

# Excepciones

## 1. ¿Qué es una excepción?

Es una condición inesperada o inusual, que surge durante la ejecución del programa y no puede ser manejada en el contexto local.

## 2. ¿Qué hacemos si surge una excepción?

En principio, si surge una excepción y el programador no la abarcó, más allá de que haya o no un manejador de excepciones, el programa aborta.

Sin embargo, si el lenguaje no tiene manejo de excepciones, el programador podría simularlas preguntando en cada punto del programa donde pudiera producirse una excepción, si ésta fue lanzada y ejecutar una serie de sentencias para salvar el error y que el programa no finalice abruptamente.

En cambio, si el lenguaje usado tiene manejo de excepciones, el desarrollador podría programarlas usando esa estructura. En este caso, el problema se resuelve de la siguiente manera:

- El caso normal se expresa de manera simple y directa.
- El flujo de control se enruta a un manejador de excepciones sólo cuando es necesario.

## 3. ¿Cómo se maneja una excepción?

Originalmente, se disponía de ejecución condicionada(algo así como si pasa esto, hace esto otro, una suerte de if), por lo que si surgía una excepción se la atendía ahí mismo y se continuaba con el programa(PL/I). Sin embargo, los lenguajes más modernos ofrecen bloques léxicos. El bloque inicial de código es para el caso normal. El bloque contiguo representa el manejador de excepciones que reemplaza la ejecución del resto del bloque inicial en caso de error.

## 4. ¿Qué analizar acerca de un lenguaje que provee excepciones?

- ¿Cómo se maneja una excepción y cuál es su ámbito?: Ocurre una excepción, hay que analizar qué es lo que hace el lenguaje. Generalmente, busca el bloque de excepciones que corresponde a esa porción de código y deriva allí la ejecución. Es útil, además, tener presente cuál es el alcance de la excepción, el cual generalmente es igual que las variables que posee el lenguaje.
- ¿Cómo se alcanza una excepción?: Hay dos formas:
  - **Implícita**: si cuando se produce una excepción que el lenguaje tiene contemplada, como puede ser la división por cero, le lanza una excepción predefinida.
  - **Explícita**: si el programador hace la invocación de la excepción a través de la instrucción que provee el lenguaje: raise, throw, etc.
- ¿Cómo especificar las unidades(manejadores de excepciones) que se han de ejecutar cuando se alcanza las excepciones?: Dependiendo del lenguaje, el bloque de excepciones se debe colocar en el código en determinado lugar.
- ¿A dónde se cede el control cuando se termina de atender las excepciones?: nuevamente, dos opciones:
  - **Terminación**: si se termina la ejecución de la unidad que alcanza la excepción y se transfiere el control a la excepción, y luego se transfiere el control al manejador.
  - **Reasunción**: si se maneja la excepción y se devuelve el control al punto siguiente dónde se invocó a la excepción, permitiendo la continuación.
- ¿Cómo se propagan las excepciones?: Debemos tener presente cómo hace el programa para buscar un manejador en caso de que el bloque no lo contenga explícitamente. Generalmente esto se realiza dinámicamente, pero podrían haber combinaciones de dinámica con estática.
- ¿Hay excepciones predefinidas?: depende del lenguaje. En el caso de ADA, por ejemplo, existe el Constraint\_Error o el Storage\_Error, entre otras.

## 5. Fases para manejar una excepción

Son básicamente dos pasos:

- Detectar e informar del error:
  - Lanzamiento de Excepciones(throw).
  - Un método detecta una condición anormal que le impide continuar con su ejecución y finaliza "lanzando" un objeto Excepción.
- Recoger el error y tratarlo:
  - Captura de Excepciones(bloque try-catch).
  - Un método recibe un objeto Excepción que le indica que otro método no ha terminado correctamente su ejecución y decide actuar en función del tipo de error.