



# Habilidades Blandas/Duras y Teamwork

Por Ariel Parra

[ Γ α = Ω 5 ]

# Habilidades Blandas (Soft Skills)

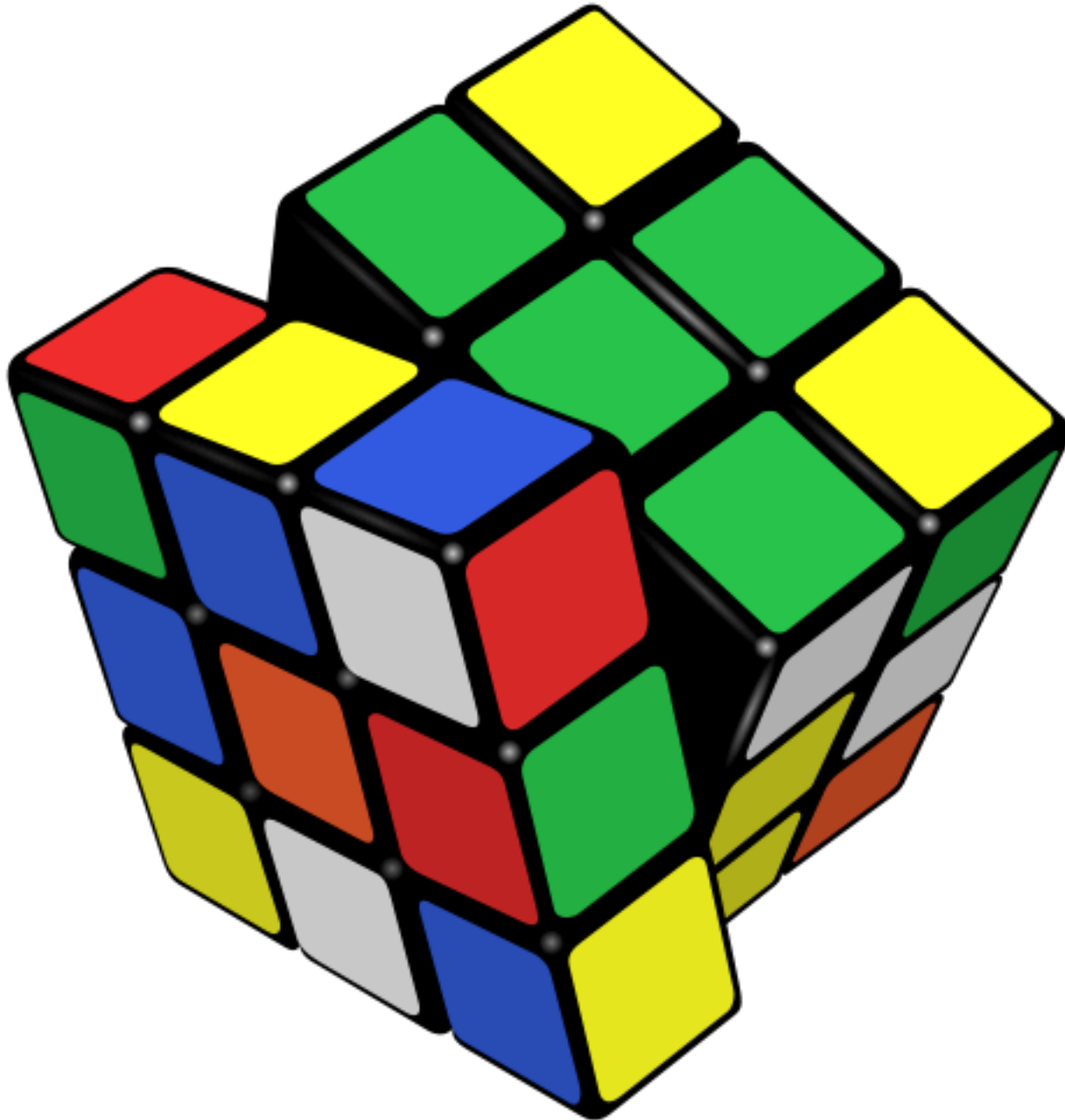
Las habilidades blandas son aquellas que reflejan cómo te relacionas, trabajas e interactúas con otras personas. Son fundamentalmente habilidades interpersonales que facilitan una comunicación y colaboración efectiva.

- Resolución de problemas
- Comunicación
- Adaptabilidad
- Gestión del tiempo
- Manejo del estrés
- Ética de trabajo
- Liderazgo
- Escucha activa

# Habilidades Duras (Hard Skills)

Las habilidades duras se adquieren a través de la educación y/o la experiencia. Son conocimientos técnicos específicos que pueden ser evaluados mediante títulos, diplomas o certificados.

- Velocidad de tecleo
- Escritura en teclado con distribución americana
- Conocimiento de idiomas (por ejemplo, inglés)
- Dominio de la sintaxis de un lenguaje de programación
- Conocimiento de Estructuras de datos y Algoritmos



# Estretegias individuales

- **Abstracción:** Es aislar conceptualmente una propiedad o función concreta de un problema y analizandolo a detalle.
- **Visualización:** Es imaginar, concpetualizar, dibujar ya sea fisicamente con lapiz y papel o con el dedo en el aire la serie de pasos para resolver el problema.
- **Analogía:** adaptación de la solución a un problema anterior que tiene características o mecanismos similares.
- **Casos base:** las condiciones de los problemas suelen contener soluciones implicitas en la declaración.
- **Dividir y conquistar:** divide un problema grande y complejo en problemas más pequeños y resolvibles.
- **Pruebas de hipótesis:** consiste en suponer una posible explicación para el problema y tratar de probar (o, en algunos contextos, refutar) esa suposición. Esto puede incluir asumir el tipo de problema y/o el paradigma de resolución.
  - **¿Tienes una teoría?** Inserta impresiones de diagnóstico o casos de prueba para verificarla.
  - **¿No tienes una teoría?** Inserta impresiones de diagnóstico o casos de prueba para obtener más información.

# Trabajo en Equipo (Teamwork)

“ Un equipo en el contexto de concursos como el ICPC suele estar compuesto por tres personas que deben colaborar de manera efectiva para maximizar el rendimiento del equipo. Existen varias estrategias de colaboración que se pueden emplear, dependiendo de las habilidades y preferencias de los miembros.

- **Terminal Man (TM)**

Un miembro del equipo, altamente competente en programación, se encarga de toda la codificación mientras los otros dos redactan pseudocódigo.

- **Ventajas:** Asegura que la programación sea eficiente, ya que una persona se concentra exclusivamente en el código.
- **Desventajas:** Puede limitar la creatividad y el análisis de problemas más complejos, ya que sólo una persona se encarga de resolverlos de manera activa.

- **Think Tank (TT) o Doble Apoyo (Double Up)**

En esta estrategia, dos miembros del equipo se dedican al análisis de los problemas mientras el tercero se concentra en la programación de subrutinas estándar y en la creación de casos de prueba. Alternativamente, dos miembros colaboran en cada problema: uno escribe el código mientras el otro genera los datos de prueba.

- **Ventajas:** Permite abordar problemas complejos desde el principio, aprovechando el análisis conjunto. Facilita una depuración más rápida gracias a la división de tareas.
- **Desventajas:** El inicio es lento y el tiempo total de resolución puede no ser óptimo. Si la evaluación falla, el programador encargado de los datos de prueba asume la responsabilidad, lo cual puede ralentizar el proceso.

- **Estrategia Simple (The Simple Strategy) o Mi Elección (My Choice)**

Cada miembro del equipo trabaja de forma independiente, maximizando el uso del "CPU humano" y minimizando los cambios de contexto.

- **Ventajas:** Ideal para equipos avanzados, donde cada miembro puede resolver cualquier problema. Reduce la dependencia y permite un enfoque flexible.
- **Desventajas:** Es una estrategia frágil, ya que el trabajo independiente puede llevar a una falta de cohesión y a la duplicación de esfuerzos.

- **Tener Reglas Claras**

En lugar de estrategias fijas, algunos equipos establecen un conjunto de reglas simples y ajustables. Por ejemplo, podrían imprimir una copia del código después de cada envío y hacer que otra persona lo revise si es rechazado. También pueden establecer que ningún problema quede sin leer después de la primera hora.

- **Ventajas:** Permite flexibilidad y adaptación continua mediante la retroalimentación de las prácticas de entrenamiento.
- **Desventajas:** Requiere tiempo de ajuste para determinar qué reglas se adaptan mejor al equipo.

- **Lluvia de Ideas**

Utilizando cualquiera de los métodos anteriores, los miembros del equipo sugieren ideas y soluciones, las combinan y desarrollan hasta encontrar la más óptima.

# Puntaje

- En la mayoría de los concursos, como el ICPC, el equipo que resuelve la mayor cantidad de problemas en el menor tiempo gana.
- **Desempates mediante Puntos de Penalización (PP):**
  - Los PP se evalúan problema por problema.
  - Se otorga 1 PP por cada minuto transcurrido desde el inicio del concurso hasta que se juzga correcta la solución de un problema.
  - Se asignan 20 PP por cada envío incorrecto a los jueces.
  - No se asignan PP a los problemas que no se resuelven.
  - Obviamente, menos PP es mejor.

“ Lo más importante es identificar el problema más fácil primero para obtener más puntos, por lo que es crucial detectarlo rápidamente.



# Distribución de la Computadora y Teclado

1. **Entender el problema antes de empezar a escribirlo:** Solo los primeros problemas fáciles pueden ser redactados directamente en el teclado.

- En las finales, probablemente habrá solo 1 problema fácil para escribir de inmediato.

2. **Después de eso, todos trabajan en papel:**

- Escribir todo el código.
- Escribir todos los casos de prueba.
- Hacer impresiones y depurar fuera del terminal.

3. **Jerarquía de Prioridades:**

- i. Solución dentro de 5 minutos.
- ii. Corrección Rápida (detallada más adelante).
- iii. Nuevo código/casos de prueba (prioridad a lo más cercano a completarse).
- iv. Código incompleto/redacción en teclado.

4. **El teclado nunca debe estar inactivo:** Se debe optimizar el uso del teclado para maximizar el rendimiento del equipo.

# Control

- **Tiempo y Estrés:**

- Se debe optimizar cada segundo en el uso del teclado, la comunicación, el pensamiento de soluciones y la prueba de casos.
- Hay un límite de tiempo y muchas tareas por hacer; es fundamental empezar tranquilizándose y respirando.
- Después, se analizan los problemas y se asignan contadores de tiempo. Alguien debe monitorear el tiempo, y si un objetivo no se cumple dentro del tiempo determinado, un participante puede seguir trabajando en él, mientras los demás continúan con el siguiente problema.

- **Liderazgo del Equipo:**

- Se suele necesitar un líder que conozca tanto las habilidades técnicas como interpersonales de cada miembro.
- Este líder toma decisiones difíciles, organiza el tiempo de resolución para cada problema y decide cuáles se abordarán primero y por quién.

# Comunicación Asertiva

- **Respeto:** No hay necesidad de insultar o menospreciar las ideas de tus compañeros.
- **Aprender a Escuchar:** No todos pueden hablar a la vez. Si un compañero tiene una idea, escúchala y analízala antes de proponer la tuya.
- **Ser Críticos:** Si no entendiste el problema, tómate el tiempo para comprenderlo y luego propón tus ideas.
- **Lenguaje Corporal:** Usa movimientos de manos o dedos para hacer más comprensiva tu idea y facilitar la comunicación.

Aquí tienes las secciones resumidas:

- **Escucha Activa:** Presta atención, mantén contacto visual y haz preguntas de seguimiento para clarificar. Fomenta un ambiente de respeto y colaboración.
- **Feedback Constructivo:** Ofrece retroalimentación positiva y sugiere mejoras. Esto ayuda al crecimiento del equipo y crea confianza.
- **Empatía:** Comprende las perspectivas y emociones de tus compañeros para facilitar una comunicación efectiva.
- **Claridad y Concisión:** Comunica tus ideas de forma clara y directa. Evita jerga y rodeos para minimizar malentendidos.

# Problema

- **1968B** Prefiquence ↗
- **330A** Cakeminator ↗

# Referencias

- Ernst, F. et al. (2000). *Teamwork in Programming*. Recuperado de <http://arantxa.ii.uam.es/~swerc/p6-ernst.pdf> ↗
- Haykal, I. (2017). *Comunicación asertiva: cómo expresarse de manera clara*. Recuperado de <https://psicologiaymente.com/social/comunicacion-asertiva> ↗
- Liu, R. (s.f.). *Training ICPC Teams: A Technical Guide*. Recuperado de <https://www.cs.montana.edu/users/paxton/classes/ravensburg/materials/TrainingICPC.pdf> ↗
- Santander Universidades. (2021). *Habilidades para currículum: cuáles son las más importantes y cómo destacarlas*. Recuperado de <https://www.santanderopenacademy.com/es/blog/habilidades-para-curriculum.html> ↗
- Van, D. (2023). *ICPC Programming Team Strategy*. Recuperado de [https://drive.google.com/drive/folders/1dQyK3OTgdh\\_HJ7WdZGKrURcMBmeXKDL1](https://drive.google.com/drive/folders/1dQyK3OTgdh_HJ7WdZGKrURcMBmeXKDL1) ↗