

Resolución Práctico 2: Programación Estructurada

Ejercicio 1: en este ejercicio debemos pedirle un año al usuario, verificar a través de un if si el año es bisiesto o no e imprimir por pantalla el resultado.

Condiciones para que el año sea bisiesto:

Tiene que ser divisible por 4 y no por 100 o tiene que ser divisible por 400.

Código:

```
public class EjercicioUno {  
  
    public static void main(String[] args) {  
        // Declaramos input para leer los datos  
        Scanner input = new Scanner(System.in);  
        int anio; // Declaramos la variable anio, es donde vamos a guardar el valor ingresado por el usuario  
  
        // Le pedimos el valor al usuario  
        System.out.println("Ingrese un año para verificar si es bisiesto");  
        anio = Integer.parseInt(input.nextLine());  
  
        // Verificamos en el condicional que sea divisible por 4 y no por 100 a excepcion que el año tambien sea divisible por 400.  
        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");  
        }  
    }  
}
```

Salida por pantalla ingresando 2024:

```
-----< com.mycompany:EjercitacionTpDos >-----  
] Building EjercitacionTpDos 1.0-SNAPSHOT  
-----[ jar ]-----  
  
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---  
  Ingrese un año para verficar si es bisiesto  
  2024  
- El año 2024 es bisiesto  
  
BUILD SUCCESS  
  
Total time: 4.516 s  
Finished at: 2025-07-17T20:37:52-03:00  
-----
```

Salida por pantalla ingresando 1900:

```
-----< com.mycompany:EjercitacionTpDos >-----  
Building EjercitacionTpDos 1.0-SNAPSHOT  
-----[ jar ]-----  
  
--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---  
Ingrese un año para verificar si es bisiesto  
1900  
El año 1900 no es bisiesto  
  
BUILD SUCCESS  
  
Total time: 4.973 s  
Finished at: 2025-07-17T20:38:24-03:00  
-----  
|
```

Ejercicio 2: En este ejercicio debíamos pedirle al usuario 3 numeros por pantalla y a través de condicionales debemos verificar cual es el mayor.

Código:

```
public class EjercicioDos {  
  
    public static void main(String[] args) {  
        // Declaramos input para leer los datos  
        Scanner input = new Scanner(System.in);  
  
        // Declaramos variables  
        int numUno, numDos, numTres, mayor;  
  
        // Le pedimos los numeros al usuario  
        System.out.println("Ingrese 3 numeros");  
        System.out.println("Primer numero: ");  
        numUno = Integer.parseInt(input.nextLine());  
        System.out.println("Segundo numero: ");  
        numDos = Integer.parseInt(input.nextLine());  
        System.out.println("Tercer numero: ");  
        numTres = Integer.parseInt(input.nextLine());  
  
        mayor = numUno; // Iniciliamos mayor igualandolo a numUno  
  
        if (numDos > mayor) { // Vericamos si numDos es mayor se guarda su valor  
            mayor = numDos;  
        }  
        if (numTres > mayor) { // En el caso que num3 sea mayor se guarda su valor  
            mayor = numTres;  
        }  
  
        // Mostramos al mayor  
        System.out.println("El mayor es: " + mayor);  
    }  
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----  
Building EjercitacionTpDos 1.0-SNAPSHOT  
-----[ jar ]-----  
  
--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---  
Ingrese 3 numeros  
Primer numero:  
8  
Segundo numero:  
12  
Tercer numero:  
5  
El mayor es: 12  
  
BUILD SUCCESS  
  
Total time: 7.513 s  
Finished at: 2025-07-17T20:41:01-03:00  
-----  
|
```

Ejercicio 3: Este ejercicio nos pedía que el usuario ingrese su edad y el programa calcule en que etapa de su vida esta.

Código:

```
public class EjercicioTres {  
    public static void main(String[] args) {  
        // Declaramos input para leer los datos  
        Scanner input = new Scanner(System.in);  
  
        // Declaramos variable para alojar la edad ingresada por el usuario y una variable para alojar la clasificacion de su etapa de vida.  
        int edad;  
        String etapaVida = "";  
  
        // Le pedimos al usuario que ingrese su edad.  
        System.out.println("Ingrese su edad:");  
        edad = Integer.parseInt(input.nextLine());  
  
        if (edad < 12) {  
            etapaVida = "Niño";  
        } else if (edad >= 12 && edad <= 17) {  
            etapaVida = "Adolescente";  
        } else if (edad >= 18 && edad <= 59) {  
            etapaVida = "Adulto";  
        } else if (edad >= 60) {  
            etapaVida = "Adulto mayor";  
        }  
  
        System.out.println("Eres un " + etapaVida);  
    }  
}
```

Salida por pantalla al ingresar 25:

```
-----< com.mycompany:EjercitacionTpDos >-----  
] Building EjercitacionTpDos 1.0-SNAPSHOT  
-----[ jar ]-----  
  
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---  
Ingrese su edad:  
25  
- Eres un Adulto  
-----  
BUILD SUCCESS  
-----  
Total time: 4.874 s  
Finished at: 2025-07-17T20:51:15-03:00  
-----  
|
```

Salida por pantalla al ingresar 10:

```
-----< com.mycompany:EjercitacionTpDos >-----  
] Building EjercitacionTpDos 1.0-SNAPSHOT  
-----[ jar ]-----  
  
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---  
Ingrese su edad:  
10  
- Eres un Niño  
-----  
BUILD SUCCESS  
-----  
Total time: 3.092 s  
Finished at: 2025-07-17T20:51:41-03:00  
-----  
|
```

Ejercicio 4: en este ejercicio debíamos pedirle un precio y una categoría(A, B o C) al usuario y en base a la categoría calcular el precio final.

Código:

```
public class EjercicioCuatro {  
  
    public static void main(String[] args) {  
        // Declaramos input para leer los datos  
        Scanner input = new Scanner(System.in);  
  
        // Declaramos las variables donde vamos a guardar el precio y la categoria(A, B o C)  
        int precio;  
        double precioConDescuento;  
        char categoria;  
  
        System.out.println("Ingrese el precio del producto");  
        precio = Integer.parseInt(input.nextLine());  
  
        System.out.println("Ingrese la categoria del producto (A, B o C)");  
        categoria = input.nextLine().charAt(0);  
  
        switch (categoria) {  
            case 'A':  
            case 'a':  
                precioConDescuento = precio -(precio * 0.10);  
                System.out.print("Descuento aplicado 10%");  
                System.out.println("");  
                System.out.print("Precio final: $" + precioConDescuento);  
                break;  
            case 'B':  
            case 'b':  
                precioConDescuento = precio -(precio * 0.15);  
                System.out.print("Descuento aplicado 15%");  
                System.out.println("");  
                System.out.print("Precio final: $" + precioConDescuento);  
                break;  
            case 'C':  
            case 'c':  
                precioConDescuento = precio -(precio * 0.20);  
                System.out.print("Descuento aplicado 20%");  
                System.out.println("");  
                System.out.print("Precio final: $" + precioConDescuento);  
                break;  
            default:  
                System.out.println("No ingreso una categoria valida");  
                break;  
        }  
    }  
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----
Building EjercitacionTpDos 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---
Ingrese el precio del producto
1000
Ingrese la categoria del producto (A, B o C)
b
Descuento aplicado 15%
Precio final: $850.0
-----
BUILD SUCCESS
-----
Total time: 6.592 s
Finished at: 2025-07-18T12:50:26-03:00
-----
```

Ejercicio 5: En este ejercicio debíamos pedirle al usuario números hasta que ingrese 0 y mostrarle la sumatoria de todos los números pares ingresados.

Código:

```
public class EjercicioCinco {
    public static void main(String[] args) {
        // Declaramos input para leer los datos
        Scanner input = new Scanner(System.in);

        // Declaramos num e inicializamos en 1 para entrar al menos una vez al ciclo
        int num = 1;

        // Declaramos sumaPares e inicializamos en 0 para acumular la sumatoria de los numeros pares
        int sumaPares = 0;

        while(num != 0){
            System.out.println("Ingrese un numero(0 para terminar)");
            num = Integer.parseInt(input.nextLine());

            // Si el resto de la division entre el numero y 2 nos da 0 entonces el numero es par
            if(num % 2 == 0){
                sumaPares = sumaPares + num; // Acumulamos la suma de numeros pares
            }
        }

        System.out.println("La suma de los numeros pares es: " + sumaPares);
    }
}
```


Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----
] Building EjercitacionTpDos 1.0-SNAPSHOT
-----[ jar ]-----

] --- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---
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
-----
Total time:  8.073 s
Finished at: 2025-07-18T13:00:56-03:00
-----
|
```

Ejercicio 6: En este ejercicio debíamos pedirle 10 números al usuario y contar cuantos negativos ingreso, cuantos positivos y cuantos 0

Código:

```
public class EjercicioSeis {
    public static void main(String[] args) {
        // Declaramos input para leer los datos
        Scanner input = new Scanner(System.in);

        // Declaramos e iniciliamos contadores en 0, y declaramos num para alojar el numero del usuario
        int contadorPositivos = 0, contadorNegativos = 0, contadorCeros = 0, num;

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

            // Verificamos si es igual a 0, menor a 0 o mayor a 0
            if(num > 0) {
                contadorPositivos++;
            } else if (num < 0) {
                contadorNegativos++;
            } else if (num == 0){
                contadorCeros++;
            }
        }

        System.out.println("Resultados");
        System.out.print("Positivos: " + contadorPositivos + "\n");
        System.out.print("Negativos: " + contadorNegativos + "\n");
        System.out.print("Ceros: " + contadorCeros);
    }
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----
- Building EjercitacionTpDos 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---
Ingrese el numero 1:
-5
Ingrese el numero 2:
3
Ingrese el numero 3:
0
Ingrese el numero 4:
-1
Ingrese el numero 5:
6
Ingrese el numero 6:
0
Ingrese el numero 7:
9
Ingrese el numero 8:
-3
Ingrese el numero 9:
4
Ingrese el numero 10:
-8
Resultados
Positivos: 4
Negativos: 4
Ceros: 2

BUILD SUCCESS

Total time: 22.276 s
Finished at: 2025-07-18T13:11:29-03:00
-----
```

Ejercicio 7: en este ejercicio debíamos pedirle una nota al usuario y verificar que este entre 0 y 10 en el caso que no cumpla con la condición pedirle hasta que la cumpla.

Código:

```
public class EjercicioSiete {

    public static void main(String[] args) {
        // Declaramos input para leer los datos
        Scanner input = new Scanner(System.in);
        int nota;

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

            // En el caso que la nota sea menor que 0 o mayor a 10 lanza un mensaje de error sino lanza un mensaje de éxito
            if(nota < 0 || nota > 10) {
                System.out.println("ERROR: nota invalida. Ingrese una nota entre 0 y 10 ");
            } else{
                System.out.println("Nota guardada correctamente");
            }
        }while(nota < 0 || nota > 10);
    }
}
```


Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----
Building EjercitacionTpDos 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---
Ingrese una nota(0-10)
15
ERROR: nota invalida. Ingrese una nota entre 0 y 10
Ingrese una nota(0-10)
-2
ERROR: nota invalida. Ingrese una nota entre 0 y 10
Ingrese una nota(0-10)
8
Nota guardada correctamente

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

Total time: 15.916 s
Finished at: 2025-07-18T13:20:47-03:00
-----
|
```

Ejercicio 8: en este ejercicio debíamos pedirle al usuario el precio de un producto, el porcentaje de impuesto y el porcentaje de descuento y en base a esto calcular a través de un método el precio final.

Código:

```
public class EjercicioOcho {

    public static void main(String[] args) {
        // Declaramos input para leer los datos
        Scanner input = new Scanner(System.in);

        // Declaramos la variables para guardar los datos ingresados por el usuario
        double precio, impuesto, descuento;

        // Le pedimos al usuario el precio del producto
        System.out.println("Ingrese el precio base del producto:");
        precio = (double)input.nextDouble();

        // Le pedimos al usuario el porcentaje de impuesto
        System.out.println("Ingrese el impuesto en porcentaje (Ejemplo: 10 para 10%):");
        impuesto = (double)input.nextDouble();

        // Le pedimos al usuario el porcentaje de descuento
        System.out.println("Ingrese el descuento en porcentaje (Ejemplo: 5 para 5%):");
        descuento = (double)input.nextDouble();

        // Alojamos el retorno del metodo en una variable
        double precioFinal = calcularPrecioFinal(precio, impuesto, descuento);

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

    /**
     * Calcula el precio final de un producto aplicando un impuesto y un descuento.
     *
     * @param precioBase El precio base del producto (sin impuesto ni descuento).
     * @param impuesto El porcentaje de impuesto a aplicar (por ejemplo, 21 para 21%).
     * @param descuento El porcentaje de descuento a aplicar (por ejemplo, 15 para 15%).
     * @return El precio final del producto con el impuesto sumado y el descuento restado.
     */
    public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento){
        double precioFinal = precioBase + ((precioBase * impuesto)/100) - ((precioBase * descuento)/100);
        return precioFinal;
    }
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----  
] Building EjercitacionTpDos 1.0-SNAPSHOT  
-----[ jar ]-----  
  
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---  
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  
-----  
Total time: 7.929 s  
Finished at: 2025-07-18T13:48:16-03:00  
-----  
|
```

Ejercicio 9: en este ejercicio debíamos realizar dos métodos uno para calcular el costo del envío de un producto según su peso y su zona, y el otro para calcular el costo total de la compra según el costo de envío y el precio del producto. Desde el main debíamos pedirle al usuario la zona de envío, el peso del paquete y el precio del producto.

Metodo calcular costo envio:

```
/**  
 * Calcula el costo de envío en función del peso y la zona de destino.  
 *  
 * @param peso Peso del paquete en kilogramos.  
 * @param zona Zona de envío: puede ser "Nacional" o "Internacional" (no  
 * distingue mayúsculas).  
 * @return El costo de envío calculado: $5 por kilo para envíos nacionales,  
 * $10 por kilo para internacionales.  
 */  
public static double calcularCostoEnvio(double peso, String zona){  
    double costoEnvio = 0;  
    if(zona.equalsIgnoreCase("Nacional")){  
        costoEnvio = peso * 5;  
    } else if (zona.equalsIgnoreCase("Internacional")){  
        costoEnvio = peso * 10;  
    }  
  
    return costoEnvio;  
}
```

Metodo calcularTotalCompra:

```
/**
 * Calcula el total de la compra sumando el precio del producto y el costo
 * de envío.
 *
 * @param precioProducto Precio base del producto sin costos adicionales.
 * @param costoEnvio Costo de envío previamente calculado.
 * @return El monto total a pagar por la compra.
 */
public static double calcularTotalCompra(double precioProducto, double costoEnvio){
    return precioProducto + costoEnvio;
}
```

Main:

```
public static void main(String[] args) {
    // Declaramos input para leer los datos
    Scanner input = new Scanner(System.in);

    // Declaramos variables para guardar los datos que pedimos al usuario
    double peso, precioProducto;
    String zona;

    // Le solicitamos el precio del producto al usuario
    System.out.println("Ingrese el precio del producto");
    precioProducto = input.nextDouble();

    // Le solicitamos el peso del paquete al usuario
    System.out.println("Ingrese el peso del paquete");
    peso = input.nextDouble();
    input.nextLine(); // Consumimos el Enter que quedó pendiente

    // Le solicitamos la zona de envío al usuario
    System.out.println("Ingrese la zona de envío(Nacional/Internacional)");
    zona = input.nextLine();

    // Llamamos al metodo para calcular costo de envío y alojamos el retorno en una variable
    double costoEnvio = calcularCostoEnvio(peso, zona);

    // Llamamos al metodo para calcular el precio total de la compra y alojamos el retorno en una variable
    double precioFinal = calcularTotalCompra(precioProducto, costoEnvio);

    // Mostramos los resultados por pantalla
    System.out.println("El costo de envío es: " + costoEnvio);
    System.out.println("El total a pagar es: " + precioFinal);
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----
Building EjercitacionTpDos 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---
Ingrese el precio 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

Total time: 12.384 s
Finished at: 2025-07-18T14:10:22-03:00
-----
```

Ejercicio 10: en este ejercicio debíamos realizar un método que calcule el stock de un producto en base a un stock actual, una cantidad vendida y una cantidad recibida ingresados por el usuario.

Metodo actualizar stock:

```
/**
 * Actualiza el stock disponible de un producto según la cantidad vendida y
 * la cantidad recibida.
 *
 * @param stockActual Cantidad actual en stock antes de la operación.
 * @param cantidadVendida Cantidad de unidades vendidas (se restan del
 * stock).
 * @param cantidadRecibida Cantidad de unidades recibidas o repuestas (se
 * suman al stock).
 * @return El nuevo stock disponible después de aplicar ventas y
 * reposiciones.
 */
public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida){
    int nuevoStock = (stockActual - cantidadVendida) + cantidadRecibida;
    return nuevoStock;
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----
] Building EjercitacionTpDos 1.0-SNAPSHOT
-----[ jar ]-----

] --- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---
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
-----
Total time: 8.441 s
Finished at: 2025-07-18T14:21:04-03:00
-----
```

Ejercicio 11: en este ejercicio debíamos declarar una variable global para almacenar un descuento especial, luego realizar un método para calcular el precio final con este descuento, y mostrar el precio final y el descuento aplicado.

Método calcular Descuento Especial:

```
/**
 * Calcula y muestra el descuento especial aplicado a un producto, junto con
 * el precio final.
 *
 * Usa la constante global DESCUENTO_ESPECIAL (10%) para calcular el
 * descuento. Imprime por pantalla: - El monto del descuento aplicado. - El
 * precio final del producto después de aplicar el descuento.
 *
 * @param precio El precio original del producto antes de aplicar el
 * descuento.
 */
public static void calcularDescuentoEspecial(double precio){
    double descuentoAplicado = precio * DESCUENTO_ESPECIAL;
    double precioFinal = precio - descuentoAplicado;

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

Main:

```
public class EjercicioOnce {
    // Declaramos la variable global
    public static final double DESCUENTO_ESPECIAL = 0.10;

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

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

        calcularDescuentoEspecial(precio);
    }
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----
[ Building EjercitacionTpDos 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---
Ingrese el precio del producto
200
El descuento aplicado es de: 20.0
El precio final del producto es: 180.0

BUILD SUCCESS

Total time: 3.446 s
Finished at: 2025-07-18T14:32:05-03:00
|
```


Ejercicio 12: en este ejercicio debíamos declarar un array de precios, modificar uno de los precios y mostrar los precios originales y los modificados.

Código:

```
public class EjercicioDoce {  
  
    public static void main(String[] args) {  
        // Declaramos el array de precios y le asignamos valores a cada posicion  
        double[] precioProducto = {199.99, 299.5, 149.75, 399.0, 89.99};  
  
        // Mostramos los precios originales recorriendo el array con un for-each  
        System.out.println("Precios originales: ");  
        for(double precio : precioProducto){  
            System.out.println("Precio: " + precio);  
        }  
  
        // Modificamos el precio de la posicion 2  
        precioProducto[2] = 129.99;  
  
        // Mostramos los precios modificados recorriendo el array con un for-each  
        System.out.println("Precios modificados: ");  
        for(double precio : precioProducto){  
            System.out.println("Precio: " + precio);  
        }  
    }  
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----  
] Building EjercitacionTpDos 1.0-SNAPSHOT  
-----[ jar ]-----  
  
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---  
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: 1.362 s  
Finished at: 2025-07-18T15:08:58-03:00  
-----  
|
```


Ejercicio Trece: en este ejercicio debíamos declarar un array double con precios, luego modificar uno de los precios, y mostrar los precios originales y los modificados con una función recursiva.

Funcion mostrar precios recursivo:

```
/**
 * Muestra los precios de un arreglo de precios de manera recursiva.
 *
 * Esta función imprime los precios contenidos en el arreglo 'precios', uno
 * por uno, en la consola. La recursividad se utiliza para iterar sobre cada
 * elemento del arreglo.
 *
 * @param precios Un arreglo de tipo 'double' que contiene los precios de
 * los productos.
 * @param indice Un índice entero que indica la posición actual en el
 * arreglo. Debe ser 0 cuando se llama a la función por primera vez.
 */
public static void mostrarPreciosRecursivo(double[] precios, int indice) {
    // Si el índice es menor que la longitud del arreglo, continúa el proceso.
    if (indice < precios.length) {
        // Imprime el precio en la posición actual del arreglo.
        System.out.println("Precio: $" + precios[indice]);
        // Llama recursivamente a la función para imprimir el siguiente precio.
        mostrarPreciosRecursivo(precios, indice + 1);
    }
}
```

Main:

```
public static void main(String[] args) {
    // Declaramos e inicializamos el array
    double[] precios = {199.99, 299.5, 149.75, 399.0, 89.99};

    // Mostramos los precios originales
    System.out.println("Precios originales:");
    mostrarPreciosRecursivo(precios, 0);

    // Modificamos el precio de la posición 2
    precios[2] = 129.99;

    // Mostramos los precios modificados
    System.out.println("Precios modificados:");
    mostrarPreciosRecursivo(precios, 0);
}
```

Salida por pantalla:

```
-----< com.mycompany:EjercitacionTpDos >-----  
Building EjercitacionTpDos 1.0-SNAPSHOT  
-----[ jar ]-----  
  
--- exec-maven-plugin:3.0.0:exec (default-cli) @ EjercitacionTpDos ---  
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: 1.430 s  
Finished at: 2025-07-18T15:18:02-03:00  
-----
```