

1. Crear una clase **PracticaExcepciones** con un método estático **dividir(int dividendo, int divisor)**. Invocarlo pasando 10 y 2 como parámetros, y luego 10 y 0. Analizar lo sucedido. Implementar algún cambio para que el programa no finalice abruptamente su ejecución y muestre un mensaje más claro para el usuario.
2. Modificar la clase **ClienteBanco** de la Práctica 1, aplicando el lanzamiento de excepciones en lugar de los mensajes por consola que informan la imposibilidad de, por ejemplo, ingresar por PIN erróneo. Manejar toda la interactividad con el usuario a través de otra clase (**BancoIU**), donde deberán manejarse las excepciones lanzadas. El objetivo es que no haya lecturas ni escrituras en la clase del dominio (ClienteBanco).
3. Crear 2 excepciones personalizadas: **SaldoInsuficiente** y **PINErroneo**. Analizar cuál clase tomar como superclase y la posibilidad de agregar algún atributo necesario. Aplicarlas en el ejercicio anterior. Implementar los cambios necesarios en las cabeceras de los métodos que las lanzan.
4. Relanzar la excepción que genera el método del primer ejercicio como una excepción propia **DivisionPorCero**. Capturarla adecuadamente.
5. Aplicar en la clase **Jugador** del ejercicio del torneo de tenis, una excepción propia **EdadInvalida** que controle que la edad se encuentre en el rango de 16 a 80 años. Utilizarla para evitar que se carguen jugadores con edades no válidas. Generar un archivo de log con los jugadores en dicha situación.
6. Modificar el ejercicio anterior para permitir la carga de los jugadores con edades inválidas, asignándole -1 para esos casos. Informar, luego de haber procesado el archivo de texto, el porcentaje de jugadores con edades inválidas, haya terminado o no correctamente dicho proceso.
7. Desarrollar una clase **Calculadora** que realice las 4 operaciones aritméticas básicas. Implementar manejo de excepciones para gestionar división por cero o ingreso de caracteres no numéricos.