

TIPOS PRINCIPALES DE EXCEPCIONE

```
graph TD; A[TIPOS PRINCIPALES DE EXCEPCIONE] --> B[EXCEPCIONES VERIFICADAS (CHECKED EXCEPTIONS)]; A --> C[EXCEPCIONES NO VERIFICADAS (UNCHECKED EXCEPTIONS)];
```

EXCEPCIONES VERIFICADAS (CHECKED EXCEPTIONS):

Aquellas que el compilador **obliga a ser capturadas o declaradas en la firma del método.**

Asociadas con situaciones que son predecibles y controlables, como operaciones de entrada/salida (E/S), manejo de archivos y conexiones de red.

Se **derivan** de la **clase Exception** (directa o indirectamente) pero **no de RuntimeException.**

Ejemplos de excepciones verificadas son

- IOException
- FileNotFoundException
- SQLException, etc.

EXCEPCIONES NO VERIFICADAS (UNCHECKED EXCEPTIONS):

Excepciones que el compilador **no obliga a ser capturadas o declaradas en la firma del método.**

Asociadas a **situaciones que son difíciles o imposibles de predecir**, como errores de programación o problemas en tiempo de ejecución que normalmente están fuera del control del programador.

Se derivan de la clase RuntimeException.

Ejemplos de excepciones no verificadas

- ArithmeticException
- NullPointerException
- ArrayIndexOutOfBoundsException, etc.

EXCEPCIONES NO VERIFICADAS (UNCHECKED EXCEPTIONS)

MULTICATCH

Permiten capturar múltiples tipos de excepciones en un solo bloque catch. Antes de Java 7, cada tipo de excepción debía capturarse por separado en bloques catch individuales. Con multicatch, puedes agrupar varias excepciones del mismo bloque de catch y manejarlas de manera similar.

```
TRY {  
    // CÓDIGO QUE PUEDE LANZAR MÚLTIPLES EXCEPCIONES  
} CATCH (EXCEPCIONTIPO1 | EXCEPCIONTIPO2 | ..) {  
    // MANEJO DE LAS EXCEPCIONES CAPTURADAS  
}
```

TRY-WITH-RESOURCES

Simplifica el trabajar con recursos que necesitan ser cerrados explícitamente, como objetos que implementan la interfaz **AutoCloseable**. Se asegura de que los recursos sean cerrados adecuadamente, incluso en situaciones de excepciones, sin necesidad de utilizar bloques finally.

```
TRY (RECURSO RECURSO = NEW RECURSO()) {  
    // CÓDIGO QUE UTILIZA EL RECURSO  
} CATCH (EXCEPCIONTIPO EX) {  
    // MANEJO DE EXCEPCIONES, SI ES NECESARIO  
}  
// SE CERRARÁ AUTOMÁTICAMENTE DESPUÉS DE SALIR DEL BLOQUE TRY
```