

## *lecture problem solving:*

te preguntarás exactamente que es pensar como un programador? ¿y cómo lo logras?  
esencialmente es todo sobre una manera más efectiva de resolver problemas

y viene la pregunta de nuevo? y como lo logras?

debes tener dos elementos:

1. Una estructura de pensamiento.
2. Practicarla (la estructura)

casi todos los empleadores se enfatizan en la capacidad de resolver problemas primero.  
(mucho más que ser competente en sintaxis, o en depurar errores y diseño)

"demostrar pensamiento computacional para descomponer problemas a gran escala es tan o más importante que las bases técnicas requeridas para un trabajo"

cita: Hacker Rank (2018 Developer Skills Report)

"el gran error que veo en nuevos programadores es que se concentran en aprender bien la sintaxis en vez de aprender a resolver problemas"

cita: V. Anton Spraul

entonces que debemos hacer cada vez que encontramos un nuevo problema, aquí están los pasos:

## 1.- ENTIENDE

Entiende exactamente lo que te están pidiendo. Los problemas más difíciles se originan de no entender lo que te piden (por eso este es el primer paso)

¿Cómo sabemos que entendimos un problema? cuando puedes explicarlo de manera simple.

recuerdas haber estado atascado en un problema, y cuando comienzas a explicarlo instantáneamente ves toda la lógica que no habías visto antes?

la mayoría de los programadores conocen este sentimiento.

por esta misma razón deberías escribir el problema, dibujar un diagrama, explicárselo a alguien (algunas personas usan el rubber duck -patito de plástico-)

"si tú no puedes explicar algo en términos simples, tu simplemente no lo entiendes"

cita: Richard Feynman

## 2.- PLAN

No te aventures a resolver un problema sin tener un plan (y de alguna manera podrás atravesarlo). Planea tu solución.

Nada puede ayudarte si no escribes los pasos exactos.

En programación no empieces a hackear al instante. Dale tiempo a tu cerebro para que analice el problema y procese la información.

Para obtener un plan acertado, responde esta pregunta:

**"Dada la entrada X, cuales son los pasos necesarios para obtener la salida Y? "**

Nota: los programadores tienen una gran herramienta para ayudar con este plan. comentarios!

### 3.-Divide

Presta atención. Este es el paso más importante de todos.

No trates de resolver un problema grandísimo. Vas a llorar.

En cambio, desglósalo en sub-problemas. Estos sub - problemas son más fáciles de resolver.

Entonces, soluciona cada sub-problema uno a uno. Comienza con el más simple. Conoces la respuesta del más simple (o ya estas más cerca de esa respuesta).

Después de haberlo resuelto significa que su solución no depende de otros sub-problemas.

Después de haber solucionado cada sub-problema, conecta los puntos y eso te dará la solución al problema original.

Esta técnica es la piedra angular para solucionar problemas.

**"si pudiera enseñarle a cada programador principiante, una habilidad de resolver problemas. sería 'la técnica de reducir problemas'.**

por ejemplo, supongamos que tú eres un nuevo programador, y te piden escribir un programa que lea 10 números y halle el tercer número más alto. Para un nuevo programador esto sería una tarea difícil aun cuando esto requiera sintaxis básica.

si estas atascado, deberías reducir el problema a algo más simple. ¿En vez de hallar el tercer número más alto, que tal si hallamos el más alto de todos? sigue siendo difícil? ¿qué tal si hallamos el número más alto entre 3 números?... o el número más alto entre 2 números?

Reducir el problema al punto donde tú los resuelves y escribes la solución. Desde ese punto expandimos el problema y reescribimos la solución que coincida. y sigue hasta que vuelvas a donde empezaste"

cita: V. Anton Spraul

#### 4.- ATASCADO?

Por ahora estarás sentado diciendo: "Hey Richard... esta súper y todo, pero que pasa si no puedo siquiera resolver un sub-problema??"

Primero, toma un respiro profundo. Eso es justo.

¡No te preocupes mi amigo, eso les pasa a todos!

La diferencia entre los mejores programadores que solucionan problemas es que se convierten en más curiosos que irritables.

De hecho, hay 3 cosas que debes hacer cuando enfrentas un golpe:

1. **Depurar:** ve paso a paso a través de tu solución intentando hallar el lugar donde te equivocaste. Los programadores llaman a esto depurar

**"El arte de depurar (debugging) es hallar aquello que realmente le dijiste a tu programa que hiciera vs lo que debe hacer, "**

cita: Andrew Singer

2. **Reevaluar:** dar un paso atrás. Mira el problema desde otra perspectiva. ¿Hay algo que pueda resumirse en un enfoque más general?

"A veces nos perdemos en los detalles de un problema dejando pasar desapercibido los principios que podrían resolver este problema a un nivel más general.

El clásico ejemplo de esto, es la suma de una larga lista de números enteros consecutivos  $1+2+3+...+n$  que un joven Gauss lo reconoció como simplemente  $n(n+1)/2$ , evitando el esfuerzo de hacer la suma"

cita: C.Jordan Ball

**Nota:** otra forma de reevaluar, es un nuevo comienzo desde cero. Probaras lo efectivo que es.

3. **Investiga:** Bueno google. No importa que problema tengas probablemente alguien lo ha resuelto. Haz esto inclusive si has logrado resolver el problema (puedes aprender de las soluciones de otras personas)

**Advertencia:** no busque una solución al gran problema. Solo busque soluciones a sub-problemas. ¿Por qué? Porque a menos que luches (aunque sea un poco), no aprenderás nada. Si no aprendes nada, perdiste el tiempo.

## Práctica

No esperes ser genial después de solo una semana. Si quieres ser un buen solucionador de problemas, ¡resuelve muchos problemas!

Práctica. Práctica. Práctica. Solo será cuestión de tiempo antes de que reconozca que "este problema podría resolverse fácilmente con <insertar concepto aquí>".

¿Cómo practicar? ¡Hay opciones fuera del wazool!

Rompecabezas de ajedrez, problemas matemáticos, Sudoku, Go, Monopoly, videojuegos, criptokitties, bla ... bla ... bla ...

De hecho, un patrón común entre las personas exitosas es su hábito de practicar la "micro resolución de problemas". Por ejemplo, Peter Thiel juega ajedrez y Elon Musk juega videojuegos.

**"Byron Reeves dijo:" Si quieres ver cómo se vería el liderazgo empresarial en tres o cinco años, mira lo que sucede en los juegos en línea ".**

**Avance rápido hasta hoy. Elon [Musk], Reid [Hoffman], Mark Zuckerberg y muchos otros dicen que los juegos han sido fundamentales para su éxito en la construcción de sus empresas ". - Mary Meeker (informe de tendencias de internet de 2017)**

¿Esto significa que solo deberías jugar videojuegos? De ningún modo.

¿Pero de qué se tratan los videojuegos? Así es, ¡resolución de problemas!

Entonces, lo que debe hacer es encontrar una salida para practicar. Algo que le permite resolver muchos micro problemas (idealmente, algo que disfrute).

Por ejemplo, disfruto los desafíos de codificación. Todos los días, trato de resolver al menos un desafío (generalmente en Coderbyte).

Como dije, todos los problemas comparten patrones similares.

## **Conclusión**

¡Eso es todo amigos!

Ahora, usted sabe mejor lo que significa "pensar como un programador".

También sabes que la resolución de problemas es una habilidad increíble para cultivar (la meta-habilidad).

Como si eso no fuera suficiente, ¡observe cómo también sabe qué hacer para practicar sus habilidades de resolución de problemas!

Uf ... ¿Genial, ¿verdad?

Finalmente, deseo que encuentres muchos problemas.

Lo has leído bien. ¡Al menos ahora sabes cómo resolverlos! (también, aprenderá que, con cada solución, mejora).

**"Justo cuando crees que has superado con éxito un obstáculo, surge otro. Pero eso es lo que mantiene la vida interesante. [...]"**

La vida es un proceso de romper estos impedimentos, una serie de líneas fortificadas que debemos romper.

Cada vez, aprenderás algo.

Cada vez, desarrollarás fuerza, sabiduría y perspectiva.

Cada vez, un poco más de la competencia se cae. Hasta que todo lo que quede sea usted: la mejor versión de usted ". – Ryan Holiday (El obstáculo es el camino)

¡Ahora, ve a resolver algunos problemas!

¿Y la mejor de las suertes?