ESTRUCTURA TRY…END TRY

Esta estructura de control del lenguaje, proporciona el medio para definir un bloque de código sensible a errores, y los correspondientes manipuladores de excepciones, en función del tipo de error producido.

```
Try
    ' código que puede provocar errores
    ' ....
    ' ....
    [Catch [Excepcion [As TipoExcepcionA]] [When Expresión]
    ' respuesta a error de tipo A
    ' ....
    ' ....
    [Exit Try]
    ]
    [Catch [Excepcion [As TipoExcepcionN]] [When Expresión]
    ' respuesta a error de tipo N
    ' ....
    ' ....
    [Exit Try]
    ]
    [Finally
    ' código posterior al control de errores
    ' ....
    ' ....
    ]
End Try
Código()
```

Analicemos con detalle los principales elementos de esta estructura.

En primer lugar nos encontramos con su declaración mediante la palabra clave Try. Todo el código que escribimos a partir de dicha palabra clave, y hasta la primera sentencia Catch, es el código que definimos como sensible a errores, o dicho de otro modo, el bloque de instrucciones sobre las que deseamos que se active el control de errores cuando se produzca algún fallo en su ejecución.

A continuación, establecemos cada uno de los manipuladores de excepción mediante la palabra clave Catch. Junto a esta palabra clave, situaremos de forma opcional, un identificador que contendrá el objeto con la excepción generada. Finalmente, y también de modo opcional, con la palabra clave When, especificaremos una condición para la captura del objeto de excepción. Podemos escribir uno o varios manipuladores Catch dentro de una estructura de control Try...End Try.

Cada vez que se produzca un error, el flujo de la ejecución saltará a la sentencia Catch más acorde con el tipo de excepción generada por el error, siempre y cuando hayamos situado varios manipuladores de excepciones en el controlador de errores.

Tal y como acaba de ver el lector en la sintaxis de la estructura Try...End Try, es posible utilizar Catch de un modo genérico, es decir, sin establecer qué tipo de excepción se ha producido. Este es el tipo de control de errores más sencillo que podemos implementar, aunque también el más limitado, ya que sólo podemos tener un manipulador de excepciones.