

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 com.mycompany.tp2;

import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioUno {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Ingresa un año: ");
        int añoIngresado = Integer.parseInt(input.nextLine());

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

```
--- exec:3.1.0:exec (default-cli) @ TP2 ---
```

```
Ingresa un año: 2024
```

```
El año 2024 es bisiesto
```

```
-----  
BUILD SUCCESS  
-----
```

```
Total time: 21.758 s
```

```
Finished at: 2025-08-20T18:32:16-03:00  
-----
```

```
Nothing to compile - all classes are up to date.
```

```
--- exec:3.1.0:exec (default-cli) @ TP2 ---
```

```
Ingresa un año: 1900
```

```
El año 1900 no es bisiesto
```

```
-----  
BUILD SUCCESS  
-----
```

```
Total time: 7.886 s
```

```
Finished at: 2025-08-20T18:32:49-03:00  
-----
```

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.

```
package com.mycompany.tp2;

import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioDos {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Ingresa el primer numero: ");
        int num1 = Integer.parseInt(input.nextLine());
        System.out.print("Ingresa el segundo numero: ");
        int num2 = Integer.parseInt(input.nextLine());
        System.out.print("Ingresa el tercer numero: ");
        int num3 = Integer.parseInt(input.nextLine());

        int numeroM = num1;

        if(numeroM < num2){
            numeroM = num2;
        } else if (numeroM < num3){
            numeroM = num3;
        }

        System.out.print("El numero mayor es " + numeroM);
    }
}
```

```
] --- exec:3.1.0:exec (default-cli) @ TP2 ---  
Ingresa el primer numero: 8  
Ingresa el segundo numero: 12  
Ingresa el tercer numero: 5  
El numero mayor es 12  
-----  
BUILD SUCCESS  
-----  
Total time: 13.399 s  
Finished at: 2025-08-20T18:34:55-03:00  
-----  
|
```

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"

```

package com.mycompany.tp2;

import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioTres {

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

        System.out.print("Ingrese su edad: ");
        int edad = Integer.parseInt(input.nextLine());

        if(edad <= 12){
            System.out.print("Niño");
        } else if (edad > 12 && edad <= 17) {
            System.out.print("Adolecente");
        } else if (edad > 18 && edad <= 59) {
            System.out.print("adulto");
        } else if (edad > 60) {
            System.out.print("Adulto Mayor");
        }
    }
}

```

```

Ingrese su edad: 25
adulto
-----
BUILD SUCCESS
-----
Total time: 6.157 s
Finished at: 2025-08-20T18:36:33-03:00
-----

```

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

```
import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioCuatro {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Ingrese el precio del producto: ");
        double precio = Integer.parseInt(input.nextLine());

        System.out.print("Ingrese la cegoria del descuento (A , B , C): ");
        char categoria = input.nextLine().charAt(0);

        switch(categoria){
            case 'A':
            case 'a':
                System.out.println("Descuento aplicado: 10%");
                System.out.print("Precio final: " + (precio * (1 - (10.0 / 100))) );
                break;
            case 'B':
            case 'b':
                System.out.println("Descuento aplicado: 15%");
                System.out.print("Precio final: " + (precio * (1 - (15.0 / 100))) );
                break;
            case 'C':
            case 'c':
                System.out.println("Descuento aplicado: 20%");
                System.out.print("Precio final: " + (precio * (1 - (20.0 / 100))) );
                break;
            default:
                System.out.println("operacion invalida");
        }
    }
}
```

```
--- exec:3.1.0:exec (default-cli) @ TP2 ---
Ingrese el precio del producto: 1000
Ingrese la cegoria del descuento (A , B , C): B
Descuento aplicado: 15%
Precio final: 850.0
-----
BUILD SUCCESS
```

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.

```
package com.mycompany.tp2;

import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioCinco {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Ingrese un numero (0 para terminar): ");
        int num = Integer.parseInt(input.nextLine());
        int suma = 0;

        while(num != 0){
            if(num % 2 == 0){
                suma += num;
            }
            System.out.print("Ingrese un numero (0 para terminar): ");
            num = Integer.parseInt(input.nextLine());
        }
        System.out.println("La suma de los numeros pares es: " + suma);
    }
}
```

```
Ingrese un numero (0 para terminar): 4
Ingrese un numero (0 para terminar): 7
Ingrese un numero (0 para terminar): 2
Ingrese un numero (0 para terminar): 0
La suma de los numeros pares es: 6
-----
BUILD SUCCESS
-----
```

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 com.mycompany.tp2;

import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioSeis {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int positivos = 0;
        int negativos = 0;
        int ceros = 0;

        for(int i = 1; i < 11; i++){
            System.out.print("Ingresa el numero: " + i + " ");
            int num = Integer.parseInt(input.nextLine());

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

        System.out.println("resultados: " + " positivos: " + positivos + " negativos: " + negativos + " ceros " + ceros);
    }
}

```

```

--- exec:3.1.0:exec (default-cli) @ TP2 ---
Ingresa el numero: 1 -5
Ingresa el numero: 2 3
Ingresa el numero: 3 0
Ingresa el numero: 4 -1
Ingresa el numero: 5 6
Ingresa el numero: 6 0
Ingresa el numero: 7 9
Ingresa el numero: 8 -3
Ingresa el numero: 9 4
Ingresa el numero: 10 -8
resultados: positivos: 4 negativos: 4 ceros 2
-----

```

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.


```

package com.mycompany.tp2;

import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioSiete {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int nota;

        do {
            System.out.print("Ingrese una nota (0-10): ");
            nota = Integer.parseInt(input.nextLine());

            if(nota > 10 || nota < 0){
                System.out.println("Error: Nota inválida. Ingrese una nota entre 0 y 10");
            }
        }while( nota > 10 || nota < 0);

        System.out.println("Nota guardada correctamente");
    }
}

```

```

--- exec:3.1.0:exec (default-cli) @ TP2 ---
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

-----
BUILD SUCCESS
-----

```

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.

```
import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioOcho {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Ingrese el precio base del producto: ");
        double precioBase = Integer.parseInt(input.nextLine());

        System.out.print("Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%): ");
        double impuesto = Integer.parseInt(input.nextLine());

        System.out.print("Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%): ");
        double descuento = Integer.parseInt(input.nextLine());

        double precioFinal = sucesionDeSimbolos(precioBase, impuesto, descuento);
        System.out.print("El precio final del producto es:" + precioFinal);
    }

    static double sucesionDeSimbolos(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;
    }
}
```

```
--- exec:3.1.0:exec (default-cli) @ TP2 ---
Ingrese el precio base del producto: 100
Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%): 10
Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%): 5
El precio final del producto es:105.0
-----
BUILD SUCCESS
-----
```

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

```
import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioNueve {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Ingrese el precio base del producto: ");
        double precioProducto = Integer.parseInt(input.nextLine());
        System.out.print("Ingrese el peso del paquete: ");
        double pesoProducto = Integer.parseInt(input.nextLine());
        System.out.print("Ingrese la zona de envío (Nacional/Internacional): ");
        String zonaProducto = input.nextLine();

        double costoEnvio = calcularCostoEnvio(pesoProducto, zonaProducto);
        double costoTotal = calcularTotalCompra(precioProducto, costoEnvio);

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

    static double calcularCostoEnvio(double peso, String zona){
        double costo = 0;

        if(zona.equalsIgnoreCase("nacional")){
            costo = 5 * peso;
        } else if (zona.equalsIgnoreCase("internacional")){
            costo = 10 * peso;
        }
        return costo;
    }

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

```
Ingrese el precio base del producto: 50
Ingrese el peso del paquete: 2
Ingrese la zona de envío (Nacional/Internacional): nacional
El costo de envío es: 10.0
El total a pagar es: 60.0
```

```
-----
BUILD SUCCESS
-----
```

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.

```
import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioDiez {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        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.println("El nuevo stock del producto es: " + nuevoStock);
    }

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

```

--- exec:3.1.0:exec (default-cli) @ TP2 ---
Ingrese el stock actual del producto: 50
Ingrese la cantidad vendida: 20
Ingrese la cantidad recibida: 30
El nuevo stock del producto es: 60
-----
BUILD SUCCESS
-----

```

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.

```

import java.util.Scanner;

/**
 *
 * @author lucko
 */
public class ejercicioOnce {
    public static final double DESCUENTO = 0.10;

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

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

        calcularDescuentoEspecial(precio);

        scanner.close();
    }

    public static void calcularDescuentoEspecial(double precio) {
        double descuentoAplicado = precio * DESCUENTO;
        double precioFinal = precio - descuentoAplicado;

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

```

```
Ingrese el precio del producto: 200
El descuento especial aplicado es: 20.0
El precio final con descuento es: 180.0
```

```
-----
BUILD SUCCESS
-----
```

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.

```
*/
public class ejercicioDoces {
    public static void main(String[] args) {

        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);
        }

        System.out.println();
        precios[2] = 129.99;

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

```
--- exec:3.1.0:exec (default-cil) @ 1F2 ---
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 SUCCESS
-----
Total time: 0.830 s
```

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.

```

public class ejercicioTrece {
    public static void main(String[] args) {
        double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};

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

        precios[2] = 129.99;
        System.out.println();

        System.out.println("Precios modificados:");
        mostrarPreciosModificados(precios, 0);
    }

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

        System.out.printf("Precio: $%.2f%n", precios[indice]);

        mostrarPreciosOriginales(precios, indice + 1);
    }

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

        System.out.printf("Precio: $%.2f%n", precios[indice]);

        mostrarPreciosModificados(precios, indice + 1);
    }
}

```

```

--- exec:3.1.0:exec (default-cli) @ TP2 ---

```

```

Precios originales:

```

```

Precio: $199,99

```

```

Precio: $299,50

```

```

Precio: $149,75

```

```

Precio: $399,00

```

```

Precio: $89,99

```

```

Precios modificados:

```

```

Precio: $199,99

```

```

Precio: $299,50

```

```

Precio: $129,99

```

```

Precio: $399,00

```

```

Precio: $89,99

```

```

-----
BUILD SUCCESS
-----

```

```

Total time: 0.795 s

```


LINK AL REPOSITORIO

<https://github.com/BarreraMariano/UTN-TUPaD-P2>