

14 Sugerencias para la resolución de problemas técnicos aparentemente inexplicables

Departamento de Sistemas Telemáticos y Computación (GSyC)

gsync-profes (arroba) gsync.es

Septiembre de 2012



©2012 GSyC
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike 3.0

Problemas aparentemente inexplicables

Especialmente trabajando con ordenadores, pero también con otro tipo de equipos, es frecuente oír

- Este fallo es imposible que se produzca
- Ayer funcionaba perfectamente, y ahora, sin cambiar nada, ya no
- Lo he probado todo y es un problema que no tiene solución

Haremos aquí algunas sugerencias, en su mayoría extraídas literalmente de

10 Consejos para la resolución de problemas técnicos inexplicables

Blog de Manuel Pereira González, octubre 2010

Sugerencia 0: Lee bien lo que se espera que leas

- En la vida *real* o en proyecto académico de cierta entidad (PFC, TFM, tesis) no hay ninguna persona que previamente haya dimensionado el problema y que haya puesto a tu alcance todos los elementos necesarios para que encuentres la solución, en un tiempo acotado
- ¡Pero en las prácticas de una asignatura o en un examen sí!
 - Lee varias veces y con mucha atención el guión, las transparencias, etc
 - Lee varias veces y con mucha atención el examen
- Lee bien el error que te de el sistema. Cópialo, es el punto de partida para encontrar la solución

Sugerencia 1. De lo que te cuenten, no te creas nada

... y de lo que veas, la mitad

- *Los pacientes siempre mienten*
Dr. Gregory House
- *Las personas mienten, las pruebas no*
Gilbert Grissom
- *Los usuarios siempre mienten*
(cualquier BOFH)

Por ejemplo, *ya no funciona pero yo no he tocado nada*
frecuentemente significa...

- *el cable está desconectado*
- *he instalado el programa XXX*
- *he borrado una carpeta que no sabía lo que era*
- *he hecho lo que el manual dice en letras verde fosforito que no hay que hacer jamás*

Es importante disimular nuestra desconfianza, para que no resulte ofensiva

Sugerencia 2. Si simplemente funciona, está mal

Si un programador dice *mi programa funciona, yo lo probé*, seguramente está lleno de errores

¿Qué pensaríamos de?

- Un médico que dice *el enfermo vive*
Un arquitecto que dice *la casa está en pie*
Un carpintero que dice *el armario no se ha caído*

Si un programador lo único que puede decir de su programa es que *funciona*, con toda probabilidad se ha limitado a hacerle un par de pruebas triviales

Especialmente aplicable al estudiante al que falla una práctica solo con moverla al ordenador donde la probará el profesor. O al novato que hace una demo a un jefe o un cliente.

Sugerencia 3: Simplifica el problema

- Principio tradicional *divide y vencerás*
- Prepara un ejemplo mínimo (programa o configuración) donde se presente el problema
- Es frecuente que preparando esto resolvamos el problema

Sugerencia 4: Piensa

Hay una técnica básica que arregla muchos problemas

- 1 Sal de la aplicación y vuelve a entrar (o reinicia el servicio)
- 2 Y si no, apaga el ordenador y vuelve a encender
- 3 Y si no, si es una red Windows, apaga todos los equipos y vuelvelos a encender, empezando por el servidor
- 4 Y si no, reinstala la aplicación
- 5 Y si no, formatea todo y reinstala

Esto está bien. Funciona muchas veces, y hasta el paso 2 podemos aplicarlo sin pensarnoslo demasiado

Pero no es la panacea. Deberíamos analizar el problema y considerar otras soluciones además de esta

Otro ejemplo:

- Programador de C que empieza a poner * y & al tuntún hasta que (aparentemente) funciona
- Programador que usa los errores del compilador para saber por dónde seguir programando. Deberías compilar solo cuando consideres posible que todo funcione

Trabaja con la cabeza, no solo con los dedos

Sugerencia 5: La máquina tiene la presunción de inocencia

Los humanos tenemos una tendencia natural a echarle la culpa a otro, no solo en el ámbito de informática y comunicaciones

- Es muy frecuente que una empresa nos diga *ha sido un fallo informático* (lo que realmente suele significar que una persona delante de un ordenador ha cometido un error)
- *Será el disco duro, será que la memoria está defectuosa, será la máquina virtual, será el sistema operativo*
- Puede ser, pero ¿Qué es más probable?
 - Un fallo en tu código / tu configuración, reciente y apenas probada
 - Un extraño fallo de hardware que solo se manifiesta en tu código
 - Un fallo en una aplicación de sistema o un sistema operativo que usan miles de personas en todo el mundo

Sugerencia 6: Si no hay más remedio, baja de nivel de abstracción

La sugerencia anterior no es infalible

A veces hay que descender: A la máquina virtual, al sistema operativo, a la bios

¡Pero el hardware es el nivel más bajo! Hay muchos escalones antes

Sugerencia 7: Si ayer funcionaba y hoy no, algo ha cambiado

- ...al menos ha cambiado la fecha
- Puede haber cambiado lo que has hecho antes

Si has cambiado a otro ordenador pero que es *igual*, luego resulta que

- *ah, que la ruta en el servidor no existe...*
- *Vaya, las variables de entorno de mi usuario ahora son...*
- *¡Anda! Si resulta que no dijimos que hacía falta tener la JDK 1.6*

Sugerencia 8: Pregúntale (bien) a Google

Es muy poco probable que seas la primer persona en el mundo a quien se le presente cierto error

Merece la pena que dedique cierto tiempo a *googlear* tu problema

- Si es un error concreto, copialo y pegalo
- Si es más abstracto, busca bien, con palabras *distintivas*, palabras que identifiquen bien ese problema
- Recuerda que el universo angloparlante es mayor que el hispano

Sugerencia 9: A veces es más fácil hallar la solución que la causa

En ocasiones no encontramos la verdadera causa, pero encontramos una solución alternativa.

atajo (en español) *workaround* (en inglés) *ñapa* (en castizo)

No es fácil dar una sugerencia absoluto ante esto

- En muchas ocasiones esto nos bastará. Es normal que haya que hacer muchas cosas en poco tiempo y *para ayer*. Satisfacer nuestra curiosidad o simplemente, aprender, puede ser un lujo
- Por otro lado, un problema que no has arreglado realmente, sigue allí y puede volver a manifestarse (o no)

¿Si funciona no lo toques?

vs

Si no sabes cómo lo has arreglado, no lo has arreglado

Sugerencia 10: Muchos ojos ven más que dos

- Por supuesto, alguien más experimentado puede ayudarte
- Pero cualquier persona nueva puede aportar un nuevo enfoque, aunque conozca la materia menos que tú
- En cualquier caso, contárselo a alguien nos ayuda a ordenar nuestras ideas
 - Incluso hablar con un oso de peluche

Sugerencia 11: A veces la mejor solución a un problema es irse a dormir

- Descansar, pensar en otro cosa, tomar el aire
- Los problemas pueden verse mejor despues de dormir ¡Incluso resolverse en sueños!

Sugerencia 12: Se metódico en las pruebas

- Piensa qué probar
- Hazlo en orden, manteniendo siempre el mismo orden
- Registra todas las pruebas

Sugerencia 13: Resuelve los problemas una sola vez

Un problema que tienes hoy, puede volver a presentarse en el futuro, cuando no recuerdes los detalles de cómo lo solucionaste

- Con un poco de suerte, dos años después podrás resolverlo en mucho menos tiempo, siguiendo los mismos pasos
- Pero es un derroche. ANOTA TODO LO QUE HAGAS, para poder consultarlo en el futuro
Y anótalo en electrónico
 - En un fichero de texto se busca inmediatamente, en un cuaderno no
 - Los papeles de toda la carrera abultarán varias cajas de papel, tal vez en casa de tus padres, hasta que acaben en el contenedor azul
 - En electrónico, la documentación que generes toda tu vida cabe sin problemas en cualquiera de tus ordenadores, ¡hasta en un *smartphone*!