

Pasos para Resolver un Ejercicio

1. **Analiza el problema:**
 - **Clases:** Identifica los sustantivos del enunciado.
 - **Métodos:** Busca las acciones clave (verbos).
 - **Relaciones:**
 - "Es un" → Herencia.
 - "Tiene un" → Composición.
2. **Planifica:**
 - Define los atributos necesarios.
 - Diseña los métodos: ¿Qué acciones debe realizar la clase?
3. **Divide y Conquista:**
 - Empieza con las partes más simples.
 - Valida y prueba cada funcionalidad antes de continuar.
4. **Valida y Maneja Errores:**
 - Identifica entradas clave que deben validarse.
 - Usa excepciones para manejar errores de forma controlada.

Relación entre Clases

1. Herencia ("Es un"):

```
java
Copiar código
class Vehiculo {
    private String marca;
}
class Coche extends Vehiculo {
    private int puertas;
}
```

2. Composición ("Tiene un"):

```
java
Copiar código
class Biblioteca {
    private Libro libro;
}
```

Inicializar Objetos

1. Definir una clase:

```
java
Copiar código
class Vehiculo {
    private String marca;
    public Vehiculo(String marca) {
        this.marca = marca;
    }
    public String getMarca() {
        return marca;
    }
}
```

```
    }  
}
```

2. Crear un objeto:

```
java  
Copiar código  
Vehiculo miCoche = new Vehiculo("Toyota");  
System.out.println("Marca: " + miCoche.getMarca());
```

Arrays y Matrices

Arrays

1. Crear sin inicializar:

```
java  
Copiar código  
int[] numeros = new int[5]; // Espacio para 5 elementos  
String[] nombres = new String[3]; // Espacio para 3 cadenas
```

2. Inicializar después de crear:

```
java  
Copiar código  
numeros[0] = 10;  
numeros[1] = 20;
```

3. Crear e inicializar al mismo tiempo:

```
java  
Copiar código  
int[] numeros = {1, 2, 3, 4};
```

4. Recorrer con For:

```
java  
Copiar código  
for (int i = 0; i < numeros.length; i++) {  
    System.out.println(numeros[i]);  
}
```

5. For-each:

```
java  
Copiar código  
for (int numero : numeros) {  
    System.out.println(numero);  
}
```

Matrices (Arrays Bidimensionales)

1. Crear sin inicializar:

```
java
Copiar código
int[][] matriz = new int[3][3]; // Matriz de 3x3
```

2. Inicializar después de crear:

```
java
Copiar código
matriz[0][0] = 1;
matriz[0][1] = 2;
```

3. Crear e inicializar al mismo tiempo:

```
java
Copiar código
int[][] matriz = {
    {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9}
};
```

4. Recorrer:

```
java
Copiar código
for (int i = 0; i < matriz.length; i++) {
    for (int j = 0; j < matriz[i].length; j++) {
        System.out.print(matriz[i][j] + " ");
    }
    System.out.println();
}
```

Operador Ternario

Condiciones compactas:

Ejemplo:

```
java
Copiar código
int edad = 20;
String mensaje = (edad >= 18) ? "Mayor de edad" : "Menor de edad";
System.out.println(mensaje);
```

Casting de Datos

1. Casting Implícito:

```
java
Copiar código
int entero = 10;
double decimal = entero; // Convierte automáticamente
```

2. Casting Explícito:

```
java
Copiar código
double decimal = 9.99;
int entero = (int) decimal; // Redondea hacia abajo
```

3. De String a Número:

```
java
Copiar código
String texto = "123";
int numero = Integer.parseInt(texto);
```

4. De Número a String:

```
java
Copiar código
int numero = 123;
String texto = String.valueOf(numero);
```

Excepciones

Uso de `throw` y `throws`

1. Lanzar una Excepción desde un Método:

```
java
Copiar código
public static void validarEdad(int edad) throws Exception {
    if (edad < 0) {
        throw new Exception("La edad no puede ser negativa.");
    }
}
```

2. Capturar Excepciones en el `main`:

```
java
Copiar código
public static void main(String[] args) {
    try {
        validarEdad(-5);
    } catch (Exception e) {
        System.out.println("Error: " + e.getMessage());
    }
}
```

Clase Validar

La clase `Validar` manejará tanto la validación como el control de los intentos. Para ello, usará un bucle dentro de los métodos de validación.

```
java
Copiar código
import java.util.Scanner;
```

```

public class Validar {

    private static final int MAX_INTENTOS = 2; // Número máximo de
    intentos

    // Validar un número con dos intentos
    public static int validarNumeroConIntentos(Scanner sc, int min,
    int max) {
        int numero = 0;
        for (int intento = 1; intento <= MAX_INTENTOS; intento++) {
            System.out.print("Introduce un número entre " + min + " y
            " + max + ": ");
            if (sc.hasNextInt()) {
                numero = sc.nextInt();
                sc.nextLine(); // Limpiar el buffer
                if (numero >= min && numero <= max) {
                    return numero; // Número válido
                } else {
                    System.out.println("El número debe estar entre " +
                    min + " y " + max + ".");
                }
            } else {
                System.out.println("Entrada no válida. Debes
                introducir un número.");
                sc.nextLine(); // Limpiar el buffer
            }
        }
        throw new IllegalArgumentException("Número no válido después
        de " + MAX_INTENTOS + " intentos.");
    }

    // Validar un texto con dos intentos
    public static String validarTextoConIntentos(Scanner sc) {
        String texto;
        for (int intento = 1; intento <= MAX_INTENTOS; intento++) {
            System.out.print("Introduce un texto no vacío: ");
            texto = sc.nextLine();
            if (texto != null && !texto.trim().isEmpty()) {
                return texto; // Texto válido
            } else {
                System.out.println("El texto no puede estar vacío.");
            }
        }
        throw new IllegalArgumentException("Texto no válido después de
        " + MAX_INTENTOS + " intentos.");
    }
}

```

Clase Principal

La clase principal solo delegará en la clase `Validar` para realizar las comprobaciones. Esto la hace más sencilla.

```

java
Copiar código
import java.util.Scanner;

public class Principal {

```

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    try {
        // Validar un número
        int numero = Validar.validarNumeroConIntentos(sc, 1, 100);
        System.out.println("Número válido: " + numero);

        // Validar un texto
        String texto = Validar.validarTextoConIntentos(sc);
        System.out.println("Texto válido: " + texto);
    } catch (IllegalArgumentException e) {
        System.out.println("Error: " + e.getMessage());
    }

    sc.close();
}
```