

Otras herramientas para el manejar errores

Nombre: Flores Estrada Abraham Miguel Angel

Codigo: 217443356

Carrera: INCO

Profesor: Michel Emanuel Lopez Franco

Sección: D06

Materia: Computación tolerante a fallas



Presentación

A la hora de crear aplicaciones o desarrollar distintos proyectos es posible que nos encontremos con errores y que estos errores destruyan nuestro código. En algunas ocasiones estos problemas pueden hacer que la ejecución se vea interrumpida y que el programa deje de funcionar.

Desarrollo

Existen varias herramientas para el manejo de errores en la programación, algunas de las cuales incluyen:

- Depuradores: son programas que te ayudan a encontrar y solucionar errores en tu código al proporcionar información detallada sobre cómo se está ejecutando tu programa.
- Monitores de excepciones: son herramientas que te ayudan a detectar y rastrear excepciones en tiempo de ejecución, proporcionando información detallada sobre cuándo y dónde se producen las excepciones.
- Analizadores de código estático: son herramientas que analizan el código fuente para detectar posibles errores o problemas antes de que el código se ejecute.
- Registros de depuración: son archivos o registros que se crean durante la ejecución del programa para proporcionar información detallada sobre la ejecución del programa y ayudar a encontrar y solucionar problemas.
- Monitores de rendimiento: son herramientas que te ayudan a medir el rendimiento y la eficiencia de tu programa, permitiéndote detectar cuellos de botella y problemas de rendimiento.
- Sistemas de seguimiento de errores: son herramientas que permiten recopilar información detallada sobre errores y excepciones que ocurren en producción, lo que facilita la detección y solución de problemas en un entorno de producción.

Cada una de estas herramientas tiene sus ventajas y desventajas, y la mejor opción dependerá del lenguaje de programación y el entorno de trabajo.

La herramienta que he encontrado es debugger en JavaScript, la cual es utilizada en varias ocasiones para encontrar y evitar fallos en el código.

Un depurador (debugger) es una herramienta utilizada para encontrar y solucionar errores en el código JavaScript. Funciona deteniendo la ejecución del código en puntos específicos para que puedas inspeccionar y analizar el estado del programa en ese momento.

Entre las funcionalidades que suelen incluir los depuradores de javascript se encuentran:

- Pausar la ejecución del código en puntos específicos

- Ver el valor de las variables en cualquier punto del código
- Ejecutar paso a paso el código
- Ver el call stack
- Modificar valores de las variables
- Evaluar expresiones
- Etc.

Estas herramientas son útiles para encontrar y solucionar errores como bucles infinitos, problemas de lógica, y problemas de memoria, entre otros. Los depuradores pueden ser utilizados en el navegador o en un entorno de desarrollo integrado (IDE) y son compatibles con diferentes navegadores como Chrome, Firefox, Safari o Edge.

La manera en que puede ser utilizado en código es la siguiente:

```
function codigoPotencialmenteDefectuoso() {
  debugger;
  // realizar paso a paso o examinar código que contiene
  // potenciales errores
}
```

En un ejemplo práctico podríamos encontrar un error lógico como el siguiente en el que a pesar de que la suma está bien podemos ver que cuando se hace la suma el resultado es correcto, por lo que en ese estado contamos con un resultado correcto. Pero cuando la función termina de ejecutarse el valor de z se vuelve a declarar por lo que el error está ocurriendo ahí.

```
1
2 var x = 5;
3 var y = 10;
4 var z = suma(x, y);
5 z=2;
6 console.log(z);
7
8 function suma(a, b) {  a = 5, b = 10
9   var resultado = a + b;  resultado = 15
10  debugger;
11  return resultado;
12 }
```

Conclusión

Esta herramienta debugger puede ser muy útil para encontrar errores, puede ayudar a encontrar errores lógicos y no necesariamente sintácticos, ya que podemos detener la ejecución del programa en diferentes estados para así encontrar el error lógico e ir analizando los estados por los que van pasando ya

sean variables u otro tipo de elementos.