



aprenderaprogramar.com

# Captura y gestión de errores. Módulos y ejemplos. Bloques try ... catch. (CU00245A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 44 del Curso Bases de la programación Nivel II

24

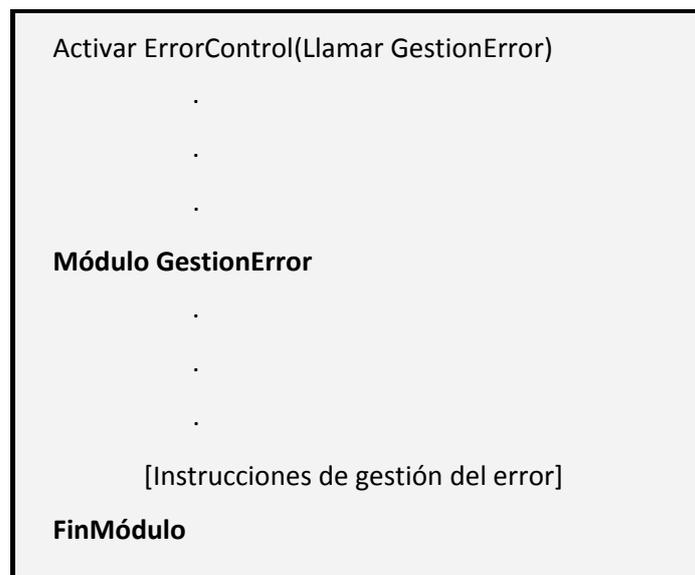
## CAPTURA Y GESTIÓN DE ERRORES

Ya hemos indicado que la gestión de errores consiste en establecer un procedimiento de tratamiento de los errores detectados durante la ejecución del programa. El proceso de detección y desvío del flujo del programa al procedimiento de gestión se llama captura del error.

El objetivo de la captura y gestión de errores es evitar los bloqueos o detenciones del programa que muchas veces obligan a reiniciar el ordenador, con la pérdida de tiempo y datos que esto puede suponer.

Muchos lenguajes permiten interceptar los errores de ejecución que se generan y darles el tratamiento adecuado para evitar bloqueos o detenciones con pérdidas de datos. La forma de hacerlo será variable. Por ejemplo, si usamos *Java* podemos encontrar que la forma de gestionar un error consiste en un bloque *try – catch* donde las instrucciones que pueden generar el error se engloban dentro del *try* y el tratamiento del error se encuentra en el bloque *catch*. Con *Visual Basic* podemos capturar cualquier error en el programa y tratarlo declarando simplemente a dónde debe saltar el flujo del programa en caso de error. Otros lenguajes pueden estar mucho más limitados, siendo la única forma de evitar la detención del programa desactivar temporalmente los mecanismos de detección de errores.

Uno de los modelos posibles se basaría en:



*Activar ErrorControl(Llamar GestionError)* indicaría al ordenador que ante un error durante la ejecución debe procederse a ejecutar el módulo *GestionError*. Dicho módulo puede concluir dando lugar a la finalización del programa a través de un *Finalizar* o bien devolviendo el control a la siguiente instrucción después de la que generó el error.

Veamos un ejemplo:

```
PROGRAMA TRATARERROR01 [Ejemplo aprenderaprogramar.com]  
  
Variables  
    Enteras: Numero  
  
1. Inicio  
    2. Activar ErrorControl(Llamar GestionError)  
    3. Mostrar "Por favor introduzca un número entero"  
    4. Pedir Numero  
    5. Mostrar "El doble es", 2 * Numero  
  
6. Fin  
  
Módulo GestionError  
    1. Mostrar "Se ha producido un error. No es posible mostrar resultados"  
    2. Finalizar  
  
FinMódulo
```

En este pseudocódigo el módulo *GestionError* da lugar a que se presente un mensaje de error y finalice el programa de forma controlada. Veamos lo que sería el caso de ejecución de un módulo *GestionError* con devolución del control al programa.

```
PROGRAMA TRATARERROR02  
  
Variables  
    Enteras: Numero  
  
1. Inicio  
    2. Activar ErrorControl(Llamar GestionError)  
    3. Mostrar "Por favor introduzca un número entero"  
    4. Pedir Numero  
    5. Mostrar "El doble es", 2 * Numero  
  
6. Fin  
  
Módulo GestionError  
    1. Mostrar "Se ha producido un error. No es posible mostrar resultados"  
  
FinMódulo
```

En esta situación el módulo *GestionError* presenta un mensaje y devuelve el control al programa, que sigue ejecutándose en la línea 5. Es de esperar que si introducimos un número tipo real en la línea 4 como 2,6318 se nos muestre el mensaje de error y a continuación “El doble es 0”. De todas formas lo que es “de esperar” depende del lenguaje y no es muy buena práctica ser confiado al respecto.

Si el error se produce dentro de un bucle donde existe una condición de salida, un módulo de gestión del error puede permitir continuar en el bucle hasta que se den las condiciones válidas para salir.

Si el error se produce dentro de un bucle con un número de iteraciones determinadas, un módulo de gestión del error puede permitir que todas las iteraciones menos una sean válidas. La trascendencia de este hecho habría que estudiarla para cada caso. Igualmente habría que estudiar la trascendencia de que existiera mayor número de iteraciones con error.

En un módulo de gestión de error normalmente dispondremos de cierta cantidad de información que se genera automáticamente relativa a un número de error, una descripción, etc. También podemos introducir desarrollos lógicos que den lugar al cambio de valor de las variables globales, interruptores, etc. Veremos estos conceptos más adelante.

**Próxima entrega: CU00246A**

**Acceso al curso completo** en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:  
[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=36&Itemid=60](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60)