

## Ejercicios excepciones

1. (ValidarFecha) Escribir un método *esFechaValida(String fecha)* que determine si la fecha representada en el String que recibe como parámetro es o no válida. La fecha estará en formato "dd/mm/aaaa". Habrá que tener en cuenta que si el String no tuviera el formato adecuado se pueden producir las excepciones siguientes: `NumberFormatException` y `StringIndexOutOfBoundsException`. En caso de producirse dichas excepciones, se considerará que la fecha no es válida y por tanto el método devolverá `false`.
2. (Nif) Escribir un método que reciba un String con un NIF y devuelva `true` si el nif es correcto y `false` si es incorrecto. Usar el método `main` para probar el método desarrollado. Consideraremos que el en la última posición del String tiene que estar la letra del NIF, y en las posiciones anteriores, el número. Tener en cuenta que si el NIF que se comprueba es, por ejemplo, "3434BBB22S" se producirá una excepción `NumberFormatException`
3. (Edades) Escribe un programa que solicite al usuario la edad de cinco personas y calcule la media.. Realiza tres versiones introduciendo el manejo de la excepción `InputMismatchException` que se produce si el usuario no teclea un entero cuando se le solicita:
  - Si se introduce mal la edad de una persona se vuelve a pedir la edad de esa persona.
  - Si se introduce mal la edad de una persona, el programa muestra un mensaje de error, no calcula la media y termina.
  - Si se introduce mal la edad de una persona, el programa vuelve a solicitar la edad de las cinco personas (comienza el proceso).

4. Indica qué se mostrará por pantalla cuando se ejecute esta clase y por qué:

```
public class Ejercicio {
    private static int metodo() {
        int valor=0;
        try {
            valor = valor + 1;
            valor = valor + Integer.parseInt("42") ;
            valor = valor + 1;
            System.out.println("Valor al final del try: " + valor);
        } catch(NumberFormatException e) {
            valor = valor + Integer.parseInt ("42");
            System.out.println("Valor al final del catch: " + valor) ;
        }
        finally {
            valor = valor + 1;
            System.out.println("Valor al final de finally: " + valor) ;
        }
        valor = valor + 1;
        System.out.println ("Valor antes del return: " + valor) ;
        return valor;
    }

    public static void main(String[] args) {
        try {
            System.out.println (metodo());
        } catch (Exception e) {
            System.err.println("Excepcion en metodo()") ;
        }
    }
}
```

5. Indica qué se mostrará por pantalla cuando se ejecute esta clase y por qué:

```
public class Ejercicio {
    private static int metodo() {
        int valor=0;
        try {
            valor = valor+1;
            valor = valor + Integer.parseInt("W");
            valor = valor + 1;
            System.out.println("Valor al final del try: " + valor);
        } catch(NumberFormatException e) {
            valor = valor + Integer.parseInt("42");
            System.out.println("Valor al final del catch: " + valor) ;
        } finally {
            valor = valor + 1;
            System.out.println("Valor al final de finally: " + valor) ;
        }
        valor = valor + 1;
        System.out.println ("Valor antes del return: " + valor) ;
        return valor ;
    }

    public static void main (String[] args) {
        try {
            System.out.println(metodo());
        } catch (Exception e) {
            System.err.println("Excepcion en metodo() ");
        }
    }
}
```

6. Indica qué se mostrará por pantalla cuando se ejecute esta clase y por qué:

```
public class Ejercicio {
    private static int metodo() {
        int valor = 0;
        try {
            valor = valor + 1;
            valor = valor + Integer.parseInt("W");
            valor = valor + 1;
            System.out.println("Valor al final del try : " + valor);
        } catch (NumberFormatException e) {
            valor = valor + Integer.parseInt("W");
            System.out.println("Valor al final del catch : " + valor);
        } finally {
            valor = valor + 1;
            System.out.println("Valor al final de finally: " + valor);
        }
        valor = valor + 1;
        System.out.println ("Valor antes del return: " + valor);
        return valor ;
    }

    public static void main (String[ ] args)
    {
        try {
            System.out.println(metodo ());
        } catch (Exception e) {
            System.err.println("Excepcion en metodo()");
        }
    }
}
```

7. Indica qué se mostrará por pantalla cuando se ejecute esta clase y por qué:

```
import java.io.*;

public class Ejercicio {
    private static int metodo() {
        int valor = 0;
        try {
            valor = valor+1;
            valor = valor + Integer.parseInt("W");
            valor = valor + 1;
            System.out.println("Valor al final del try : " + valor) ;
            throw new IOException();
        } catch (IOException e) {
            valor = valor + Integer.parseInt("42");
            System.out.println("Valor al final del catch : " + valor);
        } finally {
            valor = valor + 1;
            System.out.println("Valor al final de finally: " + valor);
        }
        valor = valor + 1;
        System.out.println ("Valor antes del return: " + valor) ;
        return valor ;
    }

    public static void main(String[] args) {
        try {
            System.out.println(metodo());
        } catch (Exception e) {
            System.err.println("Excepcion en metodo()");
        }
    }
}
```

8. Indica qué se mostrará por pantalla cuando se ejecute esta clase

- Si el programa comienza con String texto = “casa”;
- Si el programa comienza con String texto = “0”;
- Si el programa comienza con String texto = “7”;

```
public class Ejercicio {  
    public static void main(String args[]) {  
        String texto = .....  
        try {  
            int a = Integer.parseInt(texto);  
            System.out.println("a = " + a);  
            int b=42/a;  
            String c = "hola";  
            char d = c.charAt(50);  
        } catch (ArithmeticException e) {  
            System.out.println("div por 0: ");  
        } catch (IndexOutOfBoundsException e) {  
            System.out.println("Índice del String fuera de límites: ");  
        } finally {  
            System.out.println("Ejecución de finally");  
        }  
    }  
}
```

9. Indica cuál será la salida del siguiente programa y por qué

```
public class Nueve {
    public static void procA() {
        try {
            System.out.println("dentro del procA");
            throw new RuntimeException("demo");
        } finally {
            System.out.println("Finally del procA");
        }
    }

    public static void procB() {
        try {
            System.out.println("dentro del procB");
            return;
        } finally {
            System.out.println("finally del procB");
        }
    }

    public static void main(String args[]) {
        try {
            procA();
        } catch (Exception e) {
            procB();
        }
    }
}
```

10. Indica cuál será la salida del siguiente programa y por qué

```
public class Diez {  
    public static void metodo() {  
        try {  
            throw new NullPointerException("demo");  
        } catch (NullPointerException e) {  
            System.out.println("capturada en método");  
            throw e;  
        }  
    }  
  
    public static void main (String args[]) {  
        try {  
            metodo();  
        } catch (NullPointerException e) {  
            System.out.println("capturada en main ");  
        }  
    }  
}
```



