

TRABAJO PRÁCTICO N° 2

Programación Estructurada

Alumno: Ignacio Figueroa – 45.406.120

Tecnicatura Universitaria en Programación – UTN

Materia: Programación II

Comisión: 7

Objetivos

Desarrollar habilidades en programación estructurada en Java, abordando desde conceptos básicos como operadores y estructuras de control hasta temas avanzados como funciones, recursividad y estructuras de datos. Se busca fortalecer la capacidad de análisis y solución de problemas mediante un enfoque práctico.

Repositorio de mi código: [tup-utn/2do-cuatrimestre/programacion-II/programacion-estructurada/tp-programacion-estructurada/tp-3-programacion-estructurada](https://github.com/figueroaignacio/tup-utn-II/programacion-estructurada/tp-programacion-estructurada/tp-3-programacion-estructurada) at main · figueroaignacio/tup-utn

Consignas

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.

```
System.out.print("Ingrese su año: ");
int anio = scanner.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.");
}
```

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.

```
System.out.print("Ingrese el primer numero: ");
int x = scanner.nextInt();

System.out.print("Ingrese el segundo numero: ");
int y = scanner.nextInt();

System.out.print("Ingrese el tercer numero: ");
int z = scanner.nextInt();

int mayor = x;

if (y > mayor) {
    mayor = y;
}

if (z > mayor) {
    mayor = z;
}

System.out.println("El mayor es: " + mayor);
```

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"

```
System.out.print("Ingrese su edad: ");
int edad = scanner.nextInt();

if (edad < 12) {
    System.out.println("Niño");
} else if (edad ≤ 17) { // Entre 12 y 17
    System.out.println("Adolescente");
} else if (edad ≤ 59) { // Entre 18 y 59
    System.out.println("Adulto");
} else { // 60 o más
    System.out.println("Adulto mayor");
}
```

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

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

System.out.print("Ingrese la categoría del producto (A, B o C): ");
char categoria = scanner.next().toUpperCase().charAt(0);

double descuento = 0;

switch (categoria) {
    case 'A':
        descuento = 0.10;
        break;
    case 'B':
        descuento = 0.15;
        break;
    case 'C':
        descuento = 0.20;
        break;
    default:
        System.out.println("Categoría inválida.");
        scanner.close();
        return;
}

double precioDescuento = precio * descuento;
double precioFinal = precio - precioDescuento;

System.out.println("Precio original: " + precio);
System.out.println("Descuento aplicado: " + (descuento * 100) + "%");
System.out.println("Precio final: " + precioFinal);
```

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.

```
int numero;  
int sumaPares = 0;  
  
while (true) {  
    System.out.print("Ingrese un número (0 para terminar la ejecución: )");  
    numero = scanner.nextInt();  
  
    if (numero == 0) {  
        break;  
    }  
  
    if (numero % 2 == 0) {  
        sumaPares += numero;  
    }  
}  
  
System.out.println("La suma de los números pares es: " + sumaPares);
```

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 zeros.

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

for (int i = 1; i ≤ 100; i++) {
    System.out.print("Ingrese el número " + i + ": ");
    int num = scanner.nextInt();

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

System.out.println("\nResultados:");
System.out.println("Positivos: " + positivos);
System.out.println("Negativos: " + negativos);
System.out.println("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.

```
int nota;  
  
do {  
    System.out.print("Ingrese una nota (0-10): ");  
    nota = scanner.nextInt();  
  
    if (nota < 0 || nota > 10) {  
        System.out.println("Error: Nota inválida. Ingrese una nota entre 0 y 10.");  
    }  
} while (nota < 0 || nota > 10);  
  
System.out.println("Nota guardada correctamente: " + nota);
```


8. Crea un método `calcularPrecioFinal(double impuesto, double descuento)` que calcule el precio final de un producto en un e-commerce. La fórmula es: $\text{PrecioFinal} = \text{PrecioBase} + (\text{PrecioBase} \times \text{Impuesto}) - (\text{PrecioBase} \times \text{Descuento})$

Método:

```
public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento) {
    double impDecimal = impuesto / 100;
    double desDecimal = descuento / 100;

    double precioFinal = precioBase + (precioBase * impDecimal) - (precioBase * desDecimal);

    return precioFinal;
}
```

Programa:

```
System.out.print("Ingrese el precio base del producto: ");
double precioBase = scanner.nextDouble();

System.out.print("Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%): ");
double impuesto = scanner.nextDouble();

System.out.print("Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%): ");
double descuento = scanner.nextDouble();

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

System.out.print("El precio final del producto es: " + precioFinal);
```

9. 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.

Métodos:

```
// a.
public static double calcularCostoEnvio(double peso, String zona) {
    double costoPorKg;

    if (zona.equalsIgnoreCase("Nacional")) {
        costoPorKg = 5.0;
    } else if (zona.equalsIgnoreCase("Internacional")) {
        costoPorKg = 10.0;
    } else {
        System.out.println("Zona inválida. Se considerará costo 0.");
        return 0;
    }

    double costoFinal = peso * costoPorKg;

    return costoFinal;
}

//b.
public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
    double total = precioProducto + costoEnvio;
    return total;
}
```

Programa:

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

System.out.print("Ingrese el peso del paquete en kg: ");
double peso = scanner.nextDouble();

scanner.nextLine();

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

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

System.out.println("El costo de envío es: " + costoEnvio);
System.out.println("El total a pagar es: " + total);
```

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: $\text{NuevoStock} = \text{StockActual} - \text{CantidadVendida} + \text{CantidadRecibida}$

Método:

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

Programa:

```
System.out.print("Ingrese el stock actual del producto: ");
int stockActual = scanner.nextInt();

System.out.print("Ingrese la cantidad vendida: ");
int cantidadVendida = scanner.nextInt();

System.out.print("Ingrese la cantidad recibida: ");
int cantidadRecibida = scanner.nextInt();

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

System.out.print("El nuevo stock del producto es: " + nuevoStock);
```

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.

Método:

```
static double DESCUENTO_ESPECIAL = 0.10;

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);
}
```

Programa:

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

calcularDescuentoEspecial(precio);
```

12. 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.

```
double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};

System.out.println("Precios originales:");
for (double precio : precios) {
    System.out.println("Precio: $" + precio);
}
precios[2] = 129.99;

System.out.println("\nPrecios modificados:");
for (double precio : precios) {
    System.out.println("Precio: $" + precio);
}
```

13. 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.

Método:

```
public static void imprimirArray(double[] precios, int indice) {
    if (indice == precios.length) {
        return;
    }

    System.out.println("Precio: $" + precios[indice]);
    imprimirArray(precios, indice + 1);
}
```

Programa:

```
double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};

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

precios[2] = 129.99;

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

scanner.close();
```