Troubleshooting:

Este termino se usa para definir el proceso de identificar, analizar y resolver problemas, se refiere a cualquier clase de problema, no solo en IT sino de manera general. Decimos Troubleshooting cuando nos referimos a los problemas del sistema cuando corremos la aplicación.

Debugging:

Este termino se usa para definir el proceso de identificar, analizar, y remover bugs en un sistema. Decimos Debugging cuando estamos refiriéndonos a los problemas, a los bugs que tiene el código de dicha aplicación.

Debuggers:

Son herramientas que nos permiten seguir el código, línea a línea, inspeccionar cambios en asignación de variable, interrumpir el programa cuando se encuentran condiciones específicas y más.

Pasos para resolver problemas técnicos:

* Obtener información.

Reproduction case(descripción de cómo y cuando apareció el problema)

* Encontrar la causa, la raíz.
* Remidiation.

Este termino se refiere a encontrar una solución, a veces nos encontramos con que necesitamos diferentes tipos de Remidiation, uno a corto plazo(Workaround: traer el sistema de vuelta para que siga funcionando mientras se encuentra la solución definitiva), y otros a mediano y largo plazo, para que este problema ni vuelva a suceder.

Es super importante documentar todo, para que la próxima vez que aparezca un problema similar, ya sabemos cómo afrontarlo.

Herramientas de Debugging:

* strace: Rastrea las system calls que hace el programa y nos da el resultado que cual es el output resultante de la system call, podemos usar el comando **les** para verlo de a poco.

System calls: son las llamadas que hace el programa en nuestra computadora al kernel activo.

* strace -o failure.strace ./ file.py: Este comando lo que hace es crear un fichero, y en este almacena el output de este comando strace, es como hacer un pipeline. En este comando, la **-o** se refiere a la orden que le damos al comando de crear un nuevo archivo, open, **failure.strace** se refiere al nombre del archivo, done **strace**, es la extensión de dicho archivo, y luego, **file.py** es el archivo que estamos analizando.

Cuando obtenemos un reporte de que no está funcionando un servicio, lo primero que debemos hacer es ir a alas soluciones más simples primer, así nos ahorramos más tiempo. Otra estrategia que es simple pero poderosa es reproducir el problema en tu computadora, si el problema persiste, entonces no debemos buscar problemas que tengas que ver con el software del usuario, o con su conexión a internet, hay un problema real en el servicio, y este necesita ser resuelto.

Luego de esto debemos ver si los sitios web o servicios que están detenidos están ubicados en el mismo servidor, de esta manera nos aseguramos de que no es problema del servidor esto nos ayuda a aislar la causa raíz.

Si hay problemas en ese servidor, lo checamos y vemos si este esta sobrecargado, si lo esta detenemos el proceso momentáneamente.

Reproduction case: Esto se refiere a saber si el problema finalmente se arregló o no.

Si una vez que reproducimos este problema el problema en nuestra computadora, este no se manifiesta, debemos mirar en los logfiles:

Linux: var/log/syslog .xsession-errors

MasOs: /Library/Logs

Windows: se usa la herramienta **Event Viewer**

Es importante que podamos reproducir el problema en nuestra computadora tal y como lo tiene el usuario, de esta manera podemos determinar eficientemente la cause raíz del problema.

Encontrando la causa:

Podemos bien analizar el problema usando la documentación de la app, o simplemente pasando ese error en el buscador y viendo si alguien mas ha tenido este problema, de que se trata el error, y si ya fue resuelto antes, podemos intentar eso para ahorrar tiempo y esfuerzo.

Intermitent Issues:

Estos se refieren a los problemas que ocurren ocasionalmente, a veces si y a veces no. Estos son los mas problemáticos ya que muchas veces son muy difíciles de reproducir y arreglar. Lo primero es intentar encontrar patrones de cuando ocurre y cuando no.

El problema con estos es que cuando ocurren, y no los podemos reproducir, necesitamos más información sobre estos, y esta información, no la tenemos, por lo que debemos preparar nuestra computadora, modificar el código para poder recibir mas informaciones en los log-files y así poder obtener información valiosa la próxima vez que ocurra.

Muchas veces el bug es eliminado son solo agregar más información al log, esto es lo que se llaman Heisenbug bugs, en estos casos, el solo hecho de observar el fenómeno, lo altera.