

Todo en <https://github.com/manfredialdo/UTN-TUPaD-P2.git>

Ejercicio1. 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.

```
/*
 * TP2 - UTN TUPAD - PROGRAMACION2
 * consigna1 - trabajopractico2
 * @aldo manfredi 2025-2C
 */
import java.util.Scanner;

public class ejerciciouno {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("ingrese un año: ");
        int anio = sc.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");
        }
        sc.close();
    }
}
```

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.

```
/*
 * TP2 - UTN TUPAD - PROGRAMACION2
 * consigna2 - trabajopractico2
 * @aldo manfredi 2025-2C
 */
import java.util.Scanner;

public class ejerciciodos {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("ingrese un nro: ");
        int nro1 = sc.nextInt();

        System.out.print("ingrese otro nro: ");
        int nro2 = sc.nextInt();

        System.out.print("ingrese otro nro: ");
        int nro3 = sc.nextInt();

        if (nro1 > nro2 && nro1>nro3) {
            System.out.println("el numero " + nro1 + " es el mayot de entre los tres numeros");
        } else if (nro2 > nro1 && nro2>nro3){
            System.out.println("el numero " + nro2 + " es el mayot de entre los tres numeros");
        } else if (nro3 > nro1 && nro3>nro2){
            System.out.println("el numero " + nro3 + " es el mayot de entre los tres numeros");
        } else {
            System.out.println("operacion invalida.. repita las entradas con distintos numeros");
        }

        sc.close();
    }
}
```

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"

```
/*  
 * UTN TUPAD - PROGRAMACION2  
 * consigna3 - trabajopractico2  
 * @aldo manfredi 2025-2C  
 */  
  
import java.util.Scanner;  
  
public class ejerciciotres {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("ingrese un nro: ");  
        int edad = sc.nextInt();  
  
        if (edad < 12) {  
            System.out.println("Ud es un niño");  
        } else if (edad <= 17) {  
            System.out.println("Ud es un adolescente");  
        } else if (edad <= 59) {  
            System.out.println("Ud es un adulto");  
        } else if (edad >= 60) {  
            System.out.println("Ud es un adulto mayor");  
        }  
    }  
}
```

```
        sc.close();  
    }  
}
```

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

```
/*  
 * UTN TUPAD - PROGRAMACION2  
 * consigna4 - trabajopractico2  
 * @aldo manfredi 2025-2C  
 */  
  
import java.util.Scanner;  
  
public class ejerciciocuatro {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("precio del producto: ");  
        double precioOri = sc.nextDouble();  
  
        System.out.print("categoria (A, B o C): ");  
        char cate = sc.next().charAt(0);  
  
        double descu = 0.0;
```

```
switch (cate) {  
    case 'A':  
    case 'a':  
        descu = 0.10;  
        break;  
    case 'B':  
    case 'b':  
        descu = 0.15;  
        break;  
    case 'C':  
    case 'c':  
        descu = 0.20;  
        break;  
    default:  
        System.out.println("Cate no valida.");  
        break;  
}  
  
double precioFinal = precioOri - (precioOri * descu);  
  
System.out.println("Precio original: " + precioOri);  
System.out.println("Descuento aplicado: " + (descu * 100) + "%");  
System.out.println("Precio final: " + precioFinal);  
  
sc.close();  
}  
}
```

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.

Ejemplo de entrada/salida:

Ingrese un número (0 para terminar): 4

Ingrese un número (0 para terminar): 7

Ingrese un número (0 para terminar): 2

Ingrese un número (0 para terminar): 0

La suma de los números pares es: 6

```
/*
 * UTN TUPAD - PROGRAMACION2
 * consigna5 - trabajopractico2
 * @aldo manfredi 2025-2C
 */
import java.util.Scanner;

public class ejerciciocinco {
    public static void main(String[] args) {
        int nroIngresado = 0;
        int acumulador = 0;
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese un número (0 para terminar): ");
        nroIngresado = sc.nextInt();
        while (nroIngresado != 0) {
            if (nroIngresado % 2 == 0) {
```

```

        acumulador = acumulador + nroIngresado;
    }

    System.out.print("Ingrese otro número (0 para terminar): ");

    nroIngresado = sc.nextInt();

}

System.out.println("La suma de los números pares ingresados es: " + acumulador);

sc.close();

}

}

```

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.

```

/*
 * UTN TUPAD - PROGRAMACION2
 * consigna6 - trabajopractico2
 * @aldo manfredi 2025-2C
 */

import java.util.Scanner;

public class ejercicio6 {

    public static void main(String[] args) {

        int nroIngresado = 0;

        int positivos = 0;

        int negativos = 0;

        int ceros = 0;

        Scanner sc = new Scanner(System.in);

        for (int i = 1; i<=10; i++){

            System.out.print("Ingrese número " + i + ": ");

            nroIngresado = sc.nextInt();

```

```

        if (nroIngresado>0) {
            positivos = positivos+1;
        } else if (nroIngresado<0){
            negativos = negativos+1;
        } else if (nroIngresado==0){
            ceros = ceros+1;
        }
    }

    System.out.print("positivos: " + positivos + "\n");
    System.out.print("negativos: " + negativos + "\n");
    System.out.print("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.

Ejemplo de entrada/salida:

Ingrese una nota (0-10): 15

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): 8

Nota guardada correctamente.

```

/*
* UTN TUPAD - PROGRAMACION2
* consigna7 - trabajopractico2

```


* @aldo manfredi 2025-2C

*/

```
import java.util.Scanner;
```

```
public class ejerciciosiete {
```

```
    public static void main(String[] args) {
```

```
        int nota;
```

```
        Scanner sc = new Scanner(System.in);
```

```
        do {
```

```
            System.out.println("ingrese una nota (0-10):");
```

```
            nota = sc.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.");
```

```
        sc.close();
```

```
    }
```

```
}
```

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:

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

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.

/*

* UTN TUPAD - PROGRAMACION2

* consigna8 - trabajopractico2

* @aldo manfredi 2025-2C

*/

```
import java.util.Scanner;
```

```
public class ejercicioocho {
```

```
    public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento) {
```

```
        double impuestoDecimal = impuesto / 100.0;
```

```
        double descuentoDecimal = descuento / 100.0;
```

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

```
        return precioFinal;
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        double precioBase;
```

```
        double impuesto;
```

```
        double descuento;
```

```
        System.out.println("Ingrese el precio base del producto: ");
```

```
        precioBase = sc.nextDouble();
```

```
        System.out.println("Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%): ");
```

```
        impuesto = sc.nextDouble();
```

```
        System.out.println("Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%): ");
```

```
        descuento = sc.nextDouble();
```

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

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

```
        sc.close();
```

```
    }
```

```
}
```

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.

Ejemplo de entrada/salida:

Ingrese el precio del producto: 50

Ingrese el peso del paquete en kg: 2

Ingrese la zona de envío (Nacional/Internacional): Nacional

El costo de envío es: 10.0

El total a pagar es: 60.0

```
/*  
 * UTN TUPAD - PROGRAMACION2  
 * consigna9 - trabajopractico2  
 * @aldo manfredi 2025-2C  
 */  
  
import java.util.Scanner;  
  
public class ejercicionueve {  
  
    public static double calcularCostoEnvio(double peso, String zona) {  
        double costoBase = 0.0;  
        if (zona.equalsIgnoreCase("Nacional")) {  
            costoBase = 5.0;  
        } else if (zona.equalsIgnoreCase("Internacional")) {  
            costoBase = 10.0;  
        }  
    }  
}
```

```

    } else {

        System.out.println("Zona de envío no válida. El costo de envío será 0.");

        costoBase = 0.0;

    }

    return peso * costoBase;

}

public static double calcularTotalCompra(double precioProducto, double costoEnvio) {

    return precioProducto + costoEnvio;

}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    double peso;

    double precio;

    String zona;

    double costoEnvio;

    double totalCompra;

    System.out.println("Ingrese el precio del producto: ");

    precio = sc.nextDouble();

    System.out.println("Ingrese el peso del paquete en kg: ");

    peso = sc.nextDouble();

    sc.nextLine();

    System.out.println("Ingrese la zona de envío (Nacional/Internacional): ");

    zona = sc.nextLine();

    costoEnvio = calcularCostoEnvio(peso, zona);

    totalCompra = calcularTotalCompra(precio, costoEnvio);

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

    System.out.println("El total a pagar es: " + totalCompra);

```

```

        sc.close();
    }
}

```

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:

NuevoStock = StockActual – CantidadVendida + CantidadRecibida

NuevoStock = CantidadVendida + CantidadRecibida

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

Ejemplo de entrada/salida:

Ingrese el stock actual del producto: 50

Ingrese la cantidad vendida: 20

Ingrese la cantidad recibida: 30

El nuevo stock del producto es: 60

```

/*
 * UTN TUPAD - PROGRAMACION2
 * consigna10 - trabajopractico2
 * @aldo manfredi 2025-2C
 */
import java.util.Scanner;

public class ejerciciodiez {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int StockActual;

        int CantidadVendida;
    }
}

```

```

    int CantidadRecibida;

    System.out.println("Ingrese el stock actual del producto: ");
    StockActual = sc.nextInt();
    System.out.println("Ingrese la cantidad vendida: ");
    CantidadVendida = sc.nextInt();
    System.out.println("Ingrese la cantidad recibida: ");
    CantidadRecibida = sc.nextInt();

    int nuevoStock = actualizarStock(StockActual, CantidadVendida, CantidadRecibida);
    System.out.println("El nuevo stock del producto es: " + nuevoStock);
    sc.close();
}

public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
    int nuevoStock = stockActual - cantidadVendida + cantidadRecibida;
    return 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.

Ejemplo de entrada/salida:

Ingrese el precio del producto: 200

El descuento especial aplicado es: 20.0

El precio final con descuento es: 180.0

```
/*
 * UTN TUPAD - PROGRAMACION2
 * consigna11 - trabajopractico2
 * @aldo manfredi 2025-2C
 */
import java.util.Scanner;

public class ejercicioonce {

    static double variableEjemploGlobal = 0.10;

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Ingrese el precio del producto: ");

        double precio = sc.nextDouble();

        calcularDescuentoEspecial(precio);

        sc.close();

    }

    public static void calcularDescuentoEspecial(double precio) {

        double descuentoAplicado = precio * variableEjemploGlobal;

        System.out.println("El descuento especial aplicado es: " + descuentoAplicado);

        System.out.println("El precio final con descuento es: " + (precio - descuentoAplicado));

    }

}
```

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.

```
/*
 * UTN TUPAD - PROGRAMACION2
 * consigna12 - trabajopractico2
 * @aldo manfredi 2025-2C
 */
public class ejerciciodoce {
    public static void main(String[] args) {
        double [] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
        System.out.println("Precios originales:");
        for (int i = 1; i<=precios.length; i++){
            System.out.println("Precio: $" + precios[i-1]);
        }
        precios[2] = 129.99;
        System.out.println("Precios modificados:");
        for (int i = 1; i<=precios.length; i++){
            System.out.println("Precio: $" + precios[i-1]);
        }
    }
}
```

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.

```
/*
 * UTN TUPAD - PROGRAMACION2
 * consigna13 - trabajopractico2
 * @aldo manfredi 2025-2C
 */
public class ejerciciotrece {
    private static double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};
    public static void main(String[] args) {
        System.out.println("Precios originales:");
        mostrarArrayRecursiva(precios.length - 1);
        precios[2] = 129.99;
        System.out.println("\nPrecios modificados:");
        mostrarArrayRecursiva(precios.length - 1);
    }
    private static void mostrarArrayRecursiva(int i) {
        if (i < 0) {
            return;
        }
        mostrarArrayRecursiva(i - 1);
        System.out.println("Precio: $" + precios[i]);
    }
}
```