



INSTITUTO TECNOLÓGICO DE MORELIA

INGENIERIA EN SISTEMAS COMPUTACIONALES

“EXCEPCIONES EN JAVA”



PRESENTA:

DE LABASTIDA GUZMÁN ÁNGEL RAÚL,

NOVIEMBRE

En Java, existen varias excepciones básicas que representan errores comunes que pueden ocurrir durante la ejecución de un programa. Aquí te explico algunas de las excepciones más comunes junto con ejemplos de cómo pueden ocurrir y cómo manejar cada una de ellas.

1. NullPointerException

Ocurre cuando se intenta acceder a un objeto que es null. Es una de las excepciones más comunes y suele ser el resultado de intentar usar una referencia no inicializada.

```
public class EjemploNullPointerException {  
    public static void main(String[] args) {  
        String texto = null;  
        System.out.println(texto.length()); // Lanza NullPointerException  
    }  
}
```

Solución: Verifica que la referencia no sea null antes de usarla:

```
public class EjemploNullPointerException {  
    public static void main(String[] args) {  
        String texto = null;  
        if (texto != null) {  
            System.out.println(texto.length());  
        } else {  
            System.out.println("La cadena es null.");  
        }  
    }  
}
```

2. **ArrayIndexOutOfBoundsException**

Ocurre cuando se intenta acceder a un índice que está fuera de los límites de un array.

```
public class EjemploArrayIndexOutOfBoundsException {  
    public static void main(String[] args) {  
        int[] numeros = {1, 2, 3};  
        System.out.println(numeros[3]); // Lanza ArrayIndexOutOfBoundsException  
    }  
}
```

Solución: Asegúrate de que el índice está dentro del rango del array:

```
public class EjemploArrayIndexOutOfBoundsException {  
    public static void main(String[] args) {  
        int[] numeros = {1, 2, 3};  
        int indice = 3;  
        if (indice >= 0 && indice < numeros.length) {  
            System.out.println(numeros[indice]);  
        } else {  
            System.out.println("Índice fuera de los límites.");  
        }  
    }  
}
```

3. **ArithmeticException**

Se lanza cuando ocurre un error en una operación aritmética, como la división por cero.

```
public class EjemploArithmeticException {  
    public static void main(String[] args) {  
        int numero = 10;
```

```

        int resultado = numero / 0; // Lanza ArithmeticException
        System.out.println(resultado);
    }
}

```

Solución: Verifica que el divisor no sea cero antes de realizar la división:

```

public class EjemploArithmeticException {
    public static void main(String[] args) {
        int numero = 10;
        int divisor = 0;
        if (divisor != 0) {
            int resultado = numero / divisor;
            System.out.println(resultado);
        } else {
            System.out.println("No se puede dividir por cero.");
        }
    }
}

```

4. NumberFormatException

Ocurre cuando se intenta convertir una cadena en un número y el formato de la cadena no es válido.

```

public class EjemploNumberFormatException {
    public static void main(String[] args) {
        String texto = "abc";
        int numero = Integer.parseInt(texto); // Lanza NumberFormatException
        System.out.println(numero);
    }
}

```

Solución: Envuelve la conversión en un bloque try-catch y verifica que la cadena tenga un formato correcto antes de convertirla:

```
public class EjemploNumberFormatException {  
    public static void main(String[] args) {  
        String texto = "abc";  
        try {  
            int numero = Integer.parseInt(texto);  
            System.out.println(numero);  
        } catch (NumberFormatException e) {  
            System.out.println("El texto no es un número válido.");  
        }  
    }  
}
```

5. ClassCastException

Sucede cuando se intenta convertir un objeto a una clase incompatible.

```
public class EjemploClassCastException {  
    public static void main(String[] args) {  
        Object obj = "Hola";  
        Integer numero = (Integer) obj; // Lanza ClassCastException  
    }  
}
```

Solución: Verifica el tipo de la clase antes de realizar el casting:

```
public class EjemploClassCastException {  
    public static void main(String[] args) {  
        Object obj = "Hola";  
        if (obj instanceof Integer) {
```

```

        Integer numero = (Integer) obj;
        System.out.println(numero);
    } else {
        System.out.println("El objeto no es un Integer.");
    }
}
}

```

6. **IllegalArgumentException**

Se lanza cuando un método recibe un argumento inapropiado.

```

public class EjemploIllegalArgumentException {
    public static void main(String[] args) {
        setEdad(-5); // Lanza IllegalArgumentException
    }

    public static void setEdad(int edad) {
        if (edad < 0) {
            throw new IllegalArgumentException("La edad no puede ser negativa.");
        }
        System.out.println("Edad: " + edad);
    }
}

```

Solución: Valida los argumentos antes de usarlos.

```

public class EjemploIllegalArgumentException {
    public static void main(String[] args) {
        setEdad(25); // Correcto
    }
}

```

```
public static void setEdad(int edad) {  
    if (edad < 0) {  
        throw new IllegalArgumentException("La edad no puede ser negativa.");  
    }  
    System.out.println("Edad: " + edad);  
}  
}
```