



## Unidad 4: Excepciones

### **Lanzar excepción**

Crear la clase Calculadora que tenga el método `double dividir(double dividendo, double divisor)`.

En caso de que el divisor sea cero, lanzar `java.lang.Exception`, con el mensaje "No se puede dividir por cero"

### **Excepciones propias**

Crear la clase `ParseadorDeFecha`, que reciba una fecha en el formato DD/MM/YYYY y que lance la excepción `FormatoInvalidoDeFecha` si la fecha no es válida. Para parsear la fecha, usar el método `split` de la clase `String`.

### **Lanzar runtime exceptions**

Crear la clase `Circulo` con el atributo "radio". La clase debe tener 2 constructores, uno vacío y otro que reciba como parámetro el radio. Además debe tener un método `void setRadio(double radio)` para cambiar el radio del círculo. Tanto en el constructor como en el método, en caso de recibir un radio menor o igual a cero, lanzar la excepción `IllegalArgumentException`.

Además crear un método `double calcularArea()`. En caso de que el radio sea 0, lanzar la excepción `IllegalStateException`.

### **Manejo de fechas**

Para este ejercicio, primero crear la clase `Fecha`, con tres atributos enteros: día, mes y año. A continuación, crear la clase `ManejadorDeFechas` que tenga 3 métodos:

- `String formatearFecha(Fecha unaFecha)`: que dada una fecha devuelva la fecha en formato `String día(2)/mes(2)/año(4)`. Ej: 22/12/2012
- `Fecha parsearFecha(String unaFecha) throws Exception`: dada una fecha en el formato anterior, que la convierta a `Fecha`
- `int cantidadDeAnios(Fecha unaFecha)`: calcular la cantidad de años transcurridos desde el día de hoy hasta la fecha del parámetro

### **Todavía sirve**

Crear las clases `Alimento`, que representará a un alimento en particular, y `VerificadorVencimiento`, que será la encargada de verificar si un alimento se encuentra vencido o no.

De cada alimento se conoce su nombre y cuántos días restan hasta su fecha de vencimiento. Implementar en la clase `VerificadorVencimiento` el método `public void todaviaSirve(Alimento unAlimento) throws AlimentoEnMalEstadoException`, que dado un alimento, lanzará la excepción `AlimentoEnMalEstadoException` en caso de detectar que el alimento se encuentra vencido. Un alimento se considera vencido si la cantidad de días para su vencimiento es menor o igual a cero.