

PROGRAMACIÓN II

Trabajo Práctico 2:

Programación Estructurada

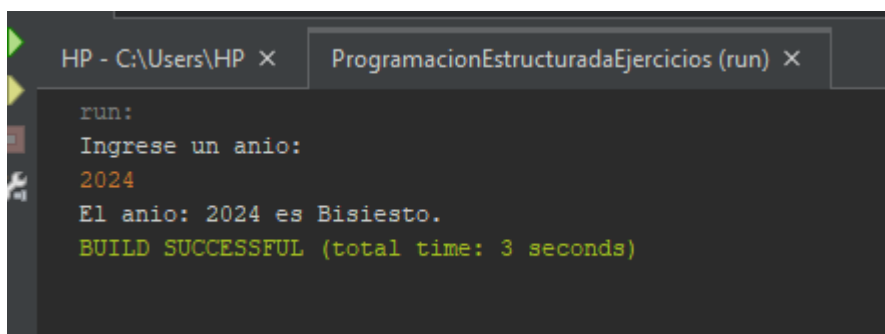
Alumna: García Cintia.

Desarrollar los siguientes ejercicios en Java utilizando el paradigma de programación estructurada. Agrupados según el tipo de estructuras o conceptos aplicados:

1) Verificación de Año Bisiesto.

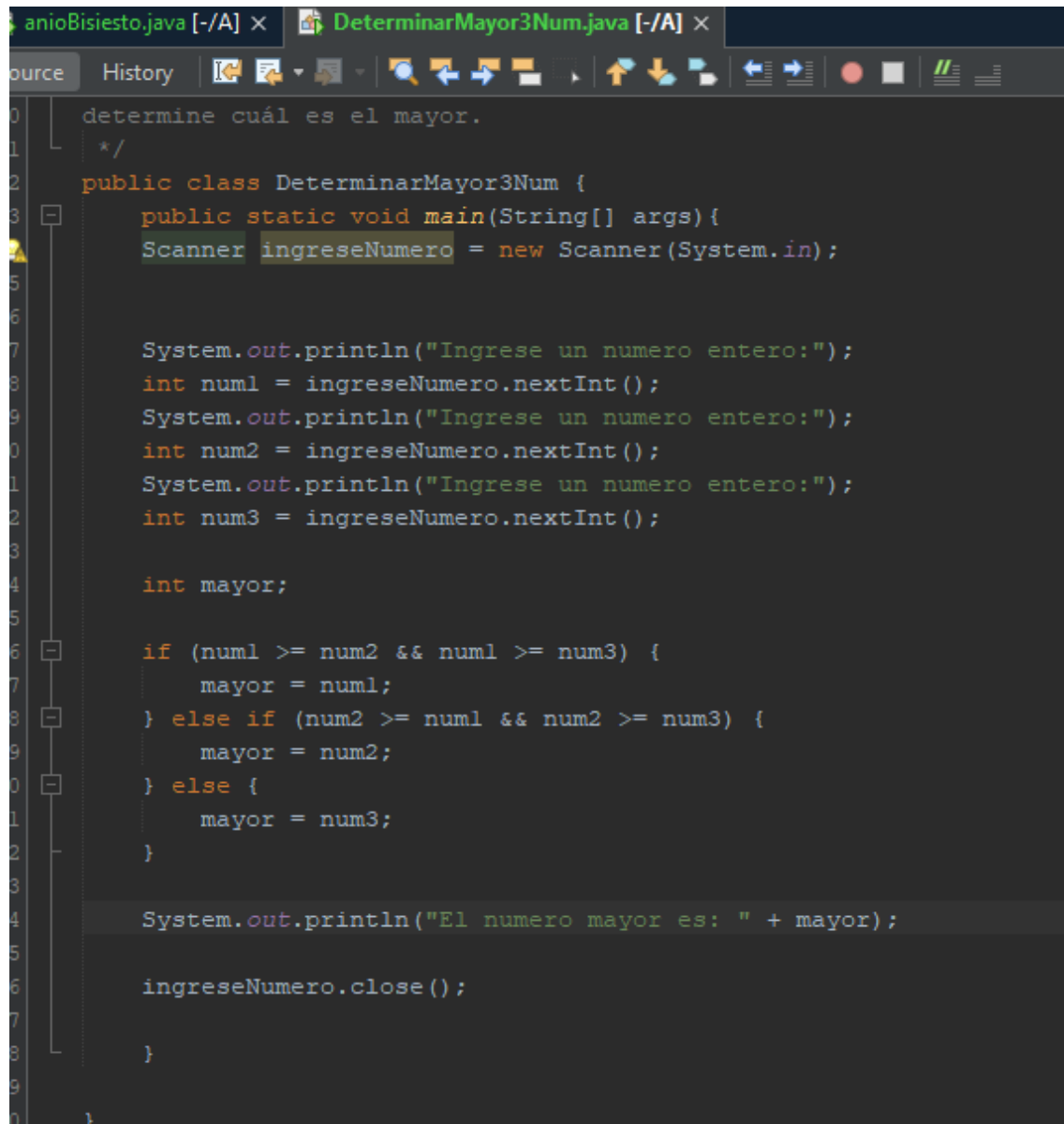
Escribe un programa en Java que solicite al usuario un año y determine si es bisiesto. Un año es bisiesto si es divisible por 4, pero no por 100, salvo que sea divisible por 400.

```
public class anioBisiesto {  
    public static void main(String[] args) {  
        Scanner esBisiesto = new Scanner(System.in);  
  
        System.out.println("Ingrese un año:");  
        int anio = esBisiesto.nextInt();  
  
        if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {  
            System.out.println("El año: " + anio + " es Bisiesto.");  
        } else {  
            System.out.println("El año: " + anio + " no es Bisiesto");  
        }  
        esBisiesto.close();  
    }  
}
```

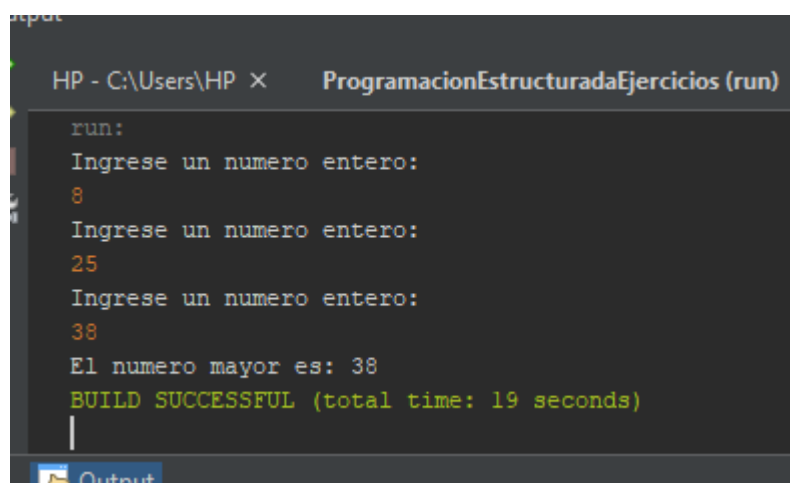


```
run:  
Ingrese un año:  
2024  
El año: 2024 es Bisiesto.  
BUILD SUCCESSFUL (total time: 3 seconds)
```

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  determine cuál es el mayor.  
2  */  
3  public class DeterminarMayor3Num {  
4      public static void main(String[] args){  
5          Scanner ingresoNumero = new Scanner(System.in);  
6  
7          System.out.println("Ingrese un numero entero:");  
8          int num1 = ingresoNumero.nextInt();  
9          System.out.println("Ingrese un numero entero:");  
10         int num2 = ingresoNumero.nextInt();  
11         System.out.println("Ingrese un numero entero:");  
12         int num3 = ingresoNumero.nextInt();  
13  
14         int mayor;  
15  
16         if (num1 >= num2 && num1 >= num3) {  
17             mayor = num1;  
18         } else if (num2 >= num1 && num2 >= num3) {  
19             mayor = num2;  
20         } else {  
21             mayor = num3;  
22         }  
23  
24         System.out.println("El numero mayor es: " + mayor);  
25  
26         ingresoNumero.close();  
27     }  
28 }  
29 }  
30 }
```



```
run:  
Ingrese un numero entero:  
8  
Ingrese un numero entero:  
25  
Ingrese un numero entero:  
38  
El numero mayor es: 38  
BUILD SUCCESSFUL (total time: 19 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"

```
public class ClasificacionEdad {  
    public static void main(String[] args) {  
        Scanner ingreseEdad = new Scanner(System.in);  
  
        System.out.println("Ingrese su edad por favor");  
        int edad = ingreseEdad.nextInt();  
  
        if (edad < 12) {  
            System.out.println("Es Niño");  
        } else if (edad >= 12 && edad <= 17) {  
            System.out.println("Adolescente");  
        } else if (edad >= 18 && edad <= 59) {  
            System.out.println("Adulto");  
        } else {  
            System.out.println("Adulto mayor");  
        }  
    }  
}
```

ProgramacionEstructuradaEjercicios.ClasificacionEdad >

run:
Ingrese su edad por favor:
18
Adulto
BUILD SUCCESSFUL (total time: 4 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

```
El programa debe mostrar el precio original, el descuento aplicado y el precio final
*/
public class CalculadoraDescuento {
    public static void main(String[] args){
        Scanner ingPrecio = new Scanner(System.in);

        System.out.println("Ingrese el precio:");
        double precioOriginal = ingPrecio.nextDouble();

        System.out.println("Ingrese su categoria: (A,B o C:) ");
        String categoria = ingPrecio.next().toUpperCase();

        double descuento = 0;

        if(categoria.equals("A")) {
            descuento = 0.10;
        } else if (categoria.equals("B")) {
            descuento = 0.15;
        } else if (categoria.equals("C")) {
            descuento = 0.20;
        } else {
            System.out.println("Categoría inválida. No se aplica descuento.");
        }

        double montoDescuento = precioOriginal * descuento;
        double precioFinal = precioOriginal - montoDescuento;

        System.out.println("\nResumen:");
        System.out.println("Precio original: $" + precioOriginal);
        System.out.println("Descuento aplicado: $" + montoDescuento);
        System.out.println("Precio final: $" + precioFinal);
    }
}
```

```
Input - ProgramacionEstructuradaEjercicios (run)
run:
Ingrese el precio:
58
Ingrese su categoria: (A,B o C:)
a

Resumen:
Precio original: $58.0
Descuento aplicado: $5.8000000000000001
Precio final: $52.2
BUILD SUCCESSFUL (total time: 8 seconds)
```

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.

```

import java.util.Scanner;

/**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. */
public class SumaNumerosPares {
    public static void main(String[] args){
        Scanner ingNumero = new Scanner(System.in);
        int numero = -1;
        int sumaPares = 0;

        System.out.print("Ingrese un numero (0 para terminar):");
        numero = ingNumero.nextInt();

        while (numero != 0){
            if (numero % 2 == 0) {
                sumaPares+=numero;
            }

            System.out.print("Ingrese otro numero (0 para terminar): ");
            numero = ingNumero.nextInt();
        }
        System.out.println("La suma de los numeros pares es: " + sumaPares);
    }
}

```

Output - ProgramacionEstructuradaEjercicios (run)

```

run:
Ingrese un numero (0 para terminar):8
Ingrese otro numero (0 para terminar): 75
Ingrese otro numero (0 para terminar): 85
Ingrese otro numero (0 para terminar): 65
Ingrese otro numero (0 para terminar): 10
Ingrese otro numero (0 para terminar): 0
La suma de los numeros pares es: 18
BUILD SUCCESSFUL (total time: 13 seconds)

```

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.

```

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

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

        for (int i = 1; i <= 10; i++){
            System.out.print("Ingrese el numero: " + i + ":");
            int numero = ingNumero.nextInt();

            if (numero > 0){
                positivos++;
            } else if (numero < 0){
                negativos++;
            } else{
                ceros++;
            }
        }
        System.out.println("Positivos: " + positivos);
        System.out.println("Negativos: " + negativos);
        System.out.println("Ceros: " + ceros);
    }
}

```

tput - ProgramacionestructuradaEjercicios (run)

```

Ingrese el numero: 4:74
Ingrese el numero: 5:0
Ingrese el numero: 6:25
Ingrese el numero: 7:-14
Ingrese el numero: 8:89
Ingrese el numero: 9:10
Ingrese el numero: 10:-104
Positivos: 6
Negativos: 3
Ceros: 1
BUILD SUCCESSFUL (total time: 29 seconds)

```

. 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.

```

import java.util.Scanner;

/**Escribe un programa que solicite al usuario una nota entre 0
usuario ingresa un número fuera de este rango, debe seguir pidiendo
hasta que ingrese un valor válido. */
public class ValidacionNota {
    public static void main(String[] args){
        Scanner ingNumero = new Scanner(System.in);
        int nota;

        do{
            System.out.print("Ingrese una nota (0-10):");
            nota = ingNumero.nextInt();

            if(nota == 0 || nota >10){
                System.out.println("No es valida:");
            }
        }while (nota < 0|| nota >10);
        System.out.println("Nota valida:");
    }
}

```

```

out - ProgramacionEstructuradaEjercicios (run)
run:
Ingrese una nota (0-10):9
Nota valida:
BUILD SUCCESSFUL (total time: 5 seconds)

```

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: $\text{PrecioFinal} = \text{PrecioBase} + (\text{PrecioBase} \times \text{Impuesto}) - (\text{PrecioBase} \times \text{Descuento})$ 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.

```

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

        System.out.println("Ingrese el precio del producto: ");
        double precioBase = calculoPrecio.nextDouble();
        System.out.println("Ingrese el pocentaje del impuesto:");
        double impuesto = calculoPrecio.nextDouble();
        System.out.println("Ingrese el descuento en porcentaje: ");
        double descuento = calculoPrecio.nextDouble();

        double precioFinal = calcularPrecioFinal(precioBase, impuesto, descuento);

        System.out.println("El Precio final del producto es: $ " + precioFinal);
    }

    //fuera del main
    public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento) {
        double impuestoDecimal = impuesto / 100;
        double descuentoDecimal = descuento / 100;

        double precioFinal = precioBase + (precioBase * impuestoDecimal) - (precioBase * descuentoDecimal)
        return precioFinal;
    }
}

```

```

run:
Ingrese el precio del producto:
96
Ingrese el pocentaje del impuesto:
23
Ingrese el descuento en porcentaje:
15
El Precio final del producto es: $ 103.68
BUILD SUCCESSFUL (total time: 10 seconds)
|

```

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.


```

// Método que suma precio del producto y costo de envío
public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
    return precioProducto + costoEnvio;
}

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

    // Solicitar datos al usuario
    System.out.print("Ingrese el peso del paquete (kg): ");
    double peso = entrada.nextDouble();
    entrada.nextLine(); //

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

    System.out.print("Ingrese el precio del producto: ");
    double precioProducto = entrada.nextDouble();

    // Calcular costo de envío y total
    double costoEnvio = calcularCostoEnvio(peso, zona);
    double total = calcularTotalCompra(precioProducto, costoEnvio);

    // Mostrar resultados
    System.out.println("\nTotal a abonar: ");
    System.out.println("Costo de envío: $" + costoEnvio);
    System.out.println("Total a pagar: $" + total);
}

```

Output - ProgramacionEstructuradaEjercicios (run)

```

run:
Ingrese el peso del paquete (kg): 18
Ingrese la zona de envío (Nacional o Internacional): nacional
Ingrese el precio del producto: 150

Total a abonar:
Costo de envío: $90.0
Total a pagar: $240.0
BUILD SUCCESSFUL (total time: 14 seconds)

```

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.

```

public static int actualizarStock (int stockActual, int cantidadVendida, int cantidadRecibida) {
    return stockActual - cantidadVendida + cantidadRecibida;
}

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

    //Se solicita datos al usuario
    System.out.println("Ingrese el stock actual: ");
    int stockActual = input.nextInt();
    System.out.println("Ingrese las ventas realizadas:");
    int cantidadVendida = input.nextInt();
    System.out.println("Ingrese las cantidades recibidas: ");
    int cantidadRecibida = input.nextInt();

    int stockFinal = actualizarStock(stockActual, cantidadVendida ,cantidadRecibida);

    System.out.println("El stock final actualizado es: " + stockFinal);
}

```

```

run:
Ingrese el stock actual:
96
Ingrese las ventas realizadas:
15
Ingrese las cantidades recibidas:
150
El stock final actualizado es: 231
BUILD SUCCESSFUL (total time: 11 seconds)
|

```

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 calculoVariableGlobal {
    public static final double DESCUENTO_ESPECIAL = 0.20;

    public static void calcularDescuento (double precio){
        double descuentoAplicado = precio * DESCUENTO_ESPECIAL;
        double precioFinal = precio - DESCUENTO_ESPECIAL;

        System.out.println("El descuento aplicado es: " + descuentoAplicado);
        System.out.println("El precio final es: " + precioFinal);
    }

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

        System.out.println("Ingrese el precio:");
        double precio=input.nextDouble();

        calcularDescuento (precio);
    }
}

```

```

run:
Ingrese el precio:
1548
El descuento aplicado es: 309.6
El precio final es: 1547.8
BUILD SUCCESSFUL (total time: 6 seconds)

```

Arrays y Recursividad: 12. Modificación de un array de precios y visualización de resultados. Crea un programa que: a. Declare e inicialice un array con los precios de algunos productos. b. Muestre los valores originales de los precios. c. Modifique el precio de un producto específico. d. Muestre los valores modificados

```

public static void main(String[] args) {

    double[] precios = {2.5,199.99,999.25,55.99};

    System.out.println("Precios originales: ");
    for (int i = 0; i < precios.length; i++){
        System.out.println("Producto :" + (i + 1) + ": $" + precios[i]);
    }
    Scanner entrada = new Scanner(System.in);
    System.out.print("\nIngrese el indice del producto a modificar (0 a 3):");
    int indice = entrada.nextInt();

    System.out.print("Ingrese el nuevo precio: ");
    double nuevoPrecio = entrada.nextDouble();

    precios[indice] = nuevoPrecio;

    System.out.println("\nPrecios modificados:");
    for (int i = 0; i < precios.length; i++) {
        System.out.println("Producto " + (i + 1) + ": $" + precios[i]);
    }
    double total = suma(precios);
    System.out.println("\nSuma total de precios: $" + total);
}

private static double suma(double[] precios) {
    double acu = 0;

```

```

run:
Precios originales:
Producto :1: $2.5
Producto :2: $199.99
Producto :3: $999.25
Producto :4: $55.99

Ingrese el indice del producto a modificar (0 a 3): 2
Ingrese el nuevo precio: 96

Precios modificados:
Producto 1: $2.5
Producto 2: $199.99
Producto 3: $96.0
Producto 4: $55.99

Suma total de precios: $354.48
BUILD SUCCESSFUL (total time: 8 seconds)

```

13. Impresión recursiva de arrays antes y después de modificar un elemento. Crea un programa que: a. Declare e inicialice un array con los precios de algunos productos. b. Use una función recursiva para mostrar los precios originales. c. Modifique el precio de un producto específico. d. Use otra función recursiva para mostrar los valores modificados.

```

        System.out.println("Precios originales: ");
        mostrarPrecios (precios,0);

        Scanner input = new Scanner(System.in);
        System.out.print("\nIngrese el indice del producto a modificar (0 a 3): ");
        int indice = input.nextInt();

        System.out.print("Ingrese el nuevo precio: ");
        double nuevoPrecio = input.nextDouble();

        precios[indice] = nuevoPrecio; // c. Modificar precio

        System.out.println("\nPrecios modificados:");
        mostrarPrecios(precios, 0);
    }

    //funcion recursiva
    public static void mostrarPrecios(double[] precios, int index) {
        if (index < precios.length) {
            System.out.println("Producto " + (index + 1) + ": $" + precios[index]);
            mostrarPrecios(precios, index + 1);
        }
    }
}

```

```

Ingrese el indice del producto a modificar (0 a 3): 1
Ingrese el nuevo precio: 25

Precios modificados:
Producto 1: $2.5
Producto 2: $25.0
Producto 3: $999.25
Producto 4: $55.99
BUILD SUCCESSFUL (total time: 6 seconds)

```

Enlace de GitHub

<https://github.com/Cigarcia1307/Programacion2-UTN-GarciaCintia.git>