar [[How to practice Competitive Programming \(Um_nik version\)](#)]

2 Competir

Como programador competitivo a nivel universitario, existen varios concursos en los cuales se puede participar para obtener experiencia, premios y gloria tanto a nivel personal como institucional (en representación de la universidad):

2.1 International Collegiate Programming Contest (ICPC)

El concurso de programación competitiva a nivel universitario más antiguo, grande y prestigioso del mundo. La participación es en equipos de tres personas, que representan a su universidad. Actualmente, la competencia consta de 3 fases: regional, macro-regional y mundial. En cada fase la competencia dura 5 horas, durante las cuales los equipos se enfrentan a 11-12 problemas algorítmicos complejos, los cuales deben resolver con eficiencia y eficacia en el menor tiempo posible.

Cabe indicar que cada equipo solo cuenta con un computador. Además, está permitido llevar material de referencia impreso (sin límites para los concursos regionales y con un máximo de 20 hojas para la mundial).

2.1.1 Criterio de puntaje (*scoring criteria*)

1. Cantidad de problemas resueltos
2. Penalidad*

¿Cómo se determina la clasificación (*ranking*) de los equipos?

Se usa el siguiente criterio de ordenamiento:

Considerar los equipos A y B , si A tiene la misma cantidad de problemas resueltos que B , entonces se ubica primero el que tenga menor penalidad. Caso contrario, se ubica primero el que tenga mayor cantidad de problemas resueltos.

*Cálculo de la penalidad:

La penalidad del equipo es la suma de las penalidades individuales para cada problema que el equipo ha resuelto. En un primer envío correcto para un problema, se suma a la penalidad:

- La cantidad de segundos desde el inicio de la competencia.
- 20 minutos por cada solución incorrecta enviada para ese problema.

A nivel de LATAM, [aquí](#) se pueden revisar los resultados de las competiciones regionales y macro-regionales de los últimos años. Asimismo, para más detalle sobre criterios de elegibilidad, revisar [este enlace](#).

2.2 IEEEXtreme

La IEEEXtreme es una competencia internacional organizada por el IEEE (Institute of Electrical and Electronics Engineers). La competencia tiene una duración de 24 horas, durante las cuales se van emitiendo nuevos problemas periódicamente (cada 1 o 2 horas). Cada equipo de 3 personas se enfrenta en total a 24 problemas de diverso grado de complejidad y con variedad de tópicos y restricciones.

2.2.1 Criterio de puntaje (*scoring criteria*)

- Se otorga puntaje por resolver cada uno de los casos de prueba (es decir, hay **puntaje parcial**).
- El puntaje otorgado es directamente proporcional a la cantidad de casos de prueba resueltos (si se resuelven todos los casos de prueba, es puntaje completo).
- Cuando es liberado, un problema otorga como máximo 100 puntos (puntaje completo). No obstante, este puntaje va decreciendo en función de la cantidad de equipos que lo resuelven (parcial o completamente). Es decir, el puntaje que otorga un problema es representativo de su dificultad relativa.
- No hay penalidad por envíos erróneos.
- Se consideran todos los casos de prueba resueltos entre todos los envíos realizados.

2.3 Universal Cup

Esta es una de las competiciones internacionales de programación competitiva más importantes en la actualidad. La participación es en equipos de tres, sin restricción respecto a edad o estatus de estudiante. Es decir, puede participar cualquier persona. La modalidad es muy similar a la de un concurso ICPC.

2.4 Competiciones individuales

Además de los concursos oficiales en equipos, hay múltiples plataformas que hospedan competencias de CP semanalmente. Para no tener que estar revisando cada una para estar actualizado con próximos concursos, hay una herramienta que lo hace por ti: [CLIST](#).

3 Entrenar

Para entrenar, se recomienda tener un equipo (no tiene que ser definitivo) con el cual participar en concursos. Esto es útil tanto para intercambiar opiniones y formas de solución, así como para acostumbrarse a la dinámica de coordinar con otras personas, comunicar asertiva y efectivamente, y usar un único computador entre todo el equipo.

Para entrenamiento orientado hacia ICPC, se recomienda usar el Gym de Codeforces, donde se puede **simular** (participar en una competencia pasada con tabla de posiciones (*standings*) emulada como si fuera en vivo) concursos ICPC pasados de diversas regiones. Se puede acceder [aquí](#). También se recomienda bastante la participación en Universal Cup **una vez que ya se tenga experiencia como equipo**.

En paralelo, para entrenar individualmente, obviamente el foco es resolver problemas. Para ello, hay varias plataformas de CP, en las cuales se puede tanto competir como resolver de entre el amplio banco de problemas pasados:

- [Codeforces](#)
- [Atcoder](#)
- [Codechef](#)
- [LeetCode](#)

4 Aprender

[USACO Guide](#) es una página muy amigable y bien organizada que apunta a ser un recurso autosuficiente para prepararse para la IOI. Por ello, es una muy buena referencia para aprender el temario de CP, con una clasificación por dificultad que responde a las divisiones bronce, plata, oro y platino. Esta guía constantemente redirige hacia otras plataformas (varias de ellas mencionadas en este documento), por lo que es muy útil como punto de inicio en CP.

Asimismo, [USACO Training Gateway](#) también es una plataforma para aprender CP de manera estructurada y progresiva. En este caso, con problemas de estilo de olimpiada de informática, con un formato secuencial donde tienes que resolver un grupo de problemas para avanzar al siguiente.

Por otra parte, para aquellos que prefieren aprender con un libro de referencia, el [Competitive Programmer's Handbook](#) acompañado por la plataforma [CSES](#) es un excelente recurso para aprender todos los temas fundamentales de CP. Otra opción es usar [Programming Challenges](#) en conjunto con el juez virtual [Online Judge | uVa](#).

Finalmente, también hay páginas web que recopilan problemas y/o recursos clasificados por temas:

- [The Ultimate Topic List](#)
- [Striver's CP Sheet](#)