

# Buscar y solucionar errores en el código

## IDENTIFICAR DÓNDE SE PRODUCE EL ERROR

- Analizar hasta qué punto el programa se ejecuta correctamente y dónde empieza a fallar.
- Detectar en qué casos se provoca el error y en cuáles no (qué datos de entrada se usan o qué pasos se siguen para reproducirlo).
- Ejecutar el código con una herramienta “debugger” poniendo los “breakpoints” necesarios justo antes del punto donde se provoca el error, observando los valores de las variables.
- Poner “prints” en lugares estratégicos para observar qué se ejecuta hasta ese punto.
- Comentar líneas de código y ejecutar sucesivamente hasta ver dónde se introdujo el error.
- Al programar, realizar pequeños cambios incrementales y ejecutar frecuentemente, para evitar tener que “rastrear” problemas en grandes porciones de código.

## SI HAY UN MENSAJE DE ERROR



- Leer e interpretar el o los mensajes de error.
- Ver qué parte del código lo provoca (qué método, función o línea de código).
- Analizar el “stack trace” para “seguir el rastro” del error.
- Buscar más detalles en “logs” de la aplicación, consola, etc.
- Si el error es en tiempo de ejecución: ¿qué datos de entrada o pasos lo provocan?
- Si el error está en español, buscar su versión en inglés para encontrar más información.
- Googlear, ver cómo lo solucionaron otras personas, preferentemente en inglés.

## SUGERENCIAS PARA CUANDO TODO PARECE PERDIDO

*Mirar a la pantalla desesperadamente no ayuda. Es mejor tomar un descanso y volver con la mente despejada.*

*Aplicar el “método del patito de goma”: buscar un patito de juguete y “explicarle” el código, línea a línea.*

*Programación de a pares: conseguir que otro programador revise el código en busca de errores.*

## PEDIR AYUDA CORRECTAMENTE

- Antes de pedir a otra persona que dedique tiempo en buscar una solución a nuestro problema, es importante demostrar que intentamos buscarla por nuestros propios medios.
- Explicar qué cosas intentamos y por qué no funcionaron.
- Formular las preguntas de forma coherente y fácilmente comprensible (evitar que la otra persona deba “descifrar” lo que se intenta decir).
- Agregar los detalles necesarios: lenguaje, compilador, frameworks, versiones, entorno...
- Acotar: mostrar la menor parte del código necesaria para analizar el error (evitar información incompleta, así como mostrar todo el código para que la persona busque la parte pertinente).
- Mostrar el mensaje de error exacto (si existe), stack trace, logs, mensajes de consola, etc.
- Evitar capturas de pantalla (copiar y pegar los fragmentos de código relativos al problema).
- No utilizar código escrito por otra persona sin comprenderlo completamente.