

UTN – MDP
Laboratorio III

Guía 07 - Excepciones

- 1.** ¿Qué está mal en la siguiente bloque de código?

```
try
{
    Persona p = baseDeDatos.buscar(datos);
    System.out.println("Los datos pertenecen a: " + p);
}
catch(Exception e) {
    // Maneja cualquiera de las excepciones comprobadas
}
catch(RuntimeException e) {
    // Maneja cualquiera de las excepciones no comprobadas
}
```

- 2.** Explique lo que sucede el siguiente código y porque.

```
public static class EjemploExcepciones{
    public static int devuelveNumero(int num) {
        try {
            if (num % 2 == 0) {
                throw new Exception("Lanzando excepcion");
            }
            return 1;
        } catch (Exception ex) {
            ex.getMessage().concat("Rompi todo y pase por Catch");
            return 2;
        } finally {
            return 3;
        }
    }
}
```

3. Escribe un programa que juegue con el usuario a adivinar un número. El ordenador debe generar un número entre 1 y 500, y el usuario tiene que intentar adivinarlo.

Para ello, cada vez que el usuario introduce un valor, el ordenador debe decirle al usuario si el número que tiene que adivinar es mayor o menor que el que ha introducido el usuario. Cuando consiga adivinarlo, debe indicarlo e imprimir en pantalla el número de veces que el usuario ha intentado adivinar el número.

Si el usuario introduce algo que no es un número, debe indicarlo en pantalla, lanzar una excepción y contarlo como un intento. Investigar qué excepción controla el tipo de datos ingresados.

4. Crea una clase colegio que almacene el listado de los alumnos (compuestos por su nombre y nacionalidad). La clase tendrá los siguientes métodos:

agregarAlumno(Alumno alumno): añade la nacionalidad de un nuevo alumno

verNacionalidad(String nacionalidad): Muestra la nacionalidad y el número de alumnos de esa nacionalidad

cuantos(): Muestra cuántas nacionalidades diferentes existen en el colegio.

borrar(Alumno alumno): Elimina los datos insertados.

VerTodos(): Muestra las distintas nacionalidades y el número de alumnos que existen por nacionalidad.

Para todos estos métodos se deben generar distintas validaciones para prevenir errores en la ejecución. Por ej: Que todos los alumnos tengan una nacionalidad; Que exista el alumno que se quiere borrar, etc.