

# Manejo de errores

## Errores en tiempo de compilación

Los errores en tiempo de compilación son errores que impiden que el programa se ejecute. Cuando se compila el código para ejecutar un programa, el código se transforma en un lenguaje binario que entiende el equipo. Si el compilador detecta código que no entiende, se produce un error en tiempo de compilación.

La mayoría de estos errores se cometen al escribir el código. Por ejemplo, puede escribir erróneamente una palabra clave u omitir alguna puntuación necesaria.

Ejemplos de errores en tiempo de compilación:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int numero = 10    //omitir punto y coma al final, produce un error.
    if(numero >=5)
    {
        pritnf("Hola\n");
    } else {
        printf("Bye bye!!!\n");
    }

    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    //int numero
    numero = 10; //Omitir la declaracion de la variable produce un error.

    if(numero >=5)
    {
        pritnf("Hola\n");
    } else {
```

```

        printf("Bye bye!!!\n");
    }

    return 0;
}

```

```

#include <stdio.h>
#include <stdlib.h> //escribir mal "include", produce un error.

int main()
{
    int numero = 10;

    if(numero >=5)
    {
        printf("Hola\n");
    } else {
        printf("Bye bye!!!\n");
    }

    return 0;
}

```

## Errores en tiempo de ejecución

Los errores en tiempo de ejecución son los que se producen durante la ejecución del proceso. Son los más difíciles de encontrar, no son detectados por el compilador, ya que son errores de lógica, no de sintaxis. Aunque al compilar un programa no de errores, el programa puede funcionar incorrectamente y/o a dar errores durante su ejecución.

Encontrar la causa que los provoca es en ocasiones complicado y tedioso, algunos editores de código traen un “Depurador” pero la mejor manera es ir comentando el código con comentarios de bloque (“/\*” para iniciar, y “\*/” para finalizar el bloque), para ir descartando sectores del código que estén provocando el error y así poder encontrar la línea que da el error. Este tipo de errores se evitan con la práctica.

Ejemplos de errores en tiempo de compilación:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int metros = 10;
    int horas = 0;
    int velocidad = metros / horas; //error de division por cero.
    printf("Velocidad: %d m/h.\n", velocidad);
    return 0;
}
```

Observación: el proceso detecta este error únicamente en tiempo de ejecución, en caso de que la variable “horas” tenga un valor válido, no se produce dicho error.

```
#include <stdio.h>
#include <stdlib.h>
#define TAM 10

int main()
{
    int i;
    int vector[TAM];
    for (i = 0; i < 12; i++)
    {
        //valor de i=10 o i=11, vector fuera de rango.
        printf("Valor[%d]: %d.\n", i, vector[i]);
    }
    return 0;
}
```

## Tipo de errores

1. Uso de variables sin valor asignado.
2. Índices de vectores fuera de rango.
3. Índices de vectores no enteros.
4. Valor de punteros sin dirección válida.
5. Diferente estructura interna usada en la lectura o grabación con las estructuras externas.
6. La estructura de un dato no es la misma para todos los módulos.
7. Variables no declaradas.

8. Variables con el mismo nombre.
9. Arreglos o cadena de caracteres no iniciados.
10. Valores iniciales no apropiados.
11. Longitudes y tipo de variables no apropiados para resolver el problema.
12. Cálculos con variables no numéricas.
13. Cálculos entre variables de diferente tipo.
14. Operaciones entre variables de diferentes longitudes.
15. Variables con longitud no apropiada al os valores en una expresión.
16. Resultados que produzcan “*overflow*”.
17. Divisiones por cero.
18. Valor de índice fuera de rango.
19. Comparaciones entre variables no consistentes por el concepto que representan.
20. Variables por comparar de diferente tipo de dato.
21. Expresiones lógicas incorrectas.
22. Mezcla de comparaciones y expresiones lógicas.
23. No termina alguna repetición.
24. No termina el programa.
25. No se ejecuta por lo menos una vez una repetición.
26. Mal los cierres de las estructuras de control.
27. Existen decisiones incompletas en su expresión lógica.
28. Número no adecuado de argumentos al llamar a una subrutina.
29. Argumentos de diferente tipo tanto en la llamada como en el encabezamiento del procedimiento.
30. No se respeta el orden de los parámetros entre la llamada y el encabezamiento.
31. Atributos incorrectos en los archivos.
32. Instrucciones incorrectas de apertura de archivos.
33. Especificación de formatos diferentes en las instrucciones de lectura y grabación.
34. Variables de diferente tamaño con respecto a los valores de los registros externos.
35. No se consideró condición de fin de archivo.
36. Errores en el texto de reportes o mensajes.
37. Son importantes los mensajes de aviso o advertencia en el caso de producirse.
38. Se omitió alguna función en la codificación.