PRACTICAS

Porter Cazares Jesus

FUNDAMENTOS DE LA PROGRAMACION 2203275018

Practica N1:

Características de un Algoritmo

1. **Finito**: Un algoritmo debe tener un número finito de pasos.
2. **Definido**: Cada paso debe estar claramente especificado y ser preciso.
3. **Entrada**: Un algoritmo recibe cero o más entradas.
4. **Salida**: Debe producir al menos una salida, que es el resultado del algoritmo.
5. **Eficiencia**: Un algoritmo debe ser eficiente en cuanto a tiempo y recursos.
6. **Correctitud**: Debe proporcionar la solución correcta para todas las entradas posibles.

Metodología de Resolución de Problemas

1. **Comprensión del problema**: Entender claramente qué se necesita resolver.
2. **Análisis**: Identificar los datos de entrada y la salida esperada.
3. **Diseño del algoritmo**: Planificar una secuencia de pasos lógicos para resolver el problema.
4. **Implementación**: Escribir el algoritmo en un lenguaje de programación.
5. **Prueba y depuración**: Verificar que el algoritmo funcione correctamente y corregir errores.

Investigación sobre Algoritmos

Un **algoritmo** es un conjunto finito de pasos o instrucciones bien definidas, ordenadas y precisas, diseñadas para resolver un problema o realizar una tarea específica. La importancia de los algoritmos radica en que permiten **automatizar procesos**, **optimizar recursos** y **garantizar la precisión** en la solución de problemas complejos. Además, son fundamentales en la programación y en la informática, ya que permiten estructurar y resolver problemas de manera sistemática.

Ejercicio 1:

Descripción del ejercicio 1:
Deseamos construir un muro de longitud Xmetros y altura Ymetros, utilizando ladrillos o bloques de tamaño desconocido. También se consideran espesores de junta horizontal y vertical, y un número Nde castillos con dimensiones específicas.

Pasos para resolverlo:

1. **Datos de entrada:**
    - Longitud del muro X(en metros).
    - Altura del muro Y(en metros).
    - Longitud y altura del ladrillo o bloque (que se desconoce).
    - Espesor de la junta horizontal y vertical.
    - Número de castillos Ny sus dimensiones.
2. **Cálculos:**
    - Calcular el área total del muro: Área del muro $= X \times Y$.
    - Calcular el área del ladrillo o bloque incluyendo el espesor de la junta.
    - Dividir el área total del muro entre el área del ladrillo para obtener el número de ladrillos necesarios.
    - Considerar los castillos y su impacto en la cantidad total de ladrillos.
3. **Salida:**
    - Número total de ladrillos necesarios.
    - Considerar también la cantidad de material adicional por los castillos.

Algoritmo (en pse int):

Inicio

- Obtener el área de la barda como la multiplicación de la longitud por la altura.
- Conocer la altura del ladrillo.
- Conocer la longitud del ladrillo.
- Conocer el espesor de la junta horizontal.
- Conocer la altura de la junta vertical.
- Calcular el área del ladrillo, incluyendo los valores de la junta.
- Calcular el número de ladrillos necesarios, dividiendo el área de la barda entre el área del ladrillo.
- Mostrar el número de ladrillos obtenido.
  Fin

Ejercicio 2:

Descripción del Ejercicio 2

Calcular la cantidad de materiales necesarios (cemento, arena, agua y grava) para construir una loza de concreto.

Datos de Entrada:

1. **Longitud (X)**: La longitud de la loza en metros.
2. **Ancho (Y)**: El ancho de la loza en metros.
3. **Espesor (N)**: El espesor de la loza en metros.

Cálculos Necesarios:

1. **Volumen de la loza**: Se calcula multiplicando la longitud por el ancho y por el espesor (Volumen = X * Y * N).
2. **Proporciones de la mezcla**: Dependiendo del tipo de mezcla de concreto, se utilizan ciertas proporciones (por ejemplo, 1:2:3 para cemento, arena y grava, y una cantidad de agua específica).
3. **Cantidad de cada material**: Basado en el volumen total y en las proporciones de la mezcla, se calcula la cantidad de cemento, arena, agua y grava.

Salida Esperada:

La cantidad total de cemento, arena, agua y grava necesaria para la loza.

Algoritmo:

Inicio

- Obtener la longitud de la loza.
- Obtener el ancho de la loza.
- Obtener el espesor de la loza.
- Calcular el volumen de la loza, multiplicando la longitud por el ancho y por el espesor.
- Calcular la cantidad de cemento, arena, grava y agua según las proporciones de la mezcla.
- Mostrar la cantidad de cada material calculado.
  Fin

PRACTICA N2: Introduccion a Pse Int y sofware de base

Primero que nada debemos saber como instalar Pse Int

La instalación de PSeInt es bastante sencilla. Te voy a guiar paso a paso:

1. **Descargar el instalador**: Ve al sitio oficial de PSeInt, que es pseint.com, y descarga la versión más reciente del instalador para tu sistema operativo (Windows, por ejemplo).
2. **Ejecutar el instalador**: Una vez descargado, abre el archivo y sigue las instrucciones del asistente de instalación. Usualmente es cuestión de aceptar los términos y elegir la carpeta de instalación.
3. **Finalizar la instalación**: Una vez que el instalador termine, puedes cerrar el asistente y, si quieres, puedes crear un acceso directo en el escritorio para mayor comodidad.
4. **Verificar la instalación**: Abre PSeInt y asegúrate de que funcione correctamente. Deberías ver la interfaz principal, lista para empezar a crear algoritmos.

Y con eso, ya estarías listo para comenzar a usar PSeInt.

Podemos empezar por entender qué es PSeInt: es una herramienta que facilita la creación de algoritmos en pseudocódigo y diagramas de flujo, ayudando a visualizar y estructurar la lógica de un programa.

Podemos enfocarnos en los puntos clave, como las reglas de diseño de diagramas de flujo, la forma de representar decisiones, procesos y datos, y también cómo utilizar PSeInt para traducir esos diagramas en pseudocódigo.

Vamos a comenzar con lo básico. En PSeInt, primero definimos las variables y luego estructuramos el algoritmo usando palabras clave como Inicio, Obtener, Calcular, Mostrar y Fin. Además, los diagramas de flujo nos ayudan a representar visualmente el algoritmo, con símbolos como óvalos para el inicio y el fin, rectángulos para procesos, rombos para decisiones, y paralelogramos para la entrada o salida de datos.

Podemos practicar diseñando un diagrama de flujo sencillo y luego traduciendo ese diagrama a pseudocódigo en PSeInt. Pero para entender que son los diagramas de flujo también ocupamos algunas reglas claras para no equivocarnos en hacerlos.

Algunas reglas fundamentales para el diseño de diagramas de flujo son:

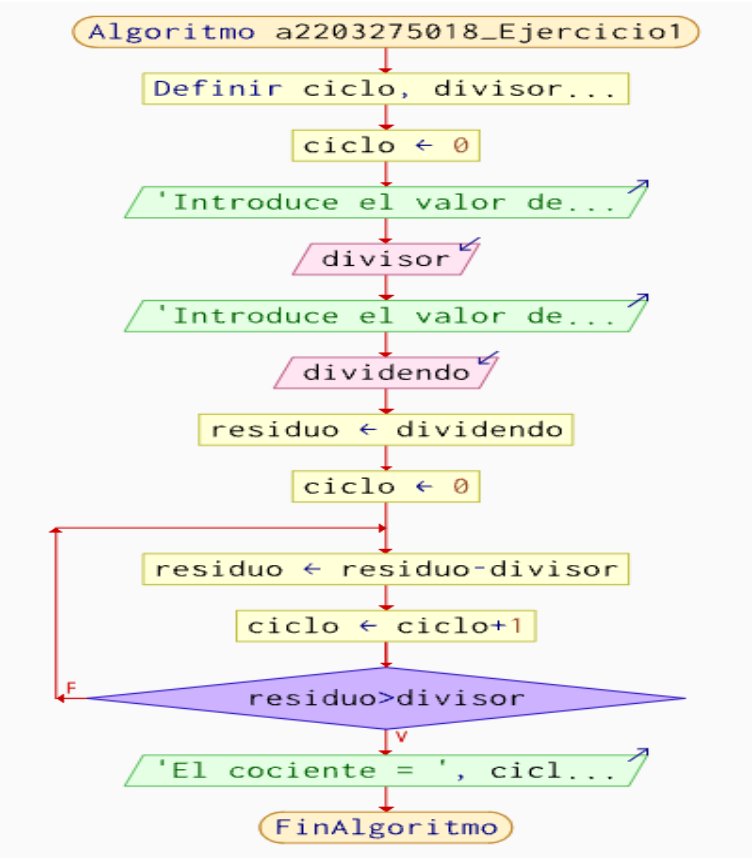
1. **Inicio y fin**: Siempre debes representar el inicio y el fin del algoritmo con un símbolo ovalado.
2. **Dirección del flujo**: El flujo de control se representa con flechas que indican el orden en que se deben seguir los pasos.
3. **Simplicidad**: Mantén el diagrama lo más simple posible, evitando cruces de flechas y asegurando que cada símbolo tenga un propósito claro.
4. **Consistencia de símbolos**: Usa los símbolos estándar: óvalos para inicio y fin, rectángulos para procesos, rombos para decisiones y paralelogramos para entrada o salida de datos.
5. **Orden lógico**: Asegúrate de que las decisiones y los procesos sigan un orden lógico y claro, evitando ambigüedades.
6. **Evitar bucles complejos**: Si necesitas un bucle, intenta representarlo de forma clara, con una entrada y salida bien definidas, para que no se vuelva confuso.

Con estas reglas, tus diagramas de flujo serán mucho más claros y fáciles de seguir.

A continuación veremos algunas actividades con diagramas de flujo y los convertiremos en pseudocodigo con su respectiva salida:

Actividad:

Ejercicio 1



Este diagrama de flujo lo convertiremos en pseudocodigo con pse int y con su respectiva salida:

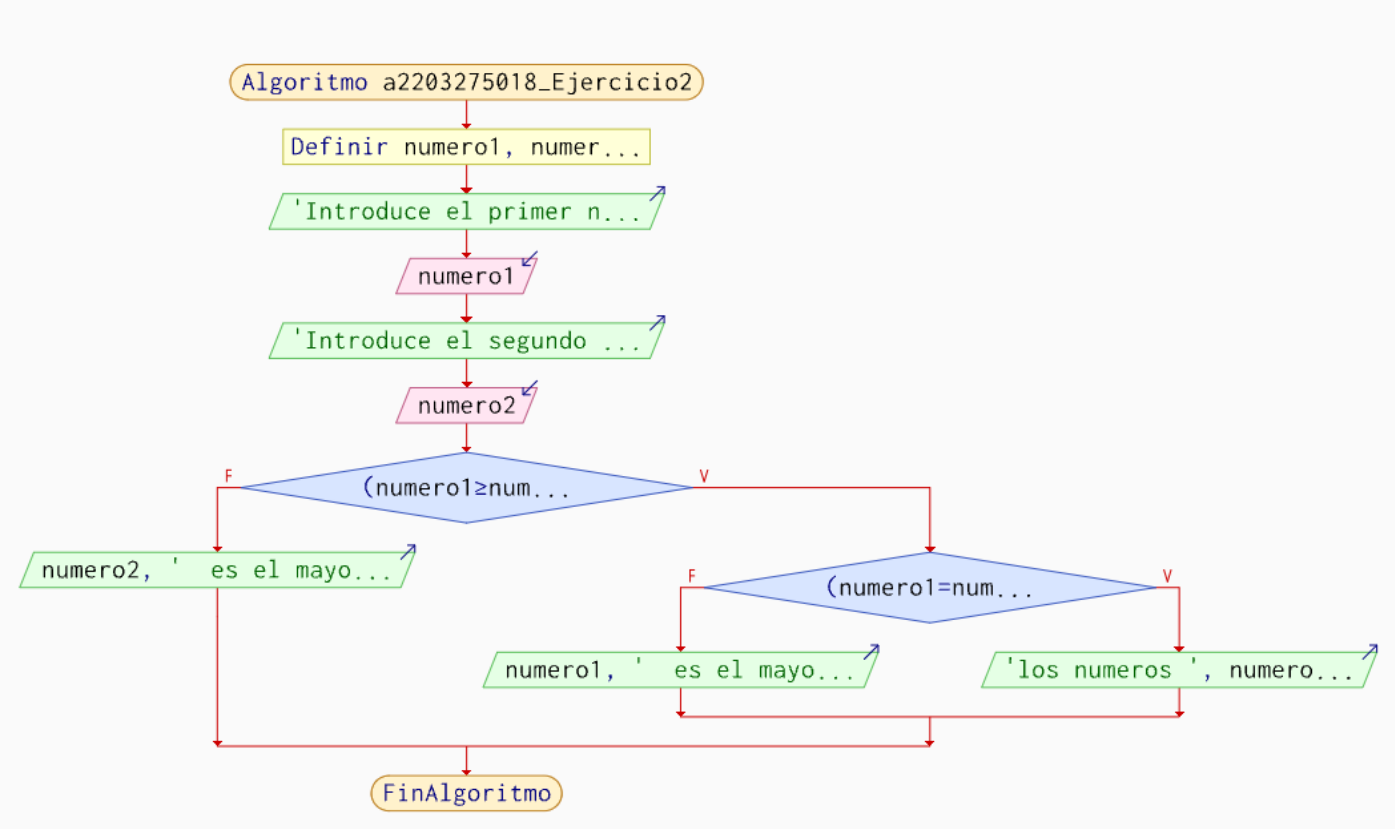
Pseudocodigo:

```
Algoritmo a2203275018_Ejercicio1
   Definir ciclo, divisor, dividendo, residuo Como Entero
   ciclo = 0
   Escribir "Introduce el valor del divisor"
   Leer divisor
   Escribir "Introduce el valor del dividendo"
   Leer dividendo
   residuo = dividendo
   ciclo = 0
         Repetir
     residuo = residuo - divisor
     ciclo = ciclo + 1
   Hasta que residuo > divisor
   Escribir "El cociente = ", ciclo, "  residuo = ", residuo
FinAlgoritmo
```

Salida:

```
*** Ejecución Iniciada. ***
Introduce el valor del divisor
> 4
Introduce el valor del dividendo
> 40
El cociente = 1  residuo = 36
*** Ejecución Finalizada. ***
```

Ejercicio 2



Haremos el mismo procedimiento que anterior ejercicio

Pseudocodigo:

Algoritmo a2203275018_Ejercicio2

Definir numero1, numero2 Como Entero

Escribir "Introduce el primer numero:"

Leer numero1

Escribir "Introduce el segundo numero: "

Leer numero2

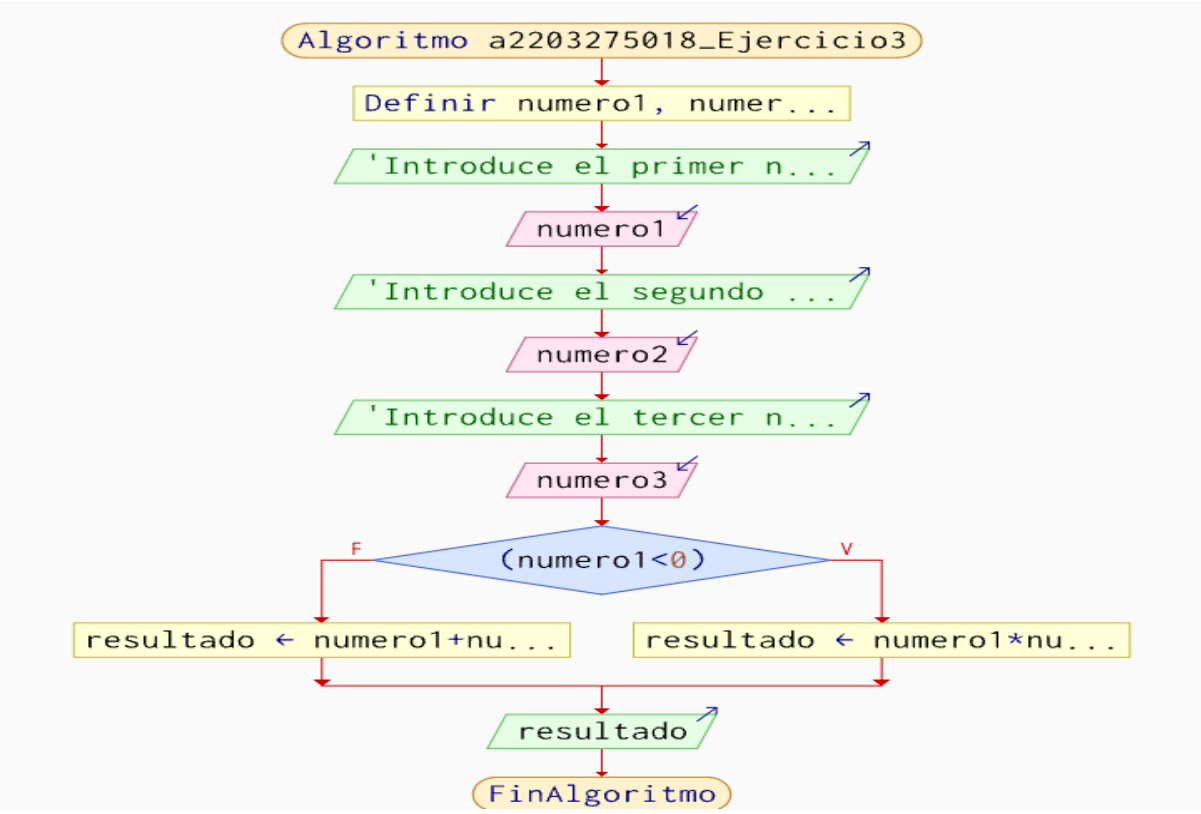
si (numero1>=numero2) Entonces

```
        si (numero1=numero2) Entonces
                Escribir "los numeros " numero1 " , " numero2 " son iguales"
        SiNo
                Escribir numero1 "  es el mayor de los dos"
        FinSi
SiNo
        Escribir numero2 "  es el mayor de los dos"
FinSi
FinAlgoritmo
```

Salida:

```
*** Ejecución Iniciada. ***
Introduce el primer numero:
> 45
Introduce el segundo numero:
> 43
45  es el mayor de los dos
*** Ejecución Finalizada. ***
```

Ejercicio 3:



se hará lo mismo que los anteriores ejemplos

Pseudocodigo:

```
Algoritmo a2203275018_Ejercicio3
        Definir numero1, numero2, numero3 Como Entero
        Escribir "Introduce el primer numero: "
        Leer numero1
        Escribir "Introduce el segundo numero: "
        Leer numero2
        Escribir "Introduce el tercer numero: "
        Leer numero3
        si (numero1<0) Entonces
                resultado=numero1*numero2*numero3
        SiNo
                resultado=numero1+numero2+numero3
        FinSi
        Escribir resultado
FinAlgoritmo
```

Salida:

```
*** Ejecucion iniciada. ***
Introduce el primer numero:
> 55
Introduce el segundo numero:
> 44
Introduce el tercer numero:
> 56
155
*** Ejecución Finalizada. ***
```

Practica N3: Algoritmo a diagramas de flujo

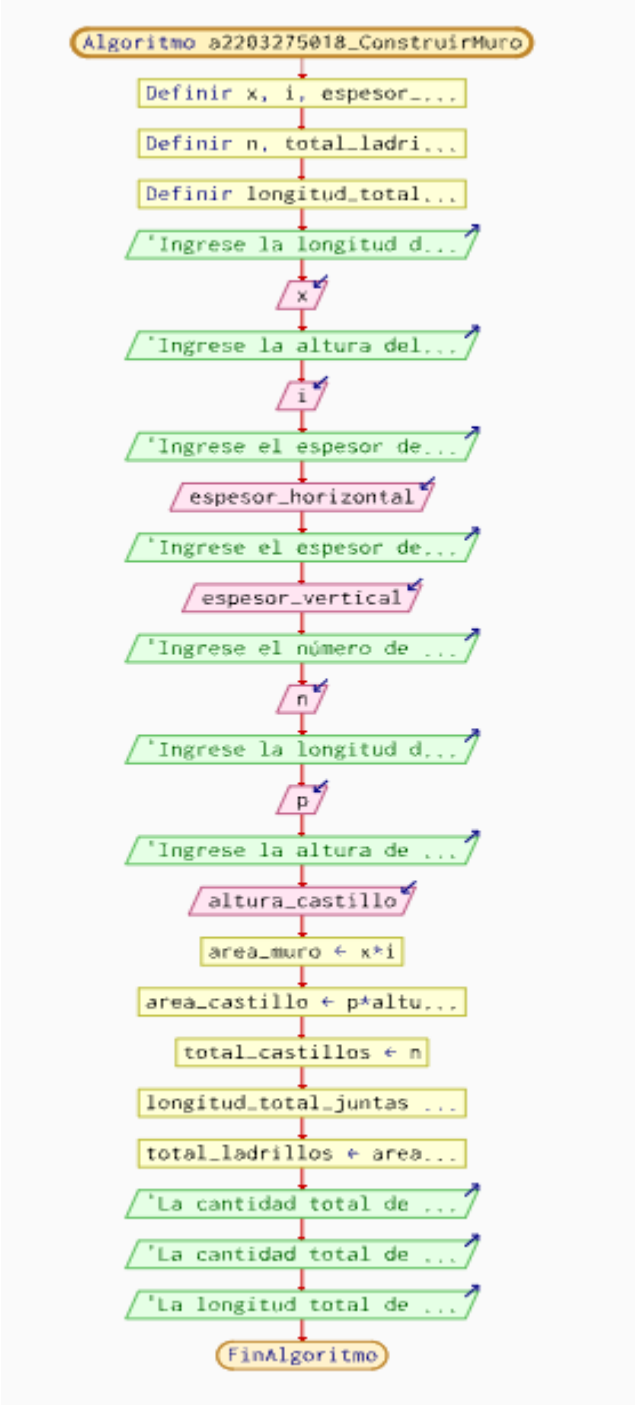
Análisis del problema:

1. **Datos de entrada:**
    - **x**: Longitud total del muro (en metros).
    - **y**: Altura total del muro (en metros).
    - **espesor_horizontal**: Espesor de la junta horizontal (en metros).
    - **espesor_vertical**: Espesor de la junta vertical (en metros).
    - **n**: Número de castillos o columnas de soporte.
    - **p**: Longitud de cada castillo (en metros).
    - **h**: Altura de cada castillo (en metros).
2. **Datos de salida:**
    - **Número total de ladrillos o bloques** necesarios para construir el muro.
    - **Número total de bloques por castillo**.
    - **Distribución de los bloques** en función de las juntas.
3. **Procesos o procesamiento:**
    - Calcular las dimensiones efectivas del muro, considerando los espesores de las juntas.
    - Determinar cuántos bloques se requieren en la longitud y en la altura, tomando en cuenta los castillos y las juntas.
    - Calcular el número total de bloques para el muro y para cada castillo.
4. **Variables a utilizar:**
    - **x (Entero o Real)**: Representa la longitud total del muro en metros. Dependiendo de la precisión, puede ser un número entero o real.
    - **i (Entero o Real)**: Representa la altura total del muro en metros. Al igual que x, puede ser entero o real.
    - **espesor_horizontal (Real)**: Espesor de la junta horizontal en metros.
    - **espesor_vertical (Real)**: Espesor de la junta vertical en metros.
    - **n (Entero)**: Número de castillos o columnas de soporte en el muro.
    - **p (Real)**: Longitud de cada castillo en metros.
    - **altura_castillo (Real)**: Altura de cada castillo en metros.
    - **total_ladrillos (Entero)**: Cantidad total de ladrillos o bloques necesarios para construir el muro.
    - **total_castillos (Entero)**: Cantidad total de castillos en el muro.
    - l**ongitud_total_juntas (Real)**: Longitud total de las juntas horizontales y verticales.
5. Descripción del algoritmo (en lenguaje natural):

- Primero, se ingresan todas las dimensiones y los espesores.
- Luego, se ajustan las dimensiones efectivas del muro, restando los espesores de las juntas.
- Se calcula cuántos bloques caben en la longitud y en la altura del muro, considerando que los castillos ocupan un espacio fijo.
- Finalmente, se suman todos los bloques necesarios para el muro y para cada castillo, y se obtiene el total.

6. Algoritmo en lenguaje orientado a la computadora:

- Definir variables y leer los datos de entrada.
- Calcular las dimensiones efectivas del muro.
- Calcular el número de bloques por longitud y por altura.
- Ajustar el número de bloques considerando los castillos.
- Calcular y mostrar el total de bloques necesarios.



1. **Inicio del diagrama de flujo**

- Se coloca el nombre del algoritmo, en este caso “ConstruirMuro”.
- Esto marca el inicio de la ejecución del programa.

2.Declaración de variables

- En esta sección, se definen todas las variables que se van a usar: por ejemplo, x, i, espesor_horizontal, espesor_vertical, n, p, altura_castillo, entre otras.
- Aquí simplemente se reservan espacios en memoria para los datos.

3.**Entrada de datos**

- Se despliegan mensajes al usuario, por ejemplo: “Ingrese la longitud del muro”.
- Luego se captura la información desde el teclado y se asigna a la variable correspondiente, por ejemplo, x para la longitud del muro.

4.**Repetición de la entrada de datos**

- Se continúa solicitando más datos, como la altura del muro, los espesores, el número de castillos, la longitud y la altura de cada castillo.
- Cada vez que se pide un dato, se muestra el mensaje y luego se asigna el valor a la variable correspondiente.

5.Procesamiento o cálculos

- Una vez que se tienen todos los datos, se realizan los cálculos necesarios, como el área del muro, el área de los castillos, la cantidad total de ladrillos, etcétera.
- Estos cálculos se asignan a sus respectivas variables.

6.**Salida de resultados**

- Se despliegan mensajes finales que muestran los resultados al usuario, como la cantidad total de ladrillos, la cantidad de castillos, y la longitud total de las juntas.
- Esto permite al usuario ver claramente el resultado final.

7.**Fin del diagrama de flujo**

- Finalmente, se llega al bloque de fin, que indica que el algoritmo ha terminado su ejecución.

```
*** Ejecución Iniciada. ***
Ingrese la longitud del muro (x):
> 45
Ingrese la altura del muro (y):
> 25
Ingrese el espesor de la junta horizontal:
> 14
Ingrese el espesor de la junta vertical:
> 25
Ingrese el número de castillos (n):
> 36
Ingrese la longitud de cada castillo (p):
> 14
Ingrese la altura de cada castillo:
> 10
La cantidad total de ladrillos necesarios es: 8.0357142857
La cantidad total de castillos es: 36
La longitud total de juntas es: 8.0357142857
*** Ejecución Finalizada. ***
```

Practica N4:

```
package practica4;
public class practica4_ejer12203275018 {
    public static void main(String[] args) {
        System.out.println("2 * 1 = 2");
        System.out.println("2 * 2 = 4");
        System.out.println("2 * 3 = 6");
        System.out.println("2 * 4 = 8");
        System.out.println("2 * 5 = 10");

    }
}
```

```
- practica4 (run)
run:
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
package practica4;
public class practic4_ejer2_2203275018 {
    public static void main(String[] args) {
        System.out.print("2 * 1 = 2");
        System.out.print("2 * 2 = 4");
        System.out.print("2 * 3 = 6");
        System.out.print("2 * 4 = 8");
        System.out.print("2 * 5 = 10");
    }
}
```

```
ut - practica4 (run)
run:
2 * 1 = 22 * 2 = 42 * 3 = 62 * 4 = 82 * 5 = 10BUILD SUCCESSFUL (total time: 0 seconds)
```

```java
package practica4;
public class practic4_ejer4_2203275018 {
    public static void main(String[] args) {
        System.out.println("2 * 1 ="+(2*1));
        System.out.println("2 * 2 ="+(2*2));
        System.out.println("2 * 3 ="+(2*3));
        System.out.println("2 * 4 ="+(2*4));
        System.out.println("2 * 5 ="+(2*5));

    }
}
```

- practica4 (run)

```
run:
2 * 1 =2
2 * 2 =4
2 * 3 =6
2 * 4 =8
2 * 5 =10
BUILD SUCCESSFUL (total time: 0 seconds)
```

```java
package practica4;
public class practc4_ejer3_2203275018 {
    public static void main(String[] args) {
        System.out.printf("2 * 1 = 2");
        System.out.printf("2 * 2 = 4");
        System.out.printf("2 * 3 = 6");
        System.out.printf("2 * 4 = 8");
        System.out.printf("2 * 5 = 10");
    }
}
```

- practica4 (run)

```
run:
2 * 1 = 22 * 2 = 42 * 3 = 62 * 4 = 82 * 5 = 10BUILD SUCCESSFUL (total time: 0 seconds)
```

```java
package practica4;
public class practic4_ejer5_2203275018 {
    public static void main(String[] args) {
        System.out.print("2 * 1 ="+(2*1));
         System.out.println("");
        System.out.print("2 * 2 ="+(2*2));
        System.out.print("2 * 3 ="+(2*3));
        System.out.print("2 * 4 ="+(2*4));
        System.out.print("2 * 5 ="+(2*5));

}
```

practica4 (run)

```
run:
2 * 1 =2
2 * 2 =42 * 3 =62 * 4 =82 * 5 =10BUILD SUCCESSFUL (total time: 0 seconds)
```

```java
package practica4;
public class practic4_ejer6_2203275018 {
    public static void main(String[] args) {
        System.out.println("coca cola de lata -> 15");
        System.out.print("cantdad vendida 10");
        System.out.print("el subtotal es ->"+(15*10)+" el iva es "+((15*10)*0.16));
        System.out.println();
        System.out.println("el total es ->"+ ((15*10)+((15*10)*016)));
    }
}
```

```
t - practica4 (run)
run:
coca cola de lata -> 15
cantdad vendida 10el subtotal es ->150 el iva es 24.0
el total es ->2250
BUILD SUCCESSFUL (total time: 0 seconds)
```

```java
ackage practica4;
blic class practic4_ejer7_2203275018 {
    public static void main(String[] args) {
        System.out.println("coca cola de lata -> 15.45");
        System.out.print("cantdad vendida 12");
        System.out.print("el subtotal es ->"+(15.45*12)+" el iva es "+((15.45*12)*0.16));
        System.out.println();
        System.out.println("el total es ->"+ ((15.45*12)+((15.45*12)*0.16)));
    }
```

```
- practica4 (run)
run:
coca cola de lata -> 15.45
cantdad vendida 12el subtotal es ->185.39999999999998 el iva es 29.663999999999998
el total es ->215.06399999999996
BUILD SUCCESSFUL (total time: 0 seconds)
```

Practica N5:

Ejercicio 12:

```java
package practica5;
import javax.swing.JOptionPane;
public class ejrcicio12 {
    public static void main(String[] args) {

        int cantidad1, cantidad2, totalproductos;
        String salida, cproducto1, cproducto2;
        double precioproducto1, precioproducto2;
        double subtotal1, subtotal2, iva1, iva2, total1, total2;
        double totalsubproductos, totaliva, totalventa;

        cantidad1 = 12;
        cproducto1 = "Coca Cola de lata 400 ml";
        precioproducto1 = 15.45;

        subtotal1 = precioproducto1 * cantidad1;
        subtotal1 = Math.round(subtotal1 * 100) / 100.0;

        iva1 = subtotal1 * 0.16;
        iva1 = Math.round(iva1 * 100) / 100.0;

        total1 = iva1 + subtotal1;
```

```
        cantidad1++;

        cantidad2 = 18;
        cproducto2 = "Coca Light de lata 400 ml";
        precioproducto2 = 14.45;

        subtotal2 = precioproducto2 * cantidad2;

        subtotal2 = (int) (subtotal2 * 100) / 100.0;

        iva2 = subtotal2 * 0.16;
        iva2 = (int) (iva2 * 100) / 100.0;

        total2 = iva2 + subtotal2;

        totalsubproductos = subtotal1 + subtotal2;
        totalsubproductos = Math.round(totalsubproductos * 100) / 100.0;

        totaliva = iva1 + iva2;
        totaliva = Math.round(totaliva * 100) / 100.0;

        totalventa = total1 + total2;
        totalventa = Math.round(totalventa * 100) / 100.0;

        totalproductos = 0;
        totalproductos += cantidad1;   // suma Cantidad 1
        totalproductos += cantidad2;   // suma Cantidad 2

        boolean umbralSuperado = totalproductos > 25;       // relacional
        boolean ventaAlta = totalventa > 100 && totaliva > 10; // lógico &&

        salida = "Producto          Cantidad     SubTotal     IVA       Total\n";
        salida += cproducto1 + "   " + cantidad1 + "       " + subtotal1 + "       " + iva1 + "       " + total1 + "\n";
        salida += cproducto2 + "   " + cantidad2 + "       " + subtotal2 + "       " + iva2 + "       " + total2 +
"\n\n";

        salida += "Subtotal de la venta del día: " + totalsubproductos + "\n";
        salida += "IVA total de la venta del día: " + totaliva + "\n";
        salida += "Total final de la venta del día: " + totalventa + "\n\n";

        salida += "Total de productos vendidos: " + totalproductos + "\n";

        if (umbralSuperado) {
            salida += "\n✓ Se superó el umbral de productos vendidos.\n";
        } else {
            salida += "\n✗ No se superó el umbral de productos vendidos.\n";
        }

        if (ventaAlta) {
            salida += "✓ Venta alta registrada.\n";
        } else {
            salida += "✗ Venta baja registrada.\n";
        }
        JOptionPane.showMessageDialog(null, salida, "Venta del Día", JOptionPane.DEFAULT_OPTION);
        System.out.println(salida);
```

```
  }
}
```

```
Producto            Cantidad     SubTotal     IVA         Total
Coca Cola de lata 400 ml   13        185.4        29.66        215.06
Coca Light de lata 400 ml   18        260.09        41.61        301.7

Subtotal de la venta del d�a: 445.49
IVA total de la venta del d�a: 71.27
Total final de la venta del d�a: 516.76

Total de productos vendidos: 31

? Se super� el umbral de productos vendidos.
? Venta alta registrada.

BUILD SUCCESSFUL (total time: 14 seconds)
```

1. ANÁLISIS DEL PROBLEMA

El programa simula la venta diaria de **dos productos**:

- Coca Cola lata 400 ml
- Coca Light lata 400 ml

Para cada producto se calcula:

- Subtotal
- IVA
- Total
- Incrementos y operadores combinados
- Cantidad total de productos
- IVA total
- Total final de la venta

Además, el programa usa:

- **Operadores aritméticos**
- **Operadores incrementales (++)**
- **Operadores combinados (+=)**
- **Operadores relacionales (> < ==)**
- **Operadores lógicos (&&)**

Finalmente se muestra toda la información en un **JOptionPane** y en la consola.

2. DATOS DE ENTRADA

Realmente **no hay entrada del usuario**, porque todos los valores están asignados en el código.
Los datos "entrada del sistema" son:

- cantidad1 = 12
- cproducto1 = "Coca Cola de lata 400 ml"
- precioproducto1 = 15.45
- cantidad2 = 18
- cproducto2 = "Coca Light de lata 400 ml"
- precioproducto2 = 14.45

3. DATOS DE SALIDA

El programa muestra:

- Nombre del producto
- Cantidad
- Subtotal

- IVA
- Total
  Y después:
- Subtotal global
- IVA global
- Total final
- Total de productos vendidos
- Mensajes adicionales según condiciones lógicas
  Todo dentro de una ventana **JOptionPane**.

4. PROCESO DEL PROBLEMA

Para cada producto:

1. Multiplicar cantidad × precio → **subtotal**
2. Sacar IVA (16%)
3. Sumar subtotal + IVA → **total del producto**
4. Incrementar cantidad1++ (operador incremental)
5. Sumar subtotales y IVAs
6. Verificar si:
   - totalproductos > 25
   - totalventa > 100 **Y** totaliva > 10

Finalmente:

- Construcción del mensaje final
- Mostrarlo con JOptionPane y imprimir en consola

5. VARIABLES (Tipo y descripción estilo PSeInt)

Variable	Tipo	Descripción
cantidad1	Entero	Cantidad del producto 1 (incrementada en el programa)
cantidad2	Entero	Cantidad del producto 2
totalproductos	Entero	Suma total de productos vendidos
cproducto1	Cadena	Nombre del producto 1
cproducto2	Cadena	Nombre del producto 2
precioproducto1	Real	Precio unitario del producto 1
precioproducto2	Real	Precio unitario del producto 2
subtotal1	Real	Subtotal del producto 1
subtotal2	Real	Subtotal del producto 2
iva1	Real	IVA del producto 1
iva2	Real	IVA del producto 2
total1	Real	Total del producto 1 (subtotal + IVA)
total2	Real	Total del producto 2
totalsubproductos	Real	Suma de subtotales de ambos productos
totaliva	Real	Suma de IVAs

Variable	Tipo	Descripción
totalventa	Real	Total global de la venta
salida	Cadena	Texto final mostrado en JOptionPane
umbralSuperado	Lógico	True si totalproductos > 25
ventaAlta	Lógico	True si totalventa > 100 y totaliva > 10

---

6. ALGORITMO EN LENGUAJE NATURAL

1. Inicializar las variables y asignar valores al producto 1.
2. Calcular subtotal, IVA y total del producto 1.
3. Incrementar la cantidad del producto 1 con ++.
4. Asignar los valores del producto 2.
5. Calcular subtotal, IVA y total del producto 2.
6. Sumar subtotales, IVAs y totales para obtener los valores globales.
7. Sumar ambas cantidades con el operador combinado +=.
8. Evaluar condiciones lógicas para:
    - Si totalproductos supera el umbral
    - Si la venta es considerada alta
9. Construir un texto con toda la información.
10. Mostrar el texto con un JOptionPane.
11. Imprimirlo también en la consola.

Ejercicio 11:

```
public class ejercicio11 {
    public static void main(String[] args) {
        int cantidad1, cantidad2, totalProductos;
        String cproducto1, cproducto2;
        double precioProducto1, precioProducto2;
        double subtotal1, subtotal2;
        double iva1, iva2;
        double total1, total2, totalVenta;

        cantidad1 = 13;
        cproducto1 = "Coca cola de lata 400 ml";
        precioProducto1 = 15.45;
        subtotal1 = precioProducto1 * cantidad1;
        iva1 = subtotal1 * 0.16;
        total1 = subtotal1 + iva1;

        cantidad1++;

        cantidad2 = 18;
        cproducto2 = "Coca Light de lata 400 ml";
        precioProducto2 = 14.45;
        subtotal2 = precioProducto2 * cantidad2;
        iva2 = subtotal2 * 0.16;
        total2 = subtotal2 + iva2;

        totalProductos = cantidad1 + cantidad2;
        totalVenta = total1 + total2;

        String productoMayor = total1 > total2 ? cproducto1 : cproducto2;

        if (totalVenta < 0) {
            totalVenta = 0;
        }

       String salida = "Producto\t\tCantidad\tSubtotal\tIva\tTotal\n";
salida += String.format("%s\t\t%d\t\t%.2f\t%.2f\t%.2f\n", cproducto1, cantidad1, subtotal1, iva1, total1);
salida += String.format("%s\t\t%d\t\t%.2f\t%.2f\t%.2f\n", cproducto2, cantidad2, subtotal2, iva2, total2);

salida += String.format("\n\t\tSubtotal de la venta del día --->\t%.2f\n", totalVenta);
salida += "\t\tEl producto con mayor venta es: " + productoMayor;


        System.out.println(salida);
    }
```

```
run:
Producto                    Cantidad        Subtotal        Iva     Total
Coca cola de lata 400 ml                    14              200,85  32,14   232,99
Coca Light de lata 400 ml                   18              260,10  41,62   301,72

                Subtotal de la venta del d�a --->       534,70
                El producto con mayor venta es: Coca Light de lata 400 ml
BUILD SUCCESSFUL (total time: 0 seconds)
```

1. Datos de entrada y salida:

- **Datos de entrada:**
  - cantidad1 y cantidad2: la cantidad vendida de cada producto (tipo int).
  - precioProducto1 y precioProducto2: el precio unitario de cada producto (tipo double).
  - cproducto1 y cproducto2: el nombre de cada producto (tipo String).
- **Datos de salida:**
  - subtotal1 y subtotal2: el subtotal antes de IVA de cada producto (tipo double).
  - iva1 y iva2: el impuesto aplicado a cada producto (tipo double).
  - total1 y total2: el total final de cada producto (tipo double).
  - totalVenta: la suma total de la venta de ambos productos (tipo double).
  - productoMayor: el producto con mayor venta (tipo String).
  - Todo esto se muestra al final en una salida formateada.

2. Proceso del problema:

El proceso consiste en:

- Asignar valores iniciales a las variables de cantidad, precios y nombres.
- Calcular el subtotal de cada producto multiplicando el precio por la cantidad.
- Calcular el IVA de cada producto como un porcentaje del subtotal.
- Calcular el total de cada producto sumando el subtotal y el IVA.
- Incrementar la cantidad del primer producto en uno.
- Comparar los totales para determinar cuál producto tuvo mayor venta.
- Validar que el total final no sea negativo.
- Mostrar toda la información formateada en pantalla.

3. Variables a utilizar:

- **int**: Para las cantidades y el total de productos, ya que son valores enteros.
- **double**: Para los precios, subtotales, IVA y totales, porque manejamos valores decimales.
- **String**: Para los nombres de los productos.

4. Algoritmo casual:

1. Definir las variables y asignarles valores iniciales (cantidad, precio, nombre).
2. Calcular subtotales, IVA y totales.
3. Incrementar la cantidad del primer producto.
4. Comparar los totales y determinar cuál producto es mayor.
5. Verificar que el total final no sea negativo.
6. Mostrar los resultados en pantalla.

5. Algoritmo computacional:

El algoritmo se traduce en el código Java que hemos visto, donde cada paso se implementa con operaciones aritméticas, lógicas y de comparación en Java.

Ejercicio 10:

```java
public class ejercicio10 {

    public static void main(String[] args) {

        double fahrenheit, celsius, celsiusf;
        String salida;

        fahrenheit = 40;

        fahrenheit++;

        fahrenheit -= 5;

        celsius = 5.0 / 9 * (fahrenheit - 32);
        celsiusf = (5.0 / 9) * (fahrenheit - 32);

        String estado = celsius < 10 ? "Hace frío" : "Temperatura agradable";

        boolean rangoValido = (celsius > -50) && (celsius < 60);

        salida = "Temperatura final en Fahrenheit: " + fahrenheit + "\n";
        salida += "Conversión usando celsius: " + Double.toString(celsius) + "\n";
        salida += "Conversión usando celsiusf: " + String.valueOf(celsiusf) + "\n";
        salida += "Estado según temperatura: " + estado + "\n";
        salida += "¿Temperatura válida?: " + rangoValido;

        // Salida final
        System.out.println(salida);
    }
}
```

```
run:
Temperatura final en Fahrenheit: 36.0
Conversi�n usando celsius: 2.2222222222222223
Conversi�n usando celsiusf: 2.2222222222222223
Estado seg�n temperatura: Hace fr�o
�Temperatura v�lida?: true
BUILD SUCCESSFUL (total time: 0 seconds)
```

1. Datos de entrada y salida:

Datos de entrada:

- fahrenheit: La temperatura en grados Fahrenheit, que es un valor numérico de tipo double.

Datos de salida:

- celsius: La temperatura convertida a grados Celsius, también de tipo double.
- celsiusf: Otra variable para almacenar la conversión a Celsius, de tipo double.
- estado: Un String que indica si la temperatura es fría o agradable.
- rangoValido: Un valor boolean que indica si la temperatura está dentro de un rango lógico.
- salida: Un String que contiene toda la información formateada para mostrar en pantalla.

2. Proceso del problema:

1. **Inicialización:** Se define la temperatura inicial en Fahrenheit.
2. **Modificaciones:** Se incrementa la temperatura en 1 grado y luego se le resta 5 grados.
3. **Conversión:** Se aplica la fórmula para convertir Fahrenheit a Celsius.
4. **Evaluación:** Se determina si la temperatura es fría o agradable con un operador relacional.
5. **Validación:** Se verifica si la temperatura está dentro de un rango lógico usando un operador lógico.
6. **Salida:** Se construye un String con todos los resultados y se imprime en pantalla.

3. Variables: tipo y descripción:

- **double fahrenheit:** Temperatura en grados Fahrenheit.
- **double celsius:** Temperatura convertida a Celsius (primera conversión).
- **double celsiusf:** Temperatura convertida a Celsius (segunda conversión).
- **String salida:** Cadena que contiene el texto final con todos los datos.
- **String estado:** Mensaje que indica si la temperatura es fría o agradable.
- **boolean rangoValido:** Indica si la temperatura está dentro de un rango lógico.

4. Algoritmo casual:

1. Inicializar la temperatura en Fahrenheit.
2. Incrementar la temperatura en 1 grado.
3. Disminuir la temperatura en 5 grados.
4. Convertir la temperatura de Fahrenheit a Celsius.
5. Determinar si la temperatura es fría o agradable.
6. Validar que la temperatura esté en un rango lógico.
7. Construir la cadena de salida con todos los datos.
8. Imprimir la información final.

Practicas N6:

ejercicio 1:

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

    double precio, subtotal, iva, total, sumaiva, sumatotal;
    int cant;

    precio = Double.parseDouble(JOptionPane.showInputDialog("Introduzca el precio del refresco coca cola light lata:"));
    cant = Integer.parseInt(JOptionPane.showInputDialog("Introduzca la cantidad comprada del refresco coca cola light la

    subtotal = precio * cant;
    iva = subtotal * 0.16; // Suponiendo un IVA del 16%
    total = subtotal + iva;

    sumaiva = iva;
    sumatotal = total;

    String salida = "Subtotal: " + subtotal + "\n" +
                    "IVA: " + iva + "\n" +
                    "Total: " + total + "\n";

    precio = Double.parseDouble(JOptionPane.showInputDialog("Introduzca el precio del 1/4 de Arroz:"));
    cant = Integer.parseInt(JOptionPane.showInputDialog("Introduzca la cantidad comprada del 1/4 de Arroz:"));

    subtotal = precio * cant;
    iva = subtotal * 0.16;
    total = subtotal + iva;

    sumaiva += iva;
    sumatotal += total;

    salida += "Subtotal: " + subtotal + "\n" +
              "IVA: " + iva + "\n" +
              "Total: " + total + "\n";

    precio = Double.parseDouble(JOptionPane.showInputDialog("Introduzca el precio de cada pieza de pan francés:"));
    cant = Integer.parseInt(JOptionPane.showInputDialog("Introduzca la cantidad comprada de piezas de pan francés:"));

    subtotal = precio * cant;
    iva = subtotal * 0.16;
    total = subtotal + iva;

    sumaiva += iva;
    sumatotal += total;

    salida += "Subtotal: " + subtotal + "\n" +
              "IVA: " + iva + "\n" +
              "Total: " + total + "\n";

    salida += "\nEl total de productos vendidos fue: " + sumaiva + "\nEl total fue: " + sumatotal;

    JOptionPane.showMessageDialog(null, salida);
```

Subtotal: 1100.0
IVA: 176.0
Total: 1276.0
Subtotal: 125.0
IVA: 20.0
Total: 145.0
Subtotal: 80.0
IVA: 12.8
Total: 92.8

El total de productos vendidos fue: 208.8
El total fue: 1513.8

1. Datos de entrada y salida:

Datos de entrada:

- **Precio de cada producto:** Se ingresan como valores decimales (tipo double).
- **Cantidad de cada producto:** Se ingresan como valores enteros (tipo int).

Datos de salida:

- **Subtotal de cada producto:** Resultado de multiplicar el precio por la cantidad (tipo double).
- **IVA de cada producto:** Calculado como un porcentaje del subtotal (tipo double).
- **Total de cada producto:** Suma del subtotal y el IVA (tipo double).
- **Suma total del IVA y del total:** Acumulado de todos los productos (tipo double).
- **Resumen final:** Cadena de texto que muestra todos los subtotales, IVAs, totales y el total general (tipo String).

2. Proceso del problema:

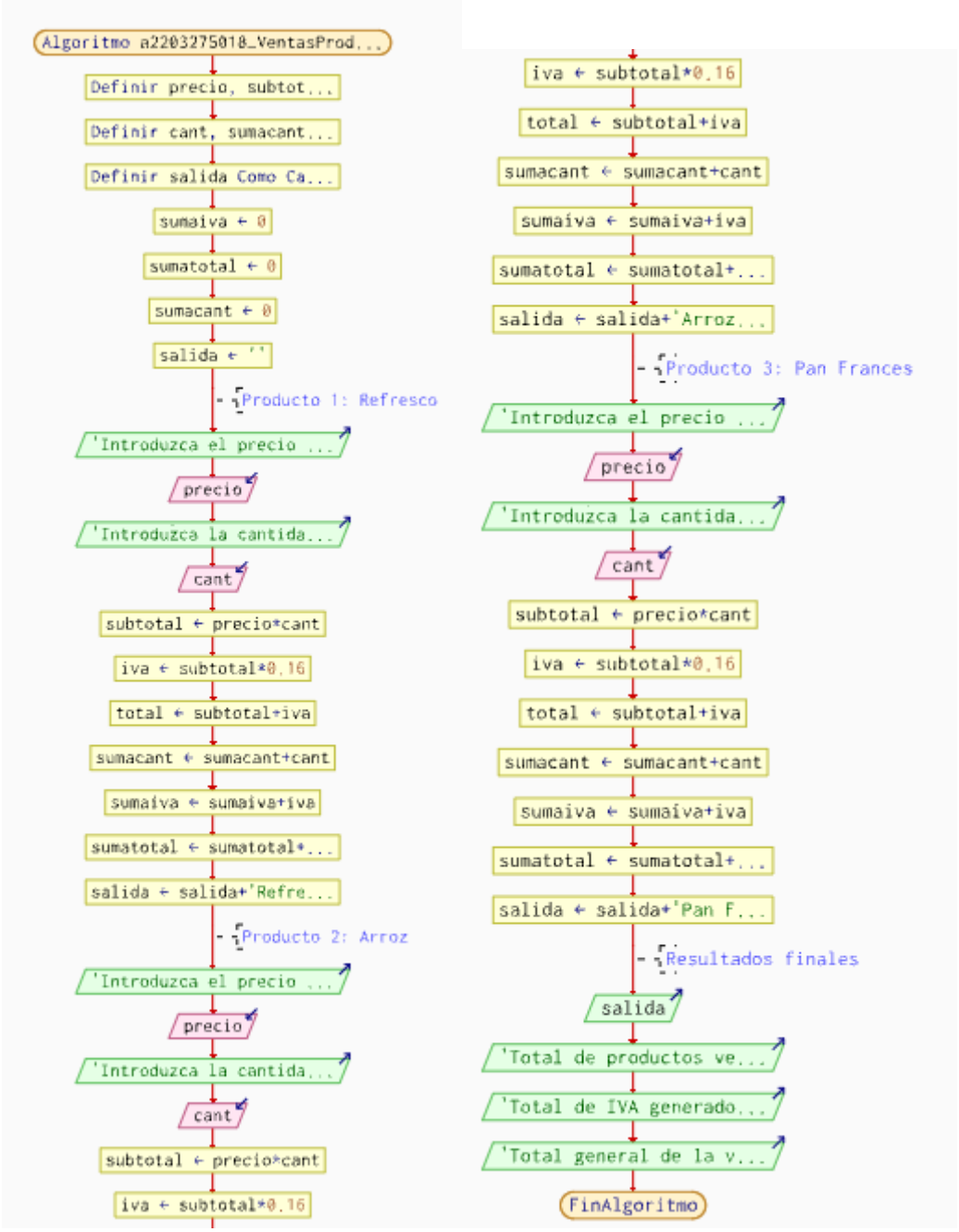
1. **Entrada de datos:** Se solicita el precio y la cantidad de cada producto mediante cuadros de diálogo o entrada por consola.
2. **Cálculos parciales:** Para cada producto, se calcula el subtotal, el IVA y el total.
3. **Acumulación:** Se suman los valores de IVA y total de todos los productos.
4. **Salida de resultados:** Se genera una cadena con todos los resultados y se muestra al usuario.

3. Variables: tipo y descripción:

- **double precio:** Precio unitario del producto (valor decimal).
- **int cant:** Cantidad del producto (valor entero).
- **double subtotal:** Resultado de la multiplicación de precio por cantidad.
- **double iva:** Impuesto calculado sobre el subtotal.
- **double total:** Suma del subtotal y el IVA.
- **double sumaiva:** Acumulado del IVA de todos los productos.
- **double sumatotal:** Acumulado del total de todos los productos.
- **String salida:** Cadena que contiene el resumen final de la venta.

4. Algoritmo casual:

1. Pedir el precio de cada producto.
2. Pedir la cantidad de cada producto.
3. Calcular el subtotal multiplicando precio por cantidad.
4. Calcular el IVA como un porcentaje del subtotal.
5. Calcular el total sumando subtotal e IVA.
6. Acumular los valores de IVA y total.
7. Repetir el proceso para cada producto.
8. Al final, mostrar un resumen con todos los cálculos.


Algoritmo a2203275018_VentasProd...
Definir precio, subtot...
Definir cant, sumacant...
Definir salida Como Ca...
sumaiva ← 0
sumatotal ← 0
sumacant ← 0
salida ← ''
Producto 1: Refresco
'Introduzca el precio ...
precio
'Introduzca la cantida...
cant
subtotal ← precio*cant
iva ← subtotal*0.16
total ← subtotal+iva
sumacant ← sumacant+cant
sumaiva ← sumaiva+iva
sumatotal ← sumatotal*...
salida ← salida+'Refre...
Producto 2: Arroz
'Introduzca el precio ...
precio
'Introduzca la cantida...
cant
subtotal ← precio*cant
iva ← subtotal*0.16
total ← subtotal+iva
sumacant ← sumacant+cant
sumaiva ← sumaiva+iva
sumatotal ← sumatotal+...
salida ← salida+'Arroz...
Producto 3: Pan Frances
'Introduzca el precio ...
precio
'Introduzca la cantida...
cant
subtotal ← precio*cant
iva ← subtotal*0.16
total ← subtotal+iva
sumacant ← sumacant+cant
sumaiva ← sumaiva+iva
sumatotal ← sumatotal+...
salida ← salida+'Pan F...
Resultados finales
salida
'Total de productos ve...
'Total de IVA generado...
'Total general de la v...
FinAlgoritmo

Ejercicio 2:

```java
package practica6;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import javax.swing.JOptionPane;
import java.util.Scanner;
public class ejercicio1 {
    public static void main(String[] args) {
        try {
            // Usando BufferedReader
            BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Ingrese un número entre 0 y 99,999 usando BufferedReader:");
            int numBufferedReader = Integer.parseInt(reader.readLine());
            procesarNumero(numBufferedReader, "BufferedReader");
            // Usando Scanner
            Scanner scanner = new Scanner(System.in);
            System.out.println("Ingrese un número entre 0 y 99,999 usando Scanner:");
            int numScanner = scanner.nextInt();
            procesarNumero(numScanner, "Scanner");
            // Usando JOptionPane
            String inputJOptionPane = JOptionPane.showInputDialog("Ingrese un número entre 0 y 99,999 usando JOptionPane:");
            int numJOptionPane = Integer.parseInt(inputJOptionPane);
            procesarNumero(numJOptionPane, "JOptionPane");
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
    public static void procesarNumero(int num, String metodo) {
        int u, d, c, um, dm;
        u = num % 10;
        num = num / 10;
        d = num % 10;
        num = num / 10;
        c = num % 10;
        num = num / 10;
        um = num % 10;
        num = num / 10;
        dm = num % 10;
        int numReconstruido = num * 10000 + um * 1000 + c * 100 + d * 10 + u;
        System.out.println("Método: " + metodo);
        System.out.println("Unidades: " + u);
        System.out.println("Decenas: " + d);
        System.out.println("Centenas: " + c);
        System.out.println("Unidades de millar: " + um);
        System.out.println("Decenas de millar: " + dm);
        System.out.println("Número reconstruido: " + numReconstruido);
        System.out.println("---------------------------------------");
    }
}
```

```
run:
Ingrese un n◆mero entre 0 y 99,999 usando BufferedReader:
55556
M◆todo: BufferedReader
Unidades: 6
Decenas: 5
Centenas: 5
Unidades de millar: 5
Decenas de millar: 5
N◆mero reconstruido: 55556
---------------------------------------
Ingrese un n◆mero entre 0 y 99,999 usando Scanner:
65923
M◆todo: Scanner
Unidades: 3
Decenas: 2
Centenas: 9
Unidades de millar: 5
Decenas de millar: 6
N◆mero reconstruido: 65923
---------------------------------------
M◆todo: JOptionPane
Unidades: 9
Decenas: 9
Centenas: 9
Unidades de millar: 8
Decenas de millar: 0
N◆mero reconstruido: 8999
---------------------------------------
BUILD SUCCESSFUL (total time: 22 seconds)
```

1. Datos de entrada

El programa recibe un número entero ingresado por el usuario, que debe estar en el rango de **0 a 99,999**.

- **Entrada principal:** num (entero).

2. Datos de salida

El programa produce los siguientes resultados:

- u: Unidades del número.
- d: Decenas del número.
- c: Centenas del número.
- um: Unidades de millar del número.
- dm: Decenas de millar del número.
- numReconstruido: El número reconstruido a partir de sus componentes.

Además, se imprime el método de entrada utilizado.

3. Proceso del problema

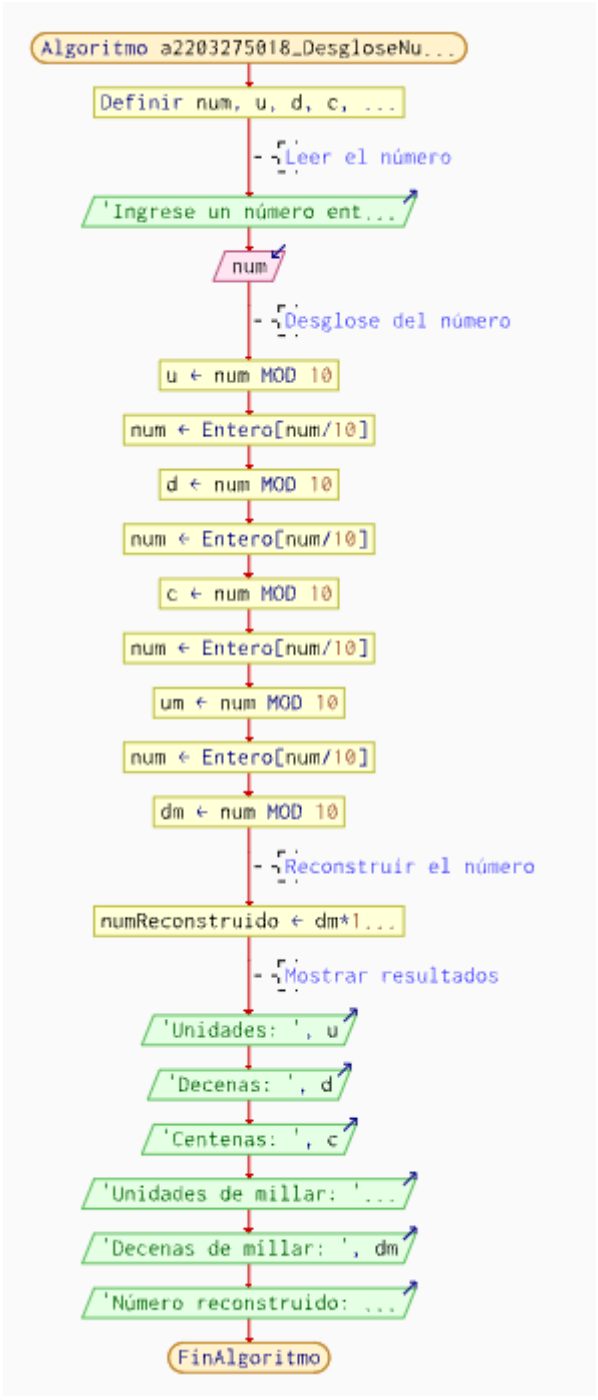
1. **Entrada del número:** Se solicita al usuario que ingrese un número mediante diferentes métodos: BufferedReader, Scanner y JOptionPane.
2. **Desglose del número:** Se descompone el número en sus unidades, decenas, centenas, unidades de millar y decenas de millar usando operaciones aritméticas (módulo y división entera).
3. **Reconstrucción del número:** Se vuelve a construir el número utilizando los componentes obtenidos.
4. **Salida de resultados:** Se muestran en consola los componentes y el número reconstruido.

4. Variables utilizadas

Variable	Tipo	Descripción
num	Entero	Número ingresado por el usuario.
u	Entero	Unidades del número.
d	Entero	Decenas del número.
c	Entero	Centenas del número.
um	Entero	Unidades de millar del número.
dm	Entero	Decenas de millar del número.
numReconstruido	Entero	Número reconstruido a partir de los componentes.

5. Algoritmo casual Inicialización: Se declara e inicializa todas las variables necesarias.

1. **Entrada del número:** Se solicita al usuario el número a analizar.
2. **Desglose del número:** Se utiliza la operación módulo y división entera para extraer cada dígito del número.
3. **Reconstrucción del número:** Se combinan los dígitos para formar el número original.
4. **Salida de resultados:** Se muestran en consola los componentes del número y el número reconstruido.


Algoritmo a2203275018_DesgloseNu...
Definir num, u, d, c, ...
Leer el número
'Ingrese un número ent...
num
Desglose del número
u ← num MOD 10
num ← Entero[num/10]
d ← num MOD 10
num ← Entero[num/10]
c ← num MOD 10
num ← Entero[num/10]
um ← num MOD 10
num ← Entero[num/10]
dm ← num MOD 10
Reconstruir el número
numReconstruido ← dm*1...
Mostrar resultados
'Unidades: ', u
'Decenas: ', d
'Centenas: ', c
'Unidades de millar: '...
'Decenas de millar: ', dm
'Número reconstruido: ...
FinAlgoritmo