

**Objetivo:** Gestionar el tratamiento de errores en tiempo de ejecución con excepciones **propias y predefinidas de Java**

### Excepciones predefinidas

1. Defina una clase llamada `DivisionNumero`. En el método `main` utilice el método `readLine()` para leer dos números en forma de cadena desde un `BufferedReader`. Utilice el método `parseInt()` para convertir las cadenas al tipo `int` y guardarlas en dos variables `int` (`numero1` y `numero2`). Divida `numero1` por `numero2` y muestre el resultado.

El Ingreso por teclado puede causar una excepción de tipo `IOException`

El método `Integer.parseInt()` convierte números de tipo `String` a tipo `int`. Si la cadena no puede convertirse a entero, arroja una `NumberFormatException`.

Al dividir un número por cero surge una `ArithmeticException`

Manipule las posibles excepciones utilizando un bloque de región segura (`try-catch`)

2. Quite el manejador de excepciones para `NumberFormatException` y `ArithmeticException` y observe el error producido durante la ejecución, al ingresar valores inadecuados (en laboratorio).

3. El bloque `try-catch` puede incluir una construcción `finally`. Este bloque de código siempre se ejecuta, independientemente de si se dio o no una excepción, y si se la atrapó o no. En la clase `DivisionNumero`, incluya un bloque `finally` y muestre un mensaje "Fin del bloque – finally".

### Excepciones definidas por el programador

4. Se desea manipular un tanque de usos múltiples (agua, nafta, etc). Se deben implementar los métodos para agregar/retirar líquido al tanque. Si excede la capacidad o el tanque está vacío, lanzar la excepción `TanqueLlenoException` o `TanqueVacioException`, según corresponda.

-

Tanque
-capacidad: int -carga: int
+tanque(int p_capacidad) +agregarCarga(int): void +retirarCarga(int): void

**Nota:** No olvidar definir las clases correspondientes, que extienden a `Exception`.

5. Cree la clase ejecutable `ControlarTanque`, que permita ingresar por teclado la capacidad del tanque, y la carga inicial. Crear una instancia (por ejemplo: `tanqueNafta`). Luego solicitar cantidad a agregar/retirar, y aplicar los métodos correspondientes a la instancia creada. Manejar las posibles excepciones en un bloque `try-catch`.