

14-8-2025

PROGRAMACION ESTRUCTURADA

Trabajo práctico II

Alumnos:

- Ezequiel Alejandro Ventura

Materia: Programación II

Profesor: Ariel Enferrel

Tutor: Tomás Ferro

Fecha de Entrega: 14/08/2025

Github:

<https://github.com/equimdq/Programacion2>

Estructuras Condicionales:

1. Verificación de Año Bisiesto.

Escribe un programa en Java que solicite al usuario un año y determine si es bisiesto. Un año es bisiesto si es divisible por 4, pero no por 100, salvo que sea divisible por 400.

```
package ejercicio01;

import java.util.Scanner;

public class Ejercicio01 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Ingrese un año para corroborar si es bisiesto: ");
        int anio = input.nextInt();

        if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
            System.out.println(anio + " es un año bisiesto.");
        } else {
            System.out.println(anio + " no es un año bisiesto.");
        }
    }
}
```

2. Determinar el Mayor de Tres Números.

Escribe un programa en Java que pida al usuario tres números enteros y determine cuál es el mayor.

```
1 package ejercicio01;
2
3
4 import java.util.Scanner;
5
6
7 public class Ejercicio01 {
8
9
10     public static void main(String[] args) {
11
12         Scanner input = new Scanner(System.in);
13
14         System.out.println("Ingrese un año para corroborar si es bisiesto: ");
15         int anio = input.nextInt();
16
17         if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
18             System.out.println(anio + " es un año bisiesto.");
19         } else {
20             System.out.println(anio + " no es un año bisiesto.");
21         }
22     }
23
24 }
25
```

Output X

Programacion2 - C:\Users\Eze\Documents\UTN - Programacion\Programacion2 X ejercicio02 (run) X

run:
Ingrese el primer número entero:
10
Ingrese el segundo número entero:
25
Ingrese el tercer número entero:
30
El mayor es: 30
BUILD SUCCESSFUL (total time: 23 seconds)

3. Clasificación de Edad.

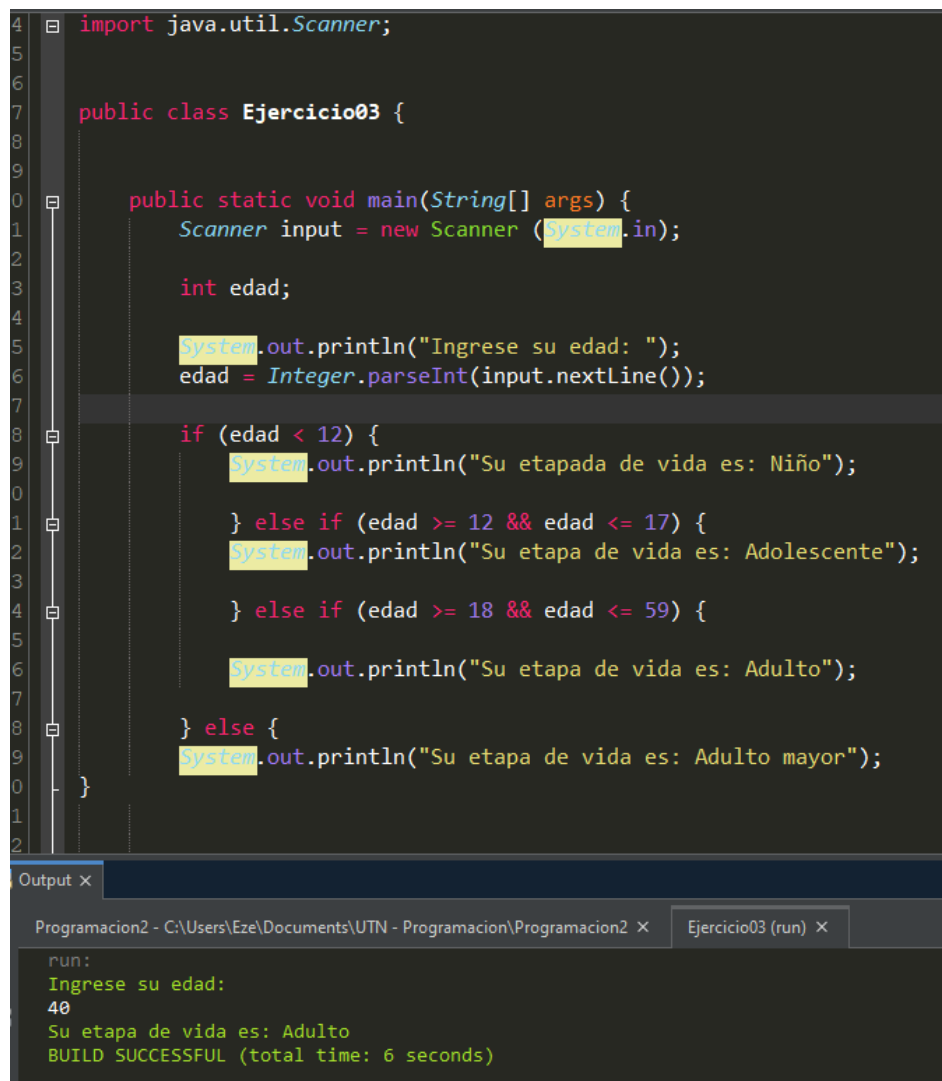
Escribe un programa en Java que solicite al usuario su edad y clasifique su etapa de vida según la siguiente tabla:

Menor de 12 años: *"Niño"*

Entre 12 y 17 años: *"Adolescente"*

Entre 18 y 59 años: *"Adulto"*

60 años o más: *"Adulto mayor"*



```
4 import java.util.Scanner;
5
6
7 public class Ejercicio03 {
8
9     public static void main(String[] args) {
10         Scanner input = new Scanner (System.in);
11
12         int edad;
13
14         System.out.println("Ingrese su edad: ");
15         edad = Integer.parseInt(input.nextLine());
16
17         if (edad < 12) {
18             System.out.println("Su etapada de vida es: Niño");
19         } else if (edad >= 12 && edad <= 17) {
20             System.out.println("Su etapa de vida es: Adolescente");
21         } else if (edad >= 18 && edad <= 59) {
22             System.out.println("Su etapa de vida es: Adulto");
23         } else {
24             System.out.println("Su etapa de vida es: Adulto mayor");
25         }
26     }
27 }
```

Output x

Programacion2 - C:\Users\Eze\Documents\UTN - Programacion\Programacion2 x Ejercicio03 (run) x

```
run:
Ingrese su edad:
40
Su etapa de vida es: Adulto
BUILD SUCCESSFUL (total time: 6 seconds)
```

4. Calculadora de Descuento según categoría.

Escribe un programa que solicite al usuario el precio de un producto y su categoría (A, B o C). Luego, aplique los siguientes descuentos: Categoría A: 10% de descuento Categoría B: 15% de descuento Categoría C: 20% de descuento El programa debe mostrar el precio original, el descuento aplicado y el precio final.

```
package ejercicio04;

import java.util.Scanner;

public class Ejercicio04 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        String categoria;
        Double precioFinal, precio;

        System.out.println("Ingrese el precio del producto: ");
        precio = Double.parseDouble(input.nextLine());

        System.out.println("Ingrese la categoria (A, B, C): ");
        categoria = input.nextLine();

        double descuento = 0;

        switch (categoria) {
            case "A" -> {descuento = 10;
            }
            case "B" -> {descuento = 15;
            }
            case "C" -> {descuento = 20;
            }
            default -> System.out.println("Categoría inválida");
        } precioFinal = precio - (precio * descuento / 100);

        System.out.println("Precio original: " + precio);
        System.out.println("Categoría del producto: " + categoria);
        System.out.println("Descuento aplicado: " + descuento + "%");
        System.out.println("Precio final: " + precioFinal);
    }
}
```

Estructuras de Repetición:

5. Suma de Números Pares (while).

Escribe un programa que solicite números al usuario y sume solo los números pares. El ciclo debe continuar hasta que el usuario ingrese el número 0, momento en el que se debe mostrar la suma total de los pares ingresados.

```
1 package ejercicio05;
2
3
4 import java.util.Scanner;
5
6 public class Ejercicio05 {
7
8     public static void main(String[] args) {
9
10         Scanner input = new Scanner(System.in);
11
12         int suma = 0;
13         int num;
14
15         System.out.println("Ingrese un número (0 para terminar): ");
16         num = Integer.parseInt(input.nextLine());
17
18         while (num != 0) {
19             if (num % 2 == 0) {
20                 suma += num;
21             } System.out.println("Ingrese un número (0 para terminar)");
22             num = Integer.parseInt(input.nextLine());
23         }
24         System.out.println("La suma de los números pares es: " + suma);
25     }
26 }
```

Output x

Programacion2 - C:\Users\Eze\Documents\UTN - Programacion\Programacion2 x Ejercicio05 (run) x Ejercicio05 (run) #2 x

run:
Ingrese un número (0 para terminar):
10
Ingrese un número (0 para terminar):
20
Ingrese un número (0 para terminar):
30
Ingrese un número (0 para terminar):
25
Ingrese un número (0 para terminar):
5
Ingrese un número (0 para terminar):
0
La suma de los números pares es: 60

6. Contador de Positivos, Negativos y Ceros (for).

Escribe un programa que pida al usuario ingresar 10 números enteros y cuente cuántos son positivos, negativos y cuántos son ceros.

```
package ejercicio06;

import java.util.Scanner;

public class Ejercicio06 {

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

        int num;
        int positivos = 0;
        int negativos = 0;
        int ceros = 0;

        System.out.println("Ingrese 10 números enteros");

        for (int i = 1; i <= 10; i++) {
            System.out.println("Ingrese el siguiente número: ");
            num = Integer.parseInt(input.nextLine());

            if (num > 0) {
                positivos++;
            } else if (num < 0) {
                negativos++;
            } else {
                ceros++;
            }
        }

        // mostrar resultados fuera del for
        System.out.println("Cantidad de números positivos: " + positivos);
        System.out.println("Cantidad de números negativos: " + negativos);
        System.out.println("Cantidad de ceros: " + ceros);
    }
}
```

7. Validación de Nota entre 0 y 10 (do-while).

Escribe un programa que solicite al usuario una nota entre 0 y 10. Si el usuario ingresa un número fuera de este rango, debe seguir pidiéndole la nota hasta que ingrese un valor válido.

```
1 package ejercicio07;
2
3 import java.util.Scanner;
4
5 public class Ejercicio07 {
6
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9         int nota;
10
11         do {
12             System.out.println("Ingrese una nota (0-10): ");
13             nota = Integer.parseInt(input.nextLine());
14
15             if (nota < 0 || nota > 10) {
16                 System.out.println("Error: Nota inválida. Ingrese una nota entre 0 y 10.");
17             }
18
19         } while (nota < 0 || nota > 10);
20
21         System.out.println("Nota guardada correctamente.");
22     }
23 }
24
25
```

Output x

Programacion2 - C:\Users\Eze\Documents\UTN - Programacion\Programacion2 x Ejercicio05 (run) x Ejercicio06 (run) x Ejercicio06 (run) #2 x

```
run:
Ingrese una nota (0-10):
12
Error: Nota inválida. Ingrese una nota entre 0 y 10.
Ingrese una nota (0-10):
-2
Error: Nota inválida. Ingrese una nota entre 0 y 10.
Ingrese una nota (0-10):
5
Nota guardada correctamente.
BUILD SUCCESSFUL (total time: 15 seconds)
```


Funciones:

8. Cálculo del Precio Final con impuesto y descuento.

Crea un método `calcularPrecioFinal(double impuesto, double descuento)` que calcule el precio final de un producto en un e-commerce.

La fórmula es: $PrecioFinal = PrecioBase + (PrecioBase \times Impuesto) - (PrecioBase \times Descuento)$
 $PrecioFinal = PrecioBase + (PrecioBase \times Impuesto) - (PrecioBase \times Descuento)$

Desde `main()`, solicita el precio base del producto, el porcentaje de impuesto y el porcentaje de descuento, llama al método y muestra el precio final.

```

1 package ejercicio08;
2
3 import java.util.Scanner;
4
5 public class Ejercicio08 {
6
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9         double precioBase, desc, precioFinal;
10        int imp;
11
12        System.out.println("Ingrese el precio base del producto: ");
13        precioBase = Double.parseDouble(input.nextLine());
14
15        System.out.println("Ingrese el impuesto en porcentaje: ");
16        imp = Integer.parseInt(input.nextLine());
17
18        System.out.println("Ingrese el descuento en porcentaje: ");
19        desc = Double.parseDouble(input.nextLine());
20
21        // Llamada al método
22        precioFinal = calcularPrecioFinal(precioBase, imp, desc);
23
24        System.out.println("El precio final del producto es: " + precioFinal);
25    }
26    public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento) {
27        double impuestoAplicado = precioBase * (impuesto / 100);
28        double descuentoAplicado = precioBase * (descuento / 100);
29        return precioBase + impuestoAplicado - descuentoAplicado;
30    }
31 }

```

9. Composición de funciones para calcular costo de envío y total de compra.

a. `calcularCostoEnvio(double peso, String zona)`: Calcula el costo de envío basado en la zona de envío (Nacional o Internacional) y el peso del paquete.

Nacional: \$5 por kg Internacional: \$10 por kg

b. `calcularTotalCompra(double precioProducto, double costoEnvio)`: Usa `calcularCostoEnvio` para sumar el costo del producto con el costo de envío. Desde `main()`, solicita el peso del paquete, la zona de envío y el precio del producto. Luego, muestra el total a pagar.

```
package ejercicio09;

import java.util.Scanner;

public class Ejercicio09 {

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

        // Programa principal
        System.out.print("Ingrese el precio del producto: ");
        double precioProducto = Double.parseDouble(input.nextLine());

        System.out.print("Ingrese el peso del paquete (kg): ");
        double peso = Double.parseDouble(input.nextLine());

        System.out.print("Ingrese la zona de envío (Nacional/Internacional): ");
        String zona = input.nextLine();

        double costoEnvio = calcularCostoEnvio(peso, zona);
        double total = calcularTotalCompra(precioProducto, costoEnvio);

        System.out.println("Costo de envío: $" + costoEnvio);
        System.out.println("Total a pagar: $" + total);
    }

    public static double calcularCostoEnvio(double peso, String zona) {
        if (zona.equalsIgnoreCase("Nacional")) {
            return peso * 5;
        } else if (zona.equalsIgnoreCase("Internacional")) {
            return peso * 10;
        } else {
            System.out.println("Zona inválida, se aplicará costo 0.");
            return 0;
        }
    }

    public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
        return precioProducto + costoEnvio;
    }
}
```

10. Actualización de stock a partir de venta y recepción de productos.

Crea un método `actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida)`, que calcule el nuevo stock después de una venta y recepción de productos: $\text{NuevoStock} = \text{StockActual} - \text{CantidadVendida} + \text{CantidadRecibida}$

Desde `main()`, solicita al usuario el stock actual, la cantidad vendida y la cantidad recibida, y muestra el stock actualizado.

```

1  package ejercicio10;
2
3  import java.util.Scanner;
4
5  public class Ejercicio10 {
6
7      //Programa Principal
8
9      public static void main(String[] args) {
10         Scanner input = new Scanner(System.in);
11
12         System.out.print("Ingrese el stock actual del producto: ");
13         int stockActual = Integer.parseInt(input.nextLine());
14
15         System.out.print("Ingrese la cantidad vendida: ");
16         int cantidadVendida = Integer.parseInt(input.nextLine());
17
18         System.out.print("Ingrese la cantidad recibida: ");
19         int cantidadRecibida = Integer.parseInt(input.nextLine());
20
21         int nuevoStock = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
22
23         if (nuevoStock != -1) {
24             System.out.println("El nuevo stock del producto es: " + nuevoStock);
25         }
26     }
27     // Función
28     public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
29
30         if (stockActual < 0 || cantidadVendida < 0 || cantidadRecibida < 0) {
31             System.out.println("Error: todos los valores deben ser números enteros positivos.");
32             return -1; //
33         }
34         int nuevoStock = stockActual - cantidadVendida + cantidadRecibida;
35         return nuevoStock;
36     }
37 }

```

11. Cálculo de descuento especial usando variable global.

Declara una variable global Ejemplo de entrada/salida: = 0.10. Luego, crea un método `calcularDescuentoEspecial(double precio)` que use la variable global para calcular el descuento especial del 10%.

Dentro del método, declara una variable local `descuentoAplicado`, almacena el valor del descuento y muestra el precio final con descuento

```
public class Ejercicio11 {  
  
    static double DESCUENTO_ESPECIAL = 0.10;  
  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        // Programa principal  
        System.out.print("Ingrese el precio del producto: ");  
        double precio = Double.parseDouble(input.nextLine());  
  
        calcularDescuentoEspecial(precio);  
    }  
  
    // Función  
    public static void calcularDescuentoEspecial(double precio) {  
        double descuentoAplicado = precio * DESCUENTO_ESPECIAL;  
        double precioFinal = precio - descuentoAplicado;  
  
        System.out.println("El descuento especial aplicado es: " + descuentoAplicado);  
        System.out.println("El precio final con descuento es: " + precioFinal);  
    }  
}
```

Input ×

programacion2 - C:\Users\Eze\Documents\UTN - Programacion\Programacion2 × Ejercicio11 (run) ×

run:
Ingrese el precio del producto: 200
El descuento especial aplicado es: 20.0
El precio final con descuento es: 180.0
BUILD SUCCESSFUL (total time: 4 seconds)

Arrays y Recursividad:

12. Modificación de un array de precios y visualización de resultados.

Crea un programa que:

- Declare e inicialice un array con los precios de algunos productos.
- Muestre los valores originales de los precios.
- Modifique el precio de un producto específico.
- Muestre los valores modificados.

```
1 package ejercicio12;
2
3 public class Ejercicio12 {
4
5     public static void main(String[] args) {
6
7         double[] precios = {199.99, 299.50, 149.75, 399.00, 89.99};
8
9         System.out.println("Precios originales:");
10        for (double precio : precios) {
11            System.out.println("Precio: $" + precio);
12        }
13
14        precios[2] = 129.99; //modificacion del precio
15
16        System.out.println("Precios modificados:");
17        for (double precio : precios) {
18            System.out.println("Precio: $" + precio);
19        }
20    }
21 }
22
```

Output X

Programacion2 - C:\Users\Eze\Documents\UTN - Programacion\Programacion2 x Ejercicio12 (run) x

```
Precios originales:
Precio: $199.99
Precio: $299.5
Precio: $149.75
Precio: $399.0
Precio: $89.99
Precios modificados:
Precio: $199.99
Precio: $299.5
Precio: $129.99
Precio: $399.0
Precio: $89.99
BUILD SUCCESSFUL (total time: 0 seconds)
```

13. Impresión recursiva de arrays antes y después de modificar un elemento.

Crea un programa que:

- Declare e inicialice un array con los precios de algunos productos.
- Use una función recursiva para mostrar los precios originales.
- Modifique el precio de un producto específico.
- Use otra función recursiva para mostrar los valores modificados.

```
1 package ejercicio13;
2
3 public class Ejercicio13 {
4
5     public static void main(String[] args) {
6
7         double[] precios = {199.99, 299.50, 149.75, 399.00, 89.99};
8
9         System.out.println("Precios originales:");
10        imprimirPreciosRecursivo(precios, 0);
11
12        precios[3] = 386.99;
13
14        System.out.println("Precios modificados:");
15        imprimirPreciosRecursivo(precios, 0);
16    }
17    // Función recursiva
18    public static void imprimirPreciosRecursivo(double[] array, int indice) {
19        if (indice < array.length) {
20            System.out.println("Precio: $" + array[indice]);
21            imprimirPreciosRecursivo(array, indice + 1);
22        }
23    }
24 }
25
```

Output x

Programacion2 - C:\Users\Eze\Documents\UTN - Programacion\Programacion2 x Ejercicio13 (run) x

```
Precios originales:
Precio: $199.99
Precio: $299.5
Precio: $149.75
Precio: $399.0
Precio: $89.99
Precios modificados:
Precio: $199.99
Precio: $299.5
Precio: $149.75
Precio: $386.99
Precio: $89.99
BUILD SUCCESSFUL (total time: 0 seconds)
```