

Trabajo práctico N° 2

“Programación Estructurada”

Alumnos: Alemis, Juan Cruz – juancruzalemis@gmail.com

Materia: Programacion II

Estructuras Condicionales:

1. Verificación de Año Bisiesto.

```
Source History
1
2 package ejerciciouno;
3 import java.util.Scanner;
4
5 public class EjercicioUno {
6
7
8     public static void main(String[] args) {
9         Scanner input = new Scanner(System.in);
10        int anio;
11
12        System.out.println("Ingrese un año para verificar si es bisiesto");
13        anio = Integer.parseInt(input.nextLine());
14
15        if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)){
16            System.out.println("El año " + anio + " es bisiesto");
17        } else {
18            System.out.println("El año " + anio + " no es bisiesto");
19        }
20    }
21 }
22
23
24
```

Output - verificaciónAñoBisiesto (run)

```
run:
Ingrese un año para verificar si es bisiesto
2024
El año 2024 es bisiesto
BUILD SUCCESSFUL (total time: 14 seconds)
```

Output - verificaciónAñoBisiesto (run)

```
run:
Ingrese un año para verificar si es bisiesto
1900
El año 1900 no es bisiesto
BUILD SUCCESSFUL (total time: 6 seconds)
```

2. Determinar el Mayor de Tres Números

```
Source History
1
2 package determinarmayortresnumeros;
3
4 import java.util.Scanner;
5
6
7 public class DeterminarMayorTresNumeros {
8
9
10     public static void main(String[] args) {
11         Scanner input = new Scanner(System.in);
12
13         int numUno, numDos, numTres, mayor;
14
15         System.out.println("Ingrese 3 números");
16         System.out.println("Primer número: ");
17         numUno = Integer.parseInt(input.nextLine());
18         System.out.println("Segundo número: ");
19         numDos = Integer.parseInt(input.nextLine());
20         System.out.println("Tercer número: ");
21         numTres = Integer.parseInt(input.nextLine());
22
23         mayor = numUno;
24
25         if (numDos > mayor) {
26             mayor = numDos;
27         }
28         if (numTres > mayor) {
29             mayor = numTres;
30         }
31         System.out.println("El mayor es: " + mayor);
32     }
33
34 }
35
```

Output - determinarMayorTresNumeros (run)

```
run:
Ingrese 3 números
Primer número:
8
Segundo número:
12
Tercer número:
5
El mayor es: 12
BUILD SUCCESSFUL (total time: 21 seconds)
```

3. Clasificación de Edad

```
Source History
1 package clasificaciónEdad;
2
3
4 import java.util.Scanner;
5
6
7 public class ClasificaciónEdad {
8
9
10 public static void main(String[] args) {
11     Scanner input = new Scanner(System.in);
12
13     int edad;
14     String etapaVida = "";
15
16     System.out.println("Ingrese su edad: ");
17     edad = Integer.parseInt(input.nextLine());
18
19
20     if (edad < 12){
21         etapaVida = "Niño";
22     } else if (edad >= 12 && edad <= 17){
23         etapaVida = "Adolescente";
24     } else if (edad >= 18 && edad <= 59){
25         etapaVida = "Adulto";
26     } else if (edad >= 60) {
27         etapaVida = "Adulto mayor";
28     }
29
30     System.out.println("Eres un " + etapaVida);
31 }
32
33
34 }
```


Output - clasificaciónEdad (run)

```
run:
Ingrese su edad:
25
Eres un Adulto
BUILD SUCCESSFUL (total time: 13 seconds)
```

Output - clasificaciónEdad (run)

```
run:
Ingrese su edad:
10
Eres un Niño
BUILD SUCCESSFUL (total time: 6 seconds)
```

4. Calculadora de Descuento según categoría

```
Source History 
4 public static void main(String[] args) {
5     Scanner input = new Scanner(System.in);
6     int precio;
7     double precioConDescuento;
8     char categoria;
9     System.out.println("Ingrese el precio del producto");
10    precio = Integer.parseInt(input.nextLine());
11
12    System.out.println("Ingrese la categoria del prodcto (A, B o C)");
13    categoria = input.nextLine().charAt(0);
14    switch(categoria){
15        case 'A':
16        case 'a':
17            precioConDescuento = precio - (precio * 0.10);
18            System.out.println("Descuento aplicada 10%");
19            System.out.println("");
20            System.out.println("Precio final: $" + precioConDescuento);
21            break;
22        case 'B':
23        case 'b':
24            precioConDescuento = precio - (precio * 0.15);
25            System.out.println("Descuento aplicada 15%");
26            System.out.println("");
27            System.out.println("Precio final: $" + precioConDescuento);
28            break;
29        case 'C':
30        case 'c':
31            precioConDescuento = precio - (precio * 0.20);
32            System.out.println("Descuento aplicada 20%");
33            System.out.println("");
34            System.out.println("Precio final: $" + precioConDescuento);
35            break;
36
37        default:
38            System.out.println("No ingreso una categoria valida");
39            break;
40    }
41 }
42 }
```

```
Output - calculadoraDescuentoCategoria (run)
run:
Ingrese el precio del producto
100
Ingrese la categoria del prodcto (A, B o C)
B
Descuento aplicada 15%


Precio final: $85.0
BUILD SUCCESSFUL (total time: 9 seconds)
```

5. Suma de Números Pares (while)




```
Source History
1
2 package sumanumerospares;
3
4 import java.util.Scanner;
5
6
7 public class SumaNumerosPares {
8
9
10 public static void main(String[] args) {
11     Scanner input = new Scanner(System.in);
12
13     int num = 1;
14
15     int sumaPares = 0;
16
17     while(num != 0){
18         System.out.println("Ingrese un numero (0 para terminar)");
19         num = Integer.parseInt(input.nextLine());
20
21         if(num % 2 == 0){
22             sumaPares = sumaPares + num;
23         }
24     }
25
26     System.out.println("La suma de los numeros pares es: " + sumaPares);
27
28 }
29
30
31 }
32
```

```
Output - sumaNumerosPares (run)
run:
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 SUCCESSFUL (total time: 16 seconds)
```

6. Contador de Positivos, Negativos y Ceros (for)

```
Source History 
1
2 package contadorpositivosnegativosceros;
3
4 import java.util.Scanner;
5
6
7 public class ContadorPositivosNegativosCeros {
8
9
10
11     public static void main(String[] args) {
12         Scanner input = new Scanner(System.in);
13
14         int contadorPositivos = 0, contadorNegativos = 0, contadorCeros = 0, num;
15
16         for (int i = 1; i <= 10; i++){
17             System.out.println("Ingrese el numero " + i + ": ");
18             num = Integer.parseInt(input.nextLine());
19
20             if(num > 0) {
21                 contadorPositivos++;
22             } else if (num < 0) {
23                 contadorNegativos++;
24             } else if (num == 0) {
25                 contadorCeros++;
26             }
27         }
28
29         System.out.println("Resultados");
30         System.out.println("Positivos: " + contadorPositivos + "\n");
31         System.out.println("Negativos: " + contadorNegativos + "\n");
32         System.out.println("Ceros: " + contadorCeros);
33     }
34 }
35
36
```

Output - contadorPositivosNegativosCeros (run)

```
 run:
 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 SUCCESSFUL (total time: 29 seconds)
```

7. Validación de Nota entre 0 y 10 (do-while)

```
Source History 
1
2 package validaciónnota;
3
4 import java.util.Scanner;
5
6
7 public class ValidaciónNota {
8
9
10
11     public static void main(String[] args) {
12         Scanner input = new Scanner(System.in);
13         int nota;
14
15         do{
16             System.out.println("Ingrese una nota (0-10)");
17             nota = Integer.parseInt(input.nextLine());
18
19             if(nota < 0 || nota > 10){
20                 System.out.println("ERROR: nota invalida. Ingrese una nota entre 0 y 10");
21             } else{
22                 System.out.println("Nota guardada correctamente");
23             }
24
25         }while(nota < 0 || nota > 10);
26
27     }
28
29 }
30
31
```

```
Output - validaciónNota (run)
run:
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 SUCCESSFUL (total time: 11 seconds)
|
```


Funciones:

8. Cálculo del Precio Final con impuesto y descuento

```
Source History
1 package calculopreciofinal;
2
3
4 import java.util.Scanner;
5
6
7 public class CalculoPrecioFinal {
8
9
10 public static void main(String[] args) {
11     Scanner input = new Scanner(System.in);
12
13     double precio, impuesto, descuento;
14
15     System.out.println("Ingrese el precio del producto");
16     precio = (double)input.nextDouble();
17
18     System.out.println("Ingrese el impuesto en porcentaje (Ej: 10 para 10%): ");
19     impuesto = (double)input.nextDouble();
20
21     System.out.println("Ingrese el impuesto en porcentaje (Ej: 5 para 5%): ");
22     descuento = (double)input.nextDouble();
23
24     double precioFinal = calcularPrecioFinal(precio, impuesto, descuento);
25
26     System.out.println("El precio final del producto es: $" + precioFinal);
27 }
28
29
30
31 public static double calcularPrecioFinal(double precioBase, double impuesto, double descuento){
32     double precioFinal = precioBase + ((precioBase * impuesto)/100) - ((precioBase * descuento)/100);
33     return precioFinal;
34 }
35
36
37 }
```

```
Output - calculoPrecioFinal (run)
run:
Ingrese el precio del producto
100
Ingrese el impuesto en porcentaje (Ej: 10 para 10%):
10
Ingrese el impuesto en porcentaje (Ej: 5 para 5%):
5
El precio final del producto es: $105.0
BUILD SUCCESSFUL (total time: 8 seconds)
```

9. Composición de funciones para calcular costo de envío y total de compra

```
Source History [Icons]

1
2 package composicioncostocompra;
3
4 import java.util.Scanner;
5
6 public class ComposicionCostoCompra {
7
8     public static void main(String[] args) {
9         Scanner input = new Scanner(System.in);
10
11         double peso, precioProducto;
12         String zona;
13
14         System.out.println("Ingrese el precio del producto:");
15         precioProducto = input.nextDouble();
16
17         System.out.println("Ingrese el peso del paquete:");
18         peso = input.nextDouble();
19         input.nextLine(); // Limpia el buffer
20
21         System.out.println("Ingrese la zona de envio (Nacional/Internacional):");
22         zona = input.nextLine();
23
24         double costoEnvio = calcularCostoEnvio(peso, zona);
25         double precioFinal = calcularTotalCompra(precioProducto, costoEnvio);
26
27         System.out.println("El costo de envio es: $" + costoEnvio);
28         System.out.println("El total a pagar es: $" + precioFinal);
29
30         input.close();
31     }
32
33     // Método para calcular el costo de envio
34     public static double calcularCostoEnvio(double peso, String zona) {
35         double costoEnvio = 0;
36         if (zona.equalsIgnoreCase("Nacional")) {
37             costoEnvio = peso * 5;
38         } else if (zona.equalsIgnoreCase("Internacional")) {
39             costoEnvio = peso * 10;
40         } else {
41             System.out.println("Zona no válida. Se asumirá costo de envio 0.");
42         }
43         return costoEnvio;
44     }
45
46     // Método para calcular el total de la compra
47     public static double calcularTotalCompra(double precioProducto, double costoEnvio) {
48         return precioProducto + costoEnvio;
49     }
50 }
51
52
```

```
Output - ComposicionCostoCompra. (run)

run:
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 SUCCESSFUL (total time: 19 seconds)
```


10. Actualización de stock a partir de venta y recepción de productos

```
Source History
1 import java.util.Scanner;
2
3 public class ActualizarStock {
4
5     public static int actualizarStock(int stockActual, int cantidadVendida, int cantidadRecibida) {
6         int nuevoStock = (stockActual - cantidadVendida) + cantidadRecibida;
7         return nuevoStock;
8     }
9
10    public static void main(String[] args) {
11        Scanner scanner = new Scanner(System.in);
12
13        System.out.print("Ingrese el stock actual: ");
14        int stockActual = scanner.nextInt();
15
16        System.out.print("Ingrese la cantidad vendida: ");
17        int cantidadVendida = scanner.nextInt();
18
19        System.out.print("Ingrese la cantidad recibida: ");
20        int cantidadRecibida = scanner.nextInt();
21
22        int resultado = actualizarStock(stockActual, cantidadVendida, cantidadRecibida);
23        System.out.println("El nuevo stock es: " + resultado);
24    }
25 }
26
```

```
Output - actualizarStock (run)

run:
Ingrese el stock actual: 50
Ingrese la cantidad vendida: 20
Ingrese la cantidad recibida: 30
El nuevo stock es: 60
BUILD SUCCESSFUL (total time: 7 seconds)
```

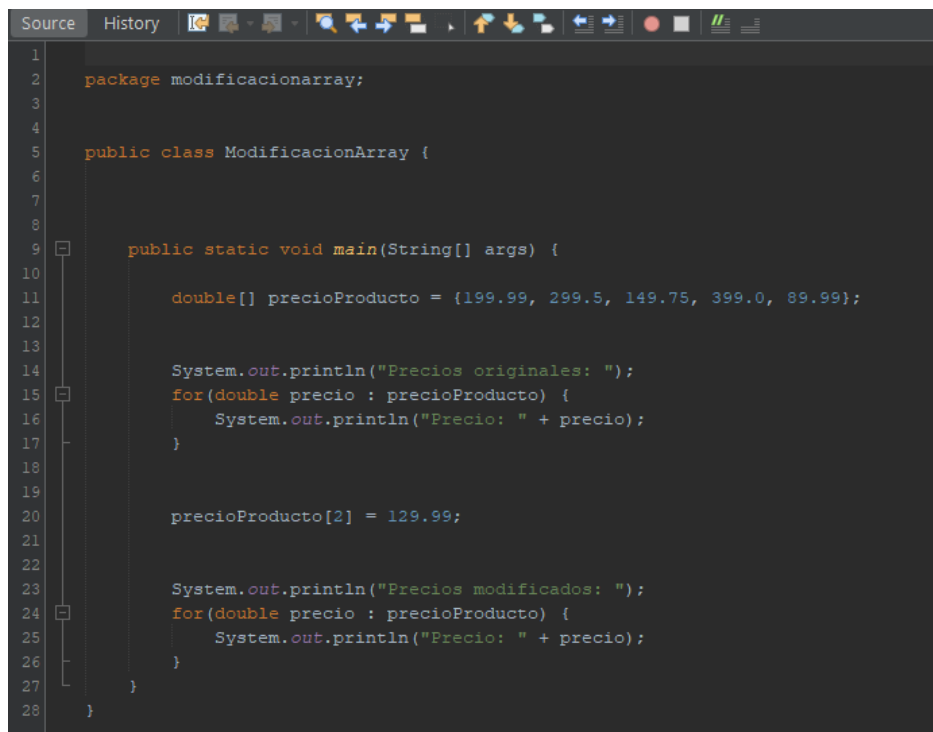
11. Cálculo de descuento especial usando variable global

```
Source History 
1  import java.util.Scanner;
2
3  public class calculoDescuentoEspecial{
4
5      public static final double DESCUENTO_ESPECIAL = 0.10;
6
7      public static void main(String[] args) {
8          Scanner input = new Scanner(System.in);
9
10         System.out.println("Ingrese el precio del producto");
11         double precio = input.nextDouble();
12
13         calcularDescuentoEspecial(precio);
14     }
15
16     public static void calcularDescuentoEspecial(double precio) {
17         double descuentoAplicado = precio * DESCUENTO_ESPECIAL;
18         double precioFinal = precio - descuentoAplicado;
19
20         System.out.println("El descuento aplicado es de: " + descuentoAplicado);
21         System.out.println("El precio final del producto es: " + precioFinal);
22     }
23 }
24
25
```

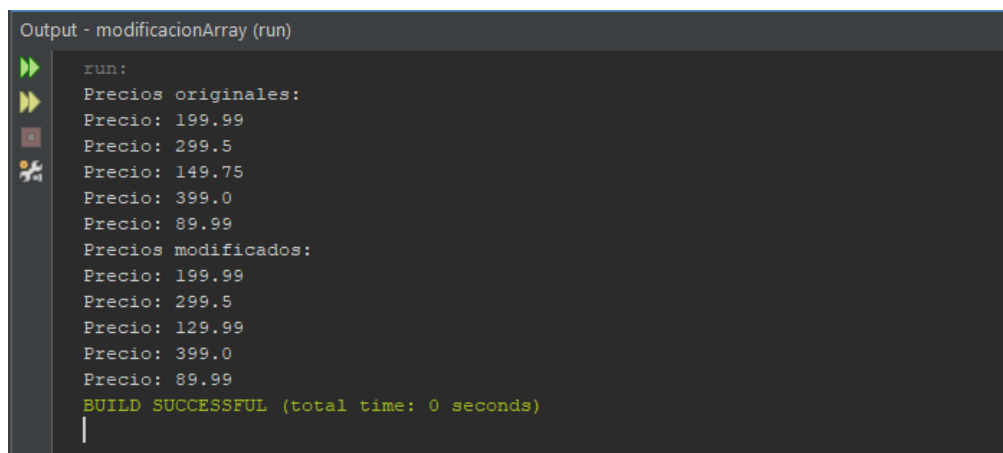
```
Output - calculoDescuentoEspecial (run)
run:
Ingrese el precio del producto
200
El descuento aplicado es de: 20.0
El precio final del producto es: 180.0
BUILD SUCCESSFUL (total time: 6 seconds)
```

Arrays y Recursividad:

12. Modificación de un array de precios y visualización de resultados



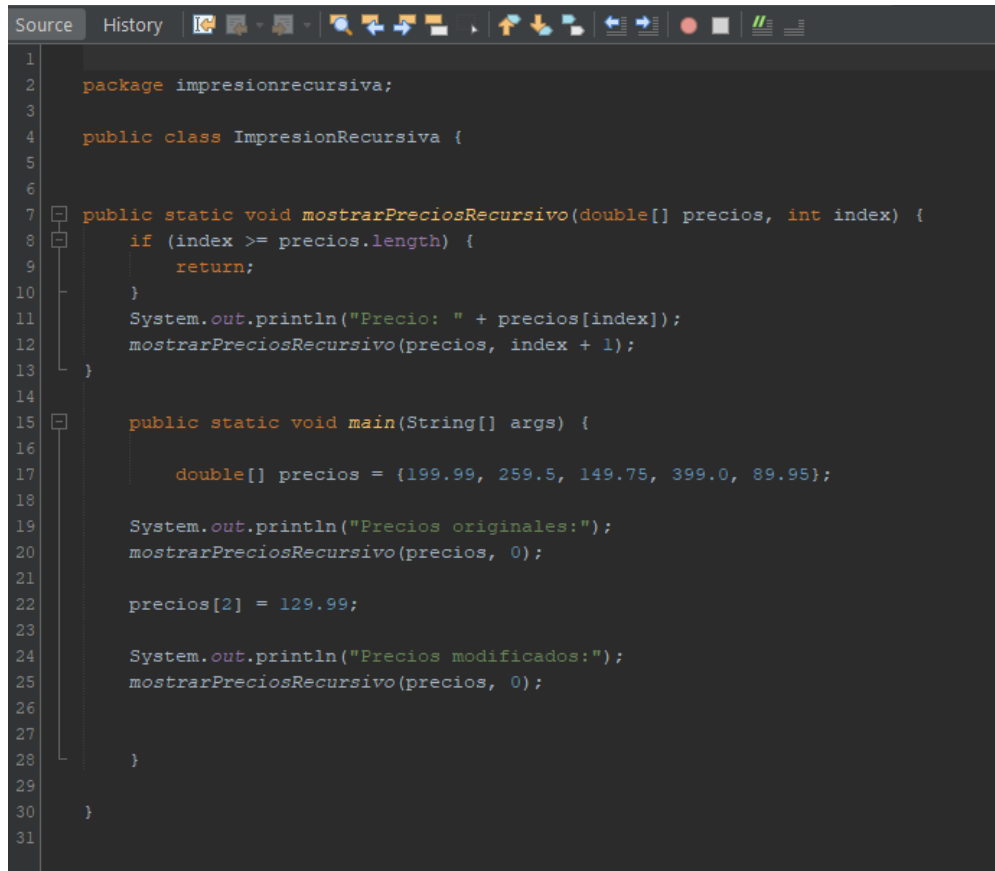
```
1
2 package modificacionarray;
3
4
5 public class ModificacionArray {
6
7
8
9     public static void main(String[] args) {
10
11         double[] precioProducto = {199.99, 299.5, 149.75, 399.0, 89.99};
12
13
14         System.out.println("Precios originales: ");
15         for(double precio : precioProducto) {
16             System.out.println("Precio: " + precio);
17         }
18
19
20         precioProducto[2] = 129.99;
21
22
23         System.out.println("Precios modificados: ");
24         for(double precio : precioProducto) {
25             System.out.println("Precio: " + precio);
26         }
27     }
28 }
```



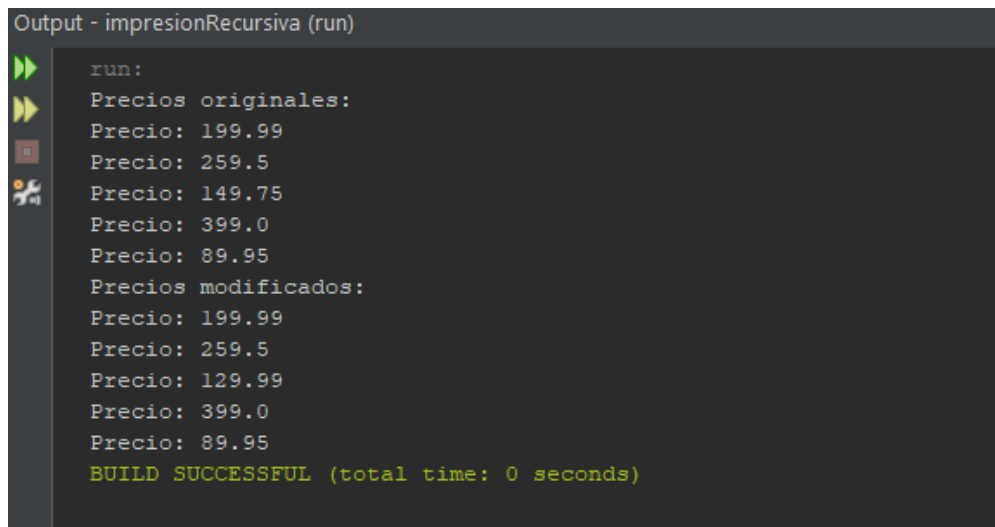
```
Output - modificacionArray (run)

run:
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 SUCCESSFUL (total time: 0 seconds)
```

13. Impresión recursiva de arrays antes y después de modificar un elemento



```
1 package impresionrecursiva;
2
3 public class ImpresionRecursiva {
4
5
6
7     public static void mostrarPreciosRecursivo(double[] precios, int index) {
8         if (index >= precios.length) {
9             return;
10        }
11        System.out.println("Precio: " + precios[index]);
12        mostrarPreciosRecursivo(precios, index + 1);
13    }
14
15    public static void main(String[] args) {
16
17        double[] precios = {199.99, 259.5, 149.75, 399.0, 89.95};
18
19        System.out.println("Precios originales:");
20        mostrarPreciosRecursivo(precios, 0);
21
22        precios[2] = 129.99;
23
24        System.out.println("Precios modificados:");
25        mostrarPreciosRecursivo(precios, 0);
26
27
28    }
29
30 }
31
```



Output - impresionRecursiva (run)

```
run:
Precios originales:
Precio: 199.99
Precio: 259.5
Precio: 149.75
Precio: 399.0
Precio: 89.95
Precios modificados:
Precio: 199.99
Precio: 259.5
Precio: 129.99
Precio: 399.0
Precio: 89.95
BUILD SUCCESSFUL (total time: 0 seconds)
```